Jens Grossklags
Jean Walrand (Eds.)

LNCS 7638

Decision and Game Theory for Security

Third International Conference, GameSec 2012
Budapest, Hungary, November 2012
Proceedings

Lecture Notes in Computer Science 7638

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Jens Grossklags Jean Walrand (Eds.)

Decision and Game Theory for Security

Third International Conference, GameSec 2012
Budapest, Hungary, November 5-6, 2012
Proceedings

Springer

Volume Editors

Jens Grossklags
The Pennsylvania State University
College of Information Sciences and Technology
329A Information Sciences and Technology Building
University Park, PA 16802, USA
E-mail: jensg@ist.psu.edu

Jean Walrand
University of California
Department of Electrical Engineering and Computer Sciences
257M Cory Hall, Berkeley, CA 94720, USA
E-mail: wlr@eecs.berkeley.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-34265-3 e-ISBN 978-3-642-34266-0
DOI 10.1007/978-3-642-34266-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012949278

CR Subject Classification (1998): C.2.0, J.1, D.4.6, K.4.4, K.6.5, H.2-3

LNCS Sublibrary: SL 4 – Security and Cryptology

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Security is a multifaceted problem area that requires a careful appreciation of many complexities regarding the underlying technical infrastructure as well as of human, economic, and social factors. Securing resources involves decision making on multiple levels of abstraction while considering variable planning horizons. At the same time, the selection of security measures needs to account for limited resources available to both malicious attackers and administrators defending networked systems. Various degrees of uncertainty and incomplete information about the intentions and capabilities of miscreants further exacerbate the struggle to select appropriate mechanisms and policies.

The GameSec conferences aim to bring together researchers who are working on the theoretical foundations and behavioral aspects of enhancing security capabilities in a principled manner. The successful previous instances of the conference series took place in 2010 in Berlin, Germany, and 2011 in College Park, Maryland, USA. Contributions at the first two meetings included analytic models based on game, information, communication, optimization, decision, and control theories that were applied to diverse security topics. In addition, researchers contributed papers which highlighted the connection between economic incentives and real-world security, reputation, trust, and privacy problems.

The Third International Conference on Decision and Game Theory for Security took place in Budapest, Hungary. We solicited papers on all economic aspects of security and privacy, and received a record number of thirty-seven submissions. The submitted papers were evaluated by the international Program Committee based on their significance, originality, technical quality, and exposition.

This edited volume contains ten contributed full papers, and eight contributed short papers that constituted the scientific part of the conference program. These articles are categorized into the following six sections:

- The section on *secret communications* includes two full papers which model the interaction between attackers and defenders in games related to practical steganography and repeated rational secret sharing.
- The second book part on the *identification of attackers* consists of three full papers on security audits, intruder classification, and the crowding out of miscreants from cybercriminal markets.
- Two full papers and one short paper form the section on *multi-step attacks* and improve our understanding of the complex behavior of adversaries.
- The section on *network security* includes one full paper and two short papers with economic models of security decision making under consideration of network topologies.

- One full paper and three short papers are focused on improved models of *system defense*. Topics include the placement of honeypots and the optimal management of moving target defense systems.
- The section on *applications security* with one full paper and two short papers addresses challenges related to security in electricity distribution, smart grid systems, and cloud-based architectures.

The contributed research papers address important challenges that security practitioners are confronted with in practice. Studying security from the economic perspective allows for generalizable insights across different types of security incidents, and strengthens the ability to formulate appropriate questions about complex security problems. This edited volume will also be of interest to experienced researchers and students who aim to contribute to the next wave of research results at the exciting intersection of economics and security.

November 2012

Jens Grossklags
Jean Walrand

Organization

Steering Committee

Tansu Alpcan	University of Melbourne, Australia
Nick Bambos	Stanford University, USA
John Baras	University of Maryland, USA
Tamer Başar	University of Illinois, Urbana-Champaign, USA
Anthony Ephremides	University of Maryland, USA
Jean-Pierre Hubaux	École Polytechnique Fédérale de Lausanne, Switzerland

Organizing Committee

Conference Co-chairs

Márk Félegyházi	Budapest University of Technology and Economics, Hungary
Levente Buttyán	Budapest University of Technology and Economics, Hungary

Program Co-chairs

Jens Grossklags	The Pennsylvania State University, USA
Jean Walrand	University of California, Berkeley, USA

Publicity Chairs

Fan Wu	Shanghai Jiao Tong University, China
Tansu Alpcan	University of Melbourne, Australia

Finance and Registration Chair

Boldizsár Bencsáth	Budapest University of Technology and Economics, Hungary

Local Chair

Tamás Holczer	Budapest University of Technology and Economics, Hungary

Web Co-Masters

Ta Vinh Thong	Budapest University of Technology and Economics, Hungary
Gergely Kótyuk	Budapest University of Technology and Economics, Hungary

Program Committee

Alessandro Acquisti	Carnegie Mellon University, USA
Tansu Alpcan	University of Melbourne, Australia
Saurabh Amin	Massachusetts Institute of Technology, USA
Ross Anderson	University of Cambridge, UK
John Baras	University of Maryland, USA
Tamer Başar	University of Illinois, Urbana-Champaign, USA
Rainer Böhme	University of Münster, Germany
Levente Buttyán	Budapest University of Technology and Economics, Hungary
Jean Camp	Indiana University, USA
Nicolas Christin	Carnegie Mellon University, USA
John Chuang	University of California, Berkeley, USA
George Cybenko	Dartmouth College, USA
Márk Félegyházi	Budapest University of Technology and Economics, Hungary
Assane Gueye	National Institute of Standard and Technologies, USA
Joseph Halpern	Cornell University, USA
Tembine Hamidou	Supélec, France
Zhu Han	University of Houston, USA
Kjell Hausken	University of Stavanger, Norway
Cormac Herley	Microsoft Research, USA
Pan Hui	Telekom Innovation Laboratories, Germany
Rahul Jain	University of Southern California, USA
Benjamin Johnson	University of California, Berkeley, USA
Murat Kantarcioglu	University of Texas at Dallas, USA
Marc Lelarge	INRIA and École Normale Supérieure, France
Patrick Loiseau	Eurecom, France
Kanta Matsuura	University of Tokyo, Japan
Patrick McDaniel	The Pennsylvania State University, USA
John Musacchio	University of California, Santa Cruz, USA
Alan Nochenson	The Pennsylvania State University, USA
Andrew Odlyzko	University of Minnesota, USA
Radha Poovendran	University of Washington, USA
Aaron Roth	University of Pennsylvania, USA
Srinivasan Raghunathan	University of Texas at Dallas, USA
Galina Schwartz	University of California, Berkeley, USA
Rahul Telang	Carnegie Mellon University, USA
Tunay Tunca	University of Maryland, USA
Jun Zhuang	University at Buffalo, SUNY, USA

External Reviewers

Amar Azad	University of California, Santa Cruz, USA
Lemonia Dritsoula	University of California, Santa Cruz, USA
Peiqiu Guan	University at Buffalo, SUNY, USA
Abhishek Gupta	University of Illinois, Urbana-Champaign, USA
Fei He	University at Buffalo, SUNY, USA
SingRu Celine Hoe	Texas A&M University-Commerce, USA
Kien Nguyen	University of Illinois, Urbana-Champaign, USA
Robert Nix	University of Texas at Dallas, USA
Ranjan Pal	University of Southern California, USA
Viet Pham	University of London, UK
Shreyas Sundaram	University of Waterloo, Canada
Krzysztof Szajowski	Wrocław University of Technology, Poland
Jie Xu	University at Buffalo, SUNY, USA

Sponsoring Institutions

Local Sponsor

GameSec 2012 was locally supported by the Laboratory of Cryptography and System Security (CrySyS Lab) at the Budapest University of Technology and Economics (BME), Hungary.

Technical Co-sponsors

IEEE Control Systems Society
ACM Special Interest Group on Security, Audit and Control

Table of Contents

Secret Communications

Identification of Attackers

Multi-step Attacks

Network Security

System Defense

Applications Security

Where to Hide the Bits?

Benjamin Johnson[1], Pascal Schöttle[2], and Rainer Böhme[2]

[1] Department of Mathematics, UC Berkeley, USA
[2] Department of Information Systems, University of Münster, Germany

Abstract. We present a stochastic two-player zero-sum game between defender and attacker related to the security of practical steganography. The defender wants to hide a message in a cover object drawn by nature. The attacker wants to distinguish plain covers from those with a hidden message. We study the case of n-bit covers, independently but not identically distributed to allow for variation in the predictability between parts of the cover. The defender knows the predictability exactly and chooses k embedding positions. The attacker may obtain side information to predict one chosen position of the cover and compare it to the observed object to make a decision. We present a unique mixed strategy Nash equilibrium for this game. It turns out that the attacker's strategy is independent of the number of hidden bits k.

Keywords: Game Theory, Information Hiding, Steganography, Security.

1 Introduction

Steganography is the art and science of hiding the very existence of a secret message by embedding it into inconspicuous cover data, such as image or audio files. A (minimal) steganographic embedding function takes as input a message and a key. It outputs a so-called stego object, which is sent to the recipient, who can extract the hidden message using the shared secret key. Steganalysis, the countermeasure, tries to detect steganography by deciding whether an observed object is cover or stego without knowing the secret key.

Prior work has established the following theory. Given a cover distribution $\mathcal{P}_0$ and a fixed embedding function, the distribution of stego objects $\mathcal{P}_1$ is completely determined. *Perfect steganography* is possible if $\mathcal{P}_0 = \mathcal{P}_1$, and efficient codes exist to embed a message [12]. For perfect steganography, a computationally unbounded steganalyst's chance to make the correct detection decision is no better than random guessing. If $\mathcal{P}_0$ is unknown, but can be sampled efficiently, then the existence of a cryptographic oneway function is sufficient to construct an embedding function that achieves $\mathcal{P}_0 \approx \mathcal{P}_1$ so that computationally bounded steganalysts have only negligible advantage at the detection decision [6]. If $\mathcal{P}_0 \neq \mathcal{P}_1$, we speak of *imperfect steganography* and the degree of imperfection can be quantified by the Kullback–Leibler divergence (KLD) between $\mathcal{P}_0$ and $\mathcal{P}_1$ [2]. All practical steganographic embedding functions proposed for real

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 1–17, 2012.

cover media belong to the class of imperfect steganography. This is so because $\mathcal{P}_0$ is unknown, arguably unknowable, and therefore it is virtually impossible to find an embedding function that preserves $\mathcal{P}_0$ exactly [1]. Security bounds for this relevant case of imperfect steganography have been derived mainly under the strong, yet conservative, assumption that the steganalyst knows $\mathcal{P}_0$ [9]. Few extensions exist for the case where both steganographer and steganalyst have incomplete knowledge of $\mathcal{P}_0$ [8]. All these studies predict and experimentally validate asymptotic detectability bounds, but they contain no constructive elements on how to design secure embedding functions.

Due to this deficit of actionable theory, engineering efforts to design practical embedding functions are dominated by heuristics and simulation experiments, often involving machine-learning techniques [10]. One rule of thumb is to minimize the embedding distortion, reflecting the conjecture that $\mathcal{P}_0$ is locally smooth around the realized cover. The actual measures of distortion vary between approaches. Another recurring thread is the idea of *content-adaptive* steganography. It is based on the observation that most cover sources produce *heterogeneous* covers, meaning that if the embedding domain of a cover object is represented by a fixed sequence of symbols, then different positions exhibit different statistical properties. For example, natural images often consist of smooth areas and gradients, but also contain some sharp edges or regions with noisy texture. Because these areas differ in predictability, e. g., in the accuracy of local statistical models, the steganalyst may find it easier to detect subtle embedding changes by deviations from the model in more predictable than in less predictable spots of the cover. To exploit differences in detectability, a content-adaptive embedding function tries to improve steganographic security by concentrating k embedding changes in less predictable areas of a cover, rather than distributing them uniformly over all n possible embedding positions [1].

Our contribution is to narrow the gap between theory and practice with a game-theoretic analysis of the optimal choice of embedding positions in the realistic regime where both steganographer and steganalyst have incomplete knowledge of $\mathcal{P}_0$. As initially pointed out in [11], game theory is the method of choice in this regime, because both players have to decide in which positions they hide or look for evidence of embedding, respectively, in anticipation of the opponent's action. The specific contribution of this work is to extend our model in [11] from the very artificial case of only two positions to covers of size n. Our results here are constructive in the sense that the equilibrium strategy can be efficiently computed for any given vector of predictability.

Here is the structure of our paper. The next Section 2 specifies the game setup and connects it to a specific interpretation of the steganographic communication model. Section 3 presents the solution of the game, starting with message sizes of $k = 1$ bits and generalizing from there. The discussion in Sect. 4 comments on the applicability of our results for the design of secure practical steganography and points to alternative interpretations in the broader field of information security. The final Section 5 concludes with an outlook on open problems.

2 Problem Definition

Let Alice be the defender (steganographer) and Eve be the attacker (steganalyst). Figure 1 visualizes the steganographic communication model. Function embed takes as input the secret message, a fresh cover, and the secret key. It outputs a stego object which is as indistinguishable as possible from covers. The stego object is communicated over the insecure channel. The recipient listening at the other end can apply function extract to retrieve the message. We do not further consider the recipient, but abstract from the necessary coding and cryptography layers which ensure that the recipient can always extract the message correctly. Our game is formulated between Alice, who implements embed, and Eve, who implements function detect. This function outputs a decision of whether the observed object is a plain cover or a stego object. Figure 1 differs from the standard model by allowing Eve to query some side information directly from the cover. Recall that practical steganalysis can often estimate such side information from the observed object. Therefore, we will elaborate below why we require this explicit interaction in our game.

To formalize the role of side information, let the embedding domain of a cover be a random sequence of n symbols with varying predictability from some kind of side information. This side information is fully available to Alice and partly available to Eve. Practical predictors, for instance, exploit spatial correlation in the local neighborhood of a symbol to estimate its most likely value. We assume that both players can exactly quantify the predictability[1] per symbol and order the symbols of the embedding domain by increasing predictability. (Both reordering and potential domain transformations are reversible so that the object on the communication channel always appears in its original order and domain.)

Therefore, we consider a vector $\boldsymbol{X} = (X_0, \ldots, X_{n-1})$ of independent random variables drawn from a binary alphabet $\mathcal{C} = \{0, 1\}$, with realizations typeset in lower case, $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$. Note that real covers may have a larger alphabet, but we settle on bits for a clearer notion of predictability. Moreover, practical embedding functions often work on a vector of binary residuals, such as the sequence of all least significant bits. The monotonically increasing function $f(i) : \{0, \ldots, n-1\} \to [\frac{1}{2}, 1]$ defines the probability of X_i taking its most likely value. Without loss of generality, let $f(i) = P(X_i = 1)$ for the analysis.

To anchor the two ends of the predictability range, we require $f(0) = \frac{1}{2} + \varepsilon$ and $f(n-1) = 1 - \varepsilon$. We need a strictly positive ε to ensure that we operate in the imperfect steganography regime. If ε were zero, Alice could embed at least one bit into x_0 without risk of detection. Similarly, if $P(X_{n-1} = 1) = 1$, embedding into x_{n-1} would allow detection with certainty.

Alice's action space is to flip k bits of a given cover realization $\boldsymbol{x}$ to embed a hidden message. There exist appropriate key-dependent codes ensuring that

[1] Our notion of *predictability* closely corresponds to the *detectability profile* in [5] or the *adaptivity criterion* in [11]. Both concepts can be interpreted as proxies to estimate the local predictability.

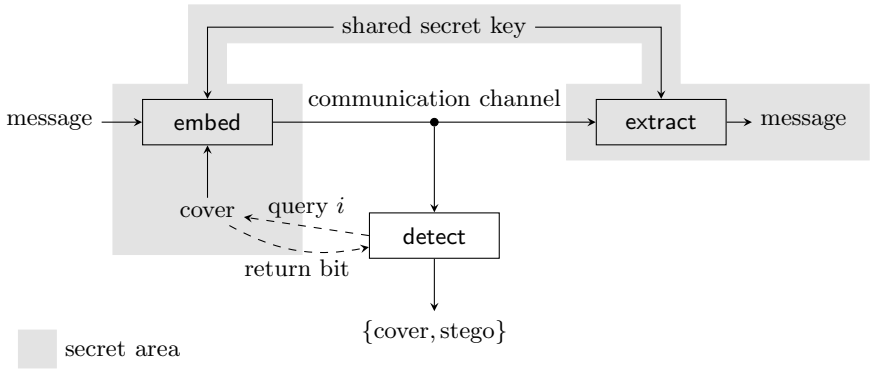


Fig. 1. Block diagram of steganographic communication system with side information

a message of length $k - \mathcal{O}(1)$ can be embedded by changing k cover values irrespective of their position such that the recipient can extract the message with the knowledge of a shared secret key (see for example [4]). Alice chooses a k-sized subset of $\{0, \ldots, n-1\}$ indicating the embedding positions. Her mixed strategy action space is a probability distribution a over all k-sized subsets of $\{0, \ldots, n-1\}$.

Eve tries to decide whether an observed bit vector is a cover or a stego object. We follow the convention in [7] and require that covers and stego objects are equally likely on the communication channel. This can be modeled by assuming that nature flips an unbiased coin to decide if Eve sees Alice's stego object or a cover drawn from the same cover source available to Alice. Eve's optimal decision rule would be a likelihood ratio test using the joint distributions over all cover and stego objects, $\mathcal{P}_0$ and $\mathcal{P}_1$. In practice, however, $\mathcal{P}_0$ and $\mathcal{P}_1$ are unknown and Eve can only make local decisions for individual symbols using a local predictor. We stipulate that Eve can use her knowledge about the marginal distributions of $\mathcal{P}_0$ and $\mathcal{P}_1$ to make optimal local decisions, although this is not always the case for practical steganalysis. While our game might be too optimistic for Eve in this respect, we contrast this by requiring that Eve only looks at one position. To justify this constraint in the basic model, we must assume that the side information necessary for prediction is only available for one position. Therefore, it cannot be estimated from the cover and we must assume an interactive query mechanisms (see Fig. 1). As a result, Eve's mixed strategy action space is a probability distribution e over all n positions for which she can query side information of variable precision, depending on the position's predictability. For all other positions, she cannot tell if $P(X_i = 0) > P(X_i = 1)$ or $P(X_i = 0) < P(X_i = 1)$ in covers. Therefore, she does not gain any information from including the values at these positions in her decision.

It is obvious that Eve's task is very hard in this setup, because if $k = 1$, her advantage over random guessing is not better than ε even if Alice deterministically embeds in the first symbol. If Alice randomizes her strategy, then Eve's

advantage shrinks. If Alice embeds more bits, Eve's advantage increases because Alice has to use more predictable (i. e., less secure) positions. Our objective is to quantify by how much, and if (and where) there is an equilibrium.

The following objective function defines a zero-sum game: Alice tries to increase her security by maximizing Eve's decision error, whereas Eve tries to minimize it. We map this to the payoff structure given in Table 1. Note that this payoff matrix induces an objective function based on the equal error rate. For practical applications, the payoff matrix might need adjustment to account for the harm caused by false positive and false negatives, respectively.

Table 1. Payoff for (Eve, Alice)

	Reality	
Eve's decision	cover	stego
cover	(1, −1)	(−1, 1)
stego	(−1, 1)	(1, −1)

Figure 2 (p. 6) summarizes the game for $k = 1$ in an extensive form graph. From left to right, first, nature draws a cover from $\mathcal{P}_0$, then Alice chooses her single (because $k = 1$) embedding position, creating a stego object (black nodes). A coin flip, invisible to Eve, decides whether she sees the stego or cover object. Then Eve chooses the position she wants to compare with a prediction to make her decision, and outputs the decision result (C for cover or S for stego). Shaded nodes indicate the cases where Eve wins, i. e., she receives positive payoff.

3 Solving the Model

3.1 Preliminaries

We begin by formulating Eve's local decision rule. Eve observes the probability $f(i)$ that bit i is 1. Since $f(i)$ is greater than $\frac{1}{2}$, the object is more likely to be a cover if the observed bit is 1, and more likely to be stego if the observed bit is 0. This constrains Eve's decision rule based on her observation at position i.

$$d(i) = \begin{cases} \text{cover} & \text{if } x_i = 1 \\ \text{stego} & \text{if } x_i = 0 \end{cases}. \tag{1}$$

To simplify the exposition of our equilibrium results, we introduce the notation

$$\tilde{f}(i) = f(i) - \frac{1}{2}. \tag{2}$$

The function $f(i)$ was introduced as the probability of seeing 1 at position i, and it measures the predictability at position i. The function $\tilde{f}(i)$ can be interpreted as measuring the bias at position i.

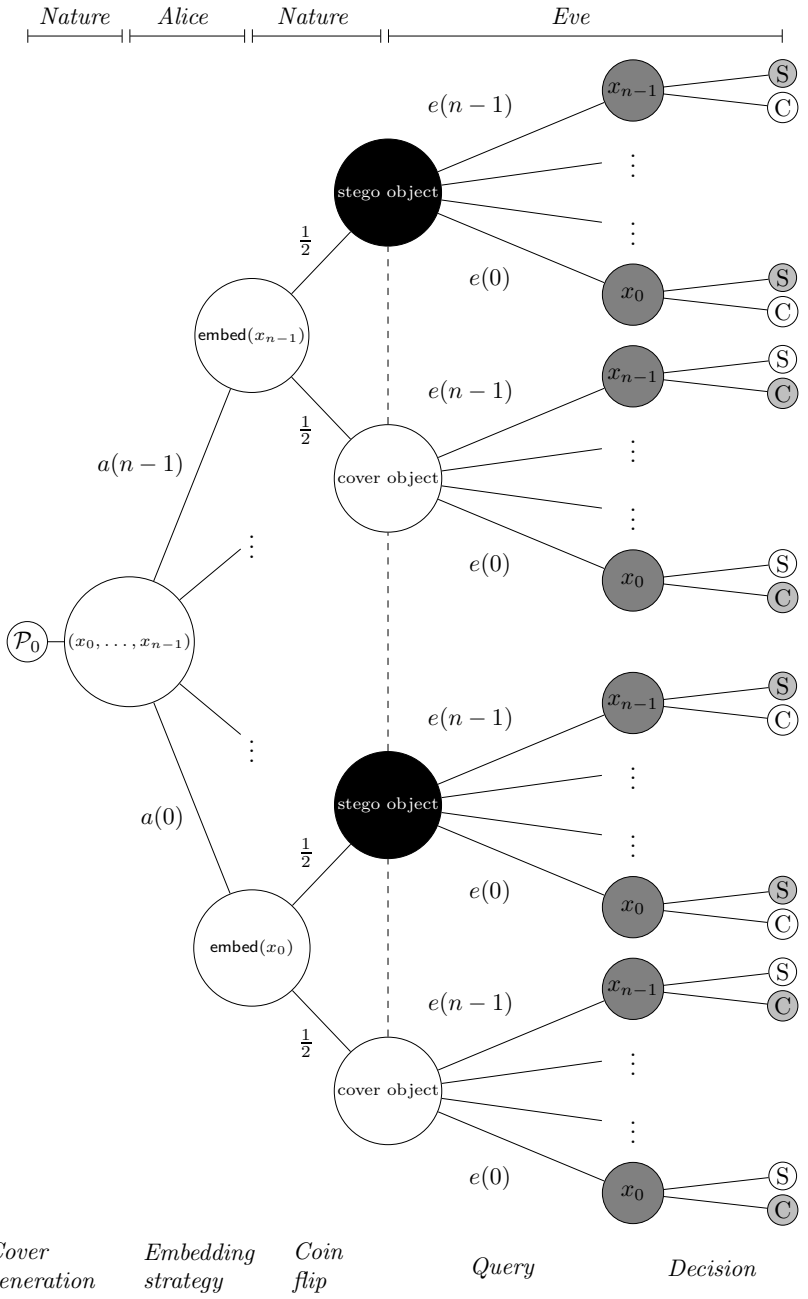


Fig. 2. Extensive form of the game for $k = 1$. The dashed line indicates Eve's information set. The dark gray nodes represent Eve's query strategy and the light gray nodes are the situations in which Eve wins the game.

Recall that Alice's mixed strategy space is a probability distribution over size-k subsets of $\{0, \ldots, n-1\}$. For a subset S of k positions, $a(S)$ is the probability that Alice embeds her bits in these k positions; and we have $\sum_S a(S) = 1$. Overloading notation, let us define the projection of Alice's mixed strategy onto positions to be the total probability that Alice embeds in position i. Formally, we define $a(i)$ for $i \in \{0, \ldots, n-1\}$ as

$$a(i) = \sum_{\{S : i \in S\}} a(S). \tag{3}$$

If Alice embeds in just one position, then $a(i) = a(\{i\})$ and $\sum_{i=0}^{n-1} a(i) = 1$. If Alice embeds k bits, then

$$\sum_{i=0}^{n-1} a(i) = k. \tag{4}$$

Eve's mixed strategy action space is a distribution over positions. Eve queries the bias at position i with probability $e(i)$ and decides stego or cover based only on her observation at position i.

3.2 Game Outcome

We quantify the payoff of Eve and Alice as a function of the bias $\tilde{f}(i)$ at each position, Eve's mixed strategy $e(i)$, and the projection of Alice's mixed strategy onto positions $a(i)$.

Theorem 1 (Game Outcome). *If $\tilde{f}$ is the bias function, e is Eve's mixed strategy, and a is Alice's mixed strategy, then the total expected payoff for (Eve, Alice) is*

$$\left(2 \sum_{i=0}^{n-1} e(i) a(i) \tilde{f}(i), -2 \sum_{i=0}^{n-1} e(i) a(i) \tilde{f}(i) \right). \tag{5}$$

Proof. First assume that Eve looks only at position i. Under this assumption, we may determine the probability she wins the game by enumerating all possible ways the world could be, and adding up the respective probabilities. We may think of the process as an orderly sequence of events. First, nature chooses whether Eve sees a cover object or a stego object by flipping an unbiased coin. The cover object is then instantiated with a realization x_i of position i, with $P(X_i = 1) = f(i)$. If nature chose stego, then Alice flips bit i with probability $a(i)$. Finally, Eve decides whether the object is cover or stego by looking at her observed bit. She decides cover if the bit is 1 and stego if the bit is 0. Table 2 records the events, probabilities, and decision outcomes for each possible case.

Table 2. Game outcome in different states of the world

	Value of x_i				
Reality	Cover	Observed	Probability	Eve's decision	Winner
C	1	1	$\frac{1}{2} \cdot f(i)$	C	Eve
C	0	0	$\frac{1}{2} \cdot (1 - f(i))$	S	Alice
S	1	0	$\frac{1}{2} \cdot f(i) \cdot a(i)$	S	Eve
S	1	1	$\frac{1}{2} \cdot f(i) \cdot (1 - a(i))$	C	Alice
S	0	1	$\frac{1}{2} \cdot (1 - f(i)) \cdot a(i)$	C	Alice
S	0	0	$\frac{1}{2} \cdot (1 - f(i)) \cdot (1 - a(i))$	S	Eve

Legend: C = cover, S = stego

Given that Eve looks only at position i, her probability of winning is

$$\frac{1}{2}\left(f(i) + f(i)a(i) + (1 - f(i))(1 - a(i))\right) \tag{6}$$

$$= \frac{1}{2}\left(f(i) + f(i)a(i) + 1 - a(i) - f(i) + f(i)a(i)\right) \tag{7}$$

$$= \frac{1}{2}\left(1 + 2f(i)a(i) - a(i)\right) \tag{8}$$

$$= \frac{1}{2} + a(i)\left(f(i) - \frac{1}{2}\right) \tag{9}$$

$$= \frac{1}{2} + a(i)\tilde{f}(i). \tag{10}$$

Hence Eve's total probability of winning is

$$\sum_{i=0}^{n-1} e(i)\left(\frac{1}{2} + a(i)\tilde{f}(i)\right) \tag{11}$$

$$= \frac{1}{2} + \sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i), \tag{12}$$

and thus Eve's total expected game payoff is

$$\mathrm{P}(\text{Eve wins}) \cdot 1 + \mathrm{P}(\text{Eve loses}) \cdot (-1) \tag{13}$$

$$= \left(\frac{1}{2} + \sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i)\right) \cdot 1 + \left(\frac{1}{2} - \sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i)\right) \cdot (-1) \tag{14}$$

$$= 2\sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i). \tag{15}$$

The total expected payoff for (Eve, Alice) is thus

$$\left(2\sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i), -2\sum_{i=0}^{n-1} e(i)a(i)\tilde{f}(i)\right). \tag{16}$$

□

3.3 Nash Equilibria

We now turn our attention to the game's Nash equilibria.

Hiding One Bit. We start with analyzing the case of $k = 1$. This simplifies Alice's mixed strategy action space to a probability distribution over the set $\{0, \ldots, n-1\}$. Recall the convention that $a(i)$ is the probability that Alice embeds into position i.

Lemma 1 (Exclusion of pure strategies). *There is no equilibrium in which either Alice or Eve assigns zero probability to any i.*

Proof. Assume Alice assigns zero probability to position i. Then Eve gains no advantage from assigning positive probability to position i. Hence, Eve's best response would assign zero probability to position i. But then Alice can completely eliminate Eve's advantage by assigning probability 1 to position i. So Alice is not in equilibrium.

Assume Eve assigns zero probability to position i, then Alice can completely eliminate Eve's advantage by assigning probability 1 to position i. But then Eve's best response would be assign probability 1 to position i. So Eve is not in equilibrium. □

It is useful to quantify Eve's advantage from looking at one position and observing the bias. The following two definitions facilitate such quantification.

Definition 1 (Eve's local advantage). *Eve's* local advantage *at position i is* $a(i) \cdot \tilde{f}(i)$.

Definition 2 (Eve's total advantage). *Eve's* total advantage *is the weighted sum over all her local advantages at positions* $0, \ldots, n-1$, *i. e.,* $\sum_{i=0}^{n-1} \left(e(i)a(i)\tilde{f}(i)\right)$.

Observe that from Theorem 1, Eve's expected game payoff is exactly twice her total advantage. Hence we may consider total advantage as a quantity of primary interest. Eve's primary objective is to increase her total advantage, while Alice's primary objective is to reduce it. Our next lemma characterizes the structure of possible equilibria in relation to Eve's local and total advantages.

Lemma 2 (Uniform local advantage condition). *A necessary condition for any equilibrium is that Eve's local advantage is uniform over $i = 0, \ldots, n-1$.*

Proof. Suppose Eve's local advantage is not uniform. Then there is at least one position i where her local advantage is not as high as it is at some other position j. I. e., $a(i) \cdot \tilde{f}(i) < a(j) \cdot \tilde{f}(j)$. Eve can then strictly increase her total advantage by setting $e(j) = e(j) + e(i)$ and then setting $e(i) = 0$. (The resulting difference in her total advantage will be $e(i)(a(j) \cdot \tilde{f}(j) - a(i) \cdot \tilde{f}(i))$, which is positive.) So the situation is not an equilibrium. □

This condition can actually be fulfilled, as shown in the next lemma.

Lemma 3 (Existence of Alice's unique strategy). *In any equilibrium, Alice's strategy to embed one bit is*

$$a(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}. \tag{17}$$

Proof. We start with the condition from Lemma 2,

$$a(i) \cdot \tilde{f}(i) = a(j) \cdot \tilde{f}(j) \quad \forall\, i \neq j. \tag{18}$$

This implies that there is a constant C with $a(i) \cdot \tilde{f}(i) = C$ for each i, and hence $a(i) = \frac{C}{\tilde{f}(i)}$ for some C.

Now by the probability axiom,

$$\sum_{i=0}^{n-1} a(i) = 1, \tag{19}$$

so that $\sum_{i=0}^{n} \frac{C}{\tilde{f}(i)} = 1$, and hence $C = \frac{1}{\sum_{i=0}^{n} \tilde{f}(i)}$. It follows that $a(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}$. I. e. the two constraints (18) and (19) completely determine $a(i)$. □

Lemma 4 (Game outcome in equilibrium). *The game's outcome for (Eve, Alice) in equilibrium is*

$$\left(\frac{2}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \frac{-2}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}} \right). \tag{20}$$

Proof. Alice's strategy fixes Eve's total advantage, which in turn fixes Eve's payoff. As Alice has only one candidate strategy in equilibrium, we know Eve's total advantage in equilibrium must be

$$\sum_{i=0}^{n-1} \left(e(i) a(i) \tilde{f}(i) \right) = \sum_{i=0}^{n-1} \left(e(i) \frac{\tilde{f}(i)}{\tilde{f}(i) \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}} \right) = \frac{1}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \tag{21}$$

hence Eve's payoff in equilibrium is $\frac{2}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}$ and the result follows. □

Turning now to Eve's strategy, we may construe her objective as preserving her total advantage.

Lemma 5 (Uniform weighted bias condition). *A necessary condition for any equilibrium is that $e(i) \cdot \tilde{f}(i)$ is uniform over $i = 0, \ldots, n-1$.*

Proof. Suppose Alice is playing her unique strategy in equilibrium from Lemma 3 and that (for the sake of contradiction) there exist $i \neq j$ with $e(i) \cdot \tilde{f}(i) < e(j) \cdot \tilde{f}(j)$. Then Alice can decrease Eve's total advantage by adopting a new strategy a^* with, $a^*(j) = 0$; $a^*(i) = a(i) + a(j)$; and $a^*(r) = a(r)$ for $r \neq i, j$.

The difference in Eve's total advantage is

$$\sum_{r=0}^{n-1} \left(e(r)a^*(r)\tilde{f}(r) \right) - \sum_{r=0}^{n-1} \left(e(r)a(r)\tilde{f}(r) \right) \tag{22}$$

$$= e(i)a^*(i)\tilde{f}(i) + w_j a^*(j)\tilde{f}(i) - (e(i)a(i)\tilde{f}(i) + e(j)a(j)\tilde{f}(i)) \tag{23}$$

$$= e(i)(a(i) + a(j))\tilde{f}(i) - (e(i)a(i)\tilde{f}(i) + e(j)a(j)\tilde{f}(j)) \tag{24}$$

$$= e(i)a(j)\tilde{f}(i) - e(j)a(j)\tilde{f}(i) \tag{25}$$

$$= a(j)(e(i)\tilde{f}(i) - e(j)\tilde{f}(j)) \tag{26}$$

$$< 0.$$

So Alice would prefer to change strategies, in violation of the equilibrium condition. □

Lemma 6 (Existence of Eve's unique strategy). *In any equilibrium for the one-bit case, Eve's probability $e(i)$ of looking at position i must be the same as Alice's probability of embedding at position i:*

$$e(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}. \tag{27}$$

Proof. The formula follows from the uniform weighted bias condition: $e(i) \cdot \tilde{f}(i) = e(j) \cdot \tilde{f}(j)$ for all $i \neq j$; and the probability constraint on Eve's mixed strategy: $\sum_{j=0}^{n-1} e(i) = 1$. The argument that these conditions uniquely determine a function is given in Lemma 3. □

Theorem 2 (Unique Nash equilibrium). *There is a unique Nash equilibrium for the one-bit game where Alice embeds in position i with probability*

$$a(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \tag{28}$$

and Eve observes position i with probability

$$e(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \tag{29}$$

and the expected payoff outcome for (Eve, Alice) is $\left(\frac{2}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, -\frac{2}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}} \right)$.

Proof. See Lemmas 3, 4, and 6. □

Hiding k Bits

Lemma 7 (Alice's k-bit strategy). *In any equilibrium, the projection of Alice's mixed strategy distribution onto singleton subsets satisfies*

$$a(i) = \frac{k}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}. \tag{30}$$

Proof. First, any equilibrium must satisfy the uniform advantage condition, as the logic from Lemma 2 applies also in the k-bit case. Thus we have

$$a(i) \cdot \tilde{f}(i) = a(j) \cdot \tilde{f}(j) \quad \forall\, i \neq j. \tag{31}$$

Since we also have

$$\sum_{i=0}^{n-1} a(i) = k, \tag{32}$$

the function $a(i)$ is completely determined as $a(i) = \frac{k}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}$. □

Lemma 8 (Eve's k-bit strategy). *In any equilibrium, Eve's mixed strategy distribution is*

$$e(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}. \tag{33}$$

Proof. Eve's strategy must satisfy the uniform weighted bias condition: $e(i) \cdot \tilde{f}(i)$ is uniform in i; as the logic from Lemma 5 still applies in the k-bit case. Since we also have $\sum_{i=0}^{n-1} e(i) = 1$, these two conditions imply $e(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}$. □

Theorem 3 (k-bit Nash equilibria). *There is a Nash equilibrium for the k-bit game where the projection of Alice's distribution onto singleton positions satisfies*

$$a(i) = \frac{k}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \tag{34}$$

and Eve observes position i with probability

$$e(i) = \frac{1}{\tilde{f}(i) \cdot \sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, \tag{35}$$

and the expected payoff outcome for (Eve, Alice) is $\left(\frac{2k}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}}, -\frac{2k}{\sum_{j=0}^{n-1} \frac{1}{\tilde{f}(j)}} \right)$.

The equilibrium is unique up to the projection of Alice's mixed strategy.

Proof. See Lemmas 7 and 8 for the strategies. For the payoffs, note that Eve's advantage in equilibrium is

$$\sum_{i=0}^{n-1}\left(e(i)a(i)\tilde{f}(i)\right)=\sum_{i=0}^{n-1}\left(e(i)\frac{k\tilde{f}(i)}{\tilde{f}(i)\sum_{j=0}^{n-1}\frac{1}{\tilde{f}(j)}}\right)=\frac{k}{\sum_{j=0}^{n-1}\frac{1}{\tilde{f}(j)}}, \tag{36}$$

so that Eve's payoff in equilibrium is $\frac{2k}{\sum_{j=0}^{n-1}\frac{1}{\tilde{f}(j)}}$.

□

The following two corollaries are easily observable.

Corollary 1. *Eve's mixed strategy in equilibrium is independent of the number of embedded bits.*

Corollary 2. *Eve's expected payoff in equilibrium increases linearly with the number of embedded bits.*

4 Discussion

4.1 Numerical Examples

Figures 3 and 4 display numerical examples of the equilibrium in our game, instantiated with the parameters $k = 1$ and $n = 100$. The red line shows the prediction function $f(i)$; note the right hand scale. The gray bars display Alice's and Eve's identical optimal strategies (left hand scale). In Figure 3, the parameter ε is set relatively high and the prediction function f is linear.

Figure 4 is more realistic. It shows a small ε and a non-linear prediction function f with the majority of positions being relatively well predictable, just like large homogeneous areas in natural images. Both figures show that the value of $a(0)$ is at its maximum. This illustrates again the advantage of content-adaptive embedding over random uniform embedding if the cover source produces heterogeneous covers. Nonetheless, the fact that $a(i) > 0$ for all i suggests that the steganographer should potentially use every available position and not only the least predictable ones, unlike what is seen in many practical schemes.

4.2 Adequacy of Eve's Constraints

We have motivated our game with practical content-adaptive steganography in heterogeneous covers. Its solution can guide the development of more secure embedding functions and detectors implementing the best response against known embedding strategies. Our results recover the conclusions of [11] for the imperfect steganography regime, namely that random uniform embedding is only optimal in homogeneous covers, and naive adaptive embedding (i. e., deterministically choosing the k least predictable symbols) is inferior to optimal adaptive embedding, the equilibrium strategy. The extension to n cover symbols presented here is an important step towards bringing more realism to the model.

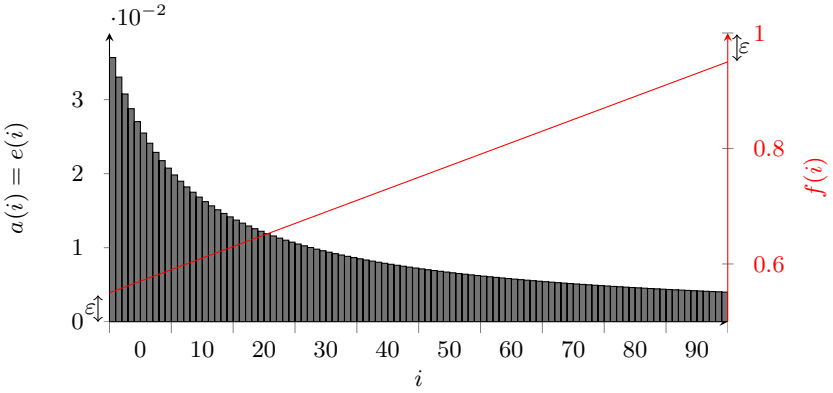


Fig. 3. Equilibrium strategies for $\varepsilon \gg 0$ and a linear function f

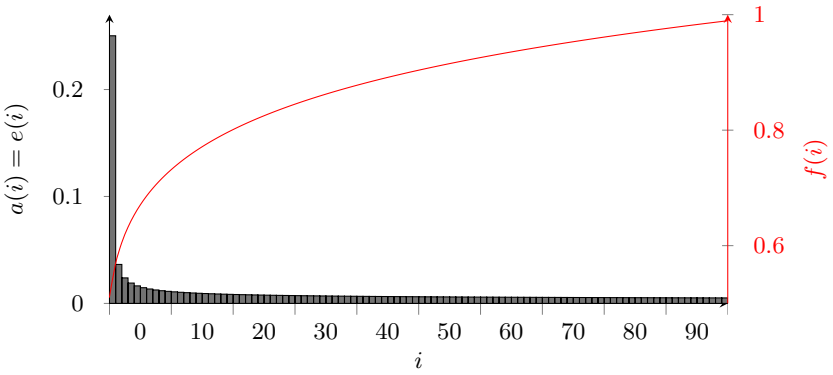


Fig. 4. Equilibrium strategies for $\varepsilon \approx 0$ and a non-linear function f

The biggest remaining obstacle (common to [11] and this paper) is the constraint that Eve can only look at one position at a time. In fact, all practical steganalysis methods are constrained to local – possibly suboptimal – decisions, but their output is an *aggregation* of local decisions for all positions in the observed sequence. Most known steganalysis methods come to a final decision using linear aggregation rules of the form,

$$D(\boldsymbol{x}) = \begin{cases} \text{cover} & \text{for} \quad \sum_0^{n-1} w_i \cdot d(x_i) > \tau \\ \text{stego} & \text{else,} \end{cases} \tag{37}$$

where $d : \mathcal{C} \to \{0, 1\}$ is the local decision function, $\boldsymbol{w}$ is a vector of weights, and τ a decision threshold to adjust the tradeoff between false negatives and false positives. Unfortunately, even a simple game defining Eve's action space as $(\boldsymbol{w}, \tau)$ deprives a straightforward analysis. Our theorems do not generalize to this case because they are based on Eve's advantage, which corresponds to the differences in expected values between cover and stego. This translates to the

difference in means of the distribution of the weighted sum of local decisions, but Eve's error rates depend also on higher moments of this distribution. More precisely, they depend on the quantile functions of the distribution for covers and stego objects, respectively. If this problem is solved, we can return to the standard model of steganographic communication as the amount of permissible side information does not need to be artificially constrained.

4.3 Alternative Interpretation

Although we motivated this game with optimal content-adaptive steganography, the underlying information hiding game is general enough to lend itself to alternative interpretations in the broader field of information security. One application is keeping secrets in an organization. Suppose Alice leads a big organization which contains a secret binary state that is extremely sensitive. Think of innovative companies ("Will the new device be a phone or not?"), central banks ("Will interest rates change or not?"), or governments ("Will they really respond to a cyber-attack with conventional warfare?"). In all these cases, Eve, an outside observer, should not be able to distinguish both possible states. However, Alice needs a team of size k to work on projects where knowledge of the state is essential. Her n staff members differ (function f) in their ability to decouple their observable behavior from their knowledge of the state, and Eve has resources to 'probe' (observe, eavesdrop, bribe, ...) exactly one staff member. Disregarding other constraints on team building, the solution to our game tells Alice how to compose her team with minimal risk of leaking the secret state.

4.4 Relation to Adversarial Classification

Our work can be seen as an example for adversarial classification, a term to the best of our knowledge coined by Dalvi et al. [3], who challenge the common assumption in data mining that the data generating process is independent of the data miner's activity. Instead, the authors study instances where the data is actively manipulated by an adversary seeking to increase the error rate in classification problems.

Like our steganographer, an adversary in their model actively manipulates data generated by nature, and a binary classifier tries to distinguish altered from unaltered objects, similar to our steganalyst. Their payoff structure is more complicated, including costs for altering and measuring features, respectively. These costs are offset by utility from successful, respectively erroneous, classifications.

The original work on adversarial classification is presented in a spam detection scenario, where spammers try to "wear out" a Bayes classifier. Nevertheless, the framework is presented in general terms, also suggesting other domains and tasks; but interestingly not steganography.

With theoretical underpinnings in feature-based machine learning theory, adversarial classification may also have the potential to deliver new insights for learning-based universal steganalysis as well as steganographic algorithms leveraging distance metrics in high-dimensional feature space [10].

5 Concluding Remarks

In this paper, we have formulated the problem of hiding multiple bits in a heterogeneous cover sequence as a stochastic two-player zero-sum game between steganographer and steganalyst. The steganographer chooses the embedding positions and the steganalyst applies a local decision rule involving side information to exactly one position. Theorem 3 states the main result: the game has a unique mixed strategy Nash equilibrium. All relevant properties to implement the equilibrium strategies can be efficiently computed from the function describing the heterogeneity in the predictability of cover bits. Corollary 1 stipulates that the steganalyst's equilibrium strategy does not depend on the number of embedded bits. This is a handy property for the construction of detectors, where no knowledge of the hidden message length must be assumed. Corollary 2 states that if the detector follows the equilibrium strategy, its success rate increases linearly with the number of embedded bits. This deviates from the square root law of steganographic capacity, which predicts asymptotically quadratic advantage even for homogeneous covers [9]. The reason for this difference is that our detector is constrained to a locally optimal decision rule.

While local decisions seem to be a good approximation of what is implemented in current steganalysis methods, other simplifications in our model may limit its validity. Most importantly, the constraint on access to side information of one symbol per cover is restrictive. Also giving the steganalyst perfect knowledge of the local predictability appears somewhat unrealistic. Practical content-adaptive embedding functions use different approximations of predictability and, depending on embedding operation and message length, the steganalyst can often recover this proxy pretty well. To account for the remaining uncertainty, future extensions of our game could equip the steganalyst with a noisy version of the true predictability profile. This can be done either by adding an independent random error term or, more realistically, by conditioning the error on the choice of embedding positions. Finally, the impact of the assumption of independent cover symbols needs to be evaluated. It remains to be seen if the useful properties established above can be maintained in a generalized game.

Acknowledgements. This research has received funding from the German national science foundation (DFG) under grant "Sichere adaptive Steganographie". Part of this work has been carried out during Benjamin Johnson's research visit to Germany, which was additionally supported by the University of Münster and the Münster School of Business and Economics, and by the National Science Foundation under ITR award CCF-0424422 (TRUST).

References

1. Böhme, R.: Advanced Statistical Steganalysis. Springer (2010)
2. Cachin, C.: An Information-Theoretic Model for Steganography. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 306–318. Springer, Heidelberg (1998)

3. Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D.: Adversarial classification. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 99–108. ACM, New York (2004)
4. Filler, T., Judas, J., Fridrich, J.J.: Minimizing Embedding Impact in Steganography Using Trellis-coded Quantization. In: Memon, N.D., Dittmann, J., Alattar, A.M., Delp, E.J. (eds.) Media Forensics and Security. SPIE Proceedings, vol. 7541, p. 754105. SPIE (2010)
5. Fridrich, J.: Minimizing the Embedding Impact in Steganography. In: Workshop on Multimedia and Security, pp. 2–10. ACM (2006)
6. Hopper, N.J., Langford, J., von Ahn, L.: Provably Secure Steganography. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 77–92. Springer, Heidelberg (2002)
7. Katzenbeisser, S., Petitcolas, F.A.P.: Defining Security in Steganographic Systems. In: Delp, E.J., Wong, P.W. (eds.) Security, Steganography and Watermarking of Multimedia Contents IV, San Jose, CA, vol. 4675, pp. 50–56 (2002)
8. Ker, A.D.: The Square Root Law in Stegosystems with Imperfect Information. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 145–160. Springer, Heidelberg (2010)
9. Ker, A.D., Pevný, T., Kodovský, J., Fridrich, J.: The Square Root Law of Steganographic Capacity. In: MM&Sec 2008: Proceedings of the 10th ACM Workshop on Multimedia and Security, pp. 107–116. ACM, New York (2008)
10. Pevný, T., Filler, T., Bas, P.: Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. In: Böhme, R., Fong, P.W.L., Safavi-Naini, R. (eds.) IH 2010. LNCS, vol. 6387, pp. 161–177. Springer, Heidelberg (2010)
11. Schöttle, P., Böhme, R.: A Game-Theoretic Approach to Content-Adaptive Steganography. In: Ghosal, D., Kirchner, M. (eds.) Information Hiding. LNCS, Springer (to appear, 2012)
12. Wang, Y., Moulin, P.: Perfectly Secure Steganography: Capacity, Error Exponents, and Code Constructions. IEEE Transactions on Information Theory 54(6), 2706–2722 (2008)

Socio-Rational Secret Sharing as a New Direction in Rational Cryptography

Mehrdad Nojoumian[1,⋆] and Douglas R. Stinson[2,⋆⋆]

[1] Department of Computer Science,
Southern Illinois University, Carbondale, Illinois, USA
nojoumian@cs.siu.edu
[2] David R. Cheriton School of Computer Science,
University of Waterloo, Waterloo, Ontario, Canada
dstinson@math.uwaterloo.ca

Abstract. Rational secret sharing was proposed by Halpern and Teague in [8]. The authors show that, in a setting with rational players, secret sharing and multiparty computation are only possible if the actual secret reconstruction round remains unknown to the players. All the subsequent works use a similar approach with different assumptions.

We change the direction by bridging cryptography, game theory, and reputation systems, and propose a "social model" for repeated rational secret sharing. We provide a novel scheme, named *socio-rational secret sharing*, in which players are invited to each game based on their reputations in the community. The players run secret sharing protocols while founding and sustaining a public trust network. As a result, new concepts such as a *rational foresighted player*, *social game*, and *social Nash equilibrium* are introduced.

To motivate our approach, consider a repeated secret sharing game such as "secure auctions", where the auctioneers receive sealed-bids from the bidders to compute the auction outcome without revealing the losing bids. If we assume each party has a reputation value, we can then penalize (or reward) the players who are selfish (or unselfish) from game to game. This social reinforcement stimulates the players to be cooperative.

Keywords: cryptography, game theory, reputation systems.

1 Introduction

The classical (t, n)-*secret sharing scheme* was proposed in [24,3], where a *dealer* distributes shares of a secret α among n players $P_1, \ldots, P_n$ for a subsequent secret recovery. In a Shamir secret sharing [24], the dealer first generates a random polynomial $f(x) \in \mathbb{Z}_q[x]$ of degree $t-1$ such that $f(0) = \alpha$ is the secret. He then sends shares $f(i)$ to player P_i for $1 \leq i \leq n$. As a result, any group of t or more players can reconstruct the secret by Lagrange interpolation whereas any group

⋆ Research supported by NSERC CGS - MSFSS Supplements Program.
⋆⋆ Research supported by NSERC Discovery Grant 203114-12.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 18–37, 2012.

of size less than t cannot gain any information about the secret. The standard assumption in traditional secret sharing is that each player is either *honest* (i.e., he follows protocols) or *malicious* (i.e., he deviates from protocols) where (1) at least t honest parties cooperate in order to recover the secret, and (2) the total number of malicious players is less than t.

A new research direction was initiated by Halpern and Teague [8] in the area of secret sharing and multiparty computation in a game-theoretic setting. In this new scheme, players are *rational* rather than being honest or malicious. This means each player selects his action (i.e., revealing his share or not revealing it) based on the utility that he can gain. As illustrated by the authors, classical secret sharing fails in this setting due to the failure of the secret reconstruction round. We should highlight that, in the context of rational secret sharing, "deviation" means that a player has not revealed his share during the reconstruction phase. Sending incorrect shares is another issue which can be prevented by having the dealer sign the shares. For a simple example of such an authentication method, see [13]. We now provide a high-level description of the problem.

If players are primarily incentivized to learn the secret, and secondly, they prefer that fewer of the other parties learn it, then it is not reasonable for each player to reveal his share in the "recovery phase". For instance, suppose players P_1, P_2, and P_3 receive shares $6, 11$, and 18 from a dealer respectively, where $f(x) = 3 + 2x + x^2 \in \mathbb{Z}_{19}[x]$ is the secret sharing polynomial. If only two players reveal their shares in the recovery phase, then the third selfish player (who has not revealed his share) can reconstruct the secret using two revealed shares and his own private share. Obviously, the other two cooperative players who have revealed their shares do not learn the secret. This justifies why the players do not reveal their shares in a rational setting, i.e., each player waits to receive shares of the other parties (see [5,11] for an overview in this direction).

To generalize this, consider the following scenario for a player P_j where the degree of the secret sharing polynomial is $t-1$. If P_i (for i less than $t-1$ or i more than $t-1$) reveal their shares, nothing changes whether P_j reveals his share or not. In the former case, no one learns the secret. In the latter case, everyone learns the secret. On the other hand, if exactly $t-1$ players P_i reveal their shares, then P_j can not only learn the secret with his own private share (i.e., t shares are sufficient to use Lagrange interpolation) but also can prevent the other players from learning the secret by not revealing his share, i.e., achieving the second preference of a self-interested player in rational secret sharing. In other words, for each P_i, revealing the share is *weakly dominated* by not revealing the share. As a result, no one reveals his share and the secret is never reconstructed.

We briefly introduce the notion of *social secret sharing* [21,22] in which players are either honest or malicious. In this protocol, weights of the players, i.e., the number of shares each player can hold, are periodically updated such that the players who cooperate receive more shares than those who defect. Although this scheme addresses a different issue compared to the secret recovery problem in a rational setting, we use its trust function in order to construct a new solution concept in rational cryptography.

1.1 Our Solution in Nutshell

In our "socio-rational" setting, the players are "selfish" similar to standard rational secret sharing. In addition, they have "concerns" about future gain or loss since our secret sharing game is repeated an unknown number of times. We term this new type of the player, a *rational foresighted player*. In the proposed scheme, each player has a reputation value which is updated according to his behavior each time the game is played. The initial reputation value is zero and its computation is public. For instance, if a player cooperates (he reveals his share), his trust value is increased, otherwise, it is decreased. A long-term utility (used by each player for action selection) and an actual utility (used for the real payment at the end of each game) are computed based on the following parameters:

1. Estimation of future gain or loss due to trust adjustment (virtual utility).
2. Learning the secret at the current time (real utility).
3. The number of other players learning the secret at the moment (real utility).

All these factors are used by each player to estimate his long-term utility and consequently to select his action, whereas only the last two items are used to compute the real payment at the end of each game. To estimate future impact, the following scenario is considered: whenever a player cooperates (or defects), we assume he can potentially gain (or lose) some extra units of utility, i.e., he has a greater (or lesser) chance to be "invited" to the future games and consequently he gains (or loses) more utility. In other words, if the reputation of P_i is decreased, he will have less chance to be invited to the future secret sharing games. Otherwise, P_i is going to be invited to more secret sharing games. To realize this scenario, in each game, the dealer selects the players based on their reputations, e.g., 50% from *reputable* players, 30% from *newcomers*, and 20% from *non-reputable* parties, where the number of players in each category varies.

This gain or loss is "virtual" at the current time but will be "realized" in the future. As an example of "future impact", consider the following statements, where $\mathrm{U} \gg \mathrm{u}$ and $\mathrm{V} \gg \mathrm{v}$:

1. As a consumer, if you buy something today (*cooperate*: lose \$u), you receive a significant discount from the producer (*rewarded* \$U) on your next purchase.
2. As a producer, if you use low-grade materials to save money (*defect*: gain \$v), you lose many of your consumers (*penalized* \$V) in the coming years.

In other words, if we construct a socio-rational model where the players can gain (or lose) more utility in the future games than the current game, depending on their behavior, we can then incentivize them to be foresighted and cooperative.

1.2 Our Motivation

In *secure multiparty computation* [7,2,4], various players cooperate to jointly compute a function based on the private data they each provide. As stated in the literature, secret sharing is a fundamental method that is used by the players

to inject their private data into a multiparty computation protocol. At the end of a multiparty computation protocol, each player has a share of the function value. Therefore, they can collaborate to reveal this value to everyone.

We refer to *sealed-bid auctions* [9] as an application of multiparty computation. In a secure auction, auctioneers receive many sealed-bids from bidders and the goal is to compute the auction outcome (i.e., the winner and selling price) without revealing the losing bids. The main reason for using sealed-bids is the fact that, if bids are not kept private, they can be used in the future auctions and negotiations by different parties, say auctioneers, to maximize their revenues, or competitors, to win a later auction. To motivate our concept of "socio-rational secret sharing", consider the following repeated game, as shown in Figure 1:

1. The bidders select a subset of auctioneers based on a non-uniform probability distribution over the auctioneers' types, i.e., reputable auctioneers have a greater chance to be selected.
2. Each bidder then acts as an independent dealer to distribute the shares of his sealed-bid among the selected auctioneers.
3. Subsequently, the auctioneers compute the selling price and determine the winner by using a multiparty computation protocol.
4. In the last phase of the multiparty computation, the auctioneers reconstruct the selling price α and report it to the seller.

In this setting, only the auctioneers who have learned and reported α to the seller, are each paid $\$\Omega$, i.e., there exists a "competition" for learning the secret. In addition, $\$\Omega$ are divided among the auctioneers who have learned the secret; each of them can therefore earn more money if fewer of them learn α. If we repeat this game an unknown number of times and choose an appropriate invitation mechanism based on the players' reputation, we can incentivize the auctioneers to be cooperative, that is, they will reveal the shares of α in the recovery phase.

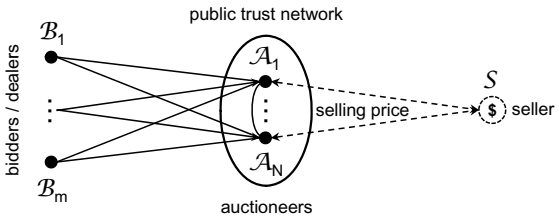


Fig. 1. Sealed-Bid Auction as a Repeated Secret Sharing Game

1.3 Our Contribution

We provide a new solution concept to the rational secret sharing problem by considering a social setting in which the players enter into a long-term interaction for executing an unknown number of independent secret sharing protocols.

In our model, a public trust network is constructed in order to incentivize the players to be cooperative. This incentive is sustained from game to game

since the players are motivated to enhance their reputations and consequently gain extra utility. In other words, they avoid a selfish behavior due to the social reinforcement of the trust network. Constructing a "social model" and inviting the players to a repeated game based on their "reputations" in the community, is a new contribution not only in rational cryptography but also in the existing game-theoretic solution concepts. We refer the reader to [17] for other discussions in this direction. Our scheme has the following desirable properties:

- It has a single secret recovery round, despite the existing solutions.
- It provides a game-theoretic solution that is always a Nash equilibrium.
- It is immune to rushing attack; it is not advantageous for players to wait.
- It prevents players from aborting the game; the case in some solutions.

The rest of this paper is organized as follows. Section 2 provides the relevant background. Section 3 reviews the literature of rational cryptography. Section 4 present our construction. Section 5 compares our solution with the existing schemes and techniques. Finally, Section 6 provides concluding remarks.

2 Preliminaries

2.1 Game-Theoretic Concepts

A *game* consists of a set of *players*, a set of *actions* and *strategies* (i.e., the way of choosing actions), and finally a *pay-off function* which is used by each participant to compute his utility. In *cooperative games*, players collaborate and split the total utility among themselves, i.e., cooperation is enforced by agreements. In *non-cooperative games*, players can not form agreements to coordinate their behavior, i.e., any cooperation must be self-enforcing. We now briefly review some game-theoretic definitions [8] for further technical discussions.

Definition 1. *Let* $\mathcal{A} \stackrel{\text{def}}{=} \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ *be an action profile for* n *players, where* $\mathcal{A}_i$ *denotes the set of possible actions of player* P_i*. A* game $\Gamma = (\mathcal{A}_i, u_i)$ *for* $1 \leq i \leq n$*, consists of* $\mathcal{A}_i$ *and a utility function* $u_i : \mathcal{A} \mapsto \mathbb{R}$ *for each player* P_i*. We refer to a vector of actions* $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{A}$ *as an* outcome of the game.

Definition 2. *The* utility function u_i *illustrates the preferences of player* P_i *over different outcomes. We say* P_i prefers *outcome* $\boldsymbol{a}$ *to* $\boldsymbol{a}'$ *iff* $u_i(\boldsymbol{a}) > u_i(\boldsymbol{a}')$*, and he* weakly prefers *outcome* $\boldsymbol{a}$ *to* $\boldsymbol{a}'$ *if* $u_i(\boldsymbol{a}) \geq u_i(\boldsymbol{a}')$*.*

In order to allow the players to follow randomized strategies (where the strategy is the way of choosing actions), we define σ_i as a probability distribution over $\mathcal{A}_i$ for a player P_i. This means that he samples $a_i \in \mathcal{A}_i$ according to σ_i. A strategy is said to be a *pure-strategy* if each σ_i assigns probability 1 to a certain action, otherwise, it is said to be a *mixed-strategy*. Let $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be the vector of players' strategies, and let $(\sigma_i', \boldsymbol{\sigma}_{-i}) \stackrel{\text{def}}{=} (\sigma_1, \ldots, \sigma_{i-1}, \sigma_i', \sigma_{i+1}, \ldots, \sigma_n)$, where P_i replaces σ_i by σ_i' and all the other players' strategies remain unchanged.

Therefore, $u_i(\boldsymbol{\sigma})$ denotes the expected utility of P_i under the strategy vector $\boldsymbol{\sigma}$. A player's goal is to maximize $u_i(\boldsymbol{\sigma})$. In the following definitions, one can substitute an action $a_i \in \mathcal{A}_i$ with its probability distribution $\sigma_i \in \mathcal{S}_i$ or vice versa.

Definition 3. *A vector of strategies $\boldsymbol{\sigma}$ is a* Nash equilibrium *if, for all i and any $\sigma_i' \neq \sigma_i$, it holds that $u_i(\sigma_i', \boldsymbol{\sigma}_{-i}) \leq u_i(\boldsymbol{\sigma})$. This means no one gains any advantage by deviating from the protocol as long as the others follow the protocol.*

Definition 4. *Let $\mathcal{S}_{-i} \stackrel{\text{def}}{=} \mathcal{S}_1 \times \cdots \times \mathcal{S}_{i-1} \times \mathcal{S}_{i+1} \times \cdots \times \mathcal{S}_n$. A strategy $\sigma_i \in \mathcal{S}_i$ (or an action) is* weakly dominated *by a strategy $\sigma_i' \in \mathcal{S}_i$ (or another action) with respect to $\mathcal{S}_{-i}$ if:*

1. *For all $\boldsymbol{\sigma}_{-i} \in \mathcal{S}_{-i}$, it holds that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) \leq u_i(\sigma_i', \boldsymbol{\sigma}_{-i})$.*
2. *There exists a $\boldsymbol{\sigma}_{-i} \in \mathcal{S}_{-i}$ such that $u_i(\sigma_i, \boldsymbol{\sigma}_{-i}) < u_i(\sigma_i', \boldsymbol{\sigma}_{-i})$.*

This means that P_i can never improve its utility by playing σ_i, and he can sometimes improve it by not playing σ_i. A strategy $\sigma_i \in \mathcal{S}_i$ is strictly dominated *if player P_i can always improve its utility by not playing σ_i.*

2.2 Rational Secret Sharing

We briefly review *rational secret sharing*, which was initiated by Halpern and Teague [8]. The scheme consists of a dealer D, who creates a secret sharing scheme with threshold t, and n players $P_1, \ldots, P_n$.

The protocol proceeds in a sequence of iterations where only one iteration is the "real" secret recovery phase (i.e., the last iteration) and the rest are just "fake" iterations for trapping selfish players. At the end of each iteration, the protocol either terminates (due to the observation of selfish behavior or cooperation for secret recovery) or it proceeds to the next iteration. Indeed, in any given round, players do not know whether the current iteration is the real recovery phase (where a player may gain more utility by being silent and not sending his share to the other players), or just a test round.

As we just stated, certain assumptions regarding the players' utility function are required for rational secret sharing to be achievable. Let $u_i(\boldsymbol{a})$ denotes the utility of P_i in a specific outcome $\boldsymbol{a}$ of the protocol. Suppose $l_i(\boldsymbol{a})$ is a bit defining whether P_i has learned the secret or not in $\boldsymbol{a}$. We then define $\delta(\boldsymbol{a}) = \sum_i l_i(\boldsymbol{a})$, which denotes the number of players who have learned the secret. The generalized assumptions of rational secret sharing are as follows:

- $l_i(\boldsymbol{a}) > l_i(\boldsymbol{a}') \Rightarrow u_i(\boldsymbol{a}) > u_i(\boldsymbol{a}')$.
- $l_i(\boldsymbol{a}) = l_i(\boldsymbol{a}')$ and $\delta(\boldsymbol{a}) < \delta(\boldsymbol{a}') \Rightarrow u_i(\boldsymbol{a}) > u_i(\boldsymbol{a}')$.

The first assumption means P_i prefers an outcome in which he learns the secret, i.e., since $l_i(\boldsymbol{a}) = 1$ and $l_i(\boldsymbol{a}') = 0$, he therefore prefers $\boldsymbol{a}$. The second assumption means P_i prefers an outcome in which the fewest number of other players learn the secret, given that P_i learns (or does not learn) the secret in both outcomes.

2.3 Social Secret Sharing

We now review *social secret sharing*, introduced by Nojoumian et al. [21], where the shares are allocated based on a player's reputation and the way she interacts with other parties. In other words, weights of players are adjusted such that participants who cooperate receive more shares compared to non-cooperative parties. This is similar to human social life where people share more secrets with whom they really trust and vice versa. In the context of social secret sharing, the players are either honest or malicious.

To quantify the reputation of each player in a social secret sharing scheme, the trust calculation method proposed in [20] is applied. In this approach, as shown in Table 1, three "types" of players (that is, $\mathcal{B}$: bad; $\mathcal{N}$: new; and $\mathcal{G}$: good) with six possible outcomes are defined, where α and β determine boundaries on the trust values used to define the different sets of players. This approach then applies functions $\mu(x)$ and $\mu'(x)$ to update the reputation $\mathcal{T}_i(p)$ of each P_i, as shown in Figure 2.

Table 1. Six Possible Actions for the Trust Management

Current Trust Value	Cooperation	Defection
$P_i \in \mathcal{B} \quad if \quad \mathcal{T}_i(p) \in [-1, \beta)$	Encourage	Penalize
$P_i \in \mathcal{N} \quad if \quad \mathcal{T}_i(p) \in [\beta, \alpha]$	Give a Chance	Take a Chance
$P_i \in \mathcal{G} \quad if \quad \mathcal{T}_i(p) \in (\alpha, +1]$	Reward	Discourage

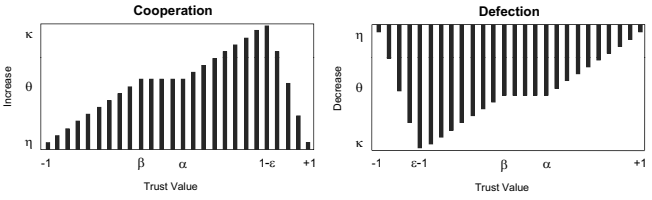


Fig. 2. Trust Adjustment by $\mu(x)$ and $\mu'(x)$ Functions

Let $\ell_i \in \{0, 1\}$ where $\ell_i = 1$ denotes that player P_i has cooperated in the current period and $\ell_i = 0$ denotes that he has defected. The proposed trust function is as follows, where $x = \mathcal{T}_i(p-1)$ (i.e., x is the previous trust value):

$$\ell_i = 1 \quad \Rightarrow \quad \mathcal{T}_i(p) = \mathcal{T}_i(p-1) + \mu(x), \text{ where}$$

$$\mu(x) = \begin{cases} \dfrac{\theta - \eta}{\beta + 1}(x+1) + \eta & P_i \in \mathcal{B} \\ \theta & P_i \in \mathcal{N} \\ \dfrac{\kappa - \theta}{1 - \epsilon - \alpha}(x - \alpha) + \theta & P_i \in \mathcal{G}, \mathcal{T}_i(p) \leq 1 - \epsilon \\ \dfrac{\kappa}{\epsilon}(1 - x - \epsilon) + \kappa & \mathcal{T}_i(p) > 1 - \epsilon \end{cases}$$

$$\ell_i = 0 \quad \Rightarrow \quad \mathcal{T}_i(p) = \mathcal{T}_i(p-1) - \mu'(x), \text{ where}$$

$$\mu'(x) = \begin{cases} \frac{\kappa}{\epsilon}(x+1) & \mathcal{T}_i(p) < \epsilon - 1 \\ \frac{\theta - \kappa}{\beta - \epsilon + 1}(x - \epsilon + 1) + \kappa & P_i \in \mathcal{B}, \mathcal{T}_i(p) \geq \epsilon - 1 \\ \theta & P_i \in \mathcal{N} \\ \frac{\eta - \theta}{1 - \alpha}(x - \alpha) + \theta & P_i \in \mathcal{G} \end{cases}$$

3 Literature Review

As we mentioned, the notion of *rational secret sharing* was introduced by Halpern and Teague [8]. Assuming the same game-theoretic model, Lysyanskaya and Triandopoulos [16] provide a solutions in a *mixed-behavior* setting in which players are either rational or malicious. Abraham et al. [1] define a notion of resistance to coalitions and present a *coalition-resistant* protocol. All these constructions use simultaneous channels (either a broadcast channel or secure private channels) that means each player must decide on the value he wants to broadcast before observing the values broadcasted by the other players; this is known as a *strategic game*.

The proposed protocols in [14,15,10] rely on *physical assumptions* such as secure envelopes and ballot boxes, which might be impossible or difficult to implement. In the same model, [19] provided a purely rational secret sharing scheme using a verifiable trusted channel. They showed that all the existing solutions not only rely on the players' rationality, but also on their beliefs. As a result, they cannot guarantee that all rational players learn the secret. For instance, suppose P_i believes that equilibrium (a, b) is played whereas P_j believes (a', b') is played, but the game leads to (a, b'), which may not be an equilibrium.

Kol and Naor [13] introduced an equilibrium notion, termed *strict Nash equilibrium*, in an information-theoretic secure setting. In a Nash equilibrium, no deviations are advantageous (i.e., there is no incentive to deviate). In its strict counterpart, all deviations are disadvantageous (i.e., there is an incentive not to deviate). They first considered both simultaneous and non-simultaneous broadcast channels and provided a new solution to avoid the simultaneous channel at the cost of increasing the round complexity.

Kol and Naor later [12] showed that all the existing computational-based protocols are susceptible to backward induction because of the cryptographic primitives used in the beginning of those protocols. That is, they can surely be broken after an exponential number of rounds. The authors then illustrate a new cryptographic coalition-resilient approach that is *immune to backward induction* by considering simultaneous as well as non-simultaneous broadcast channels.

The notion of *computational strict Nash equilibrium* was introduced in [6]. This dealer-free scheme can tolerate a coalition of size $t-1$ without using simultaneous channels. It can even be run over asynchronous point-to-point networks. Finally, it is efficient in terms of computation, share size, and round complexity.

Maleka et al. [18] presented *repeated rational secret sharing*, with the same approach proposed in [23], by considering two punishment strategies. In the former, each player reveals his share as long as the other players cooperate. As soon as the first defection is observed, the players do not reveal their shares in every subsequent game. In the latter, the players do not send their shares to the deviant for k subsequent games after observing the first defection. In the first scheme, each player not only punishes the deviant but also the other players including himself. In the second method, a player may deviate in an *expensive* secret recovery without having concern for k subsequent *cheap* recoveries.

4 Socio-Rational Secret Sharing

We first provide formal definitions of a *social game*, a *social Nash equilibrium*, and *socio-rational secret sharing*. In our model, each P_i has a public reputation value $\mathcal{T}_i$, where $\mathcal{T}_i(0) = 0$ and $-1 \leq \mathcal{T}_i(p) \leq +1$; $p = 0, 1, 2, \ldots$ denote the time periods of the games. The construction of this function is independent of our protocol, therefore, we use the existing function presented in Section 2.3. We assume each player's action $a_i \in \{\mathcal{C}, \mathcal{D}, \perp\}$, where $\mathcal{C}$ and $\mathcal{D}$ denote "cooperation" and "defection" respectively, and $\perp$ denotes P_i has not been chosen by the dealer to participate in the current game.

Definition 5. *In a society of size N, a* social game *$\Gamma = (\mathcal{A}_i, \mathcal{T}_i, u_i, u'_i)$, where $1 \leq i \leq N$, is repeatedly played an unbounded number of times among different subsets of players. Each P_i has a set of* actions *$\mathcal{A}_i$, a* reputation value *$\mathcal{T}_i$, a* long-term utility function *u_i, and an* actual utility function *u'_i. Let $\mathcal{A} \stackrel{\text{def}}{=} \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ be the action profile. In each game:*

- *A subset of $n \leq N$ players is chosen by the dealer for each new secret sharing game based on their reputation values $\mathcal{T}_i$, where more reputable players have a greater chance to be selected.*
- *Each P_i estimates his long-term utility by $u_i : \mathcal{A} \times \mathcal{T}_i \mapsto \mathbb{R}$ based on his gain in the current game and future games. Player P_i then selects his action a_i according to u_i.*
- *Let $\boldsymbol{a} = (a_1, \ldots, a_N) \in \mathcal{A}$ be the current game's outcome. The actual utility of each P_i is computed based on a function $u'_i : \mathcal{A} \mapsto \mathbb{R}$ at the end of the current game.*
- *Each player's reputation value $\mathcal{T}_i$ is publicly updated by a trust function based on each player's action in the current game, as shown in Section 2.3, except that $\mathcal{T}_i(p) = \mathcal{T}_i(p-1)$ if $a_i = \perp$.*

The *long-term utility function* u_i is used for action selection and the *actual utility function* u'_i is used to compute the "real gain" at the end of the current game.

Definition 6. *A vector of strategies $\boldsymbol{\sigma}$ is said to be a* social Nash equilibrium *in each game of a social game Γ if for all i and any $\sigma'_i \neq \sigma_i$ it holds that $u_i(\sigma'_i, \boldsymbol{\sigma}_{-i}) \leq u_i(\boldsymbol{\sigma})$. Accordingly, if $u_i(\sigma'_i, \boldsymbol{\sigma}_{-i}) < u_i(\boldsymbol{\sigma})$, it is said to be a* strict social Nash equilibrium. *That is, considering future games, a player cannot gain any benefit by deviating from the protocol in the current game.*

4.1 Utility Assumption

Let $u_i(\boldsymbol{a})$ denotes P_i's utility resulting from a list of players' actions $\boldsymbol{a}$ by considering future games, let $u'_i(\boldsymbol{a})$ denotes P_i's utility resulting from the current game, let $l_i(\boldsymbol{a}) \in \{0, 1\}$ denote if P_i has learned the secret during a given time period, and define $\delta(\boldsymbol{a}) = \sum_i l_i(\boldsymbol{a})$. Also, let $\mathcal{T}_i^{\boldsymbol{a}}(p)$ denote the reputation of P_i after outcome $\boldsymbol{a}$ in period p; each game of a social game is played in a single period. The generalized assumptions of socio-rational secret sharing are as follows:

A. $l_i(\boldsymbol{a}) = l_i(\boldsymbol{a}')$ and $\mathcal{T}_i^{\boldsymbol{a}}(p) > \mathcal{T}_i^{\boldsymbol{a}'}(p) \Rightarrow u_i(\boldsymbol{a}) > u_i(\boldsymbol{a}')$.
B. $l_i(\boldsymbol{a}) > l_i(\boldsymbol{a}') \Rightarrow u'_i(\boldsymbol{a}) > u'_i(\boldsymbol{a}')$.
C. $l_i(\boldsymbol{a}) = l_i(\boldsymbol{a}')$ and $\delta(\boldsymbol{a}) < \delta(\boldsymbol{a}') \Rightarrow u'_i(\boldsymbol{a}) > u'_i(\boldsymbol{a}')$.

The preference "*A*" illustrates that, whether player P_i learns the secret or not, P_i prefers to maintain a high reputation. The preferences "*B*" and "*C*" are the standard assumptions of rational secret sharing.

Definition 7. *In a social game, a* rational foresighted player *has prioritized assumptions: "A" (greediness) is strictly preferred to "B" and has an impact factor* ρ_1*, "B" (selfishness) is at least as good as "C" and has an impact factor* ρ_2*, and "C" (selfishness) has an impact factor* ρ_3*. We denote this using the notation* $A^{\rho_1} \succ B^{\rho_2} \succeq C^{\rho_3}$*, where* $\rho_1 \gg \rho_2 \geq \rho_3 \geq 1$.

The above definition reflects the fact that a rational foresighted player has a "long-term" vision and firstly prefers to achieve the highest level of trustworthiness. Only in this case, he will be involved in the future games and consequently gain more profits. He secondly prefers an outcome in which he learns the secret. Finally, he desires the fewest number of other players learn the secret. We next propose a long-term utility function that satisfies all three preferences.

4.2 Utility Computation

Our long-term utility function $u_i : \mathcal{A} \times \mathcal{T}_i \mapsto \mathbb{R}$ computes the utility that each player P_i potentially gains or loses by considering future games, based on assumptions "*A*", "*B*", "*C*", whereas the actual utility function $u'_i : \mathcal{A} \mapsto \mathbb{R}$ only computes the current gain or loss, based on assumptions "*B*" and "*C*".

Sample Function. We define two functions $\omega_i(\boldsymbol{a})$ and $\tau_i(\boldsymbol{a})$ for the n participating players of the current game:

$$\omega_i(\boldsymbol{a}) = \frac{3}{2 - \mathcal{T}_i^{\boldsymbol{a}}(p)} \tag{1}$$

$$\tau_i(\boldsymbol{a}) = \mathcal{T}_i^{\boldsymbol{a}}(p) - \mathcal{T}_i^{\boldsymbol{a}}(p-1). \tag{2}$$

Since $-1 \leq \mathcal{T}_i^{\boldsymbol{a}}(p) \leq +1$, then $+1 \leq \omega_i(\boldsymbol{a}) \leq +3$. Let $\Omega > 0$ be a "unit of utility", for instance, \$100. To satisfy our assumptions in Section 4.1, we define:

$$A : \frac{|\tau_i(\boldsymbol{a})|}{\tau_i(\boldsymbol{a})} \times \omega_i(\boldsymbol{a}) \times \Omega \quad \text{where} \quad \frac{|\tau_i(\boldsymbol{a})|}{\tau_i(\boldsymbol{a})} = \begin{cases} +1 \text{ if } a_i = \mathcal{C} \\ -1 \text{ if } a_i = \mathcal{D} \end{cases} \tag{3}$$

$$B : l_i(\boldsymbol{a}) \times \Omega \quad \text{where} \quad l_i(\boldsymbol{a}) \in \{0, 1\} \tag{4}$$

$$C : \frac{l_i(\boldsymbol{a})}{\delta(\boldsymbol{a}) + 1} \times \Omega \quad \text{where} \quad \delta(\boldsymbol{a}) = \sum_{i=1}^{N} l_i(\boldsymbol{a}). \tag{5}$$

- (3) will evaluate to $+\omega_i(\boldsymbol{a})\Omega$ if P_i cooperates and it will evaluate to $-\omega_i(\boldsymbol{a})\Omega$, otherwise. This means that P_i gains or loses at least 1Ω and at most 3Ω (depending on his reputation value, as reflected in ω_i) units of utility in the future games due to his current behavior.
- (4) illustrates that a player gains one unit of utility if he learns the secret in the current game and he loses this opportunity, otherwise.
- (5) results in "almost" one unit of utility being divided among all the players P_i who have learned the secret in the current game; to avoid a division by 0 when $\delta(\boldsymbol{a}) = 0$, we use $\delta(\boldsymbol{a}) + 1$ in the denominator.

We combine these three terms, weighted with their corresponding impact factors:

$$u'_i(\boldsymbol{a}) = \rho_2\Big(l_i(\boldsymbol{a}) \times \Omega\Big) + \rho_3\Big(\frac{l_i(\boldsymbol{a})}{\delta(\boldsymbol{a}) + 1} \times \Omega\Big), \text{ and} \tag{6}$$

$$\begin{aligned} u_i(\boldsymbol{a}) &= \rho_1\Big(\frac{|\tau_i(\boldsymbol{a})|}{\tau_i(\boldsymbol{a})} \times \omega_i(\boldsymbol{a}) \times \Omega\Big) + u'_i(\boldsymbol{a}) \\ &= \Omega \times \Big(\rho_1\Big(\frac{|\tau_i(\boldsymbol{a})|}{\tau_i(\boldsymbol{a})} \times \omega_i(\boldsymbol{a})\Big) + \rho_2\Big(l_i(\boldsymbol{a})\Big) + \rho_3\Big(\frac{l_i(\boldsymbol{a})}{\delta(\boldsymbol{a}) + 1}\Big)\Big). \end{aligned} \tag{7}$$

The function $u_i(\boldsymbol{a})$ shows that if player P_i, with preference factors $\rho_1 \gg \rho_2 \geq \rho_3 \geq 1$, defects (or cooperates), he may gain (or lose) $\rho_2\Omega + (\rho_3\Omega)/(\delta(\boldsymbol{a}) + 1)$ utility in the current game, but he will lose (or gain) "x" units of utility in the future games, where $\rho_1\Omega \leq x \leq 3\rho_1\Omega$. That is, future loss or gain is more important than the current loss or gain. We later show that the dealer gives a lesser (or a greater) chance of contribution to non-reputable (or reputable) players in the future games, that is, reputation remains with a player as a characteristic which continuously affects his utility.

4.3 Proposed Protocol

We now discuss our socio-rational secret sharing scheme, the details are presented in Figure 3. Suppose the public trust network has already been created. Assume we have a dealer who initiates a (t, n)-threshold secret sharing scheme. Also, assume all the players use a "pure-strategy". A *socio-rational secret sharing* game $\Gamma = (\mathcal{A}_i, \mathcal{T}_i, u_i, u'_i)$ is a social game that is played among rational foresighted players and it is based on the following elements:

Secret Sharing

1. Let ϕ be the current probability distribution over players' types $\mathcal{B}, \mathcal{N}, \mathcal{G}$, as defined in Section 2.3. The dealer D selects n out of N players, where $n \leq N$, based on this non-uniform probability distribution.
2. D then initiates a (t, n)-secret sharing scheme by selecting $f(x) \in \mathbb{Z}_q[x]$ of degree $t-1$, where $f(0) = \alpha$ is the secret. Subsequently, he sends shares $f(i)$ to P_i for the n chosen players, and leaves the scheme.

Secret Recovery

1. Each chosen player P_i computes his long-term utility function $u_i : \mathcal{A} \times \mathcal{T}_i \mapsto \mathbb{R}$, and then selects an action, i.e., revealing or not revealing his share $f(i)$.
2. If enough shares are revealed, the polynomial $f(x)$ is reconstructed through Lagrange interpolation and the secret $f(0) = \alpha$ is recovered.
3. Each chosen player P_i receives his utility $u_i' : \mathcal{A} \mapsto \mathbb{R}$ (i.e., the real payment) at the end of the reconstruction phase according to the outcome.
4. Finally, the reputation values $\mathcal{T}_i$ of all the chosen players are publicly updated according to each player's behavior and the trust function.

Fig. 3. Socio-Rational Secret Sharing Protocol

1. Set of possible actions $\mathcal{A}_i = \{\mathcal{C}, \mathcal{D}, \perp\}$, defined in Section 4.
2. Function $\mathcal{T}_i$, except that $\mathcal{T}_i(p) = \mathcal{T}_i(p-1)$ if $a_i = \perp$, defined in Section 2.3.
3. Long-term utility function $u_i : \mathcal{A} \times \mathcal{T}_i \mapsto \mathbb{R}$, defined in Section 4.2.
4. Actual utility function $u_i' : \mathcal{A} \mapsto \mathbb{R}$, defined in Section 4.2.

The *sharing phase* is similar to that of standard secret sharing. The only difference is the way that the dealer selects n out of N players for secret sharing. In other words, the dealer gives more chance to reputable players compared to unreliable parties. Although a natural approach is to invite only the reputable players, it is not fair if the dealer does not provide any opportunity for newcomers, or if he completely ignores the bad players. Once in a while, he should give a chance to the bad players so they can compensate for their past behavior. This is a realistic approach even in human society; it can be interpreted as a "forgiveness factor". The *secret recovery phase* is also similar to that of the standard secret sharing but with some extra components.

Note that since the players' reputations and the trust function are public information. Therefore, all computations associated with the reputation system can be done by any authority or a committee of the players. It is also worth mentioning that it is not required to consider unknown number of iterations for secret recovery, which is the case in all the existing rational secret sharing schemes. In fact, in a "socio-rational secret sharing" game, we have an unknown number of independent secret sharing games, whereas in "rational secret sharing", we only have one secret with an unknown number iterations for secret recovery.

Theorem 1. *In a $(2,2)$-socio-rational secret sharing, $\mathcal{C}$ strictly dominates $\mathcal{D}$, considering a long-term utility function , shown in Equation (7), which satisfies the preferences of rational foresighted players, shown in Definition 7. In other words, $\mathcal{D}$ is strictly dominated by $\mathcal{C}$. As a result, $(\mathcal{C},\mathcal{C})$ is a strict social Nash equilibrium that is a unique solution.*

Proof. We compute the utility of each outcome for P_i. Let P_j be the other player.

1. If both players cooperate, denoted by $(\mathcal{C},\mathcal{C})$, then τ_i is positive, $l_i = 1$ since P_i has learned the secret, and $\delta = 2$ because both players have learned the secret. We have:
$$\big(\tau_i > 0, l_i = 1, \delta = 2\big) \Rightarrow u_i^{(\mathcal{C},\mathcal{C})}(\boldsymbol{a}) = \Omega\Big(\rho_1\omega_i + \rho_2 + \frac{\rho_3}{3}\Big).$$
2. If only P_i cooperates, denoted by $(\mathcal{C},\mathcal{D})$, then τ_i is positive, $l_i = 0$ since P_i has not learned the secret, and $\delta = 1$ because only player P_j has learned the secret. We have:
$$\big(\tau_i > 0, l_i = 0, \delta = 1\big) \Rightarrow u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a}) = \Omega\Big(\rho_1\omega_i\Big).$$
3. If only P_j cooperates, denoted by $(\mathcal{D},\mathcal{C})$, then τ_i is negative, $l_i = 1$ since P_i has learned the secret, and $\delta = 1$ because only player P_i has learned the secret. We have:
$$\big(\tau_i < 0, l_i = 1, \delta = 1\big) \Rightarrow u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a}) = \Omega\Big(-\rho_1\omega_i + \rho_2 + \frac{\rho_3}{2}\Big).$$
4. If both players defect, denoted by $(\mathcal{D},\mathcal{D})$, then τ_i is negative, $l_i = 0$ since P_i has not learned the secret, and $\delta = 0$ because no one has learned the secret. We have:
$$\big(\tau_i < 0, l_i = 0, \delta = 0\big) \Rightarrow u_i^{(\mathcal{D},\mathcal{D})}(\boldsymbol{a}) = \Omega\Big(-\rho_1\omega_i\Big).$$

We ignore the common factor Ω. We know $1 \leq \omega_i(\boldsymbol{a}) \leq 3$ and $\rho_1 \gg \rho_2 \geq \rho_3 \geq 1$.

- First, we have:
$$u_i^{(\mathcal{C},\mathcal{C})}(\boldsymbol{a}) = \rho_1\omega_i + \rho_2 + \frac{\rho_3}{3} > \rho_1\omega_i = u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a}). \tag{8}$$
- Next, it is easy to see that
$$u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a}) = \rho_1\omega_i > -\rho_1\omega_i + \rho_2 + \frac{\rho_3}{2} = u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a}) \tag{9}$$
if and only if $2\rho_1\omega_i > \rho_2 + \frac{\rho_3}{2}$. We have:
$$\begin{aligned} 2\rho_1\omega_i &\geq 2\rho_1 \\ &> \rho_2 + \rho_3 \\ &> \rho_2 + \frac{\rho_3}{2}, \end{aligned}$$
so the desired conclusion follows.

- Finally,

$$u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a}) = -\rho_1\omega_i + \rho_2 + \frac{\rho_3}{2} > -\rho_1\omega_i = u_i^{(\mathcal{D},\mathcal{D})}(\boldsymbol{a}). \tag{10}$$

Therefore, we have the following payoff inequalities which proves the theorem:

$$\overbrace{u_i^{(\mathcal{C},\mathcal{C})}(\boldsymbol{a}) > u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a})}^{P_i \text{ cooperates}} > \overbrace{u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a}) > u_i^{(\mathcal{D},\mathcal{D})}(\boldsymbol{a})}^{P_i \text{ defects}}.$$

□

The interesting observation is the difference between the utilities $u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a})$ and $u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a})$. This means that it is better for player P_i to cooperate, even though he might not learn the secret and the other party might learn it. On the other hand, even if P_i learns the secret by deviating at a given period (using the share of the other party), he will gain less utility in the long-term. This is due to future gain or loss and the significance of being reputable, which is incorporated in our long-term utility function by considering an impact factor ρ_1. We should also note that, as ρ_1 is increased, the difference between $u_i^{(\mathcal{C},\mathcal{D})}(\boldsymbol{a})$ and $u_i^{(\mathcal{D},\mathcal{C})}(\boldsymbol{a})$ also increases, i.e., the enforcement for cooperation would be greater.

In a secret sharing scheme with selfish players, the outcome $(\mathcal{U}^-,\mathcal{U}^-)$ is a Nash equilibrium, as shown in Table 2, where $\mathcal{U}^+ > \mathcal{U} > \mathcal{U}^- > \mathcal{U}^{--}$. Rational secret sharing solves this problem by using a randomized mechanism, as presented in Section 2.2. The payoff matrix associated with socio-rational secret sharing is illustrated in Table 3. In this payoff matrix, the outcome $(\mathcal{U}^+,\mathcal{U}^+)$ is a *strict social Nash equilibrium*.

Table 2. $(2,2)$-SS with Selfish Players

P_1 \ P_2	$\mathcal{C}$ooperation	$\mathcal{D}$efection
$\mathcal{C}$ooperation	$\mathcal{U},\mathcal{U}$	$\mathcal{U}^{--},\mathcal{U}^{+}$
$\mathcal{D}$efection	$\mathcal{U}^{+},\mathcal{U}^{--}$	$\mathcal{U}^{-},\mathcal{U}^{-}$

Table 3. $(2,2)$-Socio-Rational SS

P_1 \ P_2	$\mathcal{C}$ooperation	$\mathcal{D}$efection
$\mathcal{C}$ooperation	$\mathcal{U}^{+},\mathcal{U}^{+}$	$\mathcal{U},\mathcal{U}^{-}$
$\mathcal{D}$efection	$\mathcal{U}^{-},\mathcal{U}$	$\mathcal{U}^{--},\mathcal{U}^{--}$

We should note that our socio-rational game is a non-cooperative game. In fact, cooperation is self-enforcing due to the importance of reputation as well as future concerns of a rational foresighted player. In a cooperative game, this enforcement is provided by a third party and players do not really compete. Moreover, this payoff matrix does not mean that the players never deviate. As an example, consider a scenario in which a player is involved in many independent social games. If he simultaneously receives many requests for secret recovery of various schemes, he will select the one in which he can gain more utility. This is discussed later, in Section 4.4. We now analyze our scheme for $n > 2$.

Theorem 2. *In a socio-rational secret sharing scheme with n participants and $t = 2$, $\mathcal{C}$ strictly dominates $\mathcal{D}$ for all P_i, assuming the preferences of rational foresighted parties. Consequently, the vector $\boldsymbol{a}^{\mathcal{C}} = (a_1^{\mathcal{C}}, \ldots, a_n^{\mathcal{C}})$ is a strict social Nash equilibrium that is a unique solution.*

Proof. Let $\mathcal{C}_i$ (or $\mathcal{D}_i$) denote that player P_i cooperates (or defects), and let $\mathcal{C}_{-i}$ (or $\mathcal{D}_{-i}$) denote that, excluding P_i, all the other players cooperate (or defect), and finally let $\mathcal{M}_{-i}$ denotes that, excluding P_i, some players cooperate and some of them defect. We compute the utility of each outcome based on Equation (7) for the least possible threshold $t = 2$ when $n > 2$.

1. If all the players cooperate, denoted by $(\mathcal{C}_i, \mathcal{C}_{-i})$, then τ_i is positive, $l_i = 1$ since player P_i has learned the secret, and $\delta = n$ because all the players have learned the secret. We have:
$$\big(\tau_i > 0, l_i = 1, \delta = n\big) \Rightarrow u_i^{(\mathcal{C}_i, \mathcal{C}_{-i})}(\boldsymbol{a}) = \Omega\Big(\rho_1\omega_i + \rho_2 + \frac{\rho_3}{n+1}\Big).$$
2. If player P_i cooperates but some of the other parties cooperate and some defect, denoted by $(\mathcal{C}_i, \mathcal{M}_{-i})$, then τ_i is positive, $l_i = 1$, and $\delta = n$ because all the players have learned the secret. We have:
$$\big(\tau_i > 0, l_i = 1, \delta = n\big) \Rightarrow u_i^{(\mathcal{C}_i, \mathcal{M}_{-i})}(\boldsymbol{a}) = \Omega\Big(\rho_1\omega_i + \rho_2 + \frac{\rho_3}{n+1}\Big).$$
3. If only P_i cooperates, denoted by $(\mathcal{C}_i, \mathcal{D}_{-i})$, then τ_i is positive, $l_i = 0$, and $\delta = n - 1$ since all the players, except P_i, have learned the secret. We have:
$$\big(\tau_i > 0, l_i = 0, \delta = n-1\big) \Rightarrow u_i^{(\mathcal{C}_i, \mathcal{D}_{-i})}(\boldsymbol{a}) = \Omega\Big(\rho_1\omega_i\Big).$$
4. If only P_i defects, denoted by $(\mathcal{D}_i, \mathcal{C}_{-i})$, then τ_i is negative, $l_i = 1$, and $\delta = n$ because all the players have learned the secret. We have:
$$\big(\tau_i < 0, l_i = 1, \delta = n\big) \Rightarrow u_i^{(\mathcal{D}_i, \mathcal{C}_{-i})}(\boldsymbol{a}) = \Omega\Big(-\rho_1\omega_i + \rho_2 + \frac{\rho_3}{n+1}\Big).$$
5. If P_i defects but some of the other parties cooperate and some defect, denoted by $(\mathcal{D}_i, \mathcal{M}_{-i})$, τ_i is negative, $l_i = 1$, and $\delta = n - 1$ if only one player reveals his share, or $\delta = n$ if at least two players reveal their shares. We have:
$$\big(\tau_i < 0, l_i = 1, \delta\big) \Rightarrow u_i^{(\mathcal{D}_i, \mathcal{M}_{-i})}(\boldsymbol{a}) = \Omega\Big(-\rho_1\omega_i + \rho_2 + \frac{\rho_3}{\delta+1}\Big), \delta \in \{n-1, n\}.$$
6. If all the players defect, denoted by $(\mathcal{D}_i, \mathcal{D}_{-i})$, then τ_i is negative, $l_i = 0$, and $\delta = 0$ because no one has learned the secret. We have:
$$\big(\tau_i < 0, l_i = 0, \delta = 0\big) \Rightarrow u_i^{(\mathcal{D}_i, \mathcal{D}_{-i})}(\boldsymbol{a}) = \Omega\Big(-\rho_1\omega_i\Big).$$

We now analyze these six scenarios:

- If player P_i cooperates (cases $1-3$), regardless of whether the other players cooperate or defect, then
$$u_i^{\mathcal{C}}(\boldsymbol{a}) \geq \rho_1\omega_i. \tag{11}$$
- If P_i defects (cases $4-6$), regardless of whether the other players cooperate or defect, then
$$u_i^{\mathcal{D}}(\boldsymbol{a}) \leq -\rho_1\omega_i + \rho_2 + \frac{\rho_3}{n}. \tag{12}$$

It is easy to prove that $\rho_1\omega_i > -\rho_1\omega_i + \rho_2 + \frac{\rho_3}{n}$; the proof is the same as the proof of (9) in Theorem 1. As a result, it is always in P_i's best interest to cooperate:

$$u_i^{\mathcal{C}}(\boldsymbol{a}) > u_i^{\mathcal{D}}(\boldsymbol{a}).$$

□

Remark 1. A similar analysis can be given for any threshold $t > 2$ when $n > 2$.

4.4 Expected Utility

We now illustrate how each P_i can compute his expected utility when he participates in different independent social games. Note that the *utility value* shows the connection between actions and their consequences for a player, whereas the *expected utility* of P_i is an estimation of gain or loss when he plays with P_j.

We initially show how to compute the expected utilities in a $(2,2)$-game for "cooperation" and "defection". An expected utility is computed as a linear combination of utility values and the probability of P_j's cooperation, where $\epsilon_j \in [0,1]$ denotes the probability that the opponent P_j may cooperate and $\mathcal{U}^+ > \mathcal{U} > \mathcal{U}^- > \mathcal{U}^{--}$ are the utility values from Table 3. We have:

$$\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}) = \epsilon_j\,\mathcal{U}^+ + (1-\epsilon_j)\,\mathcal{U} \tag{13}$$

$$\mathcal{EU}_i^{\mathcal{D}}(\boldsymbol{a}) = \epsilon_j\,\mathcal{U}^- + (1-\epsilon_j)\,\mathcal{U}^{--} \tag{14}$$

Theorem 3. *In a socio-rational secret sharing game with two players P_i and P_j, the expected utility of cooperation is greater than the expected utility of defection, i.e., $\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}) - \mathcal{EU}_i^{\mathcal{D}}(\boldsymbol{a}) > 0$, where ϵ_j is the probability of opponent's cooperation.*

Proof.

$$\begin{aligned}\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}) - \mathcal{EU}_i^{\mathcal{D}}(\boldsymbol{a}) &= \left(\epsilon_j\,\mathcal{U}^+ + (1-\epsilon_j)\,\mathcal{U}\right) - \left(\epsilon_j\,\mathcal{U}^- + (1-\epsilon_j)\,\mathcal{U}^{--}\right) \text{ by (13),(14)}\\ &= \epsilon_j\,(\mathcal{U}^+ - \mathcal{U}^-) + (1-\epsilon_j)\,(\mathcal{U} - \mathcal{U}^{--})\\ &> 0.\end{aligned}$$

□

We now consider the expected utilities in two independent $(2,2)$-games. Let us define $\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ij})$ and $\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ik})$ as the expected utilities of the two games, when player P_i cooperates with players P_j and P_k respectively.

Theorem 4. *Suppose P_i plays with P_j and P_k in two independent $(2,2)$-games. Player P_i then gains more utility if he collaborates with the most reputable player.*

Proof. Let P_j and P_k have different reputation values computed with the same trust function. For instance, $\epsilon_j > \epsilon_k$, which means P_j is more reputable than P_k. Suppose P_i receives the same unit of utility Ω in both games, and let $\boldsymbol{a}_{ij}, \boldsymbol{a}_{ik}$ be the outcomes of the two games. We have:

$$\begin{aligned}\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ij}) - \mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ik}) &= \big(\epsilon_j\,\mathcal{U}^+ + (1-\epsilon_j)\,\mathcal{U}\big) - \big(\epsilon_k\,\mathcal{U}^+ + (1-\epsilon_k)\,\mathcal{U}\big) \quad \text{by (13)}\\ &= \epsilon_j\,\mathcal{U}^+ - \epsilon_k\,\mathcal{U}^+ + (1-\epsilon_j)\,\mathcal{U} - (1-\epsilon_k)\,\mathcal{U}\\ &= (\epsilon_j - \epsilon_k)\,\mathcal{U}^+ + (\epsilon_j - \epsilon_k)\,\mathcal{U}\\ &> 0,\end{aligned}$$

since $\epsilon_j > \epsilon_k$. As a result, $\mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ij}) > \mathcal{EU}_i^{\mathcal{C}}(\boldsymbol{a}_{ik})$. This means that player P_i gains more utility if he collaborates with P_j rather than P_k. □

5 Comparison with Existing Techniques

Our contribution differs from *rational secret sharing* and *social secret sharing*, as shown in Figure 4. Our scheme is a repeated game that addresses the problem of secret recovery in the presence of rational foresighted parties, whereas:

- "rational secret sharing" is a one-time game with repeated rounds, and it deals with the problem of secret recovery of a secret in the presence of rational players, and
- "social secret sharing" defines how many shares each player can hold in a weighted secret sharing scheme with honest and malicious parties.

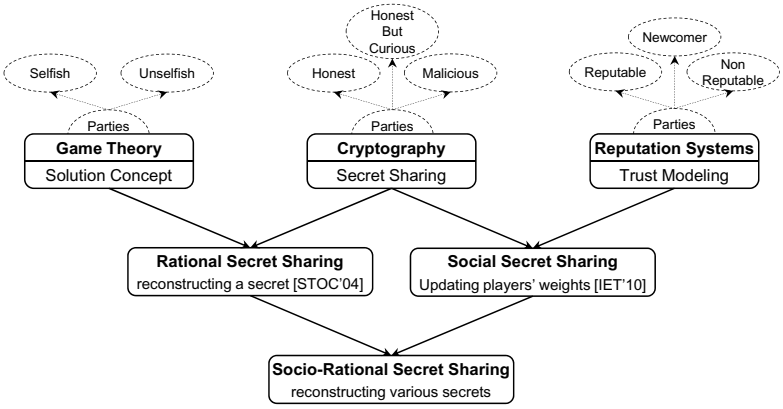


Fig. 4. Pedigree of the Socio-Rational Secret Sharing

Our contribution is also different from the *punishment strategy* used in the *repeated prisoners' dilemma* [23] where the players penalize potential deviants. As the authors have mentioned, the major point behind the repeated games is the fact that if each participant believes any deviation terminates the mutual cooperation (resulting in a subsequent loss that outweighs the short-term gain), he then prefers to cooperate. Our approach has the following advantages over the punishment strategy:

- In our model, a player is not just an abstract entity who selects actions. He also has a social characteristic reflected in his reputation that shows his trustworthiness. This attribute is solely determined by the player's actions.
- The punishment strategy is performed by selecting actions that are harmful for deviants whereas, in our model, punishment or reward (losing or gaining reputation and utility) is independent of action selection.
- Our approach avoids penalizing innocent players or the punisher himself. It also avoids being involved, to some extent, in a game with seriously selfish players who are not reputable (due to our "invitation approach").
- The punishment strategy does not consider that a game may have various levels of importance and utility weights when it is repeatedly played, e.g., a secret sharing scheme to launch a "missile" or to open a "safety box".
- The punishment strategy has a discrete penalizing approach whereas our construction has a continuous impact on the deviants. For example, it may take a long time for a player to regain lost reputation.
- Our proposed approach not only considers punishment and reward but also defines six different scenarios in order to fairly deal with various types of players, including good players, bad players, and newcomers.

Our contribution is also different from the constructions forming histories and beliefs such as *subgame perfect equilibrium* or *Bayesian equilibrium* [23]. In the former, players reassess their decisions based on the past *history*, i.e., a sequence of previous actions. In the latter, the game is designed to deal with the situations in which parties are not certain about the characteristics of each others. Therefore, they form *beliefs*, i.e., a probability distributions over actions, to anticipate any future behavior. Let P_i be a specific player, and let P_j for $1 \leq j \neq i \leq n$ denote any other player except P_i.

- In forming a belief about P_i's intentions, both parties contribute. That is, P_i is indirectly involved by his behavior, i.e., action selections, and the other players are directly involved by the methodology that they use in order to form the probability distribution over actions. A belief may or may not be common knowledge, meaning that various players may have different judgments and beliefs about P_i. On the other hand, the reputation of P_i in a trust network is solely determined by his behavior through a trust function, which is a commonly known function for reputation measurement. That is, the reputation is a direct reflection of P_i's attitude, and he knows the impact of his decision on the other players (i.e., whether he is known as a good player, a bad player, or a newcomer). He can also estimate how much extra utility he may gain or lose after his reputation's adjustment.
- Histories and beliefs are more general compared to the reputation system in a trust network. This means a belief as a probability distribution can be defined over any set of actions for any types of players. On the other hand, reputation is built over a specific set of actions, such as $\mathcal{C}$ooperation and $\mathcal{D}$efection, for specific types of players, such as good players, bad players, and newcomers. As a result, the reputation system is simpler and it is more suitable for cryptographic constructions.

- In the history and belief systems, measurements are "inside" the game-theoretic model whereas our reputation system isolates these computations from the game. For instance, two separate probability distributions can be defined over the players' types and actions by considering their past behavior. But our publicly known trust function combines these two measurements in a single reputation value outside of the game-theoretic model. In other words, the punishment or reward is embedded inside of our reputation system which continuously affects the players' utilities in the game-theoretic model, i.e., losing utility due to the reputation's decline or losing reputation and not being selected in the future secret sharing games.

6 Conclusion and Future Direction

This paper provides a multidisciplinary research connecting three major areas of computer science to propose a novel solution for a cryptographic primitive. We should note that having a trust network by considering long-term interactions can be seen as a new direction in game theory itself, specifically, the theoretical models used in social sciences such as economics and political science because elements in those frameworks are more close to human social behavior.

As our future work, we are interested to consider other complicated models. For instance, using *referral chain* in which two players who are interacting for the first time, can gain some information with respect to each other's reputation through other parties or common friends. We also would like to scrutinize the impact of a situation in which a player is involved in *various societies* while he is holding different reputation values associated with each one. It would be also interesting to construct a *hybrid model* in which both "reputation" and "belief" are considered. In this case, reputation can be seen as an estimation of the past behavior whereas belief can be viewed as an anticipation of the future activities.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.Y.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: 25th Annual ACM Symposium on Principles of Distributed Computing PODC, pp. 53–62 (2006)
2. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: 20th Annual ACM Symposium on Theory of Computing, STOC, pp. 1–10 (1988)
3. Blakley, G.: Safeguarding cryptographic keys. In: Proc. NCC, vol. 48, pp. 313–317. AFIPS Press (1979)
4. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols. In: 20th Annual ACM Symposium on Theory of Computing, STOC, pp. 11–19 (1988)
5. Dodis, Y., Halevi, S., Rabin, T.: A Cryptographic Solution to a Game Theoretic Problem. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 112–130. Springer, Heidelberg (2000)

6. Fuchsbauer, G., Katz, J., Naccache, D.: Efficient Rational Secret Sharing in Standard Communication Networks. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 419–436. Springer, Heidelberg (2010)
7. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: 19th Annual ACM Symposium on Theory of Computing, STOC, pp. 218–229 (1987)
8. Halpern, J.Y., Teague, V.: Rational secret sharing and multiparty computation: extended abstract. In: 36th Annual ACM Symposium on Theory of Computing, STOC, pp. 623–632 (2004)
9. Harkavy, M., Tygar, J.D., Kikuchi, H.: Electronic auctions with private bids. In: 3rd Conference on USENIX Workshop on Electronic Commerce, WOEC, pp. 61–74. USENIX Association (1998)
10. Izmalkov, S., Micali, S., Lepinski, M.: Rational secure computation and ideal mechanism design. In: 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 585–595 (2005)
11. Katz, J.: Bridging Game Theory and Cryptography: Recent Results and Future Directions. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 251–272. Springer, Heidelberg (2008)
12. Kol, G., Naor, M.: Cryptography and Game Theory: Designing Protocols for Exchanging Information. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 320–339. Springer, Heidelberg (2008)
13. Kol, G., Naor, M.: Games for exchanging information. In: 40th Annual ACM Symposium on Theory of Computing, STOC, pp. 423–432 (2008)
14. Lepinski, M., Micali, S., Peikert, C., Shelat, A.: Completely fair SFE and coalitionsafe cheap talk. In: 23th Annual ACM Symposium on Principles of Distributed Computing, PODC, pp. 1–10 (2004)
15. Lepinski, M., Micali, S., Shelat, A.: Collusion-free protocols. In: 37th Annual ACM Symposium on Theory of Computing, STOC, pp. 543–552 (2005)
16. Lysyanskaya, A., Triandopoulos, N.: Rationality and Adversarial Behavior in Multi-party Computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 180–197. Springer, Heidelberg (2006)
17. Mailath, G., Samuelson, L.: Repeated games and reputations: long-run relationships. Oxford University Press, USA (2006)
18. Maleka, S., Shareef, A., Rangan, C.P.: Rational Secret Sharing with Repeated Games. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 334–346. Springer, Heidelberg (2008)
19. Micali, S., shelat, a.: Purely Rational Secret Sharing (Extended Abstract). In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 54–71. Springer, Heidelberg (2009)
20. Nojoumian, M., Lethbridge, T.: A New Approach for the Trust Calculation in Social Networks. In: E-business and Telecommunication Networks: 3rd Int. Conf. on E-Business, ICE-B 2006, Best Papers, vol. 9, pp. 64–77. Springer (2008)
21. Nojoumian, M., Stinson, D., Grainger, M.: Unconditionally secure social secret sharing scheme. IET Information Security, Special Issue on Multi-Agent and Distributed Information Security 4(4), 202–211 (2010)
22. Nojoumian, M., Stinson, D.R.: Brief announcement: secret sharing based on the social behaviors of players. In: 29th ACM Symposium on Principles of Distributed Computing, PODC, pp. 239–240 (2010)
23. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT Press (1994)
24. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)

Audit Mechanisms for Provable Risk Management and Accountable Data Governance*

Jeremiah Blocki, Nicolas Christin, Anupam Datta, and Arunesh Sinha

Carnegie Mellon University, Pittsburgh, PA
{jblocki,nicolasc,danupam,aruneshs}@cmu.edu

Abstract. Organizations that collect and use large volumes of personal information are expected under the principle of *accountable data governance* to take measures to protect data subjects from risks that arise from inapproriate uses of this information. In this paper, we focus on a specific class of mechanisms—audits to identify policy violators coupled with punishments—that organizations such as hospitals, financial institutions, and Web services companies may adopt to protect data subjects from privacy and security risks stemming from inappropriate information use by insiders. We model the interaction between the organization (defender) and an insider (adversary) during the audit process as a repeated game. We then present an audit strategy for the defender. The strategy requires the defender to commit to its action and when paired with the adversary's best response to it, provably yields an *asymmetric subgame perfect equilibrium*. We then present two mechanisms for allocating the total audit budget for inspections across all games the organization plays with different insiders. The first mechanism allocates budget to maximize the utility of the organization. Observing that this mechanism protects the organization's interests but may not protect data subjects, we introduce an accountable data governance property, which requires the organization to conduct thorough audits and impose punishments on violators. The second mechanism we present achieves this property. We provide evidence that a number of parameters in the game model can be estimated from prior empirical studies and suggest specific studies that can help estimate other parameters. Finally, we use our model to predict observed practices in industry (e.g., differences in punishment rates of doctors and nurses for the same violation) and the effectiveness of policy interventions (e.g., data breach notification laws and government audits) in encouraging organizations to adopt accountable data governance practices.

* This work was partially supported by the U.S. Army Research Office contract "Perpetually Available and Secure Information Systems" (DAAD19-02-1-0389) to Carnegie Mellon CyLab, the NSF Science and Technology Center TRUST, the NSF CyberTrust grant "Privacy, Compliance and Information Risk in Complex Organizational Processes," the AFOSR MURI "Collaborative Policies and Assured Information Sharing," and HHS Grant no. HHS 90TR0003/01. Jeremiah Blocki was also partially supported by a NSF Graduate Fellowship. Arunesh Sinha was also partially supported by the CMU CIT Bertucci Fellowship. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 38–59, 2012.

1 Introduction

Organizations that collect and use large volumes of personal information are expected under the principle of *accountable data governance* to take measures to protect data subjects from risks that arise from these uses of information [1, 2]. In this paper, we focus on a specific class of mechanisms—audits to identify policy violators coupled with punishments—that organizations such as hospitals, financial institutions, and Web services companies may adopt to protect data subjects from privacy and security risks stemming from inappropriate information use by authorized insiders. Indeed, commercial audit tools are emerging to assist in the process of detecting inappropriate information use by insiders [3], and reports of privacy policy violations and associated sanctions are routinely reported in the healthcare sector [4–7].

A central challenge in this setting is the design of effective audit and punishment schemes. We assume that in each audit round audit logs are first analyzed using an automated tool that ranks actions by insiders as potential violations. Our focus is on the next step when a subset of these actions is inspected (because of budgetary constraints) to identify and punish policy violators. We seek to compute the inspection level and punishment level for an "effective" scheme.

The challenge in modeling the complex interaction between the auditor and audited agent includes making reasonable abstractions and assumptions. We model the interaction between an organization (the defender) and the insider (the adversary) as a repeated game with imperfect information (the defender does not observe the adversary's actions) and public signals (the outcome of the audit is public). The model captures a number of important economic considerations that influence the design of audit mechanisms. The game model (described in Section 3) replaces the byzantine adversary model in our previous work [8] with a *near-rational adversary model*. These adversaries act rationally with high probability and in a byzantine manner otherwise (similar to a *trembling hand* assumption [9]). Adversaries benefit from violations they commit (e.g., by selling personal data) and suffer due to punishments imposed for detected violations. The model generalizes from the situation in which the defender interacts with a single adversary to one where she interacts with multiple, non-colluding adversaries via a natural product game construction that we define. Each audit game is parametrized by a budget that the defender can use to conduct inspections.

We then present an audit strategy for the defender. This strategy when paired with the adversary's best response to it provably yields an *asymmetric approximate subgame perfect equilibrium* (Theorem 1). This equilibrium concept implies that the adversary does not gain at all from deviating from her best response strategy (see Section 4). We define this equilibrium concept by adapting the standard notion of approximate subgame perfect equilibrium, which has a symmetric flavor and permits both players to obtain small gains by unilaterally deviating from their equilibrium strategy. The symmetric equilibrium concept is unsuitable for our security application, where an adversary who deviates motivated by a small gain could cause a big loss for the organization. The defender's strategy involves committing to a level of inspection and punishment. The strategy has two desirable properties. First, the commitment results in a predictable equilibrium since the adversary plays her best response to the strategy. Second, the strategy is *deterrence dominant* over the set of maximum utility defender strategies that

result in a perfect public equilibrium, i.e., whenever such a strategy deters the adversary, so does our audit strategy (see Theorem 2 for the formal statement).

We design two mechanisms using which the defender can allocate her total audit budget across the different games to audit different insiders and types of potential violations. The first mechanism optimizes the defender's utility. Observing that this mechanism protects the organization's interests but may not protect data subjects, we introduce an accountable data governance property, which places an operational requirement on the organization to use a sufficiently effective log analysis tool and maintain sufficiently high inspection and punishment rates. The second mechanism allocates the total audit budget to achieve this property (see Section 5).

Finally, we demonstrate the usefulness of our model by predicting and explaining observed practices in industry (e.g., differences in punishment rates of doctors and nurses for the same violation) and analyzing the effectiveness of policy interventions (e.g., data breach notification laws and government audits) in encouraging organizations to adopt accountable data governance practices (see Section 6). We present comparisons to additional related work in Section 7 and conclusions and directions for future work in Section 8. The full version of this paper with proofs of theorems is available online at: https://www.cylab.cmu.edu/research/techreports/2012/tr_cylab12020.html.

2 Overview

In this section, we provide an overview of our model using a motivating scenario that will serve as a running example for this paper. Consider a "Hospital X" with employees in different roles (doctors, nurses). X conducts weekly audits to ensure that accesses to personal health records are legitimate. Given budget constraints, X cannot check every single access. The first step in the audit process is to analyze the access logs using an automated tool that ranks accesses as potential violations. Hospital X assesses the (monetary) impact of different types of violations and decides what subset to focus on by balancing the cost of audit and the expected impact ("risk") from policy violations. This type of audit mechanism is common in practice [10–13].

We provide a game model for this audit process. An employee ("adversary," $\mathcal{A}$) executes tasks, i.e., actions that are permitted as part of their job. We only consider tasks that can later be audited, e.g., through inspection of logs. For example, in X the tasks are accesses to health records. We can distinguish $\mathcal{A}$'s tasks between legitimate tasks and violations of a policy. Different *types of violations* may have different impact on the organization. We assume that there are K different types of violations that $\mathcal{A}$ can commit. Examples of violations of different types in Hospital X include inappropriate access to a celebrity's health record, or access to a health record leading to identity theft. $\mathcal{A}$ benefits by committing violations: the benefit is quantifiable using information from existing studies or by human judgment. For example, reports [14,15] indicate that on average the *personal benefit* of a hospital employee from selling a common person's health record is $\$50$. On the other hand, if $\mathcal{A}$ is caught committing a violation then she is punished according to the *punishment policy* used by $\mathcal{D}$. For example, employees could be terminated, as happened in similar recent incidents [6,7].

The organization $\mathcal{D}$ can classify each adversary's task by type. However, $\mathcal{D}$ cannot determine with certainty whether a particular task is legitimate or a violation without

investigating. Furthermore, $\mathcal{D}$ cannot inspect all of $\mathcal{A}$'s tasks due to *budgetary constraints*. As such, some violations may go undetected *internally*, but could be detected *externally*. Governmental audits, whistle-blowing, patient complaints [16, 17] are all examples of situations that could lead to external detection of violations. Externally detected violations usually cause more economic damage to the organization than internally caught violations. The 2011 Ponemon Institute report [18] states that patients whose privacy has been violated are more likely to leave (and possibly sue) a hospital if they discover the violation on their own than if the hospital detects the violation and proactively notifies the patient.

The economic impact of a violation is a combination of *direct and indirect costs*; direct costs include breach notification and remedial cost, and indirect costs include loss of customers and brand value. For example, the 2010 Ponemon Institute report [19] states that the average cost of privacy breach *per record* in health care is $\$301$ with indirect costs about two thirds of that amount. Of course, certain violations may result in much higher direct costs, e.g., $\$25,000$ per record (up to $\$250,000$ in total) in fines alone in the state of California [6]. These fines may incentivize organizations to adopt aggressive punishments policies. However, severe punishment policies create a hostile work environment resulting in economic losses for the organization due to low employee motivation and a failure to attract new talent [20].

The organization needs to balance auditing costs, potential economic damages due to violations and the economic impact of the punishment policy. The employees need to weigh their gain from violating policies against loss from getting caught by an audit and punished. The actions of one party impact the actions of the other party: if employees never violate, the organization does not need to audit; likewise, if the organization never audits, employees can violate policies in total impunity. Given this strategic interdependency, we model the auditing process as a *repeated game* between the organization and its employees, where the discrete rounds characterize audit cycles. The game is parameterized by quantifiable variables such as the personal benefit of employee, the cost of breach, and the cost of auditing, among others. The organization is engaged in multiple such games simultaneously with different employees and has to effectively allocate its total audit budget across the different games.

3 Audit Game Model

We begin by providing a high level view of the audit process, before describing the audit game in detail (Section 3). In practice, the organization is not playing a repeated audit game against a specific employee, but against all of its n employees at the same time. However, if we assume that 1) a given employee's actions for a type of task are independent of her actions for other types, and that 2) employees do not collude with other employees and act independently, we can decompose the overall game into $\mathsf{n}K$ independent *base* repeated games, that the organization plays in parallel. One base repeated game corresponds to a given type of access k by a given employee $\mathcal{A}$, and will be denoted by $\mathcal{G}_{\mathcal{A},k}$. Each game $\mathcal{G}_{\mathcal{A},k}$ is described using many parameters, e.g., *loss due to violations*, *personal benefit* for employee, etc. We abuse notation in using $\mathcal{G}_{\mathcal{A},k}$ to refer to a base repeated game of type k with any value of the parameters.

In our proposed audit process the organization follows the steps below in each audit cycle for every game $\mathcal{G}_{\mathcal{A},k}$. Assume the parameters of the game have been estimated and the equilibrium audit strategy computed for the first time auditing is performed.

before audit:
1. If any parameter changes go to step 2 else go to *audit*.
2. Estimate parameters. Compute equilibrium of $\mathcal{G}_{\mathcal{A},k}$.
audit:
3. Audit using actions of the computed equilibrium.

Note that the parameters of $\mathcal{G}_{\mathcal{A},k}$ may change for any given round of the game, resulting in a different game. However, neither $\mathcal{D}$ nor $\mathcal{A}$ knows when that will happen. As such, since the horizon of $\mathcal{G}_{\mathcal{A},k}$ with a fixed set of parameters is infinite, we can describe the interaction between the organization and its employees with an infinitely repeated game for the period in which the parameters are unchanged (see [9] for details). Thus, the game $\mathcal{G}_{\mathcal{A},k}$ is an infinitely repeated game of *imperfect information* since $\mathcal{A}$'s action is not directly observed. Instead, noisy information about the action, called a *public signal* is observed. The public signal here consists of a) the detected violations b) number of tasks by $\mathcal{A}$ and c) $\mathcal{D}$'s action. The K parallel games played between $\mathcal{A}$ and $\mathcal{D}$ can be composed in a natural manner into one repeated game (which we call $\mathcal{G}_{\mathcal{A}}$) by taking the product of action spaces and adding up utilities from the games.

Finally, analyzing data to detect changes of parameters may require the use of statistical methods [21], data mining and learning techniques. We do not delve into details of these methods as that is beyond the scope of this paper and estimating risk parameters has been studied extensively in many contexts [10–13,15]. Observe that change of parameters may change the equilibrium of the game, e.g., a lot of violations in quick succession by an employee (in spite of being inspected sufficiently) may result in the organization changing the personal benefit of the employee leading to more inspection.

Formal Description. In the remainder of this section, we focus on the base repeated games $\mathcal{G}_{\mathcal{A},k}$. We use the following notations in this paper:

- Vectors are represented with an arrow on top, e.g., $\vec{v}$ is a vector. The i^{th} component of a vector is given by $\vec{v}\,(i)$. $\vec{v} \leq \vec{a}$ means that both vectors have the same number of components and for any component i, $\vec{v}\,(i) \leq \vec{a}\,(i)$.
- Random variables are represented in boldface, e.g., $\mathbf{x}$ and $\mathbf{X}$ are random variables.
- $E(\mathbf{X})[q, r]$ denotes the expected value of random variable X, when particular parameters of the probability mass function of $\mathbf{X}$ are set to q and r.
- We will use a shorthand form by dropping $\mathcal{A}$, k and the vector notation, as we assume these are implicitly understood for the game $\mathcal{G}_{\mathcal{A},k}$, i.e., a quantity $\vec{x}_{\mathcal{A}}(k)$ will be simply denoted as x. We use this form whenever the context is restricted to game $\mathcal{G}_{\mathcal{A},k}$ only.

$\mathcal{G}_{\mathcal{A},k}$ is fully defined by the players, the time granularity at which the game is played, the actions the players can take, and the utility the players obtain as a result of the actions they take. We next discuss these different concepts in turn.

Players: The game $\mathcal{G}_{\mathcal{A},k}$ is played between the organization $\mathcal{D}$ and an adversary $\mathcal{A}$. For instance, the players are hospital X and a nurse in X.

Round of Play: In practice, audits for all employees and all types of access are performed together and usually periodically. Thus, we adopt a discrete-time model, where time points are associated with rounds. Each round of play corresponds to an audit cycle. We group together all of the $\mathcal{A}$'s actions (tasks of a given type) in a given round. All games $\mathcal{G}_{\mathcal{A},k}$ are synchronized, i.e., all rounds t in all games are played simultaneously.

Adversary Action Space: In each round, the adversary $\mathcal{A}$ chooses two quantities of type k: the number of tasks she performs, and the number of such tasks that are violations. If we denote by U_k the maximum number of type k tasks that any employee can perform, then $\mathcal{A}$'s entire action space for $\mathcal{G}_{\mathcal{A},k}$ is given by $A_k \times V_k$ with $A_k = \{u_k, \ldots, U_k\}$ ($u_k \leq U_k$) and $V_k = \{1, \ldots, U_k\}$. Let $\vec{a}^t_{\mathcal{A}}$ and $\vec{v}^t_{\mathcal{A}}$ be vectors of length K such that the components of vector $\vec{a}$ are the number of tasks of each type that $\mathcal{A}$ performs at time t, and the components of vector $\vec{v}$ are the number of violations of each type. Since violations are a subset of all tasks, we always have $\vec{v}^t_{\mathcal{A}} \leq \vec{a}^t_{\mathcal{A}}$. In a given audit cycle, $\mathcal{A}$'s action in the game $\mathcal{G}_{\mathcal{A},k}$ is defined by $\langle \vec{a}^t_{\mathcal{A}}(k), \vec{v}^t_{\mathcal{A}}(k) \rangle$, that is $\langle a^t, v^t \rangle$ in shorthand form, with $a^t \in A_k$ and $v^t \in V_k$.

Instead of being perfectly rational, we model $\mathcal{A}$ as playing with a *trembling hand* [9]. Whenever $\mathcal{A}$ chooses to commit v^t violations in as given round t, she does so with probability $1 - \epsilon_{th}$, but, with (small) probability ϵ_{th} she commits some other number of violations sampled from an unknown distribution D^t_0 over all possible violations. In other words, we allow $\mathcal{A}$ to act completely arbitrarily when she makes a mistake. For instance, a nurse in X may lose her laptop containing health records leading to a breach.

Defender Action Space: $\mathcal{D}$ also chooses two quantities of type k in each round: the number of inspections to perform, and the punishment to levy for each type-k violation detected. Let $\vec{s}^t_{\mathcal{A}}$ be the vector of length K such that components of vector $\vec{s}^t_{\mathcal{A}}$ are the number of inspections of each type that $\mathcal{D}$ performs in round t. The number of inspections that $\mathcal{D}$ can conduct is bounded by the number of tasks that $\mathcal{A}$ performs, and thus, $\vec{s}^t_{\mathcal{A}} \leq \vec{a}^t_{\mathcal{A}}$. $\mathcal{D}$ uses a log analysis tool $\mathcal{M}$ to sort accesses according to the probability of them being a violation. Then, $\mathcal{D}$ chooses the top $\vec{s}^t_{\mathcal{A}}(k) = s^t$ tasks from the sorted output of $\mathcal{M}$ to inspect in game $\mathcal{G}_{\mathcal{A},k}$. Inspection is assumed perfect, i.e., if a violation is inspected, it is detected. The number of inspections is bounded by budgetary constraints. Denoting the functions that outputs cost of inspection for each type of violation by $\vec{C}$, we have $\vec{C}(k)(\vec{s}^t_{\mathcal{A}}(k)) \leq \vec{b}^t_{\mathcal{A}}(k)$ where $\vec{b}^t_{\mathcal{A}}(k)$ defines a per-employee, per-type budget constraint. The budget allocation problem is an optimization problem depending on the audit strategy, which we discuss is Section 5.1.

$\mathcal{D}$ also chooses a punishment rate $\vec{P}^t_{\mathcal{A}}(k) = P^t$ (fine per violation of type k) in each round t to punish $\mathcal{A}$ if violations of type k are detected. P^t is bounded by a maximum punishment P_f corresponding to the employee being fired, and the game terminated.

Finally, $\mathcal{D}$'s choice of the inspection action can depend only on $\mathcal{A}$'s total number of tasks, since the number of violations is not observed. Thus, $\mathcal{D}$ can choose its strategy as a function from number of tasks to inspections and punishment even before $\mathcal{A}$ performs its action. In fact, we simulate $\mathcal{D}$ acting first and the actions are *observable* by requiring $\mathcal{D}$ to commit to a strategy and provide a proof of honoring the commitment. Specifically,

$\mathcal{D}$ computes its strategy, makes it public and provides a proof of following the strategy after auditing is done. The proof can be provided by maintaining an audit trail of the audit process itself.

Outcomes: We define the outcome of a single round of $\mathcal{G}_{\mathcal{A},k}$ as the number of violations detected in internal audit and the number of violations detected externally. We assume that there is a fixed exogenous probability p ($0 < p < 1$) of an internally undetected violation getting caught externally. Due to the probabilistic nature of all quantities, the outcome is a random variable. Let $\vec{\mathbf{O}}^t_{\mathcal{A}}$ be the vector of length K such that the $\vec{\mathbf{O}}^t_{\mathcal{A}}(k) = \mathbf{O}^t$ represents the outcome for the t^{th} round for the game $\mathcal{G}_{\mathcal{A},k}$. Then $\mathbf{O}^t$ is a tuple $\langle \mathbf{O}^t_{int}, \mathbf{O}^t_{ext} \rangle$ of violations caught internally and externally. As stated earlier, we assume the use of a log analysis tool $\mathcal{M}$ to rank the accesses with more likely violations being ranked higher. Then, the probability mass function for $\vec{\mathbf{O}}^t_{int}$ is a distribution parameterized by $\langle a^t, v^t \rangle$, s and $\mathcal{M}$. The baseline performance of $\mathcal{M}$ is when the s accesses to be inspected are chosen at random, resulting in a hyper-geometric distribution with mean $v^t\alpha^t$, where $\alpha^t = s^t/a^t$. We assume that the mean of the distribution is $\mu(\alpha^t)v^t\alpha^t$, where $\mu(\alpha^t)$ is a function dependent on α^t that measures the *performance* of $\mathcal{M}$ and $\forall \alpha^t \in [0,1]. \mu \geq \mu(\alpha^t) \geq 1$ for some constant μ (μ is overloaded here). Note that we must have $\mu(\alpha^t)\alpha^t \leq 1$, and further, we assume that $\mu(\alpha^t)$ is monotonically non-increasing in α^t. The probability mass function for $\mathbf{O}^t_{ext}$ conditioned on $\mathbf{O}^t_{int}$ is a binomial distribution parameterized by p.

Utility Functions: In a public signaling game like $\mathcal{G}_{\mathcal{A},k}$, the utilities of the players depend only on the public signal and their own action, while the strategies they choose depend on the history of public signals [22]. The utility of the repeated game is defined as a (delta-discounted) sum of the expected utilities received in each round, where the expectation is taken with respect to the distribution over histories. Let the discount factor for $\mathcal{D}$ be $\delta_{\mathcal{D}}$ and for any employee $\mathcal{A}$ be $\delta_{\mathcal{A}}$. We assume that $\mathcal{D}$ is patient, i.e., future rewards are almost as important as immediate rewards, and $\delta_{\mathcal{D}}$ is close to 1. $\mathcal{A}$ is less patient than $\mathcal{D}$ and hence $\delta_{\mathcal{A}} < \delta_{\mathcal{D}}$.

Defender utility function: $\mathcal{D}$'s utility in a round of the game $\mathcal{G}_{\mathcal{A},k}$ consists of the sum of the *cost of inspecting* $\mathcal{A}$'s actions, the monetary *loss from a high punishment rate* for $\mathcal{A}$, and *direct and indirect costs* of violations. As discussed before, inspection costs are given by $C(s^t)$ where $C = \vec{C}(k)$ is a function denoting the cost of inspecting type-k tasks. Similarly, the monetary loss from losing employee's productivity due to fear of punishment is given by $e(P^t)$, where $e = \vec{e}_A(k)$ is a function for type-k tasks. The functions in $\vec{C}$ and $\vec{e}$ must satisfy the following constraints: 1) they should be monotonically increasing in the argument and 2) $\vec{C}(k) \geq 0$, $\vec{e}_A(k) \geq 0$ for all k.

We characterize the effect of violations on the organization's indirect cost similarly to the reputation loss as in previous work [8]. Additionally, the generic function described below is capable of capturing direct costs, as shown in the example following the function specification. Specifically, we define a function r_k (r in shorthand form) that, at time t, takes as input the number of type-k violations caught internally, the number of type-k violations caught externally, and a time horizon τ, and outputs the overall loss at time $t + \tau$ due to these violations at time t. r is stationary (i.e., independent of t), and externally caught violations have a stronger impact on r than internally detected

violations. Further, $r(\langle 0,0\rangle,\tau)=0$ for any τ (undetected violations have 0 cost), and r is monotonically decreasing in τ and becomes equal to zero for $\tau \geq m$ (violations are forgotten after a finite amount of rounds). As in previous work [8], we construct the utility function at round t by immediately accounting for future losses due to violations occurring at time t. This allows us to use standard game-theory results, while at the same time, providing a close approximation of the defender's loss [8]. With these notations, $\mathcal{D}$'s utility at time t in $\mathcal{G}_{\mathcal{A},k}$ is

$$\mathbf{Rew}^t_{\mathcal{D}}(\langle s^t, P^t\rangle, \mathbf{O}^t) = -\sum_{j=0}^{m-1} \delta^j_{\mathcal{D}} r(\mathbf{O}^t, j) - C(s^t) - e(P^t)\,. \tag{1}$$

This per-round utility is always negative (or at most zero). As is typical of security games (e.g., [23,24] and related work), implementing security measures does not provide direct benefits to the defender, but is necessary to pare possible losses. Hence, the goal for the defender is to have this utility as close to zero as possible.

The above function can capture direct costs of violations as an additive term at time $\tau=0$. As a simple example [8], assuming the average direct costs for internally and externally caught violations are given by R^D_{int} and R^D_{ext}, and the function r is linear in the random variables $\vec{\mathbf{O}}^t_{int}$ and $\vec{\mathbf{O}}^t_{ext}$, r can be given by

$$r(\mathbf{O}^t,\tau)=\begin{cases}(c+R^D_{int})\mathbf{O}^t_{int}+(\psi c+R^D_{ext})\mathbf{O}^t_{ext} & \text{for } \tau=0\\ \delta^\tau c(\mathbf{O}^t_{int}+\psi\cdot\mathbf{O}^t_{ext}) & \text{for } 1\leq\tau<m\\ 0 & \text{for } \tau\geq m,\end{cases}$$

where $\delta\in(0,1)$ and $\psi\geq 1$. Then Eqn. (1) reduces to

$$\mathbf{Rew}^t_{\mathcal{D}}(\langle s^t, P^t\rangle, \mathbf{O}^t) = -R_{int}\mathbf{O}^t_{int} - R_{ext}\mathbf{O}^t_{ext} - C(s^t) - e(P^t)\,, \tag{2}$$

with $R_{int}=R^I_{int}+R^D_{int}$, $R^I_{int}=c(1-\delta^m\delta^m_{\mathcal{D}})/(1-\delta\delta_{\mathcal{D}})$ and $R_{ext}=\psi R^I_{int}+R^D_{ext}$.

Adversary utility function: We define $\mathcal{A}$'s utility as the sum of $\mathcal{A}$'s *personal benefit* gained by committing violations and the *punishment* that results due to detected violations. Personal benefit is a monetary measure of the benefit that $\mathcal{A}$ gets out of violations. It includes all kinds of benefits, e.g., curiosity, actual monetary benefit (by selling private data), revenge, etc. It is natural that true personal benefit of $\mathcal{A}$ is only known to $\mathcal{A}$. Our model of personal benefit of $\mathcal{A}$ is linear and is defined by a rate of personal benefit for each type of violation given by the vector $\vec{I}_{\mathcal{A}}$ of length K. The punishment is the vector $\vec{P}^t_{\mathcal{A}}$ of length K chosen by $\mathcal{D}$, as discussed above. Using shorthand notation, $\mathcal{A}$'s utility, for the game $\mathcal{G}_{\mathcal{A},k}$, is:

$$\mathbf{Rew}^t_{\mathcal{A}}(\langle a^t, v^t\rangle, \langle s^t, P^t\rangle, \mathbf{O}^t) = Iv^t - P^t\left(\mathbf{O}^t_{int}+\mathbf{O}^t_{ext}\right)\,.$$

Observe that the utility function of a player depends on the public signal (observed violations, $\mathcal{D}$'s action) and the action of the player, which conforms to the definition of a repeated game with imperfect information and public signaling. In such games, the *expected* utility is used in computing equilibria.

Let $\alpha^t = s^t/a^t$ and $\nu(\alpha^t) = \mu(\alpha^t)\alpha^t$. Then, $E(\mathbf{O}^t_{int}) = \nu(\alpha^t)v^t$, and $E(\mathbf{O}^t_{ext}) = pv^t(1-\nu(\alpha^t))$. The expected utilities in each round then become:

$$E(\mathbf{Rew}^t_{\mathcal{D}}) = -\sum_{j=0}^{m-1} \delta^j_{\mathcal{D}} E(r(\mathbf{O}^t, j))[v^t, a^t, \alpha^t] - C(\alpha^t a^t) - e(P^t)\ ,$$
$$E(\mathbf{Rew}^t_{\mathcal{A}}) = Iv^t - P^t v^t\left(\nu(\alpha^t) + p(1-\nu(\alpha^t))\right)\ .$$

The expected utility of $\mathcal{A}$ depends only on the level of inspection and not on the actual number of inspections. For the example loss function given by Eqn. (2), the utility function of $\mathcal{D}$ becomes:

$$E(\mathbf{Rew}^t_{\mathcal{D}}) = -v^t(R_{int}\nu(\alpha^t) + R_{ext}p(1-\nu(\alpha^t))) - C(\alpha^t a^t) - e(P^t)\ .$$

In addition to the action dependent utilities above, the players also receive a fixed utility every round, which is the salary for $\mathcal{A}$ and value generated by $\mathcal{A}$ for $\mathcal{D}$. P_f depends on these values, and is calculated in the full version. Finally, the model parameters that may change over time are R_{ext}, R_{int}, p, function C, function e, function μ and I.

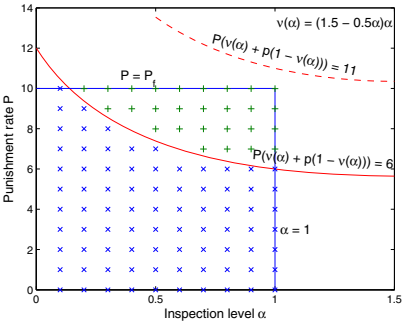


Fig. 1. Non-deterred ($\times$) and deterred ($+$) region for $I = \$6$. $I = \$11$ has empty deterred region.

Graphical representation: A graphical representation of the utilities helps illustrate the ideas presented in the next two sections. (See Figure 1). Consider the 2-dimensional plane $R^{\alpha,P}$ spanned by α^t and P^t. We define a feasible audit space in $R^{\alpha,P}$ given by $0 \le \alpha^t \le 1$ and $0 \le P^t \le P_f$. $\mathcal{D}$'s actions are points in the feasible region. The expected utility of the adversary in each round is given by $v^t(I - P^t(\nu(\alpha^t) + p(1 - \nu(\alpha^t))))$. Thus, the curve in $R^{\alpha,P}$ given by $I = P^t(\nu(\alpha^t) + p(1-\nu(\alpha^t)))$ is the separator between positive and negative expected utility regions for the adversary in each round. Within the feasible region, we call the region of positive expected utility the *non-deterred region* and the region of negative utility the *deterred region*.

$\mathcal{A}$'s utility can as well be non-linear, e.g., if $\mathcal{D}$ decides to scale punishment quadratically with violations. Technically, this partitions the feasible audit space into many regions, with each region associated with the number of violations that maximize the utility of $\mathcal{A}$ in that region. We emphasize that the equilibrium presented later can be easily extended to consider such cases. To keep the presentation simple we keep using the linear utility throughout the paper, which yields two regions associated with 0 or all violations. Similarly, it is possible to add any other relevant term to $\mathcal{D}$'s utility, e.g., if

$\mathcal{D}$ satisfies a certain accountability criteria (defined later in Section 5) then it may earn positive benefit out of increased reputation.

Estimation: Next, we describe techniques of estimating parameters of game $\mathcal{G}_{\mathcal{A},k}$, obtaining sample estimates in the process. Before getting to constant values, we state the functions that we use as concrete instances for the examples in this paper. We use simple linear functions for audit cost ($C(\alpha a) = C\alpha a$) and for punishment loss ($e(P) = eP$). The performance of $\mathcal{M}$ is dependent on the tool being used and we use a linear function for $\mu(.)$ to get $\nu(\alpha) = \mu\alpha - (\mu - 1)\alpha^2$, where μ is a constant. Further, we use the example loss function (with R_{int} and R_{ext}) stated in the last sub-section. We note that our theorems work with any function; these functions above are the simplest functions that satisfy the constraints on these functions stated in the last sub-section. Next, we gather data from industry wide studies to obtain sample estimates for parameters.

As stated in Section 2, values of direct and indirect costs of violation (average of R_{int} and R_{ext} is \$300 in healthcare [19], a detailed breakdown is present in the ANSI report [15]), maximum personal benefit I (\$50 for medical records [14,15]), etc. are available in studies. Also, in absence of studies quantitatively distinguishing externally and internally caught violations we assume $R_{int} = R_{ext} = \$300$. Many parameters depends on the employee, his role in the organization and type of violation. Keeping a track of violations and behavior within the organization offers a data source for estimating and detecting changes in these parameters. We choose values for these parameters that are not extremes, $e = \$10$, $I = \$6$, $\epsilon_{th} = 0.03$, $\delta_{\mathcal{A}} = 0.4$ and $U_k = 40$. Further, under certain assumptions we calculate P_f (in full version) to get $P_f = \$10$. Finally, the average cost of auditing C and performance factor μ of log analysis tool should be known to $\mathcal{D}$. We assume values $C = \$50$, and tool performance $\mu = 1.5$.

4 Auditing Strategy

In this section, we define a suitable equilibrium concept for the audit game (Section 4.1) and present a strategy for the defender such that the best response to that strategy by the adversary results in an equilibrium being attained (Section 4.2). Finally, we compare our equilibrium with other equilibria (Section 4.3). Recall that the equilibrium of the game occurs in the period in which the game parameters are fixed.

4.1 Equilibrium Concepts

We begin by introducing standard terminology from game theory. In a one-shot extensive form game players move in order. We assume player 1 moves first followed by player 2. An extensive form repeated game is one in which the round game is a one-shot extensive game. The history is a sequence of actions. Let H be the set of all possible histories. Let S_i be the action space of player i. A strategy of player i is a function $\sigma_i : H_i \to S_i$, where $H_i \subset H$ are the histories in which player i moves. The utility in each round is given by $r_i : S_1 \times S_2 \to \mathbb{R}$. The total utility is a δ_i-discounted sum of utilities of each round, normalized by $1 - \delta_i$.

The definition of strategies extends to extensive form repeated games with public signals. We consider a special case here that resembles our audit game. Player 1 moves

first and the action is observed by player 2, then player 2 moves, but, that action may not be perfectly observed, instead resulting in a public signal. Let the space of public signals be Y. In any round, the observed public signal is distributed according to the distribution $\Delta Y(.|s)$, i.e., $\Delta Y(y|s)$ is the probability of seeing signal y when the action profile s is played. In these games, a history is defined as an alternating sequence of player 1'a action and public signals, ending in a public signal for histories in which player 1 has to move and ending in player 1's move for histories in which player 2 has to move. The actual utility in each round is given by the function $r_i : S_i \times Y \to \mathbb{R}$. The total expected utility g_i is the expected normalized δ_i-discounted sum of utilities of each round, where the expectation is taken over the distribution over public signals and histories. For any history h, the game to be played in the future after h is called the *continuation game* of h with total utility given by $g_i(\sigma, h)$.

A strategy profile (σ_1, σ_2) is a *subgame perfect equilibrium* (SPE) of a repeated game if it is a Nash equilibrium for all continuation games given by any history h [9]. One way of determining if a strategy is a SPE is to determine whether the strategy satisfies the *single stage deviation* property, that is, any *unilateral deviation* by any player *in any single round* is not profitable. We define a natural extension of SPE, which we call *asymmetric subgame perfect equilibrium* (or (ϵ_1, ϵ_2)-SPE), which encompasses SPE as a special case when $\epsilon_1 = \epsilon_2 = 0$.

Definition 1. *((ϵ_1, ϵ_2)-SPE) Denote concatenation operator for histories as ;. Strategy profile σ is a (ϵ_1, ϵ_2)-SPE if for history h in which player 1 has to play, given $h' = h; \sigma_1(h)$ and $h'' = h; s_1$,*

$$\begin{aligned} &E(r_1(\sigma_1(h), \mathbf{y}))[\sigma_1(h), \sigma_2(h')] + \delta_1 E(g_1(\sigma, h'; \mathbf{y}))[\sigma_1(h), \sigma_2(h')] \\ &\geq E(r_1(s_1, \mathbf{y}))[s_1, \sigma_2(h'')] + \delta_1 E(g_1(\sigma, h''; \mathbf{y}))[s_1, \sigma_2(h'')] - \epsilon_1 \end{aligned}$$

for all s_1. For history h in which player 2 has to play, given $a(h)$ is the last action by player 1 in h, for all s_2

$$\begin{aligned} &E(r_2(\sigma_2(h), \mathbf{y}))[a(h), \sigma_2(h)] + \delta_2 E(g_2(\sigma, h; \mathbf{y}))[a(h), \sigma_2(h)] \\ &\geq E(r_2(s_2, \mathbf{y}))[a(h), s_2] + \delta_2 E(g_2(\sigma, h; \mathbf{y}))[a(h), s_2] - \epsilon_2 \end{aligned}$$

We are particularly interested in $(\epsilon_1, 0)$-SPE, where player 1 is the defender and player 2 is the adversary. By setting $\epsilon_2 = 0$, we ensure that a rational adversary will never deviate from the expected equilibrium behavior. Such equilibria are important in security games, since $\epsilon_2 > 0$ could incentivize the adversary to deviate from her strategy, possibly resulting in significant loss to the defender.

4.2 Equilibrium in the Audit Game

We next state an equilibrium strategy profile for the game $G_{\mathcal{A},k}$. Formally, we present a $(\epsilon_{\mathcal{A},k}, 0)$-SPE strategy profile, and calculate the value $\epsilon_{\mathcal{A},k}$. The proposed strategy relies on commitment by $\mathcal{D}$ and computation of a single round best response by $\mathcal{A}$. We accordingly refer to this strategy profile as a *simple commitment* strategy profile.

For any equilibrium to be played out with certainty, players must believe that the strategy being used by the other players is the equilibrium strategy. Our proposed strategy profile has features that aim to achieve correct beliefs for the players, even in face of partial rationality. One feature is that $\mathcal{D}$ makes its strategy publicly known, and provides a means to verify that it is playing that strategy. As noted earlier, even though $\mathcal{D}$ acts after $\mathcal{A}$ does by committing to its strategy with a verification mechanism $\mathcal{D}$ simulates a first move by making the employee believe its commitment with probability one. Thus, we envision the organization making a commitment to stick to its strategy and providing a proof that it follows the strategy. Further, $\mathcal{D}$ making its strategy publicly known follows the general security principle of not making the security mechanisms private [25]. Additionally, the simple commitment strategy profile is an approximate SPE for all values of parameters in any game $G_{\mathcal{A},k}$ and any value of $\mathcal{A}$'s discount factor $\delta_{\mathcal{A}}$. Thus, all employees observe the organization following a consistent strategy further reducing any variability in beliefs about the organization's strategy. Another important feature of the simple commitment strategy profile is the single round best response computation by $\mathcal{A}$ (yielding a single action to play), which is much simpler than optimizing over multiple rounds often yielding many strategies as the solution. Thus, the organization also trusts the employee to make the appropriate decision even if the employee is computationally constrained. The above features of the simple commitment strategy profile makes the strategy simple, which makes it more likely to be followed in the real world.

The main idea behind the definition of our strategy profile is that $\mathcal{D}$ optimizes its utility assuming the best response of $\mathcal{A}$ for a given a^t. That is, $\mathcal{D}$ assumes that $\mathcal{A}$ does not commit any violations when (P, α) is in the deterred region, and systematically commits a violation otherwise (i.e., all of $\mathcal{A}$'s tasks are violations). Further, $\mathcal{D}$ assumes the worst case when the employee (with probability ϵ_{th}) accidentally makes a mistake in the execution of their strategy; in such a case, $\mathcal{D}$ expects all of $\mathcal{A}$'s tasks to be violations, regardless of the values of (P, α). This is because the distribution D_0^t over violations when $\mathcal{A}$ makes a mistake is unknown. Thus, the expected cost function that $\mathcal{D}$ optimizes (for each total number of tasks a^t) is a linear sum of $(1 - \epsilon_{th})$ times the cost due to best response of $\mathcal{A}$ and ϵ_{th} times the cost when $\mathcal{A}$ commits all violations. The expected cost function is different in the deterred and non-deterred region due to the difference in best response of $\mathcal{A}$ in these two regions. The boundary between the deterred and non-deterred regions is conditioned by the value of the adversary's personal benefit I. We assume that $\mathcal{D}$ learns the value of the personal benefit within an error δI of its actual value, and that $\mathcal{D}$ does not choose actions (P, α) in the region of uncertainty determined by the error δI.

Formally, the expected reward is $E(\mathbf{Rew}_{\mathcal{D}}^t)[0]$ when the adversary commits no violation, and $E(\mathbf{Rew}_{\mathcal{D}}^t)[a^t]$ when all a^t tasks are violations. Both of these expected rewards are functions of P, α; we do not make that explicit for notational ease. Denote the deterred region determined by the parameter I and the budget $b_{\mathcal{A},k}^t$ as R_D^I and the non-deterred region as R_{ND}^I. Either of these regions may be empty. Denote the region (of uncertainty) between the curves determined by $I + \delta I$ and $I - \delta I$ as $R_{\delta I}^I$. Then the reduced deterred region is given by $R_D^I \backslash R_{\delta I}^I$ and the reduced non-deterred region by $R_{ND}^I \backslash R_{\delta I}^I$. The equilibrium strategy we propose is:

- For each possible number of tasks a^t that can be performed by $\mathcal{A}$, $\mathcal{D}$ constrained by budget $b^t_{\mathcal{A},k}$, assumes the expected utility

$$U_D(P,\alpha) = (1-\epsilon_{th})E(\mathbf{Rew}^t_{\mathcal{D}})[0] + \epsilon_{th}E(\mathbf{Rew}^t_{\mathcal{D}})[a^t] \text{ and}$$
$$U_{ND}(P,\alpha) = (1-\epsilon_{th})E(\mathbf{Rew}^t_{\mathcal{D}})[a^t] + \epsilon_{th}E(\mathbf{Rew}^t_{\mathcal{D}})[a^t] ,$$

  in $R^I_D \backslash R^I_{\delta I}$ and $R^I_{ND} \backslash R^I_{\delta I}$ respectively. $\mathcal{D}$ calculates the maximum expected utility across the two regions as follows:
  $- U^D_{\max} = \max_{(P,\alpha)\in R^I_D \backslash R^I_{\delta I}} U_D(P,\alpha)$, $U^{ND}_{\max} = \max_{(P,\alpha)\in R^I_{ND} \backslash R^I_{\delta I}} U_{ND}(P,\alpha)$
  $- U = \max(U^D_{\max}, U^{ND}_{\max})$
  $\mathcal{D}$ commits to the corresponding maximizer (P,α) for each a^t.
  After knowing a^t, $\mathcal{D}$ plays the corresponding (P,α).
- $\mathcal{A}$ plays her best response (based on the committed action of $\mathcal{D}$), i.e., if she is deterred for all a^t she commits no violations and if she is not deterred for some a^t then all her tasks are violations, and she chooses the a^t that maximizes her utility from violations. But, she also commits mistakes with probability ϵ_{th}, and then the action is determined by distribution D^t_0.

Let $U^{D+\delta I}_{\max} = \max_{(P,\alpha)\in R^I_D \cup R^I_{\delta I}} U_D(P,\alpha)$, $U^{ND+\delta I}_{\max} = \max_{(P,\alpha)\in R^I_{ND} \cup R^I_{\delta I}} U_{ND}(P,\alpha)$, $\delta U^D = U^{D+\delta I}_{\max} - U^D_{\max}$ and $\delta U^{ND} = U^{ND+\delta I}_{\max} - U^{ND}_{\max}$. We have the following result:

Theorem 1. *The simple commitment strategy profile (defined above) is an* $(\epsilon_{\mathcal{A},k}, 0)$*-SPE for the game* $G_{\mathcal{A},k}$*, where* $\epsilon_{\mathcal{A},k}$ *is*

$$\max\left(\max_{v^t,a^t}(\delta U^D),\ \max_{v^t,a^t}(\delta U^{ND})\right) + \epsilon_{th}\max_{\alpha\in[0,1]}\left(\sum_{j=0}^{m-1}\delta^j_{\mathcal{D}}E(r(\vec{\mathbf{O}}^t,j))[U_k,U_k,\alpha]\right)$$

Remark 1. If the value of any parameter of the game (e.g., R_{ext}, R_{int}) is perturbed in a bounded manner, then accounting for that in the analysis yields an $(\epsilon, 0)$-SPE, but, with ϵ greater than $\epsilon_{\mathcal{A},k}$. This happens because $\mathcal{D}$'s utility is continuous in the parameters.

The proof involves showing that the strategy profile has the single stage deviation property. That $\mathcal{A}$ does not profit from deviating is immediate since $\mathcal{A}$ chooses the best response in each round of the game. The bound on profit from deviation for $\mathcal{D}$ has two terms. The first term arises due to $\mathcal{D}$ ignoring the region of uncertainty in maximizing its utility. The maximum difference in utility for the deterred region is $\max_{v^t,a^t}(U^{D+\delta I}_{\max} - U^D_{\max})$ and for the undeterred region is $\max_{v^t,a^t}(U^{ND+\delta I}_{\max} - U^{ND}_{\max})$. The first term is the maximum of these quantities. The second term arises due to the use of the worst case assumption of all violations out of maximum possible U_k tasks when $\mathcal{A}$ makes a mistake as compared to the case when D^t_0 is known. Since $\mathcal{A}$'s choice only affects the violation loss part of $\mathcal{D}$'s utility and mistakes happen with probability ϵ_{th}, the second term is the maximum possible violation loss multiplied by ϵ_{th}.

Numeric applications. The above theorem can be used to calculate concrete values for $\epsilon_{\mathcal{A},k}$ when all parametric functions are instantiated. For example, with the values in Section 3, we obtain $\epsilon_{\mathcal{A},k} = \$200$. Assuming $\mathcal{A}$ performs the maximum

$U_k = 40$ number of tasks, $\epsilon_{\mathcal{A},k}$ is about 9.5% of the cost of auditing all actions of $\mathcal{A}$ with maximum punishment rate ($2100), with no violations, and about 3.3% of the cost incurred due to all violations caught externally ($6000), with no internal auditing or punishment. Similarly, if we assume 70% audit coverage with maximum punishment and four violations, the expected cost for organization is $2583, which means $\epsilon_{\mathcal{A},k}$ corresponds to about 7.7% of this cost. We present the derivation of the value of $\epsilon_{\mathcal{A},k}$ in the full version. The audit coverage here is for one employee only; hence it can be as high as 100%. Also, since $\mathcal{G}_{\mathcal{A}}$ is a parallel composition of the games $\mathcal{G}_{\mathcal{A},k}$ for all k, we claim that the simple commitment strategy profile followed for all games $\mathcal{G}_{\mathcal{A},k}$ is a $(\sum_k \epsilon_{\mathcal{A},k}, 0)$-SPE strategy profile for $G_{\mathcal{A}}$. (See full version for details.)

4.3 Comparision with Other Equilibria

In this section, we compare our proposed strategy with the set of Perfect Public Equilibrum (PPE) strategies. A PPE is the appropriate notion of equilibrium in an imperfect information repeated game with public signals and simultaneous moves. A PPE is quite similar to a SPE; the differences are that histories are sequences of public signals (instead of action profiles) and payoffs are considered in the expected sense. PPE strategy profiles also have the single stage deviation property. As pointed out already, one advantage of the simple commitment strategy is simplicity. As the set of PPE strategies is often infinite, it is difficult for players' beliefs to agree on the strategy being played. However, a commitment by one player to her part of a PPE strategy profile forces that particular PPE to be played. The organization is naturally the player who commits. A *committed utility maximizing* player is one who uses a commitment to force the PPE that yields the maximum payoff to that player. A *privacy preserving* defender is one that chooses a PPE with fewer violations when it has a choice over multiple PPE with the same payoff for the defender. The next theorem shows that simple commitment strategy deters $\mathcal{A}$ as often as the case in which the chosen PPE strategy deters $\mathcal{A}$, assuming the budget allows for deterring the employee and the organization is committed utility maximizing and privacy preserving in choosing PPE equilibrium. Stated succinctly, the simple commitment strategy profile is no worse for privacy protection than choosing the highest utility PPE in scenarios where the organization chooses a PPE strategy that deters the employee.

Theorem 2. *Assume that budget is fixed in every round and is sufficient to deter $\mathcal{A}$, and the number of tasks performed by $\mathcal{A}$ in every round in fixed. Let v_o^* be the maximum PPE payoff that $\mathcal{D}$ can obtain. Further suppose there exists a PPE E_m in which $\mathcal{D}$ always plays some action in the deterred region and the utility for $\mathcal{D}$ with E_m is v_o^*. Then a committed utility maximizing and privacy preserving $\mathcal{D}$ will choose to play E_m. Further, the action in E_m coincides with the action chosen by simple commitment strategy profile in each round.*

5 Budget Allocation

In this section we present two budget allocation mechanisms: one maximizes $\mathcal{D}$'s utility (Section 5.1) and another does the same under accountability constraints (Section 5.2).

5.1 Optimized Budget Allocation

We assume the budget available to $\mathcal{D}$ for all audits is bound by B. Then we must have $\sum_{\mathcal{A},k} \vec{b}^t_{\mathcal{A}}(k) + Cost(\mathcal{M}) \leq B$, where $Cost(\mathcal{M})$ is a fixed cost of using the log analysis tool in an audit cycle. Let $B_{\mathcal{M}} = B - Cost(\mathcal{M})$. Let $\alpha_{\mathcal{A},k}(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k)), P_{\mathcal{A},k}(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k))$ be the equilibrium in game $\mathcal{G}_{\mathcal{A},k}$ for budget $\vec{b}^t_{\mathcal{A}}(k)$ and $\mathcal{A}$'s tasks $\vec{a}^t_{\mathcal{A}}(k)$. Note that we make the dependence on $\vec{b}^t_{\mathcal{A}}(k)$, $\vec{a}^t_{\mathcal{A}}(k)$ explicit here. Let $U(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k))$ denote the corresponding expected utility in game $\mathcal{G}_{\mathcal{A},k}$. Observe that in equilibrium, when $\mathcal{A}$ is deterred for all possible $\vec{a}^t_{\mathcal{A}}(k)$ then $\mathcal{A}$ has equal preference for all possible $\vec{a}^t_{\mathcal{A}}(k)$, and otherwise $\mathcal{A}$ chooses the maximum $\vec{a}^t_{\mathcal{A}}(k)$ for which she is undeterred to maximize her utility. Thus, let $BR(\vec{b}^t_{\mathcal{A}}(k))$ be the set of number of tasks all of which are part of best responses of $\mathcal{A}$. Note that the cost functions U_D and U_{ND} in deterred and non-deterred regions are continuous in $\vec{b}^t_{\mathcal{A}}(k)$, since the regions themselves change continuously with change in $\vec{b}^t_{\mathcal{A}}(k)$. Also, by definition they are continuous in $\vec{a}^t_{\mathcal{A}}(k)$. Since U is the maximum of two continuous functions U_D and U_{ND}, using the fact that max of two functions is continuous, we get that U is continuous in both arguments. Then, the optimal allocation of budget is to solve the following non-linear optimization problem

$$\max \sum_{\mathcal{A},k} \min_{\vec{a}^t_{\mathcal{A}}(k) \in BR(\vec{b}^t_{\mathcal{A}}(k))} U(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k)) \text{ subject to } \vec{b}^t_{\mathcal{A}}(k) \geq 0 \text{ and } \sum_{\mathcal{A},k} \vec{b}^t_{\mathcal{A}}(k) \leq B_{\mathcal{M}} \ ,$$

which maximizes the minimum utility possible over $\mathcal{A}$'s possible best response actions. For example, consider a simple case with two types of tasks: celebrity records accesses and non-celebrity records accesses, and one employee. Assume the utility functions and constants as stated at the end of Section 3, except, it is assumed that it is apriori known that exactly 40 celebrity and 400 non-celebrity accesses would be made and values of some constants (in brackets) are different for celebrity type ($R_{ext} = \$4500, R_{int} = \$300, I = \$6$, $P_f = 10$) and non-celebrity type ($R_{ext} = \$90, R_{int} = \$30, I = \$0.6$, $P_f = 5$). Using discrete steps and a brute force search yields a solution of the above optimization problem in which $\mathcal{D}$ would allocate $\$1300$ to audit celebrity accesses and the remaining $\$1200$ to audit non-celebrity accesses. As the cost per inspection was assumed $\$50$ (Section 3), 0.65 fraction of celebrity accesses can be inspected and only 24 out of 400 non-celebrity accesses can be inspected. However, the equilibrium yields that no non-celebrity inspections happen as the employee is non-deterred for the level of non-celebrity inspections possible, and 0.65 fraction of celebrity accesses are inspected.

5.2 Towards Accountable Data Governance

While holding an employee responsible for the violation she causes is natural, it is difficult to define accountability for the organization, as the organization does not commit violations directly. However, the organization influences the actual violator (employee) by the choice of inspections and punishment. We use a simple definition of accountability for the organization, requiring a minimum level of inspection and punishment.

Definition 2. *($(\mathcal{M}, \vec{\alpha}, \vec{P})$-accountability) An organization satisfies $(\mathcal{M}, \vec{\alpha}, \vec{P})$-accountability if 1) its log analysis tool $\mathcal{M}'$ satisfies $\mathcal{M}' \geq \mathcal{M}$, 2) its level of inspection satisfies $\vec{\alpha}' \geq \vec{\alpha}$, and 3) its punishment rate satisfies $\vec{P}' \geq \vec{P}$.*

Our definition assumes a partial ordering over log analysis tools $\mathcal{M}$. This partial ordering could be given from empirically computed accuracy μ estimates for each log analysis tool (e.g., we could say that $\mathcal{M}_1 \geq \mathcal{M}_2$ if $\mathcal{M}_1$ is at least as accurate as $\mathcal{M}_2$ for each type of access k). The dependence of accountability on $\mathcal{M}$ is required as a better performing tool can detect the same expected number of violations as another tool with worse performance, with a lower inspection level α. We envision the above accountability being proven by the organization to a trusted third party external auditor (e.g., Government) by means of a formal proof, in the same manner as commitment is demonstrated to the employee.

To satisfy $(\mathcal{M}, \vec{\alpha}, \vec{P})$-accountability an organization must add the following constraints to its optimization problem from the last sub-section: $\min_{\vec{a}^t_{\mathcal{A}}(k) \in BR(\vec{b}^t_{\mathcal{A}})} \alpha_{\mathcal{A},k}(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k)) > \vec{\alpha}(k)$ and $\min_{\vec{a}^t_{\mathcal{A}}(k) \in BR(\vec{b}^t_{\mathcal{A}})} P_{\mathcal{A},k}(\vec{b}^t_{\mathcal{A}}(k), \vec{a}^t_{\mathcal{A}}(k)) > \vec{P}(k)$ for all $\mathcal{A}, k$. The first constraint ensures that the the minimum number of inspections divided by maximum number of tasks is greater than $\vec{\alpha}(k)$, and the second constraint ensures that the minimum punishment level is higher that $\vec{P}(k)$.

Continuing the example from last sub-section if the minimum α and P is specified as 0.1 and 1.0 for both types of accesses, then $\mathcal{D}$ would allocate $\$400$ to audit celebrity accesses and the remaining $\$2100$ to audit non-celebrity accesses. Since the cost per inspection was assumed $\$50$ (Section 3), 0.2 fraction of celebrity accesses can be inspected and 42 out of 400 non-celebrity accesses can be inspected. However, according to the equilibrium 40 non-celebrity inspections happen at punishment level of 2.0 as the employee is already deterred for that level of non-celebrity inspections. In this case, unlike the non-accountable scenario, the values $\vec{\alpha}, \vec{P}$ ensure that the privacy of common person is being protected even when the organization has more economic incentives to audit celebrity accesses more heavily.

6 Predictions and Interventions

In this section, we use our model to predict observed practices in industry and the effectiveness of public policy interventions in encouraging organizations to adopt accountable data governance practices (i.e., conduct more thorough audits) by analyzing the equilibrium audit strategy P, α under varying parameters. The explanation of observed practices provides evidence that our audit model is not far from reality. We use the values of parameters and instantiation of functions given in Section 3 (unless otherwise noted). We assume that the value of personal benefit I is learned exactly and that P and α take discrete values, with the discrete increments being 0.5 and 0.05, respectively. We also assume for sake of exposition that $u_k = U_k$, i.e., the number of tasks is fixed, there is only one type of violation and the budget is sufficient to do all possible inspections.

Average cost R_{ext} and probability p of external detection of violation. We vary R_{ext} from $\$5$ to $\$3900$, with R_{int} fixed at $\$300$. The results are shown in Figure 2. There are two cases shown in the figure: $p = 0.5$ and $p = 0.9$. The figure shows the equilibria P, α chosen for different values of R_{ext}.

Prediction 1: Increasing R_{ext} and p is an effective way to encourage organizations to audit more. In fact, when $p * R_{ext}$ is low X may not audit at all. Thus, X audits to protect itself from greater loss incurred when violations are caught externally. Surprisingly, the hospital may continue to increase inspection levels (incurring higher cost) beyond the minimum level necessary to deter a rational employee. Hospital X does so because the employee is not fully rational: even in the deterred region there is an ϵ_{th} probability of violations occurring.

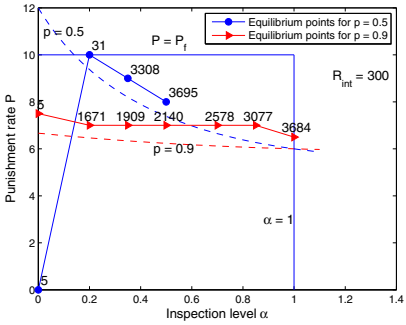


Fig. 2. Separators for two values of external detection probability p indicated by dashed lines. Equilibrium punishment and inspection rates (P, α) marked on solid lines (see legend) as the reputation loss from external detection R_{ext} varies; the R_{ext} values are labeled above the corresponding equilibrium points.

Suggested Intervention 1: Subject organizations to external audits and fines when violations are detected. For example, by awarding contracts for conducting 150 external audits by 2012 [26], HHS is moving in the right direction by effectively increasing p. This intervention is having an impact: the 2011 Ponemon study on patient privacy [27] states—"Concerns about the threat of upcoming HHS HIPAA audits and investigation has affected changes in patient data privacy and security programs, according to 55 percent of respondents."

Prediction 2: Interventions that increase the expected loss for both external and internal detection of violations are not as effective in increasing auditing as those that increase expected loss for external detection of violations only. Table 2 shows the equilibrium inspection level as R_{ext} and R_{int} are both increased at the same rate. While the inspection level may initially increase, it quickly reaches a peak. As an example, consider the principle of breach detection notification used in many data breach laws [28]. The effect of breach detection notification is to increase both R_{int} and R_{ext} since notification happens for all breaches. While there isn't sufficient data for our model to predict whether these laws are less effective than external audits (see suggested study below), prior empirical analysis [28] indicate that the benefit in breach detection from these laws is only about 6% (after adjusting for increased reporting of breaches due to the law itself).

Suggested study: An empirical study that separately reports costs incurred when violations are internally detected from those that are externally detected would be useful in quantifying and comparing the effectiveness of interventions. Existing studies either do

not speak of these distinct categories of costs [19,28] or hint at the importance of this distinction without reporting numbers [16,17].

Punishment loss factor e and personal benefit I. *Prediction 3: Employees with higher value for e (e.g., doctors have higher e; suspending a doctor is costlier for the hospital than suspending a nurse) will have lower punishment levels.* If punishments were free, i.e., $e = 0$, (an unrealistic assumption) X will always keep the punishment rate at maximum according to our model. At higher punishment rates ($e = 1000$), X will favor increasing inspections rather than increasing the punishment level P (see Table 1 in Appendix A). While we do not know of an industry-wide study on this topic, there is evidence of such phenomena occurring in hospitals. For example, in 2011 Vermont's Office of Professional Regulation, which licenses nurses, investigated 53 allegations of drug diversion by nurses and disciplined 20. In the same year, the Vermont Board of Medical Practice, which regulates doctors, listed 11 board actions against licensed physicians for a variety of offenses. However, only one doctor had his license revoked while the rest were allowed to continue practicing [7].

Prediction 4: Employees who cannot be deterred are not punished. When the personal benefit of the employee I is high, our model predicts that X chooses the punishment rate $P = 0$ (because this employee cannot be deterred at all) and increases inspection as R_{ext} increases to minimize the impact of violations by catching them inside (see Table 4 in Appendix A). Note that this is true only for violations that are not very costly (as is the case for our choice of costs). If the expected violation cost is more than the value generated by the employee, then it is better to fire the non-deterred employee (see full version).

Audit cost C and performance factor μ of log analysis tool.
Prediction 5: If audit cost C decreases or the performance μ of log analysis increases, then the equilibrium inspection level increases. The data supporting this prediction is presented in Table 3 and 5 in Appendix A. Intuitively, it is expected that if the cost of auditing goes down then organizations would audit more, given their fixed budget allocated for auditing. Similarly, a more efficient mechanized audit tool will enable the organization to increase its audit efficiency with the fixed budget. For example, MedAssets claims that Stanford Hospitals and Clinics saved $4 million by using automated tools for auditing [29].

7 Related Work

Auditing and Accountability: Prior work studies orthogonal questions of algorithmic detection of policy violations [30–33] and blame assignment [34–37]. Feigenbaum et al. [38] report work in progress on formal definitions of accountability capturing the idea that violators are punished with or without identification and mediation with non-zero probability, and punishments are determined based on an understanding of "typical" utility functions. Operational considerations of how to design an accountability mechanism that effectively manages organizational risk is not central to their work. In other work, auditing is employed to revise access control policies when unintended accesses are detected [39–41]. Another line of work uses logical methods for enforcing a

class of policies, which cannot be enforced using preventive access control mechanisms, based on evidence recorded in audit logs [42]. Cheng et al. [43,44] extend access control to by allowing agents access based on risk estimations. A game-theoretic approach of coupling access control with audits of escalated access requests in the framework of a single-shot game is studied by Zhao et al. [45]. These works are fundamentally different from our approach. We are interested in scenarios where access control is not desirable and audits are used to detect violations. We believe that a repeated game can better model the repeated interactions of auditing.

Risk Management and Data Breaches: Our work is an instance of a risk management technique [12,13] in the context of auditing and accountability. As far as we know, our technique is the first instance of managing risk in auditing using a repeated game formalism. Risk assessment has been extensively used in many areas [10,11]; the report by American National Standards Institute [15] provides a risk assessment mechanism for healthcare. Our model also models data breaches that happen due to insider attacks. Reputation has been used to study insider attacks in non-cooperative repeated games [46]; we differ from that work in that the employer-employee interaction is essentially cooperative. Also, the primary purpose of interaction between employer and employee is to accomplish some task (e.g., provide medical care). Privacy is typically a secondary concern. Our model captures this reality by considering the effect of non-audit interactions in parameters like P_f. There are quite a few empirical studies on data breaches and insider attacks [16,19,28] and qualitative models of insider attacks [47]. We use these studies to estimate parameters and evaluate the predictions of our model.

8 Conclusion and Future Work

First, as public policy and industry move towards accountability-based privacy governance, the biggest challenge is how to operationalize requirements such as internal enforcement of policies. We believe that principled audit and punishment schemes like the one presented in this paper can inform practical enforcement regimes. Second, a usual complaint against this kind of risk management approach is that there isn't data to estimate the risk parameters. We provide evidence that a number of parameters in the game model can be estimated from prior empirical studies while recognizing the need for more scientific studies with similar goals, and suggest specific studies that can help estimate other parameters. Third, our model makes an interesting prediction that merits further attention: it suggests that we should design interventions that increase the expected loss from external detection of violations significantly more than the expected loss from internal detection.

While our model captures a number of important economic considerations that influence the design of audit mechanisms, there is much room for further refinement. For example, the model does not handle colluding adversaries nor does it account for detection of violations in audit rounds other than the one in which the violation was committed. Also, our treatment of accountable data governance leaves open questions about the trade-off between utility maximization and privacy protection. Moving forward, we plan to generalize our model, explore the space of policy interventions to

encourage accountable data governance, and address normative questions such as what are appropriate levels of inspections and punishments for accountable data governance.

References

1. Center for Information Policy Leadership: Accountability-Based Privacy Governance Project (accessed May 1, 2012)
2. The White House: Consumer Data Privacy in a Networked World: A Framework for Protecting Privacy and Promoting Innovation in the Global Digital Economy (accessed May 1, 2012)
3. Fairwarning: Industry Best Practices for Patient Privacy in Electronic Health Records (April 2011)
4. Hulme, G.: Steady Bleed: State of HealthCare Data Breaches. InformationWeek (September 2010)
5. U.S. Department of Health & Human Services: HIPAA enforcement (accessed May 1,2012)
6. Ornstein, C.: Breaches in privacy cost Kaiser, http://articles.latimes.com/2009/may/15/local/me-privacy15 (May 2009)
7. Picard, K.: Are Drug-Stealing Nurses Punished More Than Doctors? (2012)
8. Blocki, J., Christin, N., Datta, A., Sinha, A.: Regret minimizing audits: A learning-theoretic basis for privacy protection. In: Computer Security Foundations Symposium, pp. 312–327 (2011)
9. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press (1991)
10. PricewaterhouseCoopers: A practical guide to risk assessment (December 2008)
11. Vellani, K.H.: Strategic Healthcare Security, Risk Assessments in the Environment of Care, Report for Wisconsin Healthcare Engineering Association (2008)
12. NIST: Guide for Conducting Risk Assessments (September 2011)
13. Cheng, P.-C., Rohatgi, P.: IT Security as Risk Management: A Reserach Perspective. IBM Research Report (April 2008)
14. Petrochko, C.: DHC: EHR Data Target for Identity Thieves (December 2011)
15. American National Standards Institute(ANSI)/The Santa Fe Group/Internet Security Alliance: The financial impact of breached protected health information (accessed May 1,2012)
16. Verizon: 2012 Data Breach Investigations Report (2012)
17. Ponemon Institute, LLC: Benchmark Study on Patient Privacy and Data Security (November 2010)
18. Ponemon Institute, LLC: 2011 Cost of Data Breach Study: United States (March 2012)
19. Ponemon Institute, LLC: 2010 Annual Study: U.S. Cost of a Data Breach (March 2011)
20. Ichniowski, C., Shaw, K., Prennushi, G.: The Effects of Human Resource Management Practices on Productivity. Technical Report 5333, National Bureau of Economic Research (November 1995)
21. Hanushek, E.A.: Statistical Methods for Social Scientists. Academic Press, New York (1977)
22. Mailath, G.J., Samuelson, L.: Repeated Games and Reputations: Long-Run Relationships. Oxford University Press, USA (2006)
23. Varian, H.: System reliability and free riding. In: Economics of Information Security (Advances in Information Security), vol. 12, pp. 1–15 (2004)
24. Grossklags, J., Christin, N., Chuang, J.: Secure or insure? A game-theoretic analysis of information security games. In: World Wide Web Conference (WWW 2008), pp. 209–218 (2008)
25. Saltzer, J.H., Schroeder, M.D.: The protection of information in computer systems. Proceedings of the IEEE 63(9), 1278–1308 (1975)
26. U.S. Department of Health & Human Services: HIPAA Privacy and Security Audit Program

27. Ponemon Institute, LLC: Second Annual Benchmark Study on Patient Privacy and Data Security (December 2011)
28. Romanosky, S., Hoffman, D., Acquisti, A.: Empirical analysis of data breach litigation. In: International Conference on Information Systems (2011)
29. MedAssets: MedAssets Case Sudy: Stanford hospital takes charge of its charge capture process, increasing net revenue by 4 million (2011)
30. Barth, A., Datta, A., Mitchell, J.C., Nissenbaum, H.: Privacy and contextual integrity: Framework and applications. In: IEEE Symposium on Security and Privacy, pp. 184–198 (2006)
31. Basin, D., Klaedtke, F., Müller, S.: Policy Monitoring in First-Order Temporal Logic. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 1–18. Springer, Heidelberg (2010)
32. Garg, D., Jia, L., Datta, A.: Policy auditing over incomplete logs: theory, implementation and applications. In: ACM Computer and Communications Security (CCS), pp. 151–162 (2011)
33. Tschantz, M.C., Datta, A., Wing, J.M.: Formalizing and enforcing purpose requirements in privacy policies. In: IEEE Symposium on Security and Privacy (2012)
34. Backes, M., Datta, A., Derek, A., Mitchell, J.C., Turuani, M.: Compositional analysis of contract-signing protocols. Theor. Comput. Sci. 367(1-2), 33–56 (2006)
35. Barth, A., Datta, A., Mitchell, J.C., Sundaram, S.: Privacy and utility in business processes. In: Computer Security Foundations Symposium (CSF), pp. 279–294 (2007)
36. Jagadeesan, R., Jeffrey, A., Pitcher, C., Riely, J.: Towards a Theory of Accountability and Audit. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 152–167. Springer, Heidelberg (2009)
37. Küsters, R., Truderung, T., Vogt, A.: Accountability: definition and relationship to verifiability. In: ACM Conference on Computer and Communications Security, pp. 526–535 (2010)
38. Feigenbaum, J., Jaggard, A.D., Wright, R.N.: Towards a formal model of accountability. In: Proceedings of the 2011 Workshop on New Security Paradigms Workshop (2011)
39. Bauer, L., Garriss, S., Reiter, M.K.: Detecting and resolving policy misconfigurations in access-control systems. In: Symposium on Access Control Models and Technologies (SACMAT), pp. 185–194 (2008)
40. Vaughan, J.A., Jia, L., Mazurak, K., Zdancewic, S.: Evidence-based audit. In: Computer Security Foundations Symposium (CSF), pp. 177–191 (2008)
41. Lampson, B.W.: Computer security in the real world. IEEE Computer 37(6), 37–46 (2004)
42. Cederquist, J.G., Corin, R., Dekker, M.A.C., Etalle, S., den Hartog, J.I., Lenzini, G.: Audit-based compliance control. Int. J. Inf. Sec. 6(2-3), 133–151 (2007)
43. Cheng, P.C., Rohatgi, P., Keser, C., Karger, P.A., Wagner, G.M., Reninger, A.S.: Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control. In: Proceedings of the IEEE Symposium on Security and Privacy (2007)
44. Cheng, P.C., Rohatgi, P.: IT Security as Risk Management: A Research Perspective. IBM Research Report RC24529 (April 2008)
45. Zhao, X., Johnson, M.E.: Access governance: Flexibility with escalation and audit. In: Hawaii International International Conference on Systems Science (HICSS), pp. 1–13 (2010)
46. Zhang, N., Yu, W., Fu, X., Das, S.K.: Towards effective defense against insider attacks: The establishment of defender's reputation. In: IEEE International Conference on Parallel and Distributed Systems, pp. 501–508 (2008)
47. Band, S.R., Cappelli, D.M., Fischer, L.F., Moore, A.P., Shaw, E.D., Trzeciak, R.F.: Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis. Technical Report CMU/SEI-2006-TR-026, Carnegie Mellon University (December 2006)

A Experimental Outcomes Supporting Predictions

Table 1. P, α for $e = 1000$

R_{ext}	P	α
5 to 443	0	0
443 to 3900	6.5	1

Table 2. P, α for constant (0) difference in R_{int}, R_{ext}

R_{ext} and R_{int}	P	α
5 to 26	0	0
26 to 3900	10	0.2

Table 3. P, α for varying C

C	P	α
10	6.5	1
20	6.5	1
30	7.0	0.85
40	7.5	0.65
50	8.0	0.5
60	9.5	0.25
70	10.0	0.2

Table 4. P, α for $I = 50$

R_{ext}	P	α
5	0	0
670	0	0.1
685	0	0.35
714	0	0.6
748	0	0.85
790	0	1.0

Table 5. P, α for varying μ

μ	P	α
1.0	10.0	0.3
1.2	9.5	0.35
1.3	9.5	0.35
1.40	9.0	0.45
1.5	9.0	0.45
1.6	8.5	0.5
1.7	8.5	0.5

A Game Theoretical Analysis of Lemonizing Cybercriminal Black Markets

SingRu (Celine) Hoe[3], Murat Kantarcioglu[1], and Alain Bensoussan[2]

[1] University of Texas at Dallas, USA
[2] University of Texas at Dallas, USA, The Hong Kong Polytechnic University, HK, and Ajou University, Korea
[3] Texas A&M University-Commerce, USA
{muratk,alain.bensoussan}@utdallas.edu, hoceline02@yahoo.com

Abstract. It is known that cybercriminal black markets that trade in illicit digital goods and services belong to markets for lemons due to the information asymmetry of quality of goods and services between sellers and buyers. Based on the seminal work of Akerlof [1], Franklin et al. [3] suggests that *"Lemonizing the Market"* be an effective way to crack down the well-developed cybercriminal underground market. In our work, we provide a game theoretical framework to analyze whether cybercriminal black markets can be effectivitely lemonized. First, we investigate if signaling quality through an extra provision, such as the offer of trial periods or a money-back guarantee, observed in this marketplace (see the Panda security report [6]) provides cybercriminals selling real illicit data (i.e., the peach group) with a solution to address the lemon market problem. We also study the relation between the market lemonization and the cost constraint on seller's implementation of signaling of quality. We find that, because of the effectiveness of resolving quality uncertainty through perfect signaling of quality, law enforcement cannot clamp down the operation of this underground economy through *"Lemonizing the Market"* by joining the group of *"pure lemons"*, that is, joining the group of sellers with no crime products offered to sell (i.e., ripoff sellers). If no information of quality is disclosed, the market demand shrinks increasingly as lemons in the market increases. However, to secure the market demand, cybercriminals with real illicit data for sale always attempt to implement quality signaling to single out their quality products, accepting a higher amount of cost constraints on applying quality signaling as the portion of lemons in the market escalates.

Recognizing that lemonizing the market through magnifying the group of ripoff sellers could not effectively shut down these underground economic activities, we extend our model to consider that law enforcement: (1) joins the "peach group" to add **"noisiness"** to quality signals, and (2) takes advantage of transactions with buyers of crime products to locate these cybercriminals for arrest. To make quality signaling noisy, law enforcement produces quality fake data with the same extra provision, such as trial periods, offered by cybercriminals selling real illicit data to lure buyers; however, once the deal proceeds further, buyers get nothing. We call law enforcement playing *"fake peaches"* in this scenario. We find that the presence of "fake peaches" makes quality signaling imperfect,

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 60–77, 2012.

which in turn disincentivizes sellers' use of quality signaling to secure demand for staying in business. When incorporating the possibility of arresting buyers of crime products, we find that the market demand decreases as a result of buyers' fear of getting arrested, leading to declines in sellers' profits. Therefore, playing "fake peaches" coupled with effectively tracing buyers for arrest is the most efficient way for law enforcement to make the signaling strategy ineffective for sellers of crime products, leading the market to resort to markets for lemons.

1 Introduction

It has been known for awhile that there is a thriving and diverse online market economy that trades in illicit digital goods and services. One of the most observed marketplaces is the *Internet Relay Chat* (IRC) market where buyers and sellers meet to buy, sell and trade goods and services in support of criminal activities such as credit card fraud, identity theft, spamming, phishing, online credential theft, the sale of botnets (compromised hosts), and among others. This online underground economy appears to mirror the real economy, which is a function of supply and demand. Cybercriminals with ability to produce malware form the market supply. They make money by selling illegal digital products and services. Miscreants who do not possess sophisticated capability of creating malware themselves and would like to profit from various forms of e-crimes, such as financial fraud, phishing, and spamming, can enter the market and buy the goods and services necessary to launch e-crimes.

Since this is a market for illicit trading, sellers' unwillingness of exposing the identity makes it almost impossible for buyers to verify the quality of perspective purchases. Therefore, this is definitely a market for lemons [1] because incentives exist for the seller to pass off low-quality goods as higher-quality ones. The existence of ripoff sellers, who do not provide the goods or services for which they have been paid, is almost assured [3,4]. In his seminal paper of "*Markets for Lemons*", using the specific example of used-car markets, Akerlof [1] shows that, due to the adverse selection resulting from asymmetrical information of product quality between buyers and sellers, good cars (peaches) may be driven out of the market by the lemons (bad cars), thus leading to market failure.

Recognizing the lemon effect, Franklin et al. [3] and Herley et al. [4] suggest that an inexpensive and simple countermeasure to this underground economy would be to ***lemonize*** the market, contrary to previous approaches focusing on standard law enforcement activities such as locating and disabling hosting infrastructure or identifying and arresting market participants [7]. The basic idea is to drive "**REAL**" crime products and services out of the market by lemons. Franklin et al. [3] propose two methods of *"Lemonizing the Market"*. One is *Sybil Attack* and the other is *Slander Attack*. Sybil attack is to add lemons to markets by first establishing Sybil identities to disrupt the market by undercutting its participant verification system, and then selling deceptive products. Slander attack is to lemonize the market by eliminating the status of sellers with crime products. For such an attack, government agency forms a

group of buyers to slander sellers with crime products. This would then lead to a market decrease by either forcing sellers with crime data to pass off deals or having buyers to deal with sellers of no crime products. Although being aware of the potential feasibility of halting the underground economy by lemonizing the market, Herley et al. [4] argues that lemonizing the market alone is not powerful enough to stop the market. They argue that lemonizing the market could possibly only discontinue the lower-tier underground economy, but not the upper-tier level. They reason that the upper-tier underground economy is presented by well-organized cybercriminals who properly integrate specialists in all channels needed to complete e-crimes, thus no incentives to trade goods and services in the cybercriminal black market.

In view of fraud transactions arising from asymmetrical information of quality, channel administrators in the IRC market provide participant verification services to facilitate honest transactions [3,4]. However, a seller's verified status does not seem to truly assure buyers of high quality of products traded. As a result, sellers with real illicit data attempt to seek more costly and perfect certification (signal) to differentiate themselves from ripoff sellers. According to Panda Security report[6], to credibly signal their product quality, sellers put some efforts on quality signaling through providing warranty periods, free replacements of invalid data ...etc. Studying the operation of this underground economy thus involves two themes, markets for lemons and the impact of signaling of product quality on the presence of markets for lemons. After Akerlof's [1] seminal work, economists have extended the study of effects of quality uncertainty on the market mechanism to consider if certification of the quality (i.e., the credible signaling of product quality) could be high-quality sellers' potential exit from markets for lemons; see for example, Viscusi [8], De and Nabar [2], Strausz [5] and among others.

Unlike [3,4] who analyze the disruption of cybercriminal black markets through lemonizing the market with intuitive reasoning, we develop a theoretically sound economic model for a comprehensive analysis of this underground economy, specifically focusing on the issue of the lemonization of this black market. In our model, sellers with real illicit data (i.e., the peach group) make their movement first by deciding whether or not to implement quality signaling, and buyers of crime products optimize their purchase decisions based on sellers' quality disclosure action as well as the prices offered. We aim to provide the rationale supporting sellers' implementation of signaling of quality through an extra provision, such as the offer of trial periods, observed in the marketplace of this underground economy (See [6]). As evidenced by the real world observation, perfect signaling of quality serves as an effective way for cybercriminals selling real illicit data to exit from lemon markets; we are, therefore, motivated to explore the question *"Can law enforcement destroy the perfect signaling effect to make quality signaling an ineffective strategy for cybercriminals selling real illicit data to leave markets for lemons?"* In response to this question, we propose to have law enforcement join the "peach group" in order to add **"noisiness"** to quality signals. Law enforcement accomplishes this task by producing quality

fake data to offer the same provision such as trial periods used as quality signaling by cybercriminals selling real illicit data to lure buyers; however, once the deal proceeds further, buyers get nothing. We call law enforcement playing *"fake peaches"* in this scenario. Due to the presence of "fake peaches", products with signaling can no longer be viewed as products of high quality, thus failing the perfect signaling of quality. This strategy is similar to the "Slander Attack" by [3] where law enforcement destroys reputation of sellers with crime products to make transactions between them and buyers impossible to reach. However, invoking the strategy of *"playing fake peaches"* by law enforcement brings an extra advantage since law enforcement can better locate buyers of crime products, with whom they establish business relations, for arrest. For example, with the offer of trial periods for purchasing stolen credit card accounts, law enforcement can trace buyers when they try to profit financially from those credit card accounts produced by the government authority during the trial period. The fear of getting arrest and the punishment following the arrest for buyers of crime products should have detrimental effect on the market demand, and we explore these issues in our study. We summarize the main results and state the contribution in the following section.

1.1 Summary of Main Results and Contribution of This Study

Our main findings include:

1. Lemonizing the cybercriminal black market by simply joining the group of ripoff sellers would not create market failure since cybercriminals selling real illicit data would remain business activities in this black market by implementing quality signaling to help them exit from markets for lemons.
2. Lemonizing the peach group of the market turns out to be effective both in discouraging sellers with real illicit data to use quality signaling to separate themselves from ripoff sellers and in reducing the profitability of sellers with real illicit data. Both of these effects may in turn lead to market failure of this underground economy.
3. Taking advantage of transactions with buyers of crime products to locate them for arrest when law enforcement joins the sellers of this black market may turn out to be the most effective way for the government authority to disrupt this black market.
4. The severity of punishment for buyers of crime products when getting arrested may significantly reduce their willingness to participate in this underground economic activity even if the probability of getting arrested is not huge. The important message to the government authority is that a credible promise of severe punishment for buyers of crime products when getting arrested may be a powerful tool to discourage this underground economy.

The remainder of the paper is organized as follows. In Section 2, we describe the model used in this study. In Section 3, we explore various scenarios by varying some parameter values of the model proposed in Sect. 2. In Section 4, we make qualitative discussions and comparisons for each scenario, and make suggestions for the government authority. We present concluding remarks in Section 5.

2 The Model

2.1 Background Information

We use a game theoretic framework to analyze whether cybercriminal black markets can be effectively lemonized. The game is formulated as follows:

1. Sellers with real illicit data simultaneously decide whether or not to signal the high quality of their products by incurring unit cost $c > 0$.
2. Based on sellers' quality disclosure action and the prices offered, buyers of crime products make their purchase decisions that maximize their utility.
3. Law enforcement joins the black market to introduce "noisiness" to the quality signal.

Before proceeding to analyses of the game, we propose the cost function of sellers and the utility function of buyers in the subsequent two sections.

2.2 Cost Function of Sellers

There are two types of suppliers of illicit products and services in the underground economy. These suppliers offer otherwise homogenous products and services at two different levels, a high one (q_H) and a low one (q_L). We assume that low-quality products (i.e., q_L) are lemons, that is, products with "zero" values. In other words, they are products provided by ripoff sellers. Each seller offers only one quality level, and the quality offered by any supplier is known only by the supplier himself.

Sellers' costs consist of fixed costs and costs of signaling if sellers decide to exercise quality signaling to differentiate their products. The fixed cost refers to the original setup cost and the cost for market entry. The setup cost relates to the cost that cybercriminals spend in creating malware to obtain illicit goods for trading whereas the market entry cost may be justified as the cost/effort of obtaining a verified status from the system administrator in the IRC market.

Assuming homogenous cost functions among cybercriminals, we can characterize the seller's cost function as:

$$C(x, \mathbf{q}) = b(\mathbf{q}) + \theta x c \tag{1}$$

where

- $\mathbf{q} = \{q_H, q_L\}$, $b(\mathbf{q})$ is the fixed cost. It depends on the quality level, and we have $b(q_H) > b(q_L)$ since a ripoff seller incurs zero cost for the setup.
- $\theta = \{0, 1\}$ is a decision variable. It indicates if sellers make efforts to signal the quality of their products by incurring unit cost $c > 0$. Without loss of generality, we can assume ripoff sellers will take $\theta = 0$ since it would be impossible for them to offer quality confirmation through some provision such as trial periods.
- x is the quantity sold.

2.3 Utility Function of Consumers

Buyers do not know the true quality of products/services traded, but they know the distribution of high quality products and low quality products traded in the market and observe sellers' signals (quality disclosure). We consider buyers are homogenous in their preferences. Buyers can gain from the transaction if the products purchased are real illicit data, and they may incur sufficient losses if they get arrested as a result of the transaction.

The consumer's utility is assumed to be of quadratic form given:

$$U(p,x,\theta;r,\rho,\xi_1,\xi_2) = -\frac{\beta}{2}[\theta\rho+(1-\theta)r]x^2+\alpha[\theta\rho+(1-\theta)r]x \\ -\alpha_1[\theta\xi_1+(1-\theta)\xi_2]x-px \qquad (2)$$

where

- r with $0 \le r \le 1$ is the portion of high quality products in the market.
- ρ with $0 \le \rho \le 1$ as the the correlation between the signaling and the high quality of the product.
- ξ_1 and ξ_2 with $\xi_1 > \xi_2$ denote [1] the probability of getting caught when buyers purchase products/services from sellers with quality disclosure and from those without quality disclosure respectively.
- p is the price.
- $\alpha > 0$, $\beta > 0$, and $\alpha_1 > 0$ are arbitrary constants. We can take α and β as some multiple of the unit value of the high quality product worth to the buyer whereas α_1 may be viewed as the unit punishment if getting arrested.
- x and θ are as previously defined.

The term, $\alpha_1(\theta\xi_1+(1-\theta)\xi_2)x$, in (2) captures the effect of the buyer's utility decrease due to arrest.

Remark 1. From (2), we observe that the linear form of utility function is a special case of the quadratic form proposed by letting $\beta = 0$.[2]

2.4 Demand Function of Consumers

Based on sellers' quality disclosure action and the prices offered, buyers make their purchase decisions that maximize their utility. By solving utility maximization from transaction using (2), consumer's demand based on θ is given:

$$D(p,\theta;\rho,r,\xi_1,\xi_2) = \frac{\alpha[\theta\rho+(1-\theta)r)]-\alpha_1[\theta\xi_1+(1-\theta)\xi_2]-p}{\beta[\theta\rho+(1-\theta)r]} \qquad (3)$$

[1] Law enforcement gets better channels to track down buyers of illicit products when playing fake peaches. For example, by the offer of trial periods, law enforcement obtains extra periods of locating buyers. Therefore, we have $\xi_1 > \xi_2$.

[2] In the analyses provided in Sect. 3, we include the discussion of the case with $\beta = 0$.

3 Analyses of Various Scenarios

In this section, we first proceed to analyze the rationale of sellers' implementation of signaling of quality through an extra provision, such as the offer of trial periods, observed in the marketplace of this underground economy. We show why sellers with real illicit data can prevent themselves from going out of business if the government tries to lemonize the market by adding more pure lemons. To obtain comprehensive results, we study two forms of consumer utility function, a quadratic form in Sect. 3.1 and a linear form in Sect. 3.2. Since perfect signaling of quality turns out to be an effective way for sellers with real illicit data to exit from markets for lemons, we study if the government could destroy the effectiveness of such an alternative by introducing **"noisiness"** to the quality signal. Moreover, we also explore the situation that buyers now face the probability of getting arrested for their transactions since law enforcement can have a better chance of tracking them down based on the established business relations. We present these topics in detail in Sect. 3.3 and Sect. 3.4.

3.1 The Basic Case I: Perfect Signaling (i.e., $\rho = 1$), No Government Participation and Arrest (i.e., $\xi_1 = \xi_2 = 0$), and Quadratic Utility Function of Consumers

In this section, we are interested in how perfect signaling (i.e., $\rho = 1$) could help cybercriminals with real illicit data to sell immune from going out of business that may arise due to the lemonization of the market. By (2), the consumer's utility is given:

$$\begin{aligned} U(p, x, \theta; r, 1, 0, 0) &= U(p, x, \theta; r) \\ &= -\frac{\beta}{2}[\theta + (1-\theta)r]x^2 + \alpha[\theta + (1-\theta)r]x - px \end{aligned} \tag{4}$$

Therefore, consumer's demand based on θ is given:

$$D(p, \theta; r) = \frac{\alpha[\theta + (1-\theta)r)] - p}{\beta[\theta + (1-\theta)r]} \tag{5}$$

For (5) to hold with economic meaning, we require $\alpha \geq p$ and $\alpha r \geq p$.

Proposition 1. *$\theta = 1$ if $c \leq \alpha(1 - \sqrt{r})$.*

Proof. If the seller with real illicit data implements the strategy of signaling of quality, that is, he takes $\theta = 1$, by (5), the demand is given

$$D(p, 1) = \frac{\alpha - p}{\beta} \tag{6}$$

The expected profit for the seller with real illicit data taking $\theta = 1$ is

$$\pi(p, 1) = pD(p, 1) - cD(p, 1) - b(q_H) \tag{7}$$

Through optimization, sellers with real illicit data taking $\theta = 1$ will charge $p(1) = \frac{\alpha + c}{2}$. The profits that the seller with real illicit data makes will be

$$\pi(p(1), 1) = \frac{(\alpha - c)^2}{4\beta} - b(q_H) \tag{8}$$

Next if the seller with real illicit data does not undertake the strategy of signaling of quality, that is, he takes $\theta = 0$, by (5), the demand is given

$$D(p, 0; r) = \frac{\alpha r - p}{\beta r} \tag{9}$$

The expected profit for the seller with real illicit data taking $\theta = 0$ is

$$\pi(p, 0) = pD(p, 0; r) - b(q_H) \tag{10}$$

Through optimization, the seller with real illicit data taking $\theta = 0$ will charge $p(0) = \frac{\alpha r}{2}$. The profits that the seller with real illicit data makes will be

$$\pi(p(0), 0; r) = \frac{\alpha^2 r}{4\beta} - b(q_H) \tag{11}$$

The seller with real illicit data will take $\theta = 1$ if and only if

$$\pi(p(1), 1) \geq \pi(p(0), 0),$$

that is,

$$\frac{(\alpha - c)^2}{4\beta} - b(q_H) \geq \frac{\alpha^2 r}{4\beta} - b(q_H),$$

leading to

$$(\alpha - c)^2 \geq \alpha^2 r;$$

we have

$$c \leq \alpha(1 - \sqrt{r}). \tag{12}$$

We note that $c \geq \alpha(1 + \sqrt{r})$ is excluded because it would lead to negative demands. Also, consistent with rational economic behavior, equation (12) shows that there is no need to exercise extra efforts to signal product quality if $r = 1$, that is, if no ripoff sellers exist. □

Proposition 1 indicates that sellers with real illicit data choose to implement the strategy of perfect signaling of quality when the signaling cost is below a threshold level, $c^* = \alpha(1 - \sqrt{r})$, determined by the multiple of unit value of the high quality product worth to the buyer, α, as well as the portion of the high quality product in the market, r. c^* decreases as the portion of sellers with real illicit data increases; that is, c^* is negatively related to r. In other words, as the portion of lemon products traded in the market increase, sellers

with real illicit data would be willing to accept higher cost of signaling in order to secure potentially shrank demand from the increasing lemonization of the market. In addition, the higher the unit value of the high quality product worth to the buyer, the larger the signal cost that sellers with real illicit data would be willing to accept to differentiate themselves from ripoff sellers. The quality signaling strategy serves as an effective alternative for the sellers with real illicit data to exit from lemon markets. The strategy of cracking down the cybecriminal black market through *"Lemonizing the Market"* with "Sybil Attack" proposed by Franklin et al. [3] may not work well if the "Sybil Attack" only involves joining the group of lemons in this market.

In sum, within an acceptable range of cost of perfect signaling, sellers with real illicit data find themselves an exit from lemon markets. That is, consistent with the phenomenon observed in the real world, suppliers of illicit products and services try to sell their products through an extra provision, such as the offer of trial periods, to differentiate themselves from lemons ([6]).

3.2 The Basic Case II: Perfect Signaling (i.e., $\rho = 1$), No Government Participation and Arrest (i.e., $\xi_1 = \xi_2 = 0$), and Linear Utility Function of Consumers (i.e., $\beta = 0$)

To gain a more robust argument that exercising perfect quality disclosure provides an effective exit from lemon markets for the cybercriminals with real illicit data to sell in this black market, we consider another form of consumer utility function. We take the linear form of consumer utility function by setting $\beta = 0$ in (2). That is the utility function now takes the form:

$$U(p, x, \theta; r) = \alpha[\theta + (1 - \theta)r]x - px \tag{13}$$

Proposition 2. *$\theta = 1$ if $c \leq \alpha(1 - r)$.*

Proof. We proceed to find the best responses. Define γ, δ_1, δ_0 as the probability of entering the deal for buyers, for cybercriminals with real illicit data exercising quality disclosure and for cybercriminals with real illicit data not exercising quality disclosure respectively. The buyer's best response given the seller's strategy is:

$$BR_B(\theta, p) = \begin{cases} \gamma = 1 & \text{if } \theta = 1 \text{ and } p \leq \alpha \\ \gamma = 1 & \text{if } \theta = 0 \text{ and } p \leq \alpha r \\ \gamma = 0 & \text{otherwise} \end{cases} \tag{14}$$

Denoting BR_1, and BR_0 as the best responses for the sellers with real illicit data with quality disclosure and for those without disclosure respectively, we have:

$$BR_1(p) = \begin{cases} \delta_1 = 1 & \text{if } p \geq c \\ \delta_1 = 0 & \text{otherwise} \end{cases} \tag{15}$$

$$BR_0(p) = \begin{cases} \delta_0 = 1 & \text{if } p \geq 0 \\ \delta_0 = 0 & \text{otherwise} \end{cases} \tag{16}$$

In a competitive market, cybercriminals selling real illicit data exercising the quality disclosure will charge up to α and those without exercising the quality disclosure will charge up to αr. Suppose $\alpha > c$, otherwise it is impossible to have cybercriminals selling real illicit data to take disclosure (signaling) actions to differentiate themselves since it's too costly for them to do so. Therefore the expected unit profit for cybercriminals selling real illicit data exercising the quality disclosure is

$$\alpha - c - b(q_H),$$

and that for cybercriminals selling real illicit data without exercising the quality disclosure is

$$\alpha r - b(q_H).$$

Cybercriminals with real illicit data will take disclosure action to differentiate themselves if and only if

$$\alpha - c - b(q_H) \geq \alpha r - b(q_H),$$

leading to

$$c \leq \alpha(1 - r). \tag{17}$$

□

As presented in Proposition 1, sellers with real illicit data choose to exercise perfect quality signals when the signal cost is below a given level, $c^* = \alpha(1 - r)$, determined by the multiple of unit value of the high quality product worth to the buyer and the degree of the market lemonization.

Combined Proposition 1 and Proposition 2, the quality disclosure strategy serves as an effective alternative for the sellers with real illicit data to exit from lemon markets. That is, this conclusion is not specific to the quadratic form of consumer utility function.

Remark 2. With the linear form of consumer utility function, sellers with real illicit data will be willing to spend larger amount of money to signal product quality since we have $\alpha(1 - r) > \alpha(1 - \sqrt{r})$.

3.3 Government Joining the Peach Group, Imperfect Signaling of Quality (i.e., $0 \leq \rho < 1$), and No Arrest (i.e., $\xi_1 = \xi_2 = 0$)

From the above two sections, we recognize that lemonizing the market by purely joining the group of lemons (i.e., joining the group of ripoff sellers) will not be an effective way to crack down this black market since cybercriminals with real illicit data to sell could survive by exercising perfect signaling of quality to differentiate themselves from lemon sellers. This is what observed in the operation of current cybercriminal black markets where sellers with real illicit data make an extra provision, such as trial periods, available in order to single out the high quality of their products (See [6]).

Therefore, we ask the questions "Can law enforcement still attempt to crack down this black market through lemonizing the market? If so, how?" Since we

understand that perfect signaling of quality provides cybercriminals selling real illicit data an effective alternative to exit from markets for lemons, we propose to have law enforcement join the peach group of the market to add noisiness to the quality signal in hopes of destroying the effectiveness of signaling of quality. That is, we propose to **"Lemonize the Peach Group of the Market"**. In order to achieve this task, law enforcement produces quality fake data to offer the same extra provision such as trial periods offered by sellers with real illicit data for the purpose of signaling of quality. However, once the deal proceeds further, buyers will obtain lemon products (i.e., get noting). As a result, the signaling is noisy, and can no longer be viewed as a confirmation of high quality. That is, law enforcement is in fact a supplier of lemons, but buyers of crime data could not distinguish them from the real peach group (i.e. sellers with real illicit data). We call law enforcement "playing fake peaches" in this scenario. To focus on the exploration of the noisy impact from the presence of fake peaches, in this section, we do not consider the situation that buyers risk getting arrested arising from the fact that they may potentially carry on deals with law enforcement and get traced. By (2), the consumer's utility function is:

$$\begin{aligned} U(p,x,\theta;\rho,r,0,0) &= U(p,x,\theta;\rho,r) \\ &= -\frac{\beta}{2}[\theta\rho+(1-\theta)r]x^2+\alpha[\theta\rho+(1-\theta)r]x-px, \end{aligned} \tag{18}$$

and demand becomes:

$$D(p,\theta;\rho,r)=\frac{\alpha[\theta\rho+(1-\theta)r)]-p}{\beta[\theta\rho+(1-\theta)r]} \tag{19}$$

For (19) to hold with economic meaning, we require $\alpha\rho \geq p$ and $\alpha r \geq p$.

Proposition 3. *$\theta = 1$ if $c \leq \alpha(\rho - \sqrt{r})$, and $\theta = 0$ if $\rho < \sqrt{r}$.*

Proof. If the seller with real illicit data exercises quality signaling, that is, he takes $\theta = 1$, by (19), the demand is given

$$D(p,1;\rho)=\frac{\alpha\rho-p}{\beta\rho} \tag{20}$$

The expected profit for the seller with real illicit data taking $\theta = 1$ is

$$\pi(p,1;\rho)=pD(p,1;\rho)+cD(p,1;\rho)-b(q_H) \tag{21}$$

Through optimization, the seller with real illicit data taking $\theta = 1$ will charge $p(1)=\dfrac{\alpha\rho+c}{2}$. The profits that the seller with real illicit data makes will be

$$\pi(p(1),1;\rho)=\frac{(\alpha\rho-c)^2}{4\beta}-b(q_H) \tag{22}$$

Next if the seller with real illicit data does not exercise quality signaling, that is, he takes $\theta = 0$, by (11), the profits that the seller with real illicit data makes

will be

$$\pi(p(0), 0; r) = \frac{\alpha^2 r}{4\beta} - b(q_H) \tag{23}$$

The seller with real illicit data will take $\theta = 1$ if and only if

$$\pi(p(1), 1; \rho) \geq \pi(p(0), 0; r),$$

that is,

$$\frac{(\alpha\rho - c)^2}{4\beta} - b(q_H) \geq \frac{\alpha^2 r}{4\beta} - b(q_H),$$

leading to

$$(\alpha\rho - c)^2 \geq \alpha^2 r;$$

we have

$$c \leq \alpha(\rho - \sqrt{r}) \tag{24}$$

Again, as in the case of perfect signaling of quality, $c \geq \alpha(\rho + \sqrt{r})$ is excluded because it would lead to negative demand. Now, from (24), we observe that if $\rho \leq \sqrt{r}$, sellers with real illicit data will choose not to exercise quality disclosure signals. □

Proposition (3) indicates that sellers with real illicit data choose to exercise quality disclosure signals when the unit signaling cost is below a threshold level, $c^* = \alpha(\rho - \sqrt{r})$. Compared with Proposition (1), the threshold c^* now is additionally determines by ρ, the correlation between the signaling and the high quality of the product, other than the multiple of unit value of the high quality product and the portion of the high quality products in the market. The inverse relation between acceptable c and r still persists. However, now sellers with real illicit data may choose not to exercise quality disclosure signals if the effectiveness of quality disclosure through signaling is less than a threshold level, $\rho^* = \sqrt{r}$, determined by the portion of the high quality products in the market.

Remark 3. Due to the joining of law enforcement, we always have $r < 1$, indicating that zero noisiness of the signaling, i.e., $\rho = 1$, is not required for sellers with real illicit data to implement signaling strategies.

Proposition 4. *The result presented in Proposition 3 is not utility function form specific. The only difference appears in the impact of r on the acceptable unit signal cost.*

Proof. Proceed similarly to the proof of Proposition 2 for a linear form of consumer utility function by taking $\beta = 0$ in (18), we arrive at

$$c \leq \alpha(\rho - r). \tag{25}$$

□

3.4 Government Joining Groups of Peaches and Lemons, Imperfect Signaling of Quality (i.e., $0 \leq \rho < 1$), and Potential Arrest (i.e., $\xi_1 > \xi_2 \geq 0$)

It is true that law enforcement would have better chance to arrest buyers of crime products when they have business relations established with these cybercriminlas. For example, with the offer of trial periods for purchasing stolen credit card accounts, law enforcement can trace buyers when they try to profit financially from those credit card accounts produced by the government authority during the trial period. As a result, we extend the model to consider the situation where law enforcement takes advantage of better locating buyers of crime products, with whom they have business relations, for arrest. That is, buyers now may risk getting caught when purchasing the crime data from the black market since now law enforcement attends the market and intends to trace them for arrest. By (2), the consumer's utility function is modified as:

$$U(p,x,\theta;\rho,r,\xi_1,\xi_2) = -\frac{\beta}{2}[\theta\rho+(1-\theta)r]x^2 + \alpha[\theta\rho+(1-\theta)r]x \\ -\alpha_1[\theta\xi_1+(1-\theta)\xi_2]x - px, \tag{26}$$

The demand becomes:

$$D(p,\theta;\rho,r,\xi_1,\xi_2) = \frac{\alpha[\theta\rho+(1-\theta)r)] - \alpha_1[\theta\xi_1+(1-\theta)\xi_2] - p}{\beta[\theta\rho+(1-\theta)r]} \tag{27}$$

Proposition 5. *$\theta = 1$ if $c \leq \alpha(\rho-\sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}})$. And $\theta = 0$ if $\rho < \sqrt{r}$ or $\rho \leq \frac{\alpha_1}{\alpha}(\xi_1 - \frac{\xi_2}{\sqrt{r}}) + \sqrt{r}$ for $\rho > \sqrt{r}$ and $\xi_1 \geq \frac{\xi_2}{\sqrt{r}}$.*

Proof. If the seller with real illicit data exercises signaling, that is, he takes $\theta = 1$, by (27), the demand is given

$$D(p,1;\rho,\xi_1) = \frac{\alpha\rho - \alpha_1\xi_1 - p}{\beta\rho} \tag{28}$$

The expected profit for sellers with real illicit data taking $\theta = 1$ is

$$\pi(p,1;\rho,\xi_1) = pD(p,1;\rho,\xi_1) - cD(p,1;\rho,\xi_1) - b(q_H) \tag{29}$$

Through optimization, the seller with real illicit data taking $\theta = 1$ will charge $p(1) = \dfrac{\alpha\rho - \alpha_1\xi_1 + c}{2}$. The profits that the seller with real illicit data makes will be

$$\pi(p(1),1;\rho) = \frac{(\alpha\rho - \alpha_1\xi_1 - c)^2}{4\beta} - b(q_H) \tag{30}$$

Next if the seller with real illicit data does not exercise signaling, that is, he takes $\theta = 0$, the demand is given

$$D(p,0;r,\xi_2) = \frac{\alpha r - \alpha_1\xi_2 - p}{\beta r} \tag{31}$$

The expected profit for the seller with real illicit data taking $\theta = 0$ is

$$\pi(p, 0; r, \xi_2) = pD(p, 0; r, \xi_2) - b(q_H) \tag{32}$$

Through optimization, the seller with real illicit data taking $\theta = 0$ will charge $p(0) = \dfrac{\alpha r - \alpha_1 \xi_2}{2}$. The profit that the seller with real illicit data makes will be

$$\pi(p(0), 0; r, \xi_2) = \frac{(\alpha r - \alpha_1 \xi_2)^2}{4\beta r} - b(q_H) \tag{33}$$

The seller with real illicit data will take $\theta = 1$ if and only if

$$\pi(p(1), 1; \rho, \xi_1) \geq \pi(p(0), 0; r, \xi_2),$$

that is,

$$\frac{(\alpha\rho - \alpha_1\xi_1 - c)^2}{4\beta} - b(q_H) \geq \frac{(\alpha r - \alpha_1 \xi_2)^2}{4\beta r} - b(q_H),$$

leading to

$$(\alpha\rho - \alpha_1\xi_1 - c)^2 \geq \frac{(\alpha r - \alpha_1 \xi_2)^2}{r};$$

we have

$$c \leq \alpha(\rho - \sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}}). \tag{34}$$

From (34), we observe that if $\rho \leq \sqrt{r}$ or $\rho \leq \frac{\alpha_1}{\alpha}(\xi_1 - \frac{\xi_2}{\sqrt{r}}) + \sqrt{r}$ for $\rho > \sqrt{r}$ and $\xi_1 \geq \frac{\xi_2}{\sqrt{r}}$, sellers with real illicit data will choose not to exercise quality disclosure signals. □

Proposition 5 shows that the threshold of the acceptable unit signaling cost for sellers with real illicit data, $c^* = \alpha(\rho - \sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}})$, now is additionally determined by ρ, the correlation between the signaling and the high quality of the product, the probability of buyers' getting caught, ξ_1 and ξ_2, and the buyer's unit loss of getting caught, α_1, other than the multiple of unit value of the high quality product and the portion of the high quality products in the market. The inverse relation between acceptable c and r still persists.

Compared with Proposition 3, cybercriminals with real data have higher chances of not choosing to exercise quality disclosure strategy since the threshold of acceptable unit signaling cost, $c^* = \alpha(\rho - \sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}})$, is smaller since $\alpha(\rho - \sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}}) \leq \alpha(\rho - \sqrt{r})$ for $\xi_1 \geq \frac{\xi_2}{\sqrt{r}}$. It indicates that quality disclosure signaling is much less effective for cybercriminals selling real illicit data to assist them to exit from lemon markets.

Table 1. Summary of Acceptable Unit Signaling Cost (USC) under Various Scenarios

Scenarios	Threshold of USC	Implement Signaling
Perfect Signaling and No Government Participation and Arrest	$c^* = \alpha(1 - \sqrt{r})$	$c \leq c^*$
Imperfect Signaling, Government Joining Peach Groups, and No Arrest	$c^* = \alpha(\rho - \sqrt{r})$	$c \leq c^*$
Imperfect Signaling, and Government Participation and Arrest	$c^* = \alpha(\rho - \sqrt{r}) - \alpha_1(\xi_1 - \frac{\xi_2}{\sqrt{r}})$	$c \leq c^*$

Table 2. Summary of Expected Profits under Various Scenarios

Scenarios	Expected Profits of Implement Signaling	Expected Profits of Not Implement Signaling
Perfect Signaling and No Government Participation and Arrest	$\frac{(\alpha - c)^2}{4\beta} - b(q_H)$	$\frac{\alpha^2 r}{4\beta} - b(q_H)$
Imperfect Signaling, Government Joining Peach Groups, and No Arrest	$\frac{(\alpha\rho - c)^2}{4\beta} - b(q_H)$	$\frac{\alpha^2 r}{4\beta} - b(q_H)$
Imperfect Signaling, and Government Participation and Arrest	$\frac{(\alpha\rho - \alpha_1\xi_1 - c)^2}{4\beta} - b(q_H)$	$\frac{(\alpha r - \alpha_1\xi_2)^2}{4\beta} - b(q_H)$

4 Discussion

Tables 1 and 2 summarize the results of acceptable unit signaling cost and expected profits with and without signaling for cybercriminals selling real illicit data under various scenarios presented in Sect. 3.

First, we discuss the relationship between the seller's acceptable unit signaling cost and law enforcement's strategy in participating in this cybercriminal market. From Table 1, we find that, in all scenarios, the multiple of unit value of high quality products worth to buyers, α, and the degree of lemonization of the market, $(1 - r)$ where r is the portion of high quality products in the market, determines the acceptable unit signaling cost for cybercriminals selling real illicit data. In all cases, the acceptable unit signaling cost is positively proportionate to the multiple of unit value of high quality products worth to buyers and positively proportionate to the degree of lemonization of the market.

When perfect signaling of quality is broken down by law enforcement's joining the peach group, imperfect signaling of quality disincentivizes sellers' willingness

to implement quality signaling since the acceptable unit signaling cost is smaller compared with the situation where the signaling is perfect. Based on the results reported in Table 1, we observe if law enforcement could make quality signaling **"sufficiently noise"**, that is, the effectiveness of quality signaling, ρ, is so ineffective such that $\rho \leq \sqrt{r}$, sellers would disrupt their usage of quality signaling to secure demand for staying in business in this black market. If law enforcement additionally takes the advantage of better locating buyers of crime products for arrest when interacting with these cybercriminals, it would further discourage sellers' undertaking quality signaling for staying in business. The reason is that the acceptable unit signaling cost further decreases as the market demand decreases reflecting the buyers' fear of getting arrested and the corresponding punishment (See Table 2) following the arrest.

Next we explore the relations between the seller's expected profits and law enforcement's strategy in participating in this cybercriminal market. From Table 2 we see that, the multiple of unit value of high quality products, α, is positively related to sellers' expected profits since the market demand is positively proportionate to α. Furthermore, we observe that ***"Lemonizing the Peach Group of the Market"*** would not only discourage sellers' use of quality signaling to separate themselves from ripoff sellers but also reduce their profitability. The latter would in turn reduce sellers' willingness to stay in this underground economic activity. Furthermore, the introduction of possibility of arresting buyers of crime products and the resulting severity of punishment would amplify sellers' unwillingness to stay in this underground economic activity due to non-profitability arising from declines in the market demand.

In what follows, we summarize and highlight the relations discussed above:

1. Law enforcement should not attempt to lemonize the cybercriminal black market by simply joining the group of ripoff sellers. Cybercriminals selling real illicit data would continue business activities in this black market by implementing quality signaling to secure demand and thus stay in business.
2. Law enforcement could discourage the black market activities by bringing down the value of high quality products worth to buyers since the acceptable unit signaling cost and the expected profit are positively proportionate to the multiple of unit value of high quality products. Lowering the value of high quality worth to buyers may force sellers with real illicit data to abandon the signaling strategy, which is used as an effective alternative for them to exit from markets for lemons. On the other hand, doing so would also disincentivize sellers' participation in this underground economy due to low profitability. Law enforcement may achieve this goal by requiring financial institutions for example to implement stricter security and monitoring guards in authorizing credit card transactions, making buyers of stolen credit card accounts harder to profit financially.
3. Law enforcement should attempt to lemonize the cybercriminal black market by joining the "peach group" to add "noisiness" to quality signals that cybercriminals selling illicit data adopt to single out their quality products. Doing so may not only force cybercriminals selling real illicit data to

abandon the use of quality signaling to differentiate themselves from ripoff sellers in hopes of staying in business but also reduce the expected profits from transactions. The former would lead the cybercriminal black market to resort to markets for lemons, and the latter may cause cybercriminals selling real illicit data to leave the business due to non-profitability.

4. Disrupting the effectiveness of quality signaling by ***"Lemonizing the Peach Group of the Market"*** while simultaneously taking advantage of locating buyers of crime products for arrest when interacting with these cybercriminals is the most efficient way to possibly shut down this black market.
5. The government authority should introduce a credible promise of severe punishment for buyers of crime products when getting arrested.

5 Conclusion

We adopt a game model to perform a comprehensive study of online market economy that trades in illicit digital goods and services. We show why sellers with real illicit data can prevent themselves from going out of business if the government tries to lemonize the market by adding more pure lemons, and analyze factors that impact the cost constraint on seller's implementation of signaling of quality. We show that law enforcement could destroy the effectiveness of quality signaling by introducing **"noisiness"** to the quality signaling, thus having cybercriminals selling real illicit data abandon the exploitation of quality signaling to resolve quality uncertainty, which in turn leads the market to resort to markets for lemons. Furthermore, we find that incorporating the force of law enforcement for potentially tracking down buyers of crime products simultaneously when law enforcement lemonizes the market by playing fake peaches and pure lemons is the most effective way to disrupt the utilization of quality signaling of sellers with real illicit data in hopes of securing businesses in this underground economic activity. Moreover, the corresponding reduction in expected profits, followed by noisy signaling and the possibility of arresting buyers of crime products, may also disincentivize cybercriminals selling real illicit data to participate in business activities in this market. Finally, a credible promise of severe punishment for buyers of crime products when getting arrested may be a powerful tool to discourage this underground economy.

In this study, to focus on the effect of lemonization, we do not consider the cost of law enforcement either when implementing the lemonization of the market by playing "fake peaching" or when implementing tracking techniques to locate buyers of crime products for arrest given established business deals. In the future work, we will extend the framework to consider these costs to solve the optimization problem for law enforcement with and without imposing budget constraints in order to provide the government authority with socially optimal actions to adopt in face of taking down this underground economic activity.

Acknowledgements. This work was partially supported by The Air Force Office of Scientific Research MURI-Grant FA-9550-08-1-0265 and Grant FA-9550-08-1-0260, National Institutes of Health Grant 1R01LM009989, National Science Foundation (NSF) Grant Career-CNS-0845803, NSF Grants CNS-0964350, CNS-1016343 and CNS-1111529, WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31 - 20007) and by the Research Grants Council of HKSAR (PolyU 5001/11P).

References

1. Akerlof, G.A.: The market for " Lemons": Quality uncertainty and the market mechanism. The Quarterly Journal of Economics 84(3), 488–500 (1970)
2. De, S., Nabar, P.: Economic implications of imperfect quality certification. Economics Letters 37, 333–337 (1991)
3. Franklin, J., Paxson, V., Perrig, A., Savage, S.: An inquiry into the nature and causes of the wealth of internet miscreants. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (2007)
4. Herley, C., Florêncio, D.: Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. In: Proceedings of the Workshop on the Economics of Information Security, WEIS (2009)
5. Strausz, R.: Separating Equilibria with Imperfect Certification. Working paper (2010)
6. The Cyber-Crime Black Market: Uncovered, http://press.pandasecurity.com/wp-content/uploads/2011/01/The-Cyber-Crime-Black-Market.pdf
7. United States Secret Service. United states secret services operation rolling stone nets multiple arrests: Ongoing undercover operation targets cyber fraudsters. Press Release (March 2006)
8. Viscusi, W.K.: A Note on " Lemons" Markets with Quality Certification. Bell Journal of Economics 9, 277–279 (1978)

Computing the Nash Equilibria of Intruder Classification Games*

Lemonia Dritsoula[1], Patrick Loiseau[2], and John Musacchio[1]

[1] UC Santa Cruz, Santa Cruz, USA
[2] EURECOM, Sophia-Antipolis, France
{lenia,johnm}@soe.ucsc.edu, patrick.loiseau@eurecom.fr

Abstract. We investigate the problem of classifying an intruder of two different types (spy or spammer). The classification is based on the number of file server and mail server attacks a network defender observes during a fixed window. The spammer naively attacks (with a known distribution) his main target: the mail server. The spy strategically selects the number of attacks on his main target: the file server. The defender strategically selects his classification policy: a threshold on the number of file server attacks. We first develop parameterized families of payoff functions for both players and analyze the Nash equilibria of the non-cooperative nonzero-sum game. We analyze the strategic interactions of the two players and the tradeoffs each one of them faces: The defender chooses a classification threshold that balances the cost of missed detections and false alarms while the spy seeks to hit the file server as much as possible while still evading detection. We give a characterization of the Nash equilibria in mixed strategies, and demonstrate how the Nash equilibria can be computed in polynomial time. We give two examples of the general model, one that involves forensics on the side of the defender and one that does not. Finally, we evaluate how investments in forensics and data logging could improve the Nash equilibrium payoff of the defender.

Keywords: Nash equilibria, intruder classification, polynomial complexity.

1 Introduction

Classifying an attacker is not an easy task. In almost every network security situation, the defender has limited resources. The defender needs to distinguish between different types of attackers (spy or spammer) and decide what kind of actions should be triggered. For example, an attack on a mail server by a spammer (causing at most network congestion) should be treated differently than an attack on a file server (possibly involving identity theft). Knowing that a defender is trying to classify attackers, the strategic spy is likely to change the way he attacks in order to make it more difficult to be classified as a spy.

* This work was supported by AFOSR grant FA9550-09-1-0049.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 78–97, 2012.

The paper focuses on the specific situation of a network with one mail server and one file server. However, the model developed is very flexible and independent of the underlying architecture. In particular, the model can fit many situations in which a strategic attacker has an incentive to blend in with or be mistaken for other more benign kinds of attackers or even as legitimate users. We first develop a generic model that guarantees the NE computation in polynomial time, but also provides insights on how the players' NE strategies are derived. Computing the NE is a tractable process, even for larger N. We propose two characteristic examples, in which the defender has made different security investments in forensic mechanisms. The analysis of these models provide us with a qualitative and quantitative view on how changes on the network parameters affect the strategies of the players. We explore the relation between the NE strategies of the two strategic players and the behavior of the non-strategic one, and we evaluate the defender's expected gain after investing in forensics.

1.1 Related Work

There is a substantially increasing body of work in the game theory community that explores and suggests equilibrium solutions in security games (see e.g., a recent survey in [4]). Particularly relevant to the present paper is the growing body of work is on the topic of intrusion detection. In [5], Alpcan and Başar present a security game between an attacker and an intrusion detection system and address some of the basic security tradeoffs, e.g., false alarms versus undetected attackers. They also provide insightful overview on how different network parameters affect the performance of the intruder detection system. Our game-theoretic framework investigates a more complex game and provides analytic expressions for the defender's NE strategies for any network size. We also investigate the way the nonstrategic player influences the spy's strategy.

Gueye, Walrand, and Anantharam [7,8] have investigated the structure of the Nash equilibria in a network topology game, in which attacker and defender select which links to attack and use for communication respectively. They consider a special case of nonzero-sum games, in which the different term in the players' payoffs is controlled only by the one player. In these games, one player optimizes his payoff against his opponent who has optimized his payoff as well. Such games are easier to analyze than general nonzero-sum games, and they give interesting insights on the strategies of the two players. Our work is using a similar payoff formulation in a different setting: the defender selects a threshold on file server attacks (not a set of links to use) and there are two different types of attackers.

This is a follow up of our recent work [1], where we investigated and characterized the Nash equilibria of a game similar to what we study in the current paper, but with a much more specific form assumed for the payoff functions of the players. In the current paper, we provide a comprehensive way to derive the strategies of the two players who have generalized payoffs. We further provide evaluation results between two different models, with different assumptions on the resources available to the defender.

1.2 Summary of Contributions

In summary, our contributions are the following:

- We propose a generic game-theoretic model to analyze the interactions between two adversaries: a classifier (defender) and a malicious attacker when a nonstrategic spammer is present (Sec. 2).
- We show how to derive the NE strategies in polynomial time (Sec. 3).
- We develop two models for intrusion detection (Sec. 4.1 and 4.2).
- By comparing the above two models, we extract key insights on the expected gain from the defender's investment in forensic capabilities. This is an example of how our methodology can be used to evaluate how changes in the strategic situation affect the equilibrium payoffs of the players. We also investigate the impact of the different network parameters on the resulting NE strategies (Sec. 5).

2 Game Model

The network we consider consists of a defender and two servers that he monitors for potential attacks: a File Server (FS) with sensitive data and a Mail Server (MS) with contents of inferior importance. The defender observes the number of hits from an attacker to each server for a fixed classification window of N time slots. The attacker may be a spy or a spammer with probabilities p and $1-p$ respectively.

The defender is a strategic player that seeks to correctly classify the potential intruder by selecting a threshold T. When he observes T or more hits on the FS, he classifies the attacker as spy; otherwise as spammer. The spy is also a strategic player that selects the number of FS attacks H he will perform. He seeks to attack the FS as frequently as possible, while evading detection. The spammer is a non-strategic player that mostly attacks the MS and adds noise to the network. He also attacks the FS Z times (Z follows a known distribution). For example, the spammer can be modeled to follow the binomial distribution, with a small probability θ_0 to attack the FS at each time slot.

Our solution captures a more general setting than the one presented above. We only require that the attacker has some cost function if he gets detected or missed. We describe the model around the example scenario in which there are two servers, one of which is of primary interest to the strategic attacker (the file server) in order to be more concrete. However, the model we develop is quite general and applicable to many settings in which there is a target of special interest to a strategic attacker but who is incentivized to mix his attack across other targets to make classification more difficult.

Notational Conventions
We use "min[$\boldsymbol{v}$]" to denote the minimum element of a vector $\boldsymbol{v}$ and "minimize" when we minimize a specific expression over some constraints. We use the *prime* sign ($'$) for transpose of matrices and vectors. All vectors are assumed to be

column vectors and are denoted by bold lowercase letters (e.g., $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$). For matrix notation we use capital greek letters (e.g., Λ). The indicator function is denoted by $\mathbb{1}_{\text{cond}}$; it is equal to 1 if "cond" holds and is equal to 0 otherwise. The column vector of ones of length N is denoted by $\mathbf{1}_N$ and the matrix of ones of dimensions $N \times M$ is denoted by $1_{N\times M}$. The norm of a vector $\boldsymbol{x}$ of length N, denoted by $\|\boldsymbol{x}\|$, always refers to the 1-norm, i.e, $\|\boldsymbol{x}\| = |x_1| + |x_2| + \ldots + |x_N|$. An overview of our notation is shown in Table 1.

2.1 Spy's Cost Function

The spy cost depends on the defender's classification decision and the number of FS hits. We denote the spy cost function when the spy is detected (i.e., $T \leq H$) by $D(H)$ and when the spy is not detected (i.e., $T > H$) by $M(H)$. Thus, the overall spy cost function is expressed as follows

$$J_A(T, H) = D(H) \cdot \mathbb{1}_{T\leq H} + M(H) \cdot \mathbb{1}_{T>H},$$

or by making the appropriate simplifications

$$J_A(T, H) = [D(H) - M(H)] \cdot \mathbb{1}_{T\leq H} + M(H).$$

2.2 Defender's Payoff Function

We now describe how the defender's expected payoff function is constructed. We distinguish two cases:

- With probability p the defender faces a spy. If the defender correctly classifies the intruder as a spy (i.e., $T \leq H$), he gains $D(H)$. If the defender misclassifies the spy (i.e., $T > H$), he gains $M(H)$.
- With probability $1-p$ the defender faces a spammer. If the defender correctly classifies the intruder as spammer (i.e., $T \geq Z$), he does not benefit. The defender incorrectly classifies the spammer with probability $\phi(T) = \Pr\{Z \geq T\}$ and in this case there is a false alarm penalty c_{fa}.

Combining these two scenarios, the defender's expected payoff is

$$\tilde{U}_D(T, H) = p \cdot [D(H) \cdot \mathbb{1}_{T\leq H} + M(H) \cdot \mathbb{1}_{T>H}] - (1 - p) \cdot c_{\text{fa}} \cdot \phi(T). \quad (1)$$

By scaling the above function, we get

$$U_D(T, H) = D(H) \cdot \mathbb{1}_{T\leq H} + M(H) \cdot \mathbb{1}_{T>H} - \mu(T),$$

where $\mu(T) = \dfrac{1-p}{p} \cdot c_{\text{fa}} \cdot \phi(T)$. Function $\phi(T)$ is decreasing on T, and we assume that it is strictly decreasing: $\Pr\{Z \geq T\} > \Pr\{Z \geq T + 1\}$.

2.3 Players' Interactions

For a classification window of N time slots, the spy has $N+1$ available actions (attack the file server $H \in \{0,\dots,N\}$ times). The defender has $N+2$ available actions (select $T \in \{0,\dots,N+1\}$ as the classification threshold).

We model our problem as a nonzero-sum game. However, the defender's payoff is different from the spy's cost function in only one term $\mu(T)$ that depends *only* on the defender's strategy ($U_D(T,H) = J_A(T,H) - \mu(T)$). These games are known as almost zero-sum games or quasi zero-sum games.

We are interested in Nash equilibria in mixed strategies for the following reason. In most cases the spy's best response to a threshold T is to attack the file server a number of times H just below T (unless the cost of being detected is so low that the spy prefers to attack as often as possible even while being detected). Likewise, in most cases, the defender's best response to an H is to choose the threshold T to be just equal with H in order to have the lowest false alarm penalty possible while still detecting the spy. Since each player wants to pick "lower" than the other, there is no pure strategy Nash equilibrium in most cases of interest, so we consider mixed strategies. The spy chooses a distribution vector $\boldsymbol{\alpha}$ on the allowed number of FS hits; $\boldsymbol{\alpha}$ is a vector of size $N+1$ (with non-negative elements that sum to 1). Similarly, the defender chooses a distribution vector $\boldsymbol{\beta}$ on the collection of possible thresholds T; $\boldsymbol{\beta}$ is a vector of size $N+2$ (with non-negative elements that sum to 1).

Let $\tilde{\Lambda}$ be a $(N+1)\times(N+2)$ matrix representing the spy's (pure) strategies' cost. We express the cost matrix of the attacker as

$$\tilde{\Lambda} = \begin{bmatrix} \delta(0) & & 0 \\ & \ddots & \\ 0 & & \delta(N) \end{bmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & 1 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \ddots & \vdots & \vdots \\ \vdots & & & \ddots & 0 & \vdots \\ 1 & \cdots & \cdots & \cdots & 1 & 0 \end{pmatrix} + \begin{bmatrix} M(0) & \cdots & M(0) \\ M(1) & \cdots & M(1) \\ \vdots & & \vdots \\ M(N-1) & \cdots & M(N-1) \\ M(N) & \cdots & M(N) \end{bmatrix},$$

where $\delta(H) = D(H) - M(H)$. Each row i of $\tilde{\Lambda}$ corresponds to one of the $N+1$ possible spy strategies. For instance, row "0" corresponds to spy attacking the FS 0 times (or $H=0$), row "1" corresponds to spy selecting $H=1$ and so on. Each column of $\tilde{\Lambda}$ corresponds to one of the $N+2$ possible defender strategies.

Let $\tilde{\Lambda}$ be defined as above, and $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, be the spy and defender distributions respectively. The attacker cost can be written as $\boldsymbol{\alpha}'\tilde{\Lambda}\boldsymbol{\beta}$ and the defender payoff can be written as $\boldsymbol{\alpha}'\tilde{\Lambda}\boldsymbol{\beta} - \boldsymbol{\mu}'\boldsymbol{\beta}$, where $\boldsymbol{\mu}$ is a strictly decreasing vector (componentwise) with μ_i be the i^{th} component of vector $\boldsymbol{\mu}$. Certain computations are simplified by using a matrix with only positive entries. We define

$$\Lambda = \tilde{\Lambda} + K \cdot 1_{(N+1)\times(N+2)},$$

where $K > 0$ is such that every matrix element is positive. Since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ must each sum to 1, the expressions $\boldsymbol{\alpha}'\Lambda\boldsymbol{\beta}$ and $\boldsymbol{\alpha}'\Lambda\boldsymbol{\beta} - \boldsymbol{\mu}'\boldsymbol{\beta}$ are respectively the

Table 1. Main Notations

p	probability for spy	$\boldsymbol{\alpha}$	spy's mixed strategy
$D(H)$	detection cost	$\boldsymbol{\beta}$	def. mixed strategy
$M(H)$	missed detection cost	$\boldsymbol{\mu}$	false alarm cost vector
$\delta(H)$	$D(H) - M(H)$	$\theta(\boldsymbol{\beta})$	defendability of $\boldsymbol{\beta}$
H	spy's strategy (# FS hits)	Λ	cost matrix of spy
T	def. strategy (threshold)	s	first tight inequality
Z	# of FS hits by spammer	f	last tight inequality

attacker cost and defender payoff shifted by a constant. Adding a constant to the players' payoff does not affect their best responses, thus from here on we will consider these expressions to be the payoff functions of each player.

3 Game-Theoretic Analysis

It is known that every finite game (finite number of players with finite number of actions for each player) has a mixed-strategy Nash equilibrium [12]. Our game is finite, thus it admits a NE in mixed strategies. In a two-player game, the players' strategies $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are a NE if each player's stategy is a best response to the other player's mixed strategy.

3.1 Best Response Analysis

Here is a roadmap of the subsequent analysis.

- Lemma 1 leads to the proof of Theorem 1 on the maximization of the defender's payoff in NE.
- Lemma 2 defines the simplified problem $\Lambda\boldsymbol{x} \geq \mathbf{1}_{N+1}$, $\boldsymbol{x} \geq \mathbf{0}$.
- Theorem 2 introduces the algorithm to compute the players' NE strategies, under certain conditions. To prove the validity of this algorithm, we prove a series of Lemmata (3 – 9).

Lemma 1. *A spy who plays a best response to a defender strategy $\boldsymbol{\beta}$ has a cost* $\min[\Lambda\boldsymbol{\beta}]$.

Proof. For a given defender strategy $\boldsymbol{\beta}$ the minimum attacker cost is achieved by putting positive probability only on strategies corresponding to the minimum entries of the vector $\Lambda\boldsymbol{\beta}$ (recall Λ is positive). Thus the spy's optimal cost is $\min[\Lambda\boldsymbol{\beta}]$. □

Definition 1 (Defendability). *The defendability of a mixed strategy $\boldsymbol{\beta}$ is defined as*

$$\theta(\boldsymbol{\beta}) = \min[\Lambda\boldsymbol{\beta}] - \boldsymbol{\mu}'\boldsymbol{\beta}. \tag{2}$$

It corresponds to the defender's payoff when the attacker plays a best response to $\boldsymbol{\beta}$.

The defendability is a measure of how good a strategy $\boldsymbol{\beta}$ is. The defendability is similar to the notion of vulnerability in [7]. An interesting fact that arises from the definition of the defendability is that the defender's payoff when the attacker plays a best response against $\boldsymbol{\beta}$ depends only on $\boldsymbol{\beta}$.

Theorem 1. *A defender-attacker strategy pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a NE, if and only if the defendability $\theta(\boldsymbol{\beta})$ is maximal.*

Proof. The intuition behind this proof is twofold. First, we prove that if the defender does not maximize his defendability, then the attacker' optimization problem in NE (in order to make the defender indifferent among the strategies in his support and not want to use strategies outside) is infeasible. Second, we prove that the attacker's optimization problem, when the spy limits the defender to the defendability $\theta(\boldsymbol{\beta})$ yields a spy cost $\min[\Lambda\boldsymbol{\beta}]$, i.e., the same one as if the spy was not limiting the defender to the defendability (as in Lemma 1). (Sketch, see [1] for full proof). □

Definition 2. ***Polyhedron****: A polyhedron is the solution set of a finite number of linear equalities and inequalities.*

Tight constraint*: An inequality constraint is tight, if it holds as an equality; otherwise, it is loose.*

Extreme point*: A point $\boldsymbol{x}$ of a polyhedron is said to be extreme if there is no $\boldsymbol{x}'$ whose set of tight constraints is a strict superset of the set of tight constraints of $\boldsymbol{x}$.*

"Corresponds"*: We say that a point on the polyhedron $\boldsymbol{x}$ corresponds to strategy $\boldsymbol{\beta}$, if $\boldsymbol{\beta} = \boldsymbol{x}/\|\boldsymbol{x}\|$.*

Lemma 2. *The defendability is maximized amongst strategies $\boldsymbol{\beta}$ corresponding to the extreme points of the polyhedron defined by $\Lambda\boldsymbol{x} \geq \mathbf{1}_{N+1}$, $\boldsymbol{x} \geq \mathbf{0}$.*

Proof. As we proved in Theorem 1, in NE, the defender maximizes the defendability, that is, he solves the following "defendability LP"

$$\begin{aligned} \underset{\boldsymbol{\beta}, z}{\text{maximize}} \quad & -\boldsymbol{\mu}'\boldsymbol{\beta} + z \\ \text{subject to} \quad & z \cdot \mathbf{1}_{N+1} \leq \Lambda\boldsymbol{\beta} \\ & \mathbf{1}'_{N+2} \cdot \boldsymbol{\beta} = 1,\ \boldsymbol{\beta} \geq \mathbf{0}. \end{aligned} \tag{3}$$

The solution for z is $z = \min[\Lambda\boldsymbol{\beta}]$ (finite and positive since Λ positive). We can make the following transformation $\boldsymbol{x} = \frac{1}{z} \cdot \boldsymbol{\beta}$, with $\|\boldsymbol{x}\| = \frac{1}{z} \cdot 1$ and get the LP

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & -\boldsymbol{\mu}'\boldsymbol{x} + 1 \\ \text{subject to} \quad & \Lambda \cdot \boldsymbol{x} \geq 1_{N+1},\ \boldsymbol{x} \geq \mathbf{0}. \end{aligned} \tag{4}$$

The intuition behind the proof is that we can rewrite the above LP (4) in terms of $\boldsymbol{\beta}$, and then impose the equality constraint $\sum_{i=0}^{i=N+1} x_i = 1$. Then the objective is linear in $\boldsymbol{\beta}$. We prove that the extreme points of the inequalities in $\boldsymbol{x}$ correspond to the extreme points of $\boldsymbol{\beta}$ in the above LP. The formal proof can be found in [1]. □

3.2 Form of Players' Strategies in NE

Since the defender maximizes his defendability in NE, the defender must solve the LP given by (4). There exist polynomial-time algorithms to solve linear programming problems [10]. Our approach not only guarantees a low-complexity algorithm to compute the NE strategies of the two players, but it also provides essential intuition about how and why the defender is behaving the way he behaves.

Defender's NE Strategy. As we saw in Lemma 2, the best response strategy of the defender is found by looking at the extreme points of the polyhedron $\Lambda \boldsymbol{x} \geq \mathbf{1}_{N+1}, \boldsymbol{x} \geq \mathbf{0}$. We call the first type "inequality" constraints and the second type "positivity" constraints. We have $N+1$ "inequality"- and $N+2$ "positivity" constraints. We assume that $\delta(H), M(H)$ are positive functions. If they are not, we can add a constant parameter and render them positive without affecting the Nash equilibria of the game. Writing down the "inequality" constraints, we get

$$\begin{aligned}
\delta(0) \cdot x_0 + M(0)\|\boldsymbol{x}\| &\geq 1 \\
\delta(1) \cdot (x_0 + x_1) + M(1)\|\boldsymbol{x}\| &\geq 1 \\
&\vdots \\
\delta(i) \cdot (x_0 + \cdots + x_i) + M(i)\|\boldsymbol{x}\| &\geq 1 \\
&\vdots \\
\delta(N) \cdot (x_0 + x_1 + \ldots + x_N) + M(N)\|\boldsymbol{x}\| &\geq 1.
\end{aligned}$$

Our goal is to eliminate nonextreme and other points that are not selected by a defender in NE, so that we reduce the number of points we have to check. Depending on the nature of the attacker's cost functions δ and M, we are able to compute analytically the defender's NE strategies in polynomial time. We will consider the following conditions for the subsequent analysis.

Condition 1: $\forall s \in \{0, \ldots, N-1\}$, where $\Delta_k g(i) = g(i+k) - g(i)$,

1. $\Delta_1 \delta(s+1) \geq \Delta_1 \delta(s)$, and
2. $\Delta_1 M(s+1) \geq \Delta_1 M(s)$

Condition 1 suggests that the difference between the cost of the spy upon detection and his cost upon misdetection is non decreasing with respect to H. It also suggests that the marginal cost for the spy when he is not detected is smaller for smaller values of H. We use this condition to prove that the inequalities are violated, unless there is a contiguous block of tight inequalities (see Lemma 5).

Condition 2:

1. $D(H)$ is monotone with respect to the number of attacks to the FS H.
2. $M(H)$ is a decreasing function with respect to H.

Theorem 2 summarizes our results on the computation of Nash equilibria for the intruder classification games.

Theorem 2. *Under condition 1, there exists a defender NE strategy that satisfies a contiguous block (by index) of tight inequalities (indexed s through f). Under condition 2, the contiguous block will finish at index $f = N$, or we only have pure NE. When $f = N$, we search amongst different β_{N+1} for the defender strategies $\boldsymbol{\beta}$ that maximize the defendability. The remaining vector $\boldsymbol{\beta}$ is the result of the solution of the tight inequalities with the maximum allowed integer s. The attacker's strategy is the solution of the LP given by (8).*

We now develop a series of lemmata that lead to Theorem 2. The proof is provided in the Appendix.

Lemma 3. *Two points $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ on the polyhedron, with $\|\boldsymbol{x_1}\| = \|\boldsymbol{x_2}\|$, correspond to defender strategies $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_2}$ respectively with detection cost $\min[\Lambda\boldsymbol{\beta_1}] = \min[\Lambda\boldsymbol{\beta_2}]$ against a best responding attacker.*

Proof. We showed in Lemma 2 that a defender NE strategy $\boldsymbol{\beta}$ corresponds to one of the extreme points of a polyhedron defined by $\Lambda\boldsymbol{x} \geq \mathbf{1}_{N+1}$, $\boldsymbol{x} \geq \mathbf{0}$, with $\|\boldsymbol{x}\| = 1/z = 1/\min[\Lambda\boldsymbol{\beta}]$. Thus, for the same the norm $\|\boldsymbol{x}\|$, we get the same detection cost against a best responding attacker, i.e., $\min[\Lambda\boldsymbol{\beta_1}] = \min[\Lambda\boldsymbol{\beta_2}]$. □

Lemma 4. *If $\|\boldsymbol{x_1}\| = \|\boldsymbol{x_2}\|$ and $\boldsymbol{\mu}'\boldsymbol{x_1} < \boldsymbol{\mu}'\boldsymbol{x_2}$, then $\boldsymbol{x_1}$ corresponds to a defender strategy $\boldsymbol{\beta_1}$ with a better defendability, i.e., $\theta(\boldsymbol{\beta_1}) > \theta(\boldsymbol{\beta_2})$.*

Proof. From the definition of the defendability (Definition 2), we get

$$\begin{aligned}\theta(\boldsymbol{\beta_1}) - \theta(\boldsymbol{\beta_2}) &= \min[\Lambda\boldsymbol{\beta_1}] - \boldsymbol{\mu}'\boldsymbol{\beta_1} - (\min[\Lambda\boldsymbol{\beta_2}] - \boldsymbol{\mu}'\boldsymbol{\beta_2}) \\ &= \boldsymbol{\mu}'\boldsymbol{\beta_2} - \boldsymbol{\mu}'\boldsymbol{\beta_1} \qquad (5) \\ &> 0, \qquad (6)\end{aligned}$$

where (5) results from Lemma 3 (since $\|\boldsymbol{x_1}\| = \|\boldsymbol{x_2}\|$, $\min[\Lambda\boldsymbol{\beta_1}] = \min[\Lambda\boldsymbol{\beta_2}]$), and (6) follows the assumption $\boldsymbol{\mu}'\boldsymbol{x_1} < \boldsymbol{\mu}'\boldsymbol{x_2}$. The point $\boldsymbol{x_1}$ corresponds to a defender strategy $\boldsymbol{\beta_1}$ with a smaller false alarm cost, i.e., $\boldsymbol{\mu}'\boldsymbol{\beta_2} > \boldsymbol{\mu}'\boldsymbol{\beta_1}$. Hence $\theta(\boldsymbol{\beta_1}) > \theta(\boldsymbol{\beta_2})$. □

Lemma 5. *Under condition 1, an extreme point $\boldsymbol{x}$ corresponding to a defender NE strategy $\boldsymbol{\beta}$ satisfies exactly one contiguous set (of indices) of tight inequalities.*

The proof is provided in the Appendix. Let s, f be the indices of the first and last tight inequalities (of the contiguous block of tight ones) respectively.

Lemma 6. *Under condition 1, an extreme point $\boldsymbol{x}$ that corresponds to a defender NE strategy $\boldsymbol{\beta}$ has zeros before s and after $f + 1$, i.e.,*

$$x_i = 0, \forall i \in \{0, \ldots, s-1\} \cup \{f+2, \ldots, N+1\}.$$

Proof. We first show that $x_i = 0, \forall i < s$. If $\exists i \in \{0, \ldots, s-1\}$, s.t. $x_i > 0$, we reduce x_i to $\hat{x}_i$ until either $\hat{x}_i = 0$ or i^{th} inequality is tight, and increase x_{i+1} by the same amount. We maintain $\|\boldsymbol{x}\|$ constant, and in case that $x_{i+1} > 0$ we get one more tight constraint. Thus the original point is not extreme, as we can find another point whose tight constraints is a strict superset of those of the original. In case that $x_{i+1} = 0$, the new $\hat{\boldsymbol{x}}$ corresponds to a defender NE strategy with a better defendability.

We now show that $x_i = 0, \forall i > f+1$. If $\exists i \in \{f+2, \ldots, N+1\}$, s.t. $x_i > 0$, we reduce x_i until $\hat{x}_i = 0$ and increase x_{f+1} by the same amount. We again keep the norm $\|\boldsymbol{x}\|$ constant but $\hat{\boldsymbol{x}}$ has one more tight constraint, thus $\boldsymbol{x}$ was not extreme. □

Lemma 7. *In any Nash equilibrium, under conditions 1 and 2,*

1. *$f = N$, when D is non increasing.*
2. *$f = N$ or $s = f$, when D is increasing.*

The proof is provided in the Appendix.

Lemma 8. *Amongst different defender mixed strategies $\boldsymbol{\beta}$ with the same component β_{N+1}, the detection cost against a best responding attacker is the same, under conditions 1 and 2.*

Proof. By Lemma 7 under conditions 1 and 2, $f = N$ or we have pure strategies NE. By Lemma 3, the points with the same norm $\|\boldsymbol{x}\|$ correspond to defender strategies with the same detection cost ($\min[\Lambda\boldsymbol{\beta}] = 1/\|\boldsymbol{x}\|$). Scaling the last tight inequality N with the norm and since $\boldsymbol{\beta}$ is a distribution, we get $\delta(N)(1 - \boldsymbol{\beta}_{N+1}) + M(N) = \frac{1}{\|\boldsymbol{x}\|}$. Thus for the same β_{N+1}, the norm is the same, which results in the same detection cost against a best responding attacker. □

Lemma 9. *Under conditions 1 and 2, amongst defender mixed strategies with different s and same β_{N+1}, the defendability is maximal when s is maximal.*

The proof is provided in the Appendix.

Note 1. There might be more than one maximizers of the defendability. In this case, we have multiple NE strategies for the defender. But, by small perturbations of the game parameters we can prevent ties. For instance, let $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ be two maximizers of the defendability, with different detection costs ($\min[\Lambda\boldsymbol{\beta}_1] > \min[\Lambda\boldsymbol{\beta}_2]$, and false alarm costs $\boldsymbol{\mu}'\boldsymbol{\beta}_1 < \boldsymbol{\mu}'\boldsymbol{\beta}_2$. Perturbing the $\boldsymbol{\mu}$ such that $\boldsymbol{\mu}'\boldsymbol{\beta}_1 \geq \boldsymbol{\mu}'\boldsymbol{\beta}_2$, we get a unique maximizer of the defendability. We can follow the same approach to break the ties among multiple defender strategies $\boldsymbol{\beta}$. This way, Nash's theorem of NE existence guarantees an attacker's NE strategy.

Attacker's NE Strategy. Having computed and analyzed the defender NE strategy, we now explore the spy's attack strategy. Let Λ_r be a reduced matrix, after keeping only the defender strategies in his support (columns). Similarly, let μ_{r} be the reduced false alarm vector. Then the payoff of the defender must be the same for all strategies in his support, and greater (or equal) with his respective

payoff fox all strategies outside his support. Thus, the attacker is solving the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & 0 \\ \text{subject to} \quad & \boldsymbol{\alpha}' \cdot \Lambda \leq \theta \cdot \mathbf{1}_{N+2} + \boldsymbol{\mu}, \\ & \boldsymbol{\alpha}' \cdot \Lambda_r = \theta \cdot \mathbf{1} + \boldsymbol{\mu}_r \\ & \mathbf{1}'_{N+1} \cdot \boldsymbol{a} = 1, \\ & \boldsymbol{a} \geq \mathbf{0}. \end{aligned} \tag{7}$$

Since this is an LP, it can be solved in polynomial time [10]. Using the Big M method (with M big) we can transform the above problem into the following one (that is more robust with respect to noise and / or small perturbations).

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{maximize}} \quad & -M(\alpha_s + \alpha_x) \\ \text{subject to} \quad & \boldsymbol{\alpha}' \cdot \Lambda \leq \theta \cdot \mathbf{1}_{N+2} + \boldsymbol{\mu} \\ & \boldsymbol{\alpha}' \cdot \Lambda_r + \alpha'_s \cdot 1_{(N+1)\times R} \leq \theta \cdot \mathbf{1}'_R + \boldsymbol{\mu}_r \\ & \mathbf{1}'_{N+1} \cdot \boldsymbol{a} + \alpha_x \leq 1, \\ & \boldsymbol{a} \geq \mathbf{0}, \alpha_s \geq \mathbf{0}, \alpha_x \geq 0. \end{aligned} \tag{8}$$

To solve problem (8) we use `CVX`, a package for specifying and solving convex programs [13,14]. `CVX` is using the simplex method to find the solution. From the Nash Equilibrium Theorem, we know that a solution exists, since the attacker will play a best response.

Depending on the degrees of freedom N and the number of defender NE strategies that are given nonzero probability in NE (R), the above procedure might give a unique or multiple $\boldsymbol{\alpha}$. This $\boldsymbol{\alpha}$ must be a valid probability distribution (sum to one and have nonnegative elements) for if otherwise, it would contradict Nash's existence theorem.

4 Evaluation with Model Examples

In this section, we present two characteristic examples of the above general problem and evaluate them in terms of the expected defender NE payoff.

4.1 Example Model 1

In the first model, which is analyzed in [1], the spy's cost function in case of detection is $D(H) = c_{\mathrm{d}} - H \cdot c_{\mathrm{a}}$. There is a constant cost c_{d} associated with the detection and a benefit proportional to the number of attacks H. In case of missed detection, the spy gets the benefit from the attacks, without suffering from the detection cost, thus $M(H) = -H \cdot c_{\mathrm{a}}$, where c_{a} is the cost associated with a single FS attack. The spy cost is

$$J_A(T, H) = c_{\mathrm{d}} \cdot \mathbb{1}_{T \leq H} - c_{\mathrm{a}} \cdot H.$$

The defender's expected reward function depends on the true type of the attacker and following the general model analysis and is given by

$$U_D(T, H) = J_A(T, H) - \mu(T),$$

where $\mu(T) = \frac{1-p}{p} \cdot c_{\text{fa}} \cdot \phi(T)$. All lemmata that were proved in Sec. 3 hold since conditions 1 and 2 hold. Note that $M(H) = -H \cdot c_{\text{a}}$ is a decreasing function with respect to H and $\delta(H) = c_{\text{d}}$ is constant. Thus there is a contiguous block of tight inequalities starting at index s and finishing at index N with $x_i = 0$, $\forall i \in \{0, \ldots, s-1\}$, or we have pure NE. Furthermore, the defender's NE strategy exists amongst the two forms in Table 2. The proof is given in [1].

Table 2. Defender's strategy in NE ($\beta_m = c_{\text{a}}/c_{\text{d}}$)

#	...	β_s	β_{s+1}	...	β_N	β_{N+1}
1.	0	0	β_m	β_m	β_m	$1-(N-s)\beta_m$
2.	0	$1-(N-s)\beta_m$	β_m	β_m	β_m	0

4.2 Example Model 2

In this second variation of the model, we assume that the defender maintains some logs on the type of occurred attacks. When a spy is detected, the defender has the appropriate tools to investigate the attacker's behavior. This way, the defender has the opportunity to learn about the spy's true intentions (which specific target/information he seeks to extract from the file server), his location or identity and his future attack pattern, in case he is not immediately expelled.

Each of the H FS hits now gives the spy a benefit of c_{a} only if he evades detection. In case he is correctly identified, each FS attack yields a cost of c_{a} for the spy, as they reveal the intentions of the spy. Thus $D(H) = c_{\text{d}} + H \cdot c_{\text{a}}$, and $M(H) = -H \cdot c_{\text{a}}$, giving the spy a cost function of

$$J_A(T, H) = (c_{\text{d}} + 2c_{\text{a}} \cdot H) \cdot \mathbb{1}_{T \le H} - c_{\text{a}} \cdot H.$$

Following the analysis for the general model, the defender payoff function is

$$U_D(T, H) = J_A(T, H) - \mu(T).$$

All lemmata that were proved in Sec. 3 hold since conditions 1 and 2 hold. Note that $M(H) = -c_{\text{a}} \cdot H$ is a decreasing function with respect to H and $\delta(H) = D(H) - M(H) = c_{\text{d}} + 2Hc_{\text{a}}$ is increasing with respect to H. Thus there is a contiguous block of tight inequalities starting at index s and finishing at index N with $x_i = 0$, $\forall i \in \{0, \ldots, s-1\}$, or we have pure NE.

Defender's strategy in example model 2. After subtracting the two tight inequalities N and $N-1$, we get $\beta_{N+1} \ge 1/2$, because a tight inequality $(N-1)$ also suggests that $\beta_{N-1} \ge 0$. Thus in either case, it must be that $\beta_{N+1} \ge 1/2$. The upper bound for $\boldsymbol{\beta}$ is 1. But since the index of the first tight inequality is an

integer, only certain values of β_{N+1} result in an optimal s, which is also an integer.

By Theorem 2, given a certain β_{N+1} for the defender NE strategy, we need to find the largest possible s such that inequality s is tight and $(s-1)^{\text{th}}$ is loose, with $\beta_0 = \ldots = \beta_{s-1} = 0$. Subtracting the tight inequality N from the loose inequality $s-1$, we get $s \leq \frac{(c_a - Nc_a)+(c_d+2Nc_a)\beta_{N+1}-c_d}{c_a}$. Since s must be an integer, $\beta_{N+1} = \frac{(N-1+k)c_a+c_d}{c_d+2Nc_a}$, with k integer. Thus the search over the optimal β_{N+1} has a linear complexity with respect to N, with $\beta_{N+1}^{\min} = 1/2$, $\beta_{N+1}^{\max} = 1$ and $step = \frac{(N-1+k)c_a+c_d}{c_d+2Nc_a}$. Alternatively, solving for the integer k, we get $k^{\min} = 1 - N - c_d/c_a$ and $k^{\max} = N+1$.

5 Parameter Effects in the Game

The two previously presented models have an essential difference: While in the first model, the spy benefits from the FS attacks regardless of the defender's classification decision, in the second model, the spy benefits from the FS attacks only when he is misclassified. The assumption under the second model is that the defender has invested in forensic techniques, and is able to identify, preserve and analyze attacks within the network. This way, each FS attack reveals information about the identity and the intention of the attacker.

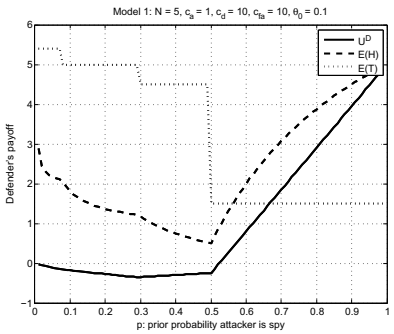
Computer forensics is a costly investment, thus the defender needs to decide under which circumstances he should develop such tools. By comparing the two above models, and essentially the defender's expected payoff in NE, we extract key insights on the expected gain from the forensics. The crucial parameters are prevalent in both models, like p, c_{fa}, θ_0, c_a, c_d. The critical difference is that the spy's cost function in case of detection $D(H)$ is decreasing in the first model and increasing in the second model, with respect to H.

Some network parameters are correlated in sets of combinations, in the sense that a change in any element on the set alters the Nash equilibrium and payoffs of the players toward the same direction. For instance, looking at the defender's payoff function for the first model, we observe that a change in p or c_{fa} affects the false alarm penalty, thus changing p and keeping c_{fa} constant will provide us with the same implications as if we kept p the same and changed c_a. Thus it makes sense to investigate the impact of only a subset of the parameters.

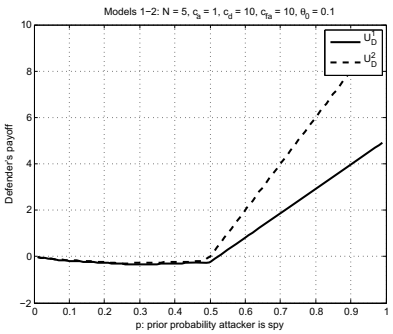
5.1 Effect of the Probability of the Spy p and False Alarm Cost c_{fa}

We expect that when p is small, the defender will suffer a small cost from potential FS attacks. As p increases it becomes more difficult to distinguish between spy and spammer and the defender's payoff will be decreasing. When p becomes larger, the defender will classify him correctly and receive a higher payoff.

Indeed, in Fig. 1(a) we observe two areas of different behavior. When $p < 0,5$, the defender payoff function is decreasing whereas it is increasing as the spy's probability reaches $p = 0.5$. We also observe that as p increases, the spy's attack



(a) NE defender payoff first decreases and then increases on p



(b) Comparison of models 1-2

Fig. 1. As p increases, the NE defender gain with model 2 increases

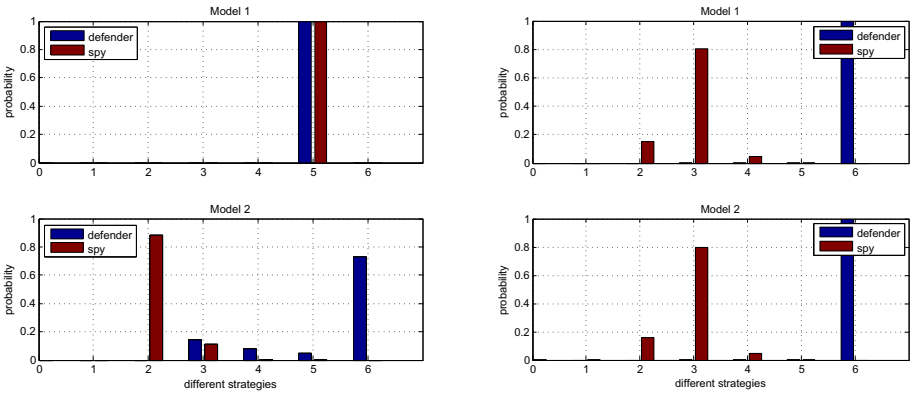
policy becomes more aggressive and the defender reduces his threshold to catch the more-probable-to-exist spy. In Fig. 1(b) we note that as p increases, the benefit of investing in forensics (and employ model 2) is an increasing function on p. We note here that the depicted values for the defender payoff are the unscaled and unshifted initial payoffs, as expressed in (1).

5.2 Effect of the Detection Cost c_{d}, the Classification Window N and Single FS Attack Cost c_{a}

In Model 1, when the cost of detection c_{d} is small compared with the maximum achievable gain from the FS attacks ($N \cdot c_{\mathrm{a}}$), the spy does not care about getting detected and is attacking with his maximum allowed strength (N times). On the contrary, in Model 2, where the spy suffers a cost proportional to his attack aggressiveness in case of detection, the spy is more conservative with his attacks. This difference is depicted in Fig. 2(a).

As we can see, in model 1, the cost of detection is so small, than the attacker always attacks N times. On the other hand, the defender selects a threshold equal to the pure strategy of the spy and detects him. If the defender selected $T = N + 1$ or $T = N - 1$ instead of $T = N$ as his classification threshold, he would miss the spy and would have smaller payoff due to the increased false alarm, respectively. In the second model, though, the spy takes into consideration the potential benefit his FS attacks would give the defender. The spy is less aggressive, and attacks fewer times. Other parameters of the game are $N = 5$, $c_{\mathrm{d}} = 1$, $c_{\mathrm{a}} = 1$, $p = 0.1$, $c_{\mathrm{fa}} = 10$, and $\theta_0 = 0.1$. We also note here that the spy's strategy is a weighted truncated binomial distribution. Every defender's strategy in his NE support gives the defender the same payoff. Thus the difference in the false alarm penalty for the different thresholds matches the difference in the misdetection cost. For instance $\Pr\{H = 3\} = \frac{(1-p)\cdot c_{\mathrm{fa}}\cdot[\phi(3)-\phi(4)]}{c_{\mathrm{d}}+2\cdot 3\cdot c_{\mathrm{a}}} = 0.1041$.

When c_{a} is small, (or else when c_{d} is most important than $N \cdot c_{\mathrm{a}}$), we observe that the two models result in the same strategies for the two players (Fig. 2(b)).



(a) small c_{d} (b) small c_{a}

Fig. 2. A Difference of the two models

Indeed, when the spy expects not to reveal a lot of information to the defender if he gets detected, he will act as if there was not risk (as in model 1). Thus, when the defender expects to lose little from the FS hits, he will avoid investing in forensics to learn more about the intentions of the spy.

5.3 Effect of the Spammer's Distribution Parameter θ_0

In these two models we have assumed a specific distribution on the FS attacks for the naive player, i.e., the spammer. Each time slot (period) of the available N time slots, the spammer attacks the FS with a frequency of θ_0. In the case that θ_0 is small (the spammer is mostly interested in attacking the MS instead of FS) the task of the defender to differentiate between the two types of attackers becomes easier.

On the contrary, if the spammer is attacking with a high θ_0 each period, then the defender is hurt from the false alarms, since he will be confused from the large number of FS hits and will classify the attacker as spy. We can see this difference in the defender NE payoff as θ_0 increases.

In Fig. 3 we see the effect of the spammer's strategy, essentially θ_0, on the two players' NE strategies. In both models, as θ_0 increases, the spy becomes more aggressive (to imitate the spammer's behavior). As θ_0 increases, the spammer attacks the FS more frequently, and it is more difficult for the defender to distinguish the two types of attacker. The spy then exploits this uncertainty to increase his payoff (by attacking more times). When θ_0 is small, the defender sets the threshold low for spy classification. As θ_0 increases, his false alarm penalty gets smaller and the defender assigns a larger weight to the "always classify as a spammer" strategy.

In Fig. 4 we see the effect of the θ_0 on the defender's payoff for the two models, for various values of the prior probability of the spy p. In model 1, depicted in Fig. 4(a), 4(b), we observe that as θ_0 increases, the defender's NE

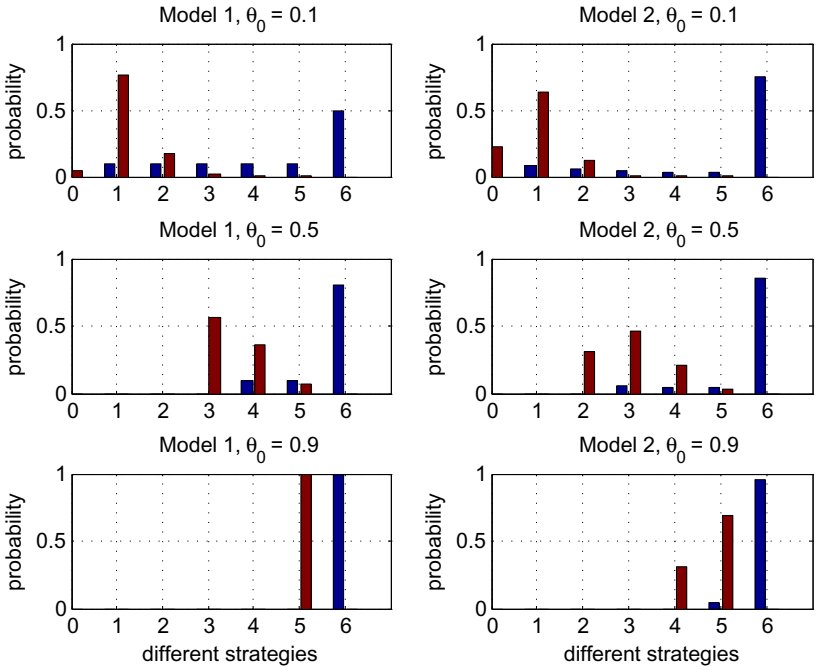
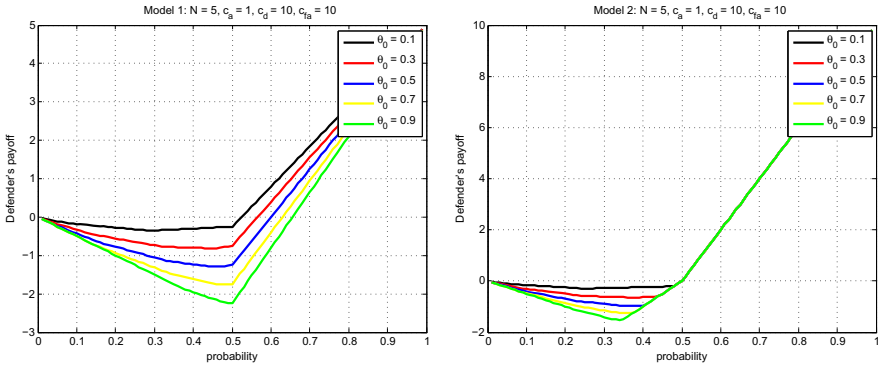


Fig. 3. Effect of θ_0 on the players' NE strategies ($N = 5$, $c_{\mathrm{a}} = 1$, $c_{\mathrm{d}} = c_{\mathrm{fa}} = 10$, $p = 0.3$). The bar left and right of numeral represents the defender and the spy respectively.



(a) Model 1 (b) Model 2

Fig. 4. The defender's NE payoff decreases as θ_0 increases for all values of p for model 1, but only for $p < 0.5$ for model 2

payoff decreases for any value of p, because higher θ_0 signifies a higher false alarm penalty for the defender. In contrast, the second model depicted in Fig. 4(b), the above rule applies only for the ranges of θ_0 below $\theta_0 = 0.5$. For $p > 0.5$, the defender will always select the same pure strategy, that yields the same payoff.

6 Conclusion

We investigate a classification game, where a network administrator (defender) seeks to classify an attacker as a strategic spy or a naive spammer. We first prove that a nonzero-sum game with general payoff functions that satisfy some conditions can lead to a NE computation in polynomial time. Our approach characterizes the structure of the best response strategies of the two players and explains the intuition for the resulting strategies. We investigate two specific game models: model 1 is a simpler game, where the spy benefits from his attacks, regardless of the defender's classification decision. In model 2, the defender is equipped with forensic tools and the spy only benefits from his attacks upon a misclassification. By analyzing these two games, we extract important information about when the defender should invest in forensics and how the strategies of two players in NE are affected by the various control parameters of the game.

References

1. Dritsoula, L., Loiseau, P., Musacchio, J.: A game-theoretic approach for finding optimal strategies in an intruder classification game. To Appear in Proc. of the 51th IEEE Conf. Decision and Control (CDC) (December 2012)
2. Cyber Security Research Report, Bit9 (2012)
3. TMT Global Security Study Key Findings, Deloitte (2011)
4. Manshaei, M.H., Zhu, Q., Alpcan, T., Basar, T., Hubaux, J.-P.: Game Theory Meets Network Security and Privacy, Ecole Polytechnique Federale de Lausanne (EPFL). Tech. Rep. EPFL-REPORT-151965 (April 2011)
5. Alpcan, T., Başar, T.: A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection. In: Proc. of the 42nd IEEE Conf. Decision and Control, pp. 2595–2600 (December 2003)
6. Chen, L., Leneutre, J.: A game theoretical framework on intrusion detection in heterogeneous networks. IEEE Transactions on Information Forensics and Security 4(2), 165–178 (2009)
7. Gueye, A., Walrand, J.C., Anantharam, V.: A Network Topology Design Game: How to Choose Communication Links in an Adversarial Environment? In: GameNets (April 2011)
8. Gueye, A.: A Game Theoretical Approach to Communication Security. PhD dissertation. University of California, Berkeley, Electrical Engineering and Computer Sciences (March 2011)
9. Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D.: Adversarial classification. In: Proc. of the ACM SIGKDD, pp. 99–108 (2004)
10. Luenberger, D.G.: Linear and Nonlinear Programming, 2nd edn. Addison-Wesley (1984)

11. Gambit, Gambit game theory analysis software and tools, http://www.hss.caltech.edu/gambit (2002)
12. Nash, J.: Non-Cooperative Games. The Annals of Mathematics 54(2), 286–295 (1951)
13. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. ../../cvx (April 2011)
14. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (eds.) Recent Advances in Learning and Control. LNCIS, vol. 371, pp. 95–110. Springer, Heidelberg (2008)

Appendix: Omitted Proofs from Section 3

Proof (Proof of Lemma 5). An extreme point $\boldsymbol{x}$ satisfies at least one tight inequality. If none of the inequalities are tight, we scale the vector $\boldsymbol{x}$ down until one inequality becomes tight. The new vector's set of tight inequalities is a strict superset of those of the original vector, thus the point with no tight inequalities is not extreme. Let there be two tight inequalities with indices s and $s+k$ and let all their intermediate inequalities be loose. There exist two possible cases:

1. $\exists i \in \{1,\ldots,k-1\}$, with $x_{s+i} > 0$. We make the following transformation that increases the defendability. We reduce x_{s+i} by a small amount $\epsilon > 0$ and increase x_{s+i+1} by the same amount, maintaining the same norm $\|\boldsymbol{x}\|$. All the inequalities before and after the one with index $(s+i)$ are intact, while the previously loose inequality $(s+i)$ is now tight. For the new vector $\hat{\boldsymbol{x}}$ it holds that $\boldsymbol{\mu}'\hat{\boldsymbol{x}} < \boldsymbol{\mu}'\boldsymbol{x}$, since $\boldsymbol{\mu}$ is a vector with decreasing values and we have shifted some weight from x_{s+i} to x_{s+i+1}. By Lemma 4, the new point corresponds to a defender NE strategy with a better defendability. We continue the above procedure until there are no loose inequalities between the initial tight ones, or until $x_{s+i} = 0, \forall i \in \{1,\ldots,k-1\}$.
2. $x_{s+i} = 0$, $\forall i \in \{1,\ldots,k-1\}$. Subtracting the first tight inequality (of index s) from any loose inequality of index $s+i$, with $i \in \{1,\ldots,k-1\}$, we get

$$\begin{aligned} \Delta_1\delta(s)\cdot(x_0+\ldots+x_s)+\Delta_1 M(s)\|\boldsymbol{x}\| &> 0 \\ &\vdots \\ \Delta_{k-1}\delta(s)\cdot(x_0+\ldots+x_s)+\Delta_{k-1}M(s)\|\boldsymbol{x}\| &> 0. \end{aligned} \tag{9}$$

Similarly, subtracting the last tight inequality $(s+k)$ from all the loose inequalities of index $s+i, \forall i \in \{1,\ldots,k-1\}$, we get

$$\begin{aligned} \Delta_{k-1}\delta(s+1)\cdot(x_0+\cdots+x_s)+\Delta_{k-1}M(s+1)\|\boldsymbol{x}\| &< -\delta(s+k)x_{s+k} \\ &\vdots \\ \Delta_1\delta(s+k-1)\cdot(x_0+\cdots+x_s)+\Delta_1 M(s+k-1)\|\boldsymbol{x}\| &< -\delta(s+k)x_{s+k}. \end{aligned} \tag{10}$$

Under condition 1, the set of equations (9) and (10) cannot be satisfied simultaneously. Indeed, the last equation of (10) gives $0 > \Delta_1\delta(s+k-1)\cdot(x_0+\ldots+x_s)+\Delta_1 M(s+k-1)\|\boldsymbol{x}\| > \Delta_1\delta(s)\cdot(x_0+\ldots+x_s)+\Delta_1 M(s)\|\boldsymbol{x}\|$, which contradicts the first equation of (9). □

Proof (Lemma 7). Suppose that $f < N$. Then the inequality of index $(f+1)$ exists, is loose and all positivity constraints are satisfied. Subtracting the tight inequality of index f from the loose inequality of index $(f+1)$, we get

$$x_{f+1} > \frac{[D(f) - D(f+1)] \cdot \|\boldsymbol{x}\|}{\delta(f)} \triangleq C. \tag{11}$$

1. If D is non increasing, since δ is positive, $C \geq 0$ and $x_{f+1} > 0$. We consider the following transformation

$$\hat{x}_i = \begin{cases} x_i & \text{for } i \in \{0, \ldots, f\} \cup \{f+3, \ldots, N+1\} \\ C & \text{for } i = f+1 \\ x_{f+1} - C & \text{for } i = f+2. \end{cases} \tag{12}$$

   With the above transformation we get

$$\begin{aligned} \boldsymbol{\mu}'(\hat{\boldsymbol{x}} - \boldsymbol{x}) &= \mu_{f+1} \cdot (\hat{x}_{f+1} - x_{f+1}) + \mu_{f+2} \cdot (\hat{x}_{f+2} - x_{f+2}) \\ &= \mu_{f+1} \cdot (C - x_{f+1}) + \mu_{f+2} \cdot (x_{f+1} - C - 0) \\ &= (x_{f+1} - C) \cdot (\mu_{f+2} - \mu_{f+1}) \\ &< 0, \end{aligned}$$

   since $x_{f+2} = 0$, $x_{f+1} > C$, and $\boldsymbol{\mu}$ is a strictly decreasing vector ($\mu_{f+2} < \mu_{f+1}$). Hence, for the new point $\hat{\boldsymbol{x}}$, $\|\hat{\boldsymbol{x}}\| = \|\boldsymbol{x}\|$, but $\boldsymbol{\mu}'\hat{\boldsymbol{x}} < \boldsymbol{\mu}'\boldsymbol{x}$. By Lemma 4 point $\hat{\boldsymbol{x}}$ corresponds to a defender NE strategy with a better defendability. We can continue making the above transformation until $f = N$.
2. If D is increasing, then $C < 0$ and $x_{f+1} \geq 0$. If $x_{f+1} > 0$, while $f < N$ we can shift a small amount ϵ from x_{f+1} to x_{f+2}, keeping the same norm but getting a better defendability. We keep making the above transformation until $f = N$. If $x_{f+1} = 0$, then $\|\boldsymbol{x}\| = x_s + \ldots + x_f$. Subtracting the two tight inequalities (s) and (f) and since $D(H)$ is an increasing function,

$$x_s = \frac{[D(f) - M(s)] \cdot \|\boldsymbol{x}\|}{\delta(s)} > \frac{[D(s) - M(s)] \cdot \|\boldsymbol{x}\|}{\delta(s)} = \frac{\delta(s) \cdot \|\boldsymbol{x}\|}{\delta(s)} = \|\boldsymbol{x}\|,$$

   or $x_s > \|\boldsymbol{x}\|$. Contradiction, unless $s = f$. □

Proof (Lemma 9). Let $\boldsymbol{\beta}$, $\hat{\boldsymbol{\beta}}$ be two different defender NE strategies with $\beta_{N+1} = \hat{\beta}_{N+1}$. By Lemma 8, since β_{N+1} is the same for both vectors, the cost of detection is the same. Let $\hat{s} = s-1$. We will show that the false alarm penalty for the largest index s is larger, i.e., $\boldsymbol{\mu}' \cdot (\beta_s + \ldots + \beta_{N+1}) < \boldsymbol{\mu}' \cdot (\hat{\beta}_{\hat{s}} + \ldots + \beta_{N+1})$. Subtracting the tight inequalities N and $(N-1)$, results in $\beta_N = \hat{\beta}_N$. Similarly, iteratively subtracting the tight inequalities $(s+k)$ and $(s+k+1)$, $\forall k \in \{1, \ldots, N-s-1\}$ results in $\beta_{s+k} = \hat{\beta}_{s+k}$. By Lemma 6, $\beta_{s-1} = \ldots = \beta_0 = 0$ and $\hat{\beta}_{s-2} = \ldots = \hat{\beta}_0 = 0$. Thus the two different NE strategies differ only in β_{s-1}, and β_s. The remaining weight is the same for both vectors ($\beta_{s-1} + \beta_s = \hat{\beta}_{s-1} + \hat{\beta}_s = 1 - \sum_{i=s+1}^{N+1} \beta_i$). In the case of the vector $\hat{\boldsymbol{\beta}}$, this weight is divided into two different components ($\hat{\beta}_{s-1}$

and $\hat{\beta}_s$) while in the case of $\boldsymbol{\beta}$ it is all assigned into the component with index s. Since $\mu_s < \mu_{s-1}$, the vector $\boldsymbol{\beta}$ with the largest index of the first tight inequality s will provide a smaller false alarm cost, and hence a greater defendability. □

Proof (Theorem 2). Depending on the nature of the cost functions, there are two potential constructions for the defender NE strategy $\boldsymbol{\beta}$. We select the one that yields the maximal defendability.

1. Mixed strategies NE with $f = N$. By Lemma 9, defender strategies $\boldsymbol{\beta}$ with the same β_{N+1} yield the maximal defendability when s is maximal. Thus, we need to find the largest possible s such that the inequalities 0 through $(s-1)$ are loose and $x_0 = \ldots = x_{s-1} = 0$. Since we are in mixed NE strategies, there exist at least two tight inequalities. Starting from the last tight inequality N and subtracting the next tight inequality $N-1$, we compute β_N. In general, subtracting the i, $(i-1)$ inequalities, we compute $\beta_i = \frac{D(i-1)-D(i)+[\delta(i)-\delta(i-1)]\cdot\sum_i^{N+1}\beta_i}{\delta(i-1)}$. In every step we check whether the previous inequality $(s-1)$ can be loose. If this is possible, then we assign all the remaining weight to $\beta_s (\beta_s = 1 - \sum_{i=s+1}^{N+1} \beta_i)$. Since the block of tight inequalities that ranges from s through N (integers) is unique, only a certain number of selections on β_{N+1} will produce valid vectors $\boldsymbol{\beta}$ (with unit norm and nonnegative weights). Thus we need to solve the following equations

$$\begin{gathered}\delta(s-1)\cdot 0 + M(s-1) > 1/\|\boldsymbol{x}\| \\ \delta(s)\cdot\beta_s + M(s) = 1/\|\boldsymbol{x}\| \\ \vdots \\ \delta(N)\cdot(1-\beta_{N+1}) + M(N) = 1/\|\boldsymbol{x}\|.\end{gathered}$$

   Subtracting the tight inequality N from the $(s-1)$ loose inequality we get $M(s-1) > \delta(N)\cdot(1-\beta_{N+1}) + M(N)$. Solving for the integer s, we compute the increments of β_{N+1} that give a valid distribution $\boldsymbol{\beta}$.
2. Pure NE with $s = f$. This case implies that when D is an increasing function, a pure defender strategy maximizes the defendability. For each selection of s in $\{0, \ldots, N\}$, we compute the defendability of the resulting strategy $\boldsymbol{\beta}$ $(\beta_s = 1)$, and select the strategy that maximizes the defendability.

Given the defender strategy $\boldsymbol{\beta}$, the attacker is solving his LP (8) and selects his strategy $\boldsymbol{\alpha}$. Nash's existence theorem guarantees a Nash equilibrium, thus the LP will always provide a valid solution. □

Upper Bounds for Adversaries' Utility in Attack Trees

Ahto Buldas[1,2,3,⋆] and Roman Stepanenko[2]

[1] Cybernetica AS
[2] Tallinn University of Technology
[3] Guardtime AS

Abstract. Attack trees model the decision making process of an adversary who plans to attack a certain system. Attack-trees help to visualize possible attacks as Boolean combinations of atomic attacks and to compute attack-related parameters such as cost, success probability and likelihood. The known methods of estimating adversarie's utility are of high complexity and set many unnatural restrictions on adversaries' behavior. Hence, their estimations are incorrect—even if the computed utility is negative, there may still exist beneficial ways of attacking the system. For avoiding unnatural restrictions, we study *fully adaptive adversaries* that are allowed to try atomic attacks in arbitrary order, depending on the results of the previous trials. At the same time, we want the algorithms to be efficient. To achieve both goals, we do not try to measure the exact utility of adversaries but only *upper bounds*. If adversaries' utility has a negative upper bound, it is safe to conclude that there are no beneficial ways of attacking the system, assuming that all reasonable atomic attacks are captured by the attack tree.

1 Introduction

We live in the world where information is extremely valuable. Many of our activities depend on access to information which is correct and up to date. Even minor discrepancies in such things as on-line traffic schedules can cause huge inconveniences. It is crystal clear that information security is of utmost importance to governments and enterprises. Leakage of state secrets can cause conflicts between countries, and for commercial entities loss of their trade secrets may cost not only huge sums of money but also cause them to go bankrupt. Many security features have been introduced into modern information systems. It could be possible to talk about encryption, authentication and authorization schemes, various other technical solutions like firewalls, intrusion detection systems and so on. However even having introduced all those security measures it is difficult to give a quantitative answer how secure the information protected by them really is. There are many techniques of risk assessment available, however most of

⋆ This research has been supported by Estonian Science Foundation, grant No. 8124, and by the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS).

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 98–117, 2012.

them are not suitable for applying to information systems. For example, using statistical data to assess the likelihood of a threat may turn out to be impossible in practice—the very field is quite new and victims usually do not make security incidents public, which means that no statistical data is available. But this doesn't mean there is no hope in finding useful methods for information security.

Attack tree analysis, which is quite similar to fault tree analysis [6], is one of the promising methods. The idea behind attack trees is that multi-stage attacks against information systems can be decomposed into simple atomic attacks against the components of the system. Provided with the security metrics for the atomic attacks and a computational model it could be possible to estimate adversaries' expected utility which would allow us to talk about quantitative security of the system. Attack tree analysis has been used to analyze the Border Gateway Protocol [3], online banking systems [5], as well as large-scale e-voting systems [2].

There exists a handful of quantitative attack tree models, however they are plagued by important problems. The ones that appeared the earliest do not account for economical feasibility of attacks, while the more recent ones put unnatural restrictions on the actions available to the adversary. A good example of those restrictions is that some of the models require adversaries to fix the order of their (atomic) attacks in advance and do not allow adjustments of attacking strategies, while it is more natural to expect that the adversary chooses the next atomic attack adaptively, by taking into account the results of the previously tried atomic attacks. It is evident that such kind of model does not cover all attack possibilities and does not guarantee the absence of beneficial attacks against the system, even if all reasonable atomic attacks were taken into account in the model.

Only by being able to capture all reasonable ways of breaking the system, we could prove that the system is secure, and since the earlier models do not have this quality, a new approach to the problem is needed. Instead of computation-extensive methods for finding exact utilities of restricted adversaries, we have to find computationally lightweight methods for computing upper bounds of the utility of fully-adaptive adversaries. This way by showing that if the largest possible average utility of an economically oriented adversary is negative, we prove that the system is secure.

The aim of this paper is to introduce some of the available models for attack tree evaluation and to comment on their flaws as well as to present a new fully-adaptive model for computing upper bounds of the adversaries utility which is free of those problems. In Section 2, we outline the state of the art in the field of attack tree models, explain our motivation and sum up the main results. Section 3 outlines the main theoretical concepts of attack trees with fully adaptive adversaries. In Section 4, we present and analyze two composition rules that can be used in order to find efficiently computable upper bounds for the utility of fully adaptive adversaries. In Section 5, we describe another method that strengthens the adversary by assuming that every attack can be repeated arbitrary number of times. Some numerical examples are presented in Appendix A.

2 State of the Art, Motivation and Results

2.1 Attack Trees and Computational Models

Attack trees are models in which event algebra is used to visualize the decision making process of an adversary who decided to attack a certain system. In each step of the attack tree analysis, an attack $\mathcal{A}$ (as an event) is decomposed into several simpler attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$, that are defined to be the *child-attacks* of $\mathcal{A}$. In the visual description, $\mathcal{A}$ is represented as a node with $\mathcal{A}_1, \ldots, \mathcal{A}_n$ to be its child nodes. There are two types of decompositions used in the attack tree:

- AND-decomposition $\mathcal{A} = \mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$ means that $\mathcal{A}$ happens (as an event) if and only if all child attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ succeed.
- OR-decomposition $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ means that $\mathcal{A}$ happens if and only if at least one of the child attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ succeed.

A tree-like structure (in general, a directed acyclic graph) is obtained when these two rules are used recursively several times to decompose $\mathcal{A}$ into simpler attacks. Attacks that are not decomposed in such a recursive process, are called *atomic attacks*. They correspond to the leaves of the attack-tree. To summarize, the attack tree analysis represent an attack $\mathcal{A}$ as a monotone Boolean formula $\mathcal{A} = \mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ of the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$.

Attack trees are useful not only for visualization, but also for computing several attack-related parameters such as cost, success probability, feasibility and likelihood, as shown by by Weiss [13] and Schneier [11]. Mauw and Oostdijk [9] presented general soundness rules for the computational semantics of attack-trees, which state that the semantics must be invariant under any transformation of the formula $\mathcal{A} = \mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ that does not change its Boolean function. Though already the early works used attack trees for many many different security-related parameters, they estimated just one of them at a time. Buldas et al. [1] presented a multi-parameter game-theoretic model to estimate the expected utility $U(\mathcal{A})$ of an adversary who tries to make $\mathcal{A}$ happen. Protecting a system against rational adversaries means that the security measures of the system should guarantee that $U(\mathcal{A}) \leq 0$ for all reasonable attacks $\mathcal{A}$.

To estimate $U(\mathcal{A})$, the model of [1] uses computational rules for AND and OR nodes to compute the game theoretic parameters of nodes based on the parameters of their child nodes. Their algorithm works in time linear in the number of nodes in the attack tree (i.e. the size of the Boolean formula $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$). Jürgenson and Willemson [8,7] showed that the computational semantics of the model does not satisfy the general requirements of Mauw and Oostdijk [9]. Jürgenson and Willemson proposed two new consistent models for computing exact utility of the adversary. In their so-called *parallel model* [8], the adversary tries to satisfy the Boolean function $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ by choosing a subset $\mathcal{S} \subseteq \{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$ of atomic attacks and trying all of them independently in parallel. In [7], they refined their model by assuming that the atomic attacks (of the subset) are tried in certain (optimal) order σ and the adversary may skip the atomic attacks in the order in case they would not increase the probability

of materializing the root attack. They showed that the outcome $U_\sigma^{\mathrm{JW}}(\mathcal{A})$ of the adversary in their new (so-called *serial*) model always supersedes the outcome $U^{\mathrm{P}}(\mathcal{A})$ in the parallel model for any ordering σ, i.e. $U^{\mathrm{P}}(\mathcal{A}) \leq U_\sigma^{\mathrm{JW}}(\mathcal{A})$. They payed with the cost though, because while the parallel algorithm works in time $O(2^m)$ (number of terms in a DNF), the serial model uses time $O(m!)$.

Niitsoo [10] showed that the attack-skipping rule of the serial model of Jürgenson and Willemson [7] is not optimal and proposed a new rule inspired by standard *decision theory* by which an atomic attack is skipped if and only if this increases the expected outcome. Niitsoo showed that in his so-called *decision-theoretical model* the adversary's utility $U^{\mathrm{DT}}(\mathcal{A})$ is at least as large as in the serial model of [7], i.e. $U_\sigma^{\mathrm{JW}}(\mathcal{A}) \leq U_\sigma^{\mathrm{DT}}(\mathcal{A})$ for any order σ. He also showed that in case of a certain fixed natural order σ of the atomic attacks, the exact utility can be computed in time liner in the size of the attack tree.

2.2 Shortcomings of the Previous Computational Models

None of the three models [8,7,10] captures all possibilities of the adversary. In both the serial model [7] and the decision-theoretic model [10], the order σ of the atomic attacks is fixed and cannot be adjusted by the adversary during the attack. It is more logical for the adversary to choose the next attack based on the results of the previous trials. Obviously, $\max_\sigma U_\sigma^{\mathrm{DT}}(\mathcal{A}) \leq U^{\mathrm{FA}}(\mathcal{A})$, for the utility $U^{\mathrm{FA}}(\mathcal{A})$ of the adversary in such a *fully adaptive model*, but $U^{\mathrm{FA}}(\mathcal{A})$ was considered in [7] to be too complex to estimate. As the inequality may be strict, it might be that $\max_\sigma U_\sigma^{\mathrm{DT}}(\mathcal{A}) < 0$, but still $U^{\mathrm{FA}}(\mathcal{A}) > 0$, which means that negative utility upper bounds in terms of the serial and decision-theoretic models [7,10] do not guarantee that there are no beneficial adaptive attacks.

2.3 Our Motivation and Goals

The main goal of the attack tree analysis is to justify that a system is secure, assuming that the attack-tree captures all reasonable attacks. The computational models proposed so far are not quite suitable for such analysis because of unnatural restrictions on the behavior of adversaries. Instead of computation-extensive methods for finding exact utilities of restricted adversaries, we have to find *computationally lightweight* methods for computing *upper bounds* of the utility of *fully-adaptive* (or even artificially overpowered) adversaries.

2.4 Main Results of this Work

The starting point of this work is that even though the exact value of $U(\mathcal{A}) = U^{\mathrm{FA}}(\mathcal{A})$ is hard to compute, there might exist rough but easily computable upper bounds for $U(\mathcal{A})$. We first turn back to the method of AND- and OR-rules that was first proposed in [1] and study the following two natural *negativity rules*:

- AND-rule: If $U(\mathcal{A}_i) \leq 0$ for an $i \in \{1, \ldots, n\}$, then $U(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n) \leq 0$.
- OR-rule: If $U(\mathcal{A}_i) \leq 0$ for every $i \in \{1, \ldots, n\}$, then $U(\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n) \leq 0$.

We prove that the AND-rule holds universally and that the OR-rule holds if $\mathcal{A}_1, \ldots, \mathcal{A}_n$ do not contain common atomic attacks. We show that the OR-rule does hot hold in the general case. The main reason is that if $\mathcal{A}_1$ and $\mathcal{A}_2$ contain a common atomic attack $\mathcal{X}$, then trying $\mathcal{X}$ contributes to both attacks $\mathcal{A}_1$ and $\mathcal{A}_2$ and there may exist attacking strategies for $\mathcal{A}_1 \vee \mathcal{A}_2$ that play $\mathcal{A}_1$ and $\mathcal{A}_2$ "in parallel" and has utility larger than $\max\{U(\mathcal{A}_1), U(\mathcal{A}_2)\}$.

To make the general OR-rule work in the general case, we introduce the so-called *cost reduction* technique, that uses the fact that the statement $U(\mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n) \leq 0$ will follow from somewhat stronger assumptions $U(\mathcal{A}'_1) \leq 0$, ... , $U(\mathcal{A}'_n) \leq 0$, where $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ are attacks in which the cost parameters of the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$ have been artificially lowered. For example, if $\mathcal{X}'_1$ is the same atomic attack as $\mathcal{X}_1$ the cost of which is half the original cost, then

$$U(\mathcal{X}_2 \wedge \mathcal{X}'_1) \leq 0 \text{ and } U(\mathcal{X}'_1 \wedge \mathcal{X}_3) \leq 0 \;\; \Rightarrow \;\; U((\mathcal{X}_2 \wedge \mathcal{X}_1) \vee (\mathcal{X}_1 \wedge \mathcal{X}_3)) \leq 0 \;\; .$$

We also show that there is an $O(m \log m)$ algorithm to determine the optimal attacking strategy in the case $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$, where $\mathcal{X}_1, \ldots, \mathcal{X}_m$ are atomic. This is possible because there exists an easily computable invariant r (so-called *cost-nonsuccess ratio*), such that $\mathcal{X}_i$ must be tried before $\mathcal{X}_j$ if and only if $r(\mathcal{X}_i) < r(\mathcal{X}_j)$. The question of existence of such invariants was left open in [10] and hence we completely solved this open question.

Together with the cost-reduction technique this will give us the following method of determining upper bounds of $U(\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m))$. First, represent the Boolean function $\mathcal{A}(\mathcal{X}_1, \ldots, \mathcal{X}_m)$ is a Disjunctive Normal Form (DNF), reduce the cost of those atomic attacks that belong to more than one term of the DNF, determine the utility of each term in $O(m \log m)$ time. Though, the number of terms in a DNF can be large, it is still much less than the number $m!$ of all orderings of the atomic attacks.

Finally, we generalize the concept of an adversary so that it is possible to retry some of the atomic attacks in the case of failure. We assume that each atomic attack is either not repeatable or can be repeated arbitrarily many times. We show that we can reduce this kind of adversaries to the case of ordinary non-repeatable model by just modifying the parameters of repeatable atomic attacks. In such a model, if the utility of an adversary is denoted by $U_\infty(\mathcal{A})$, then $U(\mathcal{A}) = U^{\mathrm{FA}}(\mathcal{A}) \leq U_\infty(\mathcal{A})$. Hence, if we prove that $U_\infty(\mathcal{A}) \leq 0$, this also implies $U(\mathcal{A}) \leq 0$. We also show (Theorem 8) that in the model where all attacks are repeatable, we can use the DNF-method without cost reduction, which means that though the adversary is only mildly strengthened, we are able to compute upper bounds in the *fully adaptive model* with approximately the same cost as that of computing $U^{\mathrm{P}}(\mathcal{A})$ in the parallel model of Jürgenson and Willemson [8].

3 Attack Trees with Fully Adaptive Adversaries

3.1 Notation

If $\mathcal{F}(x_1, \ldots, x_m)$ is a Boolean formula and $v \in \{0, 1\}$, then by $\mathcal{F}_{x_j=v}$ we mean a Boolean formula $\mathcal{F}'(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_m)$ derived from $\mathcal{F}$ by the

assignment $x_j := v$. By $\mathcal{F} \equiv 1$ we mean that $\mathcal{F}$ is identically true (i.e. is a tautology), and by $\mathcal{F} \equiv 0$, we mean that $\mathcal{F}$ is identically false. By a *min-term* of a Boolean formula $\mathcal{F}(x_1, \ldots, x_m)$, we mean a Boolean formula $M(x_1, \ldots, x_m)$ of type $x_{i_1} \wedge x_{i_2} \wedge \ldots \wedge x_{i_k}$ such that $M(x_1, \ldots, x_m) \Rightarrow \mathcal{F}(x_1, \ldots, x_m)$ is a tautology. We say that M is a *critical min-term* of M if non of the sub-terms $M'(x_1, \ldots, x_m) = x_{i_1} \wedge \ldots \wedge x_{i_{j-1}} \wedge x_{i_{j+1}} \wedge \ldots \wedge x_{i_k}$ is a min-term of M.

3.2 Attack Trees, Strategies and Utility

Definition 1 (Attack tree). *An attack tree $\mathcal{A}$ consists of the next components:*

- *A finite number of* atomic attacks $\mathcal{X}_1 \ldots \mathcal{X}_m$*, each attack $\mathcal{X}_i$ having the following parameters:* success probability p_i, failure probability q_i, (preparation) cost C_i *(a real number), and* penalty Π_i *(a real number).*
- *A negation-free (monotone) Boolean formula $\mathcal{F}(x_1, \ldots, x_m)$, where x_j are the input variables that correspond to the atomic attacks $\mathcal{X}_j$.*
- *The prize P (a non-negative real number).*

Definition 2 (Subtree). *By a* subtree $\mathcal{B}$ *of an attack tree $\mathcal{A}$ with Boolean formula $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m)$ we mean an attack tree with the same parameters as $\mathcal{A}$, except that the Boolean formula $\mathcal{F}_{\mathcal{B}}$ of $\mathcal{B}$ is in the form*

$$\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) = \mathcal{F}'(\mathcal{F}_{\mathcal{B}}(x_1, \ldots, x_m), x_1, \ldots, x_m) \ , \tag{1}$$

where $\mathcal{F}'$ is a negation-free Boolean formula $\mathcal{F}'$. Note that (1) is an identity between Boolean formulae*, not just between Boolean functions.*

Each attack tree $\mathcal{A}$ represents a *one-player game*, where the adversary (the player) can choose and execute atomic attacks one by one. At any stage of the game, the adversary is always allowed to *give up*, i.e. stop playing the game.

Definition 3 (Attack game). *By an* attack game *we mean a one-player game, every instance of which is an attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$), whereas*

- *if $\mathcal{F} \equiv 1$, the game is* won *and the adversary gets the prize P;*
- *if $\mathcal{F} \equiv 0$, the game is* lost *and the adversary does not get the prize.*

A game $\mathcal{A}$ that is neither won nor lost, the adversary may choose an atomic attack $\mathcal{X}_j$ after which it has to pay the costs C_j of $\mathcal{X}_j$ and the following happens:

- *with probability p_j, the game is reduced to $\mathcal{A}_{x_j=1}$ (with formula $\mathcal{F}_{x_j=1}$);*
- *with probability q_j, the game is reduced to $\mathcal{A}_{x_j=0}$ (with formula $\mathcal{F}_{x_j=0}$); and*
- *with probability $1 - p_j - q_j$, the adversary* gets caught*, i.e. it has to pay the penalty Π_j and the game is over. Formally, we denote this case by $x_j = \perp$.*

Definition 4 (Strategy). *A* strategy *S for an attack tree $\mathcal{A}$ is a rule that for any sequence of assignments $\langle x_{j_1} = v_1, \ldots, x_{j_k} = v_k \rangle$ (where $v_j \in \{0, 1\}$) that represent the previous moves, and possibly some auxiliary information, either points to the next atomic attack $\mathcal{X}_{j_{k+1}}$ to try, or decides to give up the game.*

Definition 5 (Strategy-tree). *A strategy can be represented as a tree, each node of which represents an atomic attack $\mathcal{X}_j$ and each node may have two or less successors, that correspond to the choice of the next move in two cases $x_j = 0$ and $x_j = 1$. The root node of the strategy-tree represents the first move.*

Definition 6 (Empty strategy). *A strategy S may suggest not to play the attack game of $\mathcal{A}$ at all. Such a strategy can be represented as an empty tree and is denoted by $\emptyset$.*

Definition 7 (Branch of a strategy). *By a* branch *β of the strategy S for an attack tree $\mathcal{A}$, we mean a sequence of assignments*

$$\beta = \langle x_{i_1} = v_1, \ldots, x_{i_{k-1}} = v_{k-1}, x_{i_k} = v_k \rangle \ , \tag{2}$$

where $v_1, \ldots, v_{k-1} \in \{0, 1\}$, $v_k \in \{0, 1, \perp\}$ that may occur when the attack game of $\mathcal{A}$ is played according to S. A branch can also be viewed as a sequence of nodes from the root to a leaf in the strategy-tree together with the outcome v_k of the last-tried atomic attack. Let $\beta \Rightarrow \mathcal{A}$ denote the proposition that the assignments (2) of β imply $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) = 1$.

Every branch represents a possible sequence of events when playing the attack game with strategy S. We associate to each branch β the following parameters: the cost C_β, the penalty Π_β and the prize P_β. For the branch (2), we have:

$$C_\beta = \sum_{i=1}^{m} C_i \cdot [x_i \in \beta] = C_{i_1} + C_{i_2} + \ldots + C_{i_k}$$

$$\Pi_\beta = \sum_{i=1}^{m} \Pi_i \cdot [x_i \in \beta] \cdot [x_i = \perp] = \begin{cases} \Pi_{i_k} & \text{if } w_k = \perp, \\ 0 & \text{otherwise} \end{cases}$$

$$P_\beta = P \cdot [\beta \Rightarrow \mathcal{A}] = \begin{cases} P \text{ if } x_{i_1} = w_1, \ldots, x_{i_k} = w_k \text{ imply } \mathcal{F}(x_1, \ldots, x_m) = 1, \\ 0 \ \text{ otherwise} \end{cases}$$

where the parentheses [] denote the so-called *Iverson symbol*—for any proposition $\mathcal{P}$, the Iverson symbol $[\mathcal{P}] = 1$ if $\mathcal{P}$ is true, and $[\mathcal{P}] = 0$, otherwise.

Definition 8 (Utility of a strategy). *By the* utility *of a strategy S for an attack tree $\mathcal{A}$, we mean*

$$U(\mathcal{A}; S) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \ , \tag{3}$$

where $\mathsf{P}[\beta]$ is the probability that β occurs during the attack game played with strategy S. For example, for the branch (2), $\mathsf{P}[\beta] = P_{i_1} \cdot P_{i_2} \cdot \ldots \cdot P_{i_k}$, where

$$P_{i_j} = \begin{cases} p_{i_j} & \text{if } w_j = 1 \\ q_{i_j} & \text{if } w_j = 0 \\ 1 - p_{i_j} - q_{i_j} & \text{if } w_j = \perp \end{cases}$$

For the empty strategy $\emptyset$ we have $U(\mathcal{A}; \emptyset) = 0$ for every attack tree $\emptyset$.

Example: For the attack tree and the strategy of Fig. 1, we have the following seven branches $\beta_1, \ldots, \beta_7$ listed in Tab. 1. Hence, by (3) the utility $U(\mathcal{A}; S)$ of the strategy is computed as follows:

$$\begin{aligned} U(\mathcal{A}; S) = {} & p_1 p_2 \cdot (-C_1 - C_2 + P) + p_1(1 - p_2 - q_2) \cdot (-C_1 - C_2 - \Pi_2) + \\ & + p_1 q_2 p_3 \cdot (-C_1 - C_2 - C_3 + P) + \\ & + p_1 q_2 (1 - p_3 - q_3) \cdot (-C_1 - C_2 - C_3 - \Pi_3) - \\ & - p_1 q_2 q_3 \cdot (C_1 + C_2 + C_3) - (1 - p_1 - q_1) \cdot (C_1 + \Pi_1) - q_1 C_1 \ . \end{aligned}$$

Table 1. The branches of the strategy of Fig. 1 and their parameters

β	Assignments	Probability $\mathsf{P}[\beta]$	Cost C_β	Penalty Π_β	prize P_β
β_1	$x_1 = 1, x_2 = 1$	$p_1 p_2$	$C_1 + C_2$	0	P
β_2	$x_1 = 1, x_2 = \bot$	$p_1(1 - p_2 - q_2)$	$C_1 + C_2$	Π_2	0
β_3	$x_1 = 1, x_2 = 0, x_3 = 1$	$p_1 q_2 p_3$	$C_1 + C_2 + C_3$	0	P
β_4	$x_1 = 1, x_2 = 0, x_3 = \bot$	$p_1 q_2 (1 - p_3 - q_3)$	$C_1 + C_2 + C_3$	Π_3	0
β_5	$x_1 = 1, x_2 = 0, x_3 = 0$	$p_1 q_2 q_3$	$C_1 + C_2 + C_3$	0	0
β_6	$x_1 = \bot$	$1 - p_1 - q_1$	C_1	Π_1	0
β_7	$x_1 = 0$	q_1	C_1	0	0

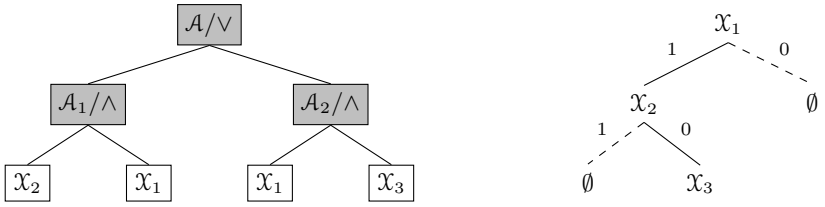


Fig. 1. An attack tree $\mathcal{A}$ (left) and a strategy (right)

Definition 9 (Utility of an attack tree). *By the* utility *of an attack tree $\mathcal{A}$ we mean the limit $U(\mathcal{A}) = \sup_S U(\mathcal{A}; S)$, which exists due to the bound $U(\mathcal{A}; S) \leq P$ (where P is the prize) for any strategy S.*

Corollary 1. *$U(\mathcal{A}) \geq 0$ for any $\mathcal{A}$, as $U(\mathcal{A}) = \sup_S U(\mathcal{A}; S) \geq U(\mathcal{A}; \emptyset) = 0$.*

Theorem 1 (Optimal strategy). *For any attack tree $\mathcal{A}$, there exists an* optimal strategy*, i.e. a strategy S for which $U(\mathcal{A}; S) = U(\mathcal{A})$.*

Proof. We use induction on the number m of atomic attacks in $\mathcal{A}$. The statement is clearly true for $m = 0$. Assume that every attack tree $\mathcal{A}'$ with $m - 1$ atomic attacks has an optimal strategy. Let $\mathcal{A}$ be an attack tree with atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$. Let $S_\mathcal{A}$ be the strategy that first finds $\mathcal{X}_j$ that maximizes the value:

$$u_j = -C_j - (1 - p_j - q_j) \cdot \Pi_j + q_j \cdot U(\mathcal{A}_{x_j=0}) + p_j \cdot U(\mathcal{A}_{x_j=1}) \ ,$$

and chooses $\mathcal{X}_j$ as the next move if $u_j > 0$, or gives up if $u_j \leq 0$. After that:

– if $x_j = 0$, $S_{\mathcal{A}}$ uses an optimal strategy S'_0 for $\mathcal{A}_{x_j=0}$ (induction hypothesis);
– if $x_j = 1$, $S_{\mathcal{A}}$ uses an optimal strategy S'_1 for $\mathcal{A}_{x_j=1}$.

Clearly, $S_{\mathcal{A}}$ is optimal for $\mathcal{A}$. □

Corollary 2 (Algorithm for Exact Utility). *The exact utility in the fully adaptive model can be computed by using the following recursive relation*

$$U(\mathcal{A}) = \max_j\{0, -C_j - (1 - p_j - q_j) \cdot \Pi_j + p_j U(\mathcal{A}_{x_j=1}) + q_j U(\mathcal{A}_{x_j=0})\} \ , \quad (4)$$

with initial conditions $U(\mathbf{1}) = P$ and $U(\mathbf{0}) = 0$, where $\mathbf{1}$ and $\mathbf{0}$ denote attack games with Boolean functions $\mathcal{F} \equiv 1$ and $\mathcal{F} \equiv 0$, respectively.

This algorithm runs in time $O(m!)$, where m is the number of atomic attacks, and is hence unsuitable if m is large.

3.3 Simulated Strategies

The concept of simulated strategies is a useful tool for drawing implications about the utility $U(\mathcal{A})$ of an attack tree $\mathcal{A}$ based on the utilities of its subtrees $\mathcal{B}$ (Def. 2). Let $\mathcal{A}$ be an attack tree $\mathcal{A}$ and $\mathcal{B}$ be a subtree of $\mathcal{A}$.

Definition 10 (Simulated strategy). *Every strategy S for $\mathcal{A}$ can be modified to a* simulated strategy $S\,|\,\mathcal{B}$ *for $\mathcal{B}$, in the following way:*

– *Whenever S decides to try an atomic attack in $\mathcal{B}$, then so does strategy $S\,|\,\mathcal{B}$.*
– *If S decides to try an atomic attack $\mathcal{X}_i$ that $\mathcal{B}$ does not involve, then $\mathcal{X}_i$ is* simulated *by $S\,|\,\mathcal{B}$ (without actually investing into it) and the results are hold as auxiliary information a.*

Let C_β and $C_{\beta|\mathcal{B}}$ be the costs of S and $S\,|\,\mathcal{B}$ respectively in branch β of S. Then

$$C_{\beta|\mathcal{B}} = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{B}] \le \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta \ . \quad (5)$$

If Π_β and $\Pi_{\beta|\mathcal{B}}$ are the penalties of S and $S\,|\,\mathcal{B}$ (in case of β) respectively, then

$$\Pi_{\beta|\mathcal{B}} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \perp] \cdot [x_j \in \mathcal{B}] \le \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \perp] = \Pi_\beta \ . \quad (6)$$

Let P_β and $P_{\beta|\mathcal{B}}$ be the prize of S and $S\,|\,\mathcal{B}$ respectively in β. Without any additional assumptions about $\mathcal{A}$ and $\mathcal{B}$, we do not know the relationship between $P_\beta = P \cdot [\beta \Rightarrow \mathcal{A}]$ and $P_{\beta|\mathcal{B}} = P \cdot [\beta \Rightarrow \mathcal{B}]$. Lemma 1 is an important special case.

Lemma 1. *If $\mathcal{B}$ is a subtree of $\mathcal{A}$ and $\mathcal{F}_{\mathcal{A}}(x_1, \ldots, x_m) \Rightarrow \mathcal{F}_{\mathcal{B}}(x_1, \ldots, x_m)$ is a tautology, then $U(\mathcal{A}) \le U(\mathcal{B})$.*

Proof. As $\beta \Rightarrow \mathcal{A}$ implies $\beta \Rightarrow \mathcal{B}$, then $P_\beta \,|\, \mathcal{B} \geq P_\beta$, and by (5) and (6) we have:

$$U(\mathcal{A}; S) = \sum_\beta \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \leq \sum_\beta \mathsf{P}[\beta] \cdot (-C_{\beta|\mathcal{B}} - \Pi_{\beta|\mathcal{B}} + P_{\beta|\mathcal{B}})$$
$$= U(\mathcal{B}; S \,|\, \mathcal{B}) \leq U(\mathcal{B}) \ ,$$

for any strategy S for $\mathcal{A}$ and the corresponding simulated strategy for $\mathcal{B}$. Hence, in case S is an optimal strategy for $\mathcal{A}$, we have $U(\mathcal{A}) = U(\mathcal{A}; S) \leq U(\mathcal{B})$. □

4 Efficient Decomposition Rules

4.1 The AND-Rule

An attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$) is a $\wedge$-*composition* of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (with Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$, respectively) and write $\mathcal{A} = \mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n$, if

$$\mathcal{F}(x_1, \ldots, x_m) \equiv \mathcal{F}_1(x_1, \ldots, x_m) \wedge \ldots \wedge \mathcal{F}_n(x_1, \ldots, x_m) \ , \tag{7}$$

where $x_1, \ldots, x_m$ represent the atomic attacks $\mathcal{X}_1, \ldots, \mathcal{X}_m$ that the trees contain.

Theorem 2 (AND-rule). $U(\mathcal{A}_1 \wedge \ldots \wedge \mathcal{A}_n) \leq \min\{U(\mathcal{A}_1), U(\mathcal{A}_2), \ldots, U(\mathcal{A}_n)\}$.

Proof. As $\mathcal{F}(x_1, \ldots, x_m) \Rightarrow \mathcal{F}_i(x_1, \ldots, x_m)$ is a tautology for every $i = 1..n$, by Lemma 1, $U(\mathcal{A}) \leq U(\mathcal{A}_i)$ and hence $U(\mathcal{A}) \leq \min\{U(\mathcal{A}_1), \ldots, U(\mathcal{A}_n)\}$. □

Definition 11 (Non-Stop Strategy). *A strategy S for an attack tree $\mathcal{A}$ is called a* non-stop strategy*, if for any branch β, either: (1) $\beta \Rightarrow \mathcal{A}$, (2) $\beta \Rightarrow \neg\mathcal{A}$, or (3) β contains an assignment $(x_j, \bot)$.*

Let $U_{\mathrm{ns}}(\mathcal{A}) = \sup_N U(\mathcal{A}; N)$, where N varies over all non-stop strategies. Clearly, $U_{\mathrm{ns}}(\mathcal{A}) \leq U(\mathcal{A})$.

Lemma 2. *Let $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$, where $\mathcal{X}_1, \ldots, \mathcal{X}_m$ are atomic. Then* $U(\mathcal{A}) = \begin{cases} U_{\mathrm{ns}}(\mathcal{A}) & , \text{if } U_{\mathrm{ns}}(\mathcal{A}) > 0 \\ 0 & \text{otherwise.} \end{cases}$

Proof. We use induction on m. The statement is clearly true for $m = 1$. Let $c_i = C_i + (1 - q_i - p_i)\Pi_i$. Assume that the statement is true for $m - 1$ and let $\mathcal{O}$ be an optimal strategy for playing $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$. Assume without loss of generality that $U(\mathcal{A}) > 0$ and $\mathcal{X}_m$ was the first atomic attack $\mathcal{O}$ tries. Let $\mathcal{A}' = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_{m-1}$. Hence, $U(\mathcal{A}) = U(\mathcal{A}; \mathcal{O}) = -c_m + p_m U(\mathcal{A}') > 0$, and hence also $U(\mathcal{A}') > 0$. By the induction assumption, $U(\mathcal{A}') = U_{\mathrm{ns}}(\mathcal{A}')$ and hence,

$$U_{\mathrm{ns}}(\mathcal{A}) \leq U(\mathcal{A}) = U(\mathcal{A}; \mathcal{O}) = -c_m + p_m U(\mathcal{A}') = -c_m + p_m U_{\mathrm{ns}}(\mathcal{A}') \leq U_{\mathrm{ns}}(\mathcal{A}) \ ,$$

which implies $U(\mathcal{A}) = U_{\mathrm{ns}}(\mathcal{A})$. □

Theorem 3 (Atomic AND Case). *If $\mathcal{A} = \mathcal{X}_1 \wedge \ldots \wedge \mathcal{X}_m$ then the best first move of the adversary is to try $\mathcal{X}_i$ with the smallest cost-nonsuccess ratio $\frac{c_i}{1-p_i}$.*

Proof. If $U(\mathcal{A}) > 0$ then $U(\mathcal{A}) = U_{\text{ns}}(\mathcal{A})$. Let S and S' be non-stop strategies that are identical except that in S' the k-th and $(k+1)$-th trials are exchanged. Let $c_{1,k-1}$ denote the average cost of the first $k-1$ atomic attacks, which is the same for S and S', let $p_{1,k-1}$ denote the probability that the first $k-1$ trials are all successful, and let $U_{k+2}(\mathcal{A})$ be the utility of $\mathfrak{X}_{k+2} \wedge \mathfrak{X}_{k+3} \wedge \ldots \wedge \mathfrak{X}_m$. As

$$\begin{aligned} U(\mathcal{A}; S) &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_k + p_k(-c_{k+1} + p_{k+1} \cdot U_{k+2}(\mathcal{A}))] \\ &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_k - p_k c_{k+1} + p_k p_{k+1} \cdot U_{k+2}(\mathcal{A})] \\ U(\mathcal{A}; S') &= -c_{1,k-1} + p_{1,k-1} \cdot [-c_{k+1} - p_{k+1} c_k + p_k p_{k+1} \cdot U_{k+2}(\mathcal{A})] \ , \end{aligned}$$

$U(\mathcal{A}; S) > U(\mathcal{A}; S')$ iff $-c_k - p_k c_{k+1} > -c_{k+1} - p_{k+1} c_k$, i.e. $\frac{c_k}{1-p_k} < \frac{c_{k+1}}{1-p_{k+1}}$. □

Corollary 3. *For any attack game of the form $\mathcal{A} = \mathfrak{X}_1 \wedge \ldots \wedge \mathfrak{X}_m$ there is a $O(m \log m)$-time algorithm for finding the optimal order of the atomic attacks, because by Thm. 3, to find the optimal order, we have to sort the atomic attacks by their cost-nonsuccess ratio and sorting requires $O(m \log m)$-time.*

4.2 OR-Rule for Independent Trees

An attack tree $\mathcal{A}$ (with Boolean formula $\mathcal{F}$) is a $\vee$-*composition* of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (with Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$, respectively) and write $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$, if

$$\mathcal{F}(x_1, \ldots, x_m) \equiv \mathcal{F}_1(x_1, \ldots, x_m) \vee \ldots \vee \mathcal{F}_n(x_1, \ldots, x_m) \ , \tag{8}$$

where $x_1, \ldots, x_m$ represent the atomic attacks $\mathfrak{X}_1, \ldots, \mathfrak{X}_m$ that the trees contain.

Definition 12 (Independent Trees). *Attack trees $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are said to be* independent*, if they do not contain common atomic attacks, i.e. if their Boolean formulae $\mathcal{F}_1, \ldots, \mathcal{F}_n$ do not have common variables.*

For example, $\mathcal{F}_1(x_1, \ldots, x_4) = x_1 \wedge x_2$ and $\mathcal{F}_2(x_1, \ldots, x_4) = x_3 \wedge x_4$ are independent but $x_1 \wedge x_2$ and $x_1 \wedge x_3$ (Fig. 11) are not as x_1 is their common variable. It turns out that if $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are independent and negative (i.e. $U(\mathcal{A}_i) \leq 0$ for all i), then also $U(\mathcal{A}) \leq 0$. The independence is necessary for the negativity rule to hold—we give a counter-example with attack trees that are not independent.

Theorem 4 (OR Rule for Independent Trees). *Let $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ where $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are independent. Then $U(\mathcal{A}) \leq U(\mathcal{A}_1) + U(\mathcal{A}_2) + \ldots + U(\mathcal{A}_n)$.*

Proof. Let S be a optimal strategy for $\mathcal{A}$ and $S\,|\,\mathcal{A}_i$ be the simulated strategy for $\mathcal{A}_i$. Due to $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ we have $\sum_{i=1}^n [\beta \Rightarrow \mathcal{A}_i] \geq [\beta \Rightarrow \mathcal{A}]$, and because of independence, $\sum_{i=1}^n [x_j \in \mathcal{A}_i] = 1$ for every variable x_j that $\mathcal{A}$ contains. Hence,

$$\sum_{i=1}^{n} P_{\beta|\mathcal{A}_i} = P \cdot \sum_{i=1}^{n} [\beta \Rightarrow \mathcal{A}_i] \geq P \cdot [\beta \Rightarrow \mathcal{A}] = P_\beta \ ,$$

$$\sum_{i=1}^{n} C_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n} \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot \sum_{i=1}^{n} [x_j \in \mathcal{A}_i]$$

$$= \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta \ ,$$

$$\sum_{i=1}^{n} \Pi_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \cdot \sum_{i=1}^{n} [x_j \in \mathcal{A}_i]$$

$$= \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \Pi_\beta \ .$$

Therefore,

$$U(\mathcal{A}) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \leq \sum_{i=1}^{n} \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_{\beta|\mathcal{A}_i} - \Pi_{\beta|\mathcal{A}_i} + P_{\beta|\mathcal{A}_i})$$

$$= U(\mathcal{A}_1; S \,|\, \mathcal{A}_1) + \ldots + U(\mathcal{A}_n; S \,|\, \mathcal{A}_n) \leq U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n) \ . \qquad \square$$

4.3 Counterexample with Common Atomic Attacks

If in the case $\mathcal{A} = \mathcal{A}_1 \vee ... \vee \mathcal{A}_n$, where the attacks may have common atomic attacks. It turns out that the OR-rule (If all $\mathcal{A}_i$ are negative then $\mathcal{A}$ is negative) does not hold in this case. There is the following counter-example. Let $\mathcal{A} = \mathcal{A}_1 \vee \mathcal{A}_2$, where $\mathcal{A}_1 = \mathcal{X}_1 \wedge \mathcal{X}_2$, and $\mathcal{A}_2 = \mathcal{X}_1 \wedge \mathcal{X}_3$ (Fig. 1). Let all the three atomic attacks $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ have the same parameters $c = C + (1-p-q) \cdot \Pi = 1$, $p = q = 0.5$; and let $P = 5$. Then,

$$U(\mathcal{X}_1 \wedge \mathcal{X}_2) = U(\mathcal{X}_1 \wedge \mathcal{X}_3) = -1 + 0.5(-1 + 0.5 \cdot 5) = -1.5 + 1.25 = -0.25 < 0$$

but as $U(\mathcal{X}_1) = U(\mathcal{X}_2) = U(\mathcal{X}_3) = -1 + 0.5 \cdot 5 = 1.5$ and $U(\mathcal{X}_2 \vee \mathcal{X}_3) = U(\mathcal{X}_2) + q \cdot U(\mathcal{X}_3) = (1+q) \cdot 1.5 = 1.5 \cdot 1.5 = 2.25$, we have (by trying $\mathcal{X}_1$ first)

$$U(\mathcal{A}) \geq -c + p \cdot U(\mathcal{X}_2 \vee \mathcal{X}_3) = -1 + 0.5 \cdot 2.25 = 0.125 > 0 \ .$$

This means that in the proof of the OR-rule, we certainly have to assume that the attacks $\mathcal{A}_1, \ldots, \mathcal{A}_n$ do not have common atomic attacks.

4.4 General OR-Rule: Cost Reduction

Instead of proving $U(\mathcal{A}) \leq U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n)$, which is not true in general, we modify the attack (sub-) trees $\mathcal{A}_1, \ldots \mathcal{A}_n$ by artificially reducing the costs C_j and the penalties Π_j (thereby, making the attacks easier to perform), and prove

$$U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n) \ ,$$

where $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ are the modified attack trees. The philosophy behind this is that if the system is secure even if some of the attacks are assumed to be easier then they really are, then also the attacks against the real system are infeasible.

In order to see, how the costs should be reduced, we study the reasons why the inequality $U(\mathcal{A}) \le U(\mathcal{A}_1) + \ldots + U(\mathcal{A}_n)$ fails. If $\mathcal{A}_1, \ldots, \mathcal{A}_n$ are allowed to have common atomic attacks $\mathfrak{X}_j$, then for some atomic attacks $\mathfrak{X}_j$ we may have $\sum_{i=1}^{n}[x_j \in \mathcal{A}_i] = k_j > 1$, where k_j is the number of attacks among $\mathcal{A}_1, \ldots, \mathcal{A}_n$ that contain $\mathfrak{X}_j$. By using the same simulation concept as before, we have:

$$\sum_{i=1}^{n} P_{\beta|\mathcal{A}_i} = P \cdot \sum_{i=1}^{n}[\beta \Rightarrow \mathcal{A}_i] \ge P \cdot [\beta \Rightarrow \mathcal{A}] = P_\beta$$

$$\sum_{i=1}^{n} C_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] \cdot \sum_{i=1}^{n}[x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} k_j C_j \cdot [x_j \in \beta] \not\le C_\beta \quad (9)$$

$$\sum_{i=1}^{n} \Pi_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \cdot \sum_{i=1}^{n}[x_j \in \mathcal{A}_i] \quad (10)$$

$$= \sum_{j=1}^{m} k_j \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] \not\le \Pi_\beta$$

We see that the main reason for failure is that we have the terms $k_j C_j$ and $k_j \Pi_j$ instead of C_j and Π_j. This inspires the following theorem.

Theorem 5 (Uniform Cost Reduction). *Let $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ and let $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ be sub-trees with the cost and penalties reduced by the rule $C'_j = \frac{C_j}{k_j}$ and $\Pi'_j = \frac{\Pi_j}{k_j}$. Then $U(\mathcal{A}) \le U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$.*

Proof. Let S be an optimal strategy for $\mathcal{A}$ and let $S\,|\,\mathcal{A}_i$ be the simulated strategy for $\mathcal{A}'_i$. By (9) we have

$$\sum_{i=1}^{n} C'_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} k_j C'_j \cdot [x_j \in \beta] = \sum_{j=1}^{m} C_j \cdot [x_j \in \beta] = C_\beta$$

$$\sum_{i=1}^{n} \Pi'_{\beta|\mathcal{A}_i} = \sum_{j=1}^{m} k_j \Pi'_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \sum_{j=1}^{m} \Pi_j \cdot [x_j \in \beta] \cdot [x_j = \bot] = \Pi_\beta \ .$$

Hence,

$$U(\mathcal{A}) = \sum_{\beta} \mathsf{P}[\beta] \cdot (-C_\beta - \Pi_\beta + P_\beta) \le \sum_{i=1}^{n} \sum_{\beta} \mathsf{P}[\beta] \cdot (-C'_{\beta|\mathcal{A}_i} - \Pi'_{\beta|\mathcal{A}_i} + P'_{\beta|\mathcal{A}_i})$$

$$= U(\mathcal{A}'_1; S\,|\,\mathcal{A}_1) + \ldots + U(\mathcal{A}'_n; S\,|\,\mathcal{A}_n) \le U(\mathcal{A}'_1) + \ldots + U(\mathcal{A}'_n) \ . \quad \square$$

Cost reduction can be generalized so that the costs and penalties in the attack games $\mathcal{A}'_i$ are reduced in different ways, i.e. the reduction amount may depend on i.

Theorem 6 (General Cost Reduction). *Let* $\mathcal{A} = \mathcal{A}_1 \vee \ldots \vee \mathcal{A}_n$ *and let* $\mathcal{A}'_1, \ldots, \mathcal{A}'_n$ *be sub-trees with the cost and penalties reduced in* $\mathcal{A}'_i$ *by general rules* $C_j \mapsto C'_{j,i}$ *and* $\Pi_j \mapsto \Pi'_{j,i}$*, so that*

$$\sum_{i=1}^{n} C'_{j,i} \cdot [x_j \in \mathcal{A}_i] = C_j \ , \qquad \text{and} \qquad \sum_{i=1}^{n} \Pi'_{j,i} \cdot [x_j \in \mathcal{A}_i] = \Pi_j \ . \tag{11}$$

Then $U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$*. The Iverson symbol* $[x_j \in \mathcal{A}_i]$ *can be omitted in (11) if we assume that* $C'_{j,i} = \Pi'_{j,i} = 0$ *if* $\mathcal{A}_i$ *does not contain* x_j*.*

Proof. Because of

$$\sum_{i=1}^{n} C'_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} C'_{j,i} \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} [x_j \in \beta] \cdot \underbrace{\sum_{i=1}^{n} C'_{j,i} \cdot [x_j \in \mathcal{A}_i]}_{C_j}$$

$$= C_\beta \ ,$$

$$\sum_{i=1}^{n} \Pi'_{\beta|\mathcal{A}_i} = \sum_{i=1}^{n}\sum_{j=1}^{m} \Pi'_{j,i} \cdot [x_j \in \beta] \cdot [x_j \in \mathcal{A}_i] = \sum_{j=1}^{m} [x_j \in \beta] \cdot \underbrace{\sum_{i=1}^{n} \Pi'_{j,i} \cdot [x_j \in \mathcal{A}_i]}_{\Pi_j}$$

$$= \Pi_\beta \ ,$$

we have $U(\mathcal{A}) \leq U(\mathcal{A}'_1) + U(\mathcal{A}'_2) + \ldots + U(\mathcal{A}'_n)$, like the proof of Thm. 5. □

4.5 Algorithm 1: Iterated AND/OR Rules

The first algorithm uses the AND-rule and the OR-rule at every node of the attack-tree. In order to make the OR-rule work in general case, cost-reduction is used. The cost of every every atomic attack is reduced at every OR-node of the attack tree. The cost reduction step starts from the root vertex and ends in the root vertices. For example, in case we have an attack tree with Boolean formula

$$\mathcal{F} = (\mathcal{X} \wedge \mathcal{X}_1) \vee (\mathcal{X} \wedge \mathcal{X}_2) \vee (((\mathcal{X} \wedge \mathcal{X}_3) \vee (\mathcal{X} \wedge \mathcal{X}_4)) \wedge \mathcal{X}_5) \ ,$$

which is depicted in Fig. 2 (left), the cost c of the atomic attack $\mathcal{X}$ must be used in two places: (1) in the root node we divide the cost by 3; and (2) in the subtree $(\mathcal{X} \wedge \mathcal{X}_3) \vee (\mathcal{X} \wedge \mathcal{X}_4)$, we have to divide the cost again by 2, and hence, the cost c of $\mathcal{X}$ reduces to $c/6$ in this subtree, while in subtrees $\mathcal{X} \wedge \mathcal{X}_1$ and $\mathcal{X} \wedge \mathcal{X}_2$ it reduces to $c/3$.

After the costs of all atomic attacks are reduced in this way, we apply the AND, OR negativity rules starting from the leaves of the tree (the atomic attacks) and ending with the root vertex.

The algorithm works in time which is roughly linear in the number of vertices of the tree, and hence is very fast. The main drawback of this algorithm is that in case of "deep" attack trees where atomic attacks appear many times, the cost reduction rules will reduce the cost too much, i.e. it is practically impossible to find practical security measures and apply them in the real system so that the system is still secure if reduced costs are assumed.

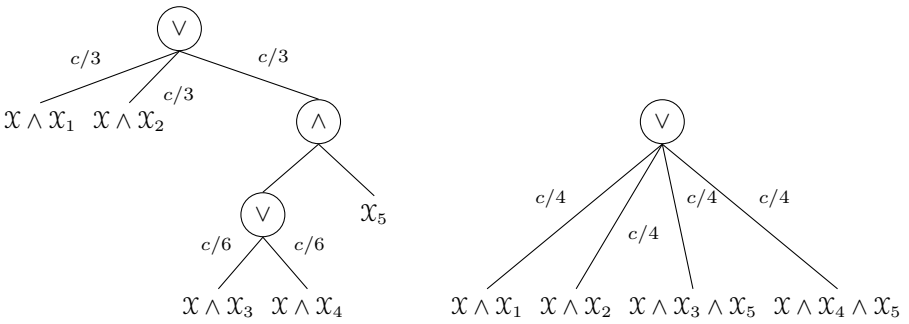


Fig. 2. Iterated cost reduction based on the tree structure (left) and cost reduction based on the DNF (right)

4.6 Algorithm 2: DNF with Cost Reduction

The second algorithm first constructs a DNF of the Boolean function of the attack tree, reduces the cost of each atomic attack by dividing it by the number of min-terms in which the atomic attack appears. For example, if an atomic attack appears in four min-terms of the DNF, then its cost c is reduced to $c/4$ (Fig. 2, right). Then, for each min-term, Algorithm 2 finds the optimal order of the attack by arranging the atomic attacks by their cost-nonsuccess ratio (Theorem 3). Finally, it applies the OR-rule, i.e. checks if all min-terms are of negative utility.

The second algorithm has running time $O(2^m)$, were m is the number of atomic attacks, because constructing a DNF is of exponential complexity. However, it terms of practical applicability, the second algorithm has a big advantage over the first one. The reason is how the min-terms are handled by the two algorithms. To imply the negative utility of a min-term, the first algorithm has to find an atomic attack in the min-term that is of negative utility (and then apply the AND-rule). Such an atomic attack may not exist in the min-term, while the min-term as a whole still has negative utility. The second algorithm computes the exact utility of the min-term instead and hence determines the negativity of min-terms without errors.

The main drawback of Algorithm 2 is that an atomic attack may appear in a considerable fraction of the (exponentially large) set of min-terms and hence, its cost may reduce from a high value to a negligible one.

5 Infinite Repetition Model

The models proposed so far assume that the atomic attacks are tried just once by the adversary. This may not be the case in the real world. If an adversary fails with an atomic attack $\mathcal{X}_j$, then it might be that $\mathcal{X}_j$ can be tried again by the adversary, i.e. in case of failure (with probability q_j) no assignments are made and the position of the attack game remains the same. Such a game is called

attack game with repetition. Note that repetition only makes the attacks easier, i.e. if $U_\infty(\mathcal{A})$ denotes the utility of $\mathcal{A}$ in an attack game with repetition, then $U(\mathcal{A}) \leq U_\infty(\mathcal{A})$. Hence, if we manage to prove that $U_\infty(\mathcal{A}) \leq 0$, then it implies $U(\mathcal{A}) \leq 0$. Hence, it is safe to assume that repetition is always allowed.

5.1 Conversion to Infinite Repetition and Failure-Free Models

If an optimal strategy S chooses an atomic attack $\mathcal{X}$ as the next move and fails then the game remains the same and S chooses $\mathcal{X}$ again. Hence, any atomic attack is iterated until success or getting caught.

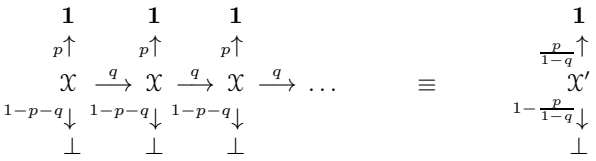


Fig. 3. A repeatable atomic attack $\mathcal{X}$ with parameters (p, q, C, Π) is equivalent to a non-repeatable attack $\mathcal{X}'$ with parameters $(\frac{p}{1-q}, 0, \frac{C}{1-q}, \Pi)$

Theorem 7. *Every repeatable atomic attack $\mathcal{X}$ with parameters (p, q, C, Π) can be replaced with a unrepeatable atomic attack $\mathcal{X}'$ with parameters $(\frac{p}{1-q}, 0, \frac{C}{1-q}, \Pi)$ without changing the utility of the game.*

Proof. The winning probability p' while "iterating" $\mathcal{X}$ (Fig. 3) is $p' = p + qp + q^2p + \ldots = p \cdot (1 + q + q^2 + \ldots) = \frac{p}{1-q}$, because the success probability of the first trial is p, the probability of success at the second trial is $q \cdot p$, the prob. that we win at the third trial is $q^2 \cdot p$, etc. The average cost C' during the iteration is

$$C' = (1-q) \cdot C + q(1-q) \cdot 2C + q^2(1-q) \cdot 3C + \ldots = C + qC + q^2C + \ldots = \frac{C}{1-q} \ .$$

Indeed, the probability that the game ends (with win or penalty) at the first trial is $1-q$, the probability of stopping at the second trial is $q(1-q)$, etc. If the game ends at the first trial, the costs are C. If the game ends at the second trial, the costs are $2C$, etc. The event that the game never ends is a null event. □

Hence, if we apply such a transformation to all atomic attacks, we get a so-called *failure-free* attack tree, in which $q_j = 0$ every atomic attack $\mathcal{X}_j$.

Theorem 8. *Failure-free attack trees $\mathcal{A}$ have optimal strategies that are non-adaptive, i.e. there is a fixed ordering $\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k}$ of atomic attacks that the optimal strategy follows. Moreover, the formula $x_{i_1} \wedge \ldots \wedge x_{i_k}$ is a critical min-term of the Boolean formula $\mathcal{F}$ of $\mathcal{A}$.*

Proof. Let S be an optimal strategy for $\mathcal{A}$ and $\mathcal{X}_{i_1}$ be the best move. As $\mathcal{A}$ is failure-free, there may be two possible outcomes of trying $\mathcal{X}_{i_1}$:

– the adversary "gets caught", the game ends;
– $\mathcal{X}_{i_1}$ is successful and the next game to play is $\mathcal{A}_{x_{i_1}=1}$.

Hence, if the game does not end in the first move, the next move to play is the best move x_{i_2} of the game $\mathcal{A}_{x_{i_1}=1}$. Let $(\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k})$ be the order of trials suggested by S. The formula $x_{i_1} \wedge \ldots \wedge x_{i_k}$ is a min-term of $\mathcal{F}$ (the Boolean formula of $\mathcal{A}$), because otherwise S never wins and $U(\mathcal{A})$ would be negative, which is impossible due to Corollary 1 of Thm. 1. If there is x_{x_j} such that $x_{i_1} \wedge \ldots \wedge x_{i_{j-1}} \wedge x_{i_{j+1}} \wedge \ldots \wedge x_{i_k}$ is still a min-term of $\mathcal{F}$, the atomic attack $\mathcal{X}_{i_j}$ can be skipped and we have a strategy S' with $U(S') > U(S)$, which is impossible because S is optimal. Hence, $x_{i_1} \wedge \ldots \wedge x_{i_k}$ is a critical min-term. □

5.2 Algorithm 3: Exact Utility in the Infinite Repetition Model

Theorem 9. *The exact utility in the fully adaptive model can be computed as follows: (1) find the atomic attack $\mathcal{X}$ with the smallest ratio $\frac{c}{1-q-p}$ (where $c = C + (1-q-p)\Pi$); and (2) compute recursively*

$$U_\infty(\mathcal{A}) = \max\left\{0,\ \ \frac{-c}{1-q} + \frac{p}{1-q} \cdot U_\infty(\mathcal{A}_{x=1}),\ \ U_\infty(\mathcal{A}_{x=0})\right\}\ , \qquad (12)$$

with initial conditions $U_\infty(\mathbf{1}) = P$ and $U_\infty(\mathbf{0}) = 0$, where $\mathbf{1}$ and $\mathbf{0}$ denote attack games with Boolean functions $\mathcal{F} \equiv 1$ and $\mathcal{F} \equiv 0$, respectively.

Proof. If $U_\infty(\mathcal{A}) > 0$, then by Theorems 3, 7, 8, every optimal strategy in the infinite repetition model is non-adaptive and associated with a critical min-term $x_{i_1} \wedge \ldots \wedge x_{i_k}$ of the corresponding Boolean function and the first move of which is to try the atomic attack $\mathcal{X} \in \{\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k}\}$ with the smallest cost-nonsuccess ratio $\frac{c'}{1-p'} = \frac{c}{1-q-p}$, where $p' = \frac{p}{1-q}$ and $c' = \frac{c}{1-q}$ are the transformed parameters. So, it is sufficient to generate all critical min-terms the variables of which are ordered according to the ratios of the corresponding atomic games. If the optimal strategy is a min-term that contains the atomic game $\mathcal{X}$ with the smallest ratio, then $\mathcal{X}$ is the first move and the utility is $\frac{-c}{1-q} + \frac{p}{1-q} \cdot U_\infty(\mathcal{A}_{x=1})$. If the optimal strategy does not involve $\mathcal{X}$, then the utility is $U_\infty(\mathcal{A}_{x=0})$. □

Algorithm 3 runs in time $O(2^m)$, where m is the number of atomic games, and is much more efficient compared to the algorithm that finds exact outcome in the fully adaptive model without repetition. While having approximately the same complexity as Algorithm 2, Algorithm 3 seems to have a big advantage, because it avoids the most important drawback of Algorithm 2—the change of parameters in Algorithm 3 is moderate and does not depend on the DNF size.

6 Open Questions and Further Work

Algorithm 3 has several advantages over the previous methods of estimating adversaries' utility: it has the same complexity than the parallel algorithm of

Jürgenson-Willemson [8] but the bound it gives is much more reliable. Still, the exponential complexity is unsuitable in many practical cases where attack-trees are large. It would be interesting to study algorithms that combine the AND-OR rules (for larger trees) and Algorithm 3 for sufficiently small subtrees of the attack-tree. Such an approach seems promising because attack trees for real systems are "modular", i.e. they consist of subtrees of moderate size that are relatively independent of each other (contain a small number of common atomic attacks). It might be the case that there are better AND-OR rules in the infinite repetition model than in the model without repetition. This needs more research.

Even having some outstanding qualities the model we propose still relies on the ability of analysts to construct precise attack trees that capture all attack vectors. If some atomic attacks are forgotten and not included in the attack tree, they may define a profitable attack suite, while the answer given by the model may imply that the system is secure. This means that security has to be a cyclic process when the list of threats and vulnerabilities is revised constantly.

All attack tree models depend on the metrics assigned to the leaves of the attack tree. Unfortunately there are no good frameworks for metric estimation. Even though some effort has been made to establish methods [4,12,14] for metric calculation, a lot of work has to be done in this field before quantitative attack tree models become as useful as they potentially can be.

References

1. Buldas, A., Laud, P., Priisalu, J., Saarepera, M., Willemson, J.: Rational Choice of Security Measures Via Multi-parameter Attack Trees. In: López, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 235–248. Springer, Heidelberg (2006)
2. Buldas, A., Mägi, T.: Practical Security Analysis of E-Voting Systems. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 320–335. Springer, Heidelberg (2007)
3. Convery, S., Cook, D., Franz, M.: An attack tree for the Border Gateway Protocol (2004)
4. Downs, D.D., Haddad, R.: Penetration testing—the gold standard for security rating and ranking. In: Proceedings of the 1st Workshop on Information-Security-System Rating and Ranking (WISSRR), Williamsburg, Virginia, USA (2001)
5. Edge, K.S.: A framework for analyzing and mitigating the vulnerabilities of complex systems via attack and protection trees. Ph.D. thesis, Air Force Institute of Technology, Ohio (2007)
6. Ericson, C.: Fault tree analysis—a history. In: The 17th International System Safety Conference (1999)
7. Jürgenson, A., Willemson, J.: Serial Model for Attack Tree Computations. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 118–128. Springer, Heidelberg (2010)
8. Jürgenson, A., Willemson, J.: Computing Exact Outcomes of Multi-parameter Attack Trees. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 1036–1051. Springer, Heidelberg (2008)
9. Mauw, S., Oostdijk, M.: Foundations of Attack Trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006)

10. Niitsoo, M.: Optimal Adversary Behavior for the Serial Model of Financial Attack Trees. In: Echizen, I., Kunihiro, N., Sasaki, R. (eds.) IWSEC 2010. LNCS, vol. 6434, pp. 354–370. Springer, Heidelberg (2010)
11. Schneier, B.: Attack trees: Modeling security threats. Dr. Dobbs Journal 24(12), 21–29 (1999)
12. Schudel, G., Wood, B.: Adversary Work Factor As a Metric for Information Assurance. In: Proceedings of the 2000 Workshop on New Security Paradigms, Ballycotton, County Cork, Ireland, pp. 23–30 (2000)
13. Weiss, J.D.: A system security engineering process. In: Proc. of the 14th National Computer Security Conf., pp. 572–581 (1991)
14. Wood, B., Bouchard, J.: Read team work factor as a security measurement. In: Proc. of the 1st Workshop on Information-Security-System Rating and Ranking (WISSRR 2001), Williamsburg, Virginia, USA (2001)

A Computational Examples

To show how the proposed models may be applied, we analyze the attack tree of Fig. 1 with the parameters given in Tab. 2. We assume that $\mathcal{X}_1$ can be repeated multiple times, and let the prize be $P = 80$.

Table 2. Atomic attack parameters of the attack tree shown in Fig. 1

Atomic attack	Cost	Success probability	Failure probability	Repeatability
$\mathcal{X}_1$	$c_1 = 6$	$p_1 = 0.2$	$q_1 = 0.2$	Repeatable
$\mathcal{X}_2$	$c_2 = 6$	$p_2 = 0.2$	$q_2 = 0.2$	Non-repeatable
$\mathcal{X}_3$	$c_3 = 3$	$p_3 = 0.2$	$q_3 = 0.2$	Non-repeatable

A.1 Uniform Cost Reduction

First, the repeatable atomic attack $\mathcal{X}_1$ has to be substituted with unrepeatable version $\mathcal{X}_1'$ of itself with the parameters $c_1' = \frac{c_1}{1-q_1} = \frac{6}{0.8} = 7.5$, $p_1' = \frac{p_1}{1-q_1} = \frac{0.2}{0.8} = 0.25$. As $\mathcal{X}_1$ is involved in both $\mathcal{A}_1$ and $\mathcal{A}_2$, the cost of $\mathcal{X}_1'$ is reduced according to the rules of uniform cost reduction $c_1'^{red} = \frac{c_1'}{2} = 3.75$, producing the atomic attack $\mathcal{X}_1'^{red}$. The atomic attacks in $\mathcal{A}_1' = \mathcal{X}_1'^{red} \wedge \mathcal{X}_2$ and $\mathcal{A}_2' = \mathcal{X}_1'^{red} \wedge \mathcal{X}_3$ have to be sorted according to the increasing cost-nonsuccess ratio.

$$\frac{c_1'^{red}}{1-p_1'} = \frac{3.75}{0.75} = 5 \ , \quad \frac{c_2}{1-p_2} = \frac{6}{0.8} = 7.5 \ , \quad \frac{c_3}{1-p_3} = \frac{3}{0.8} = 3.75 \ .$$

In $\mathcal{X}_1'$, the attack $\mathcal{X}_1'^{red}$ must be tried first (because $5 < 7.5$), and in $\mathcal{X}_2'$, we have to try $\mathcal{X}_3$ first (because $3.75 < 5$). Hence,

$$U(\mathcal{A}_1) = -c_1'^{red} + p_1'(-c_2 + p_2 P) = -3.75 + 0.25(-6 + 0.2 \cdot 80) = -1.25 \ ,$$
$$U(\mathcal{A}_2) = -c_3 + p_3(-c_1'^{red} + p_1' P) = -3 + 0.2(-3.75 + 0.25 \cdot 80) = 0.25 \ .$$

Since $U(\mathcal{A}_2) > 0$, according to the OR-rule of attack trees, $\mathcal{A}$ may have positive utility and a profitable attack suite.

A.2 Non-Uniform Cost Reduction

We reduce c'_1 in $\mathcal{A}_1$ to $\frac{c'_1}{3}$ and in $\mathcal{A}_2$ to $\frac{2c'_1}{3}$. The ratio $\frac{c_1^{'red}}{1-p'_1}$ will then be $\frac{10}{3} \approx 3.33$ in $\mathcal{A}'_1$ and $\frac{20}{3} \approx 6.67$ in $\mathcal{A}'_2$. This means that the optimal order of atomic attacks in $\mathcal{A}'_1$ and $\mathcal{A}'_2$ remains the same as in the uniform cost reduction, and we have:

$$U(\mathcal{A}_1) = -c_1^{'red} + p'_1(-c_2 + p_2 P) = -7.5/3 + 0.25(-6 + 0.2 \cdot 80) = 0 \ ,$$
$$U(\mathcal{A}_2) = -c_3 + p_3(-c_1^{'red} + p'_1 P) = -3 + 0.2(-5 + 0.25 \cdot 80) = 0 \ .$$

The non-uniform cost reduction shows that there are no utilities larger than zero and no beneficial attacking strategies, so the system which is being analyzed is secure against rational adversaries. Since by using the reduced costs we give additional power to adversaries, using non-uniform cost reduction means that we control how this additional power is redistributed, however we still get an artificial state which is more favorable to the adversary than the original one, hence it gives a valid result.

A.3 Infinite Repetition Model

All atomic attacks $\mathfrak{X}_1, \mathfrak{X}_2, \mathfrak{X}_3$ must be substituted with multiple repetition versions of themselves $\mathfrak{X}'_1, \mathfrak{X}'_2, \mathfrak{X}'_3$ with the following parameters:

$$c'_1 = c'_2 = \frac{6}{0.8} = 7.5, \quad c'_3 = \frac{3}{0.8} = 3.75, \quad p'_1 = p'_2 = p'_3 = \frac{0.2}{0.8} = 0.25 \ .$$

The ratios of the atomic attacks are the following:

$$\frac{c'_1}{1-p'_1} = \frac{7.5}{0.75} = 10 \ , \quad \frac{c'_2}{1-p'_2} = \frac{7.5}{0.75} = 10 \ , \quad \frac{c'_3}{1-p'_3} = \frac{3.75}{0.75} = 5 \ .$$

Hence, the utility of $\mathcal{A}_1$ and $\mathcal{A}_2$ is computed as follows:

$$U_\infty(\mathcal{A}_1) = -7.5 + 0.25(-7.5 + 0.25 \cdot 80) = -4.375 \ ,$$
$$U_\infty(\mathcal{A}_2) = -3.75 + 0.25(-7.5 + 0.25 \cdot 80) = -0.625 \ .$$

Since $U_\infty(\mathcal{A}_2) = -0.625$ is the larger of the two utilities, this means that $U_\infty(\mathcal{A}) = -0.625$ and it represents an upper bound of the utility $U(\mathcal{A})$ of the original attack tree $\mathcal{A}$ (where $\mathfrak{X}_2$ and $\mathfrak{X}_3$ are not repeatable. Hence, $U(\mathcal{A})$ must also be negative.

Using Signaling Games to Model the Multi-step Attack-Defense Scenarios on Confidentiality

Jingqiang Lin[1], Peng Liu[2], and Jiwu Jing[1]

[1] State Key Lab of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China
[2] College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802, USA

Abstract. In the multi-step attack-defense scenarios (MSADSs), each rational player (the attacker or the defender) tries to maximize his payoff, but the uncertainty about his opponent prevents him from taking the suitable actions. The defender doesn't know the attacker's target list, and may deploy unnecessary but costly defenses to protect machines not in the target list. Similarly, the attacker doesn't know the deployed protections, and may spend lots of time and effort on a well-protected machine. We develop a repeated two-way signaling game to model the MSADSs on confidentiality, and show how to find the actions maximizing the expected payoffs through the equilibrium. In the proposed model, on receiving each intrusion detection system alert (i.e., a signal), the defender follows the equilibrium to gradually reduce the uncertainty about the attacker's targets and calculate the defenses maximizing his expected payoff.

Keywords: Attack graph, game theory, multi-step attack-defense scenario, signaling game.

1 Introduction

Game theory has been used to model the interactions of defenders and attackers in networked systems. Assumed to be rational, both the defender and the attacker try to maximize their payoffs in the attack-defense scenario (or game). Thus, they will follow an equilibrium of the game. In such a two-player game, an equilibrium is a pair of strategies $(\mathcal{S}_a, \mathcal{S}_d)$ [8]: if the attacker follows $\mathcal{S}_a$, the defender has to follow $\mathcal{S}_d$ to maximize his payoff; and vice versa. By applying a game to model the attack-defense scenarios and analyzing the equilibrium, we can infer the attacker's intent, objectives and strategies [11], find the suitable defenses [9, 14, 21, 31], and configure intrusion detection systems (IDSs) [13, 30].

This paper focuses on the *multi-step* attack-defense scenarios (MSADSs) on *confidentiality*. In such scenarios, the attacked system consists of machines with vulnerabilities, which can be exploited to intrude them step by step. A vulnerable machine stores confidential information or not, and different attackers are interested in the confidential information on different machines. Some attackers stop once their targets are reached (even if there are uncompromised but vulnerable machines), while others try to compromise all machines for a different

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 118–137, 2012.

target. The MSADSs on confidentiality differ from others on security properties such as availability and integrity. The effective defenses for confidentiality are limited to *prevention* that prevents attack actions, while recovery is useless (e.g., killing malicious processes or clearing infected files), because the attacker probably has stolen the confidential information once he intrudes it. Moreover, if the attacker wants to compromise availability or integrity, his payoff is usually related to how long the system is compromised. Therefore, such scenarios can play infinitely: the attacker compromises the service and the defender recovers it again and again. However, an attack-defense scenario on confidentiality will end after a finite number of actions, because the rational attacker doesn't spend infinite effort to compromise confidentiality.

In the MSADSs, *uncertainty* about the attacker prevents the defender from deploying the suitable defenses. Because the defenses consume time and effort, and often bring inconvenience to users, they are not deployed unconditionally and shall be done based on the comprehensive estimation of defense costs, attack targets and the current stage of the MSADS. However, the defender doesn't know the precise attack target list, and may deploy costly defenses to protect machines not in the list. Similarly, the attacker may spend lots of time and effort on a well-protected machine, because he doesn't know which defenses are deployed.

The *signaling* in MSADSs reduces the defender's uncertainty and help him find the suitable actions. The defender installs an IDS and receives alerts (i.e., signals about the attacker's actions), helping him to infer the attack targets. Similarly, the attacker also receives signals about the defenses; i.e., the attacker learns the result (success or failure) of each attack action and gradually obtains information about the defenses deployed during the scenarios.

When game theory is applied to analyze an attack-defense scenario, the essential features shall be considered; otherwise, the analysis results do not match the results in the real world. In this paper, we propose a game model for the MSADSs on confidentiality, and the following features are reflected: (*a*) *irregular repetition*, the model is composed of a finite number of basic signaling games and the players don't take actions synchronously or by turns; (*b*) *two-way signaling*, both the defender and the attacker receive signals helping them to choose suitable actions, and they also trigger signals to their opponents during the scenario; and (*c*) *uncertainty*, each player has uncertainty about his opponent and the signaling mechanism introduces additional uncertainty.

The proposed model helps to reduce the uncertainty about the attack targets and find the suitable defenses in the MSADSs. To the best of our knowledge, it is the first time to apply signaling games to analyze the procedure that the defender and the attacker collect information (i.e., receive signals) to gradually reduce the uncertainty and find the suitable actions in multi-step attacks. In a basic signaling game, the defender receives a signal (i.e., IDS alert) and finds the action maximizing his expected payoffs in the *whole* scenario through the equilibrium of the basic game. As more signals are received by the defender, the uncertainty about the attacker is reduced step by step and the defenses are optimized gradually.

The rest of this paper is organized as follows. Section 2 describes the MSADSs on confidentiality, and Section 3 presents the repeated two-way signaling game model. A case study is presented in Section 4. We discuss the related work and conclude the proposed game model in Sections 5 and 6, respectively.

2 Multi-step Attack-Defense Scenario on Confidentiality

In this section, the MSADS on confidentiality is described. We present the assumptions, the uncertainty and the signaling mechanism of these scenarios.

2.1 Attack Graph and Pruned Attack Graph

A MSADS is performed by the attacker and the defender, based on an *attack graph* [2, 19, 20, 25]. An attack graph depicts the ways in which the attacker can exploit vulnerabilities to break into the system. Figure 1(a) is an example borrowed from [20], where `webServer`, `fileServer` and `workStation` store different information. Assume that the attacker aims at `workStation`. He firstly intrudes `webServer` by exploiting vulnerability `CVE-2002-0392`. Since `webServer` accesses `fileServer` through the NFS protocol, he can then modify data on `fileServer`. There are two ways to achieve this. If there are vulnerabilities in the NFS service, he can exploit them and get local access on the server; or if the NFS export table isn't configured appropriately, he can modify files by programs like NFS Shell. Once he can modify files on `fileServer`, the attacker installs a Trojan-horse in the executable binaries on `fileServer` that are mounted by `workStation`. The attacker then waits for a user on `workStation` to execute it and obtains the control of this machine to steal the confidential information.

The ways to intrude the system are depicted by the attack graph (denoted as $\widetilde{G}$) in Figure 1(b). It is composed of *fact nodes* (denoted as N_i, $i = 1, 2, \cdots, I$) and *causality relations* (denoted as R_j, $j = 1, 2, \cdots, J$). Each N_i is labeled with a logical statement about the system, and N_i is (a) *reached* by the attacker if the statement is true, or (b) *unreached* otherwise. Each R_j involves an action and represents the derivation of two fact nodes $(N_i, N_{i'})$: if N_i is reached and the action of R_j is performed successfully, the logical statement of $N_{i'}$ becomes

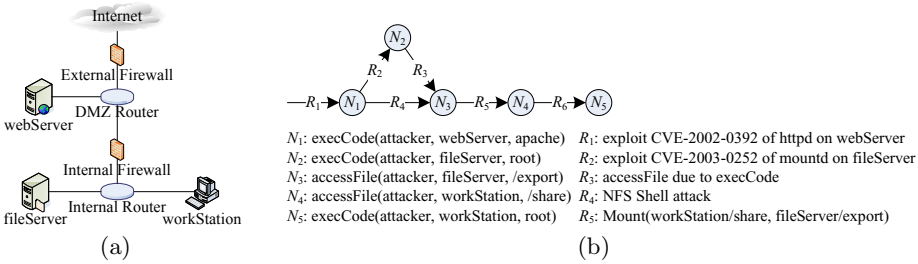


Fig. 1. Attack Graph

true. For example, if the attacker can execute codes on `webServer` (i.e., reaches N_1), he will access files on `fileServer` and reach N_3 through NFS Shell attacks.

In order to reach his interested fact nodes, the attacker needs to activate causality relations step by step. Some causality relations involve *attack actions* that are taken only by the attacker, while others involve actions that take effect *automatically* under the system configuration and setting. The attack action involved in R_j is denoted as A_j. In Figure 1(b), R_1, R_2, R_4 and R_6 involve attack actions. So, the attacker needs to successfully take $\{A_1, A_2, A_6\}$ or $\{A_1, A_4, A_6\}$, before he reaches N_5.

An attack action by the attacker doesn't succeed always (even when there is no defense), and failed actions don't activate the related causality relation. An attack action usually exploits vulnerabilities; e.g., A_1 exploits vulnerability `CVE-2002-0392`. For a certain attack action, two factors affect its success rate: the attacker's knowledge about the vulnerability and the complexity of exploiting the vulnerability [15, 23]. If the vulnerability is very complex or the attacker has little knowledge about it, it is almost impossible to take the action successfully.

Defenses are deployed to reduce attack losses during the MSADSs, such as firewall rules, recovering and patching vulnerabilities. In general, these defenses can be classified as (a) *prevention* that disables or deletes a causality relation and (b) *recovery* that makes a (reached) fact node be unreached. For example, the defender can unmount the shared directory of `workStation` to disable R_5, or recover `webServer` to the uncompromised state when it is suspected to be controlled by the attacker. As mentioned in Section 1, only prevention is effective in the attack-defense scenarios on confidentiality.

Applying prevention to the initial attack graph, results in *pruned attack graphs*. In the example system, the available prevention defenses and the corresponding pruned attack graphs are presented in Figure 2. The prevention disabling R_j, is denoted as D_j. Additional, the defender can deploy a set of defenses simultaneously, not shown in the figure; e.g., configure a firewall rule to drop packets of RPC-100005 between `webServer` and `fileServer` and unmount the shared directories of `workStation` (i.e., deploy the defense-set $\{D_2, D_5\}$).

Definition 1 (Multi-step Attack-Defense Scenario on Confidentiality). *A multi-step attack-defense scenario on confidentiality in $\widetilde{G}$ is a limited sequence of actions taken from $\mathbb{A}$ by the attacker and from $\mathbb{D}$ by the defender.*

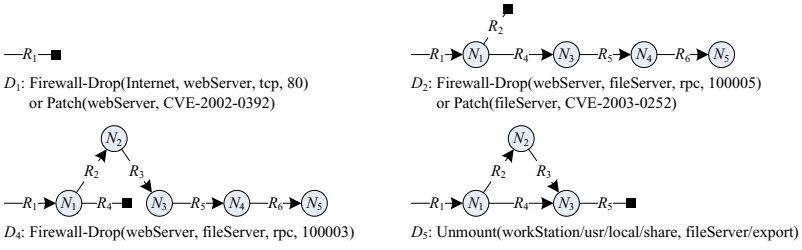


Fig. 2. Examples of defenses and pruned attack graphs

A MSADS on confidentiality ends after a limited number of attack actions, when the attacker (a) obtains the confidential information interested or (b) quits due to the prevention and the difficulties to attack. In this definition, $\widetilde{G} = (\mathbb{N}, \mathbb{R}, \mathbb{A}, \mathbb{D}) = (\{N_i\}, \{R_j\}, \{A_j\}, \{D_j\})$, and $\mathbb{A}$ is the collection of attack actions involved in causality relations. Note that not every R_j involves an attack action. $\mathbb{D}$ is the collection of prevention defenses to disable causality relations, and not every R_j can be disabled by the defender (e.g., R_3).

Assumption 1. *$\widetilde{G}$ is known to the defender, and the attacker knows $\widetilde{G}$ except the detailed vulnerabilities exploited by $\mathbb{A}$.*

Firstly, it is reasonable to assume that the defender who is responsible for system management and security, knows the attack graph. Secondly, a real-world attacker often has some limited knowledge about the attacked system. For example, he can learns the network topology and the location of his interested confidential information. Such information is usually consistent with common sense; e.g., in the example, `webServer` only accesses files for routine operations, sensitive data are stored on `fileServer`, and the most confidential information is on `workStation` and protected by the internal firewall. Sometimes, attackers can even have the system configurations from betrayed operators. Finally, the attacker usually needs to try malicious packets for several times before a successful attack action, even if he knows the detailed vulnerabilities. To reflect this feature, we assume that the attacker knows the attack graph except the detailed vulnerabilities, so the success rate of attack actions is rather low in this model.

This assumption allows the attacker to find the attack paths but requires he to try several times before taking a successful attack action. It gives the defender chance and time to play a game with the attacker. However, this assumption does not tell the attacker which defenses are deployed after the attack-defense scenario starts, as in the real-world MSADSs.

2.2 Target List

When a fact node is reached, attackers have the capability to compromise services of the system, but he needs to take some *supplemental actions* to compromise it. The supplemental actions are not included in the attack graph or involved in any causality relation, and the attacker doesn't reach any more fact nodes by the supplemental actions. For example, the attacker can compromise the WWW service if N_1 is reached. He may shut down the machine, modify files or search sensitive data to compromise availability, integrity or confidentiality.

In the real world, different attackers usually have different targets even when they are attacking a same system. Not all compromises can bring benefits to a specific attackers. Therefore, an attacker may choose to keep the WWW service (and continue to intrude `fileServer`) after reaching N_1, because compromising it doesn't bring any benefits. Moreover, the supplemental actions make the attack be detected and responded more quickly.

Definition 2 (Target List). *A fact node is in the attacker's target list,* if and only if *the attacker obtains benefits (by taking supplemental actions) after reaching this fact node and before reaching any other fact nodes.*

A rational and intended attacker has a limited number of target nodes, denoted as $\mathbb{T} \subseteq \mathbb{N}$. For the example system in Figure 1, the attack targets may include the confidential information on `webServer`, `fileServer` and/or `workStation`.

2.3 Payoff

In the MSADS, each rational player tries to maximize his (expected) payoff. The attacker's payoff is determined by (a) the benefits from reached target fact nodes and (b) the cost of all attack actions. When N_i is reached, the attack benefit is B_i if $N_i \in \mathbb{T}$; otherwise, he obtains nothing. The attack cost is determined by the effort and time of each action (denoted as E_j) and the number of attack actions, either successful or failed. Assume the success rate of A_j is S_j, and the expected cost of a successful A_j after $k-1$ failed attempts (provided that there is no defense) is $E_j \sum_{k=1}^{\infty} k(1-S_j)^{k-1} S_j = \frac{E_j}{S_j}$. And it costs nothing to activate the causality relations involving automatic actions; e.g., R_3 and R_5 in the example.

Assumption 2. *B_i, E_j and S_j are known to both the attacker and the defender.*

B_i can be roughly viewed as the price of the confidential information on each machine, which is publicly known. B_i is a parameter about the fact node, whether N_i is in the target list or not. So, some attackers obtain B_i by reaching N_i while others obtain nothing. Additionally, the attacker usually has experience to estimate E_j and S_j, so can the defender based on statistics, experience and experiments.

The defender's payoff is determined by (a) the defense costs to disable causality relations and (b) the losses due to the reached target fact nodes. The defender maximizes his payoff by *minimizing* the defense costs and attack losses. The cost of D_j is denoted as C_j. When N_i is reached by the attacker, the defender suffers a loss (denoted as L_i) if and only if $N_i \in \mathbb{T}$. Note that a reached fact node does not automatically cause a loss to the defender; if $N_i \notin \mathbb{T}$ is reached, the defender does not suffer any loss.

Assumption 3. *D_j and C_j are known to both the attacker and the defender.*

The defender who is responsible for the system, knows the available prevention defenses and the costs. For the attacker, this assumption is consistent with Assumption 1. Without knowing the detailed vulnerability of R_j, the attacker still can judge whether a causality relation can be disabled by the defender or not. Since the prevention causes only unavailability for users, it is possible for the attacker to estimate C_j.

2.4 Uncertainty

Uncertainty 1. *$\mathbb{T}$ is only known to the attacker, and the defender maintains the probability distribution of $\mathbb{T}$.*

Attackers with different target lists can take a same sequence of actions and the defender cannot understand their targets easily. To reach his targets, an attacker usually has to reach some fact nodes not in the target list. For example, when the attacker is only interested in the confidential information on `workStation` (i.e., $\mathbb{T} = \{N_5\}$), he has to firstly reach N_1, N_3 and N_4 even if the attacker obtains nothing by compromising `webServer` and `fileServer`. For the defender, the uncertainty about the target list is represented as the probability distribution of $\mathbb{T} \subseteq \mathbb{N}$, denoted as $P(\mathbb{T}_k)$ $(k = 1, 2, \cdots, K)$ and $\sum_k P(\mathbb{T}_k) = 1$.

Uncertainty 2. *L_i is only known to the defender, and the attacker maintains the probability function of L_i.*

It is very difficulty for the attacker to estimate L_i. L_i is not equal to and usually much greater than B_i, because it includes indirect and direct losses. Typical indirect losses in the scenarios on confidentiality are compensation to clients, market incompetitiveness, business discontinuity and the expense to rebuild the system, which are unknown to the attackers. We assume that, based on his knowledge and the public information about the attacked system, the attacker can estimate the range of L_i: $L_{i[M]} \geq L_i \geq L_{i[m]}$, and the probability function of L_i is $F(L_i) = \begin{cases} \frac{1}{L_{i[M]} - L_{i[m]}} & \text{if } L_{i[M]} \geq L_i \geq L_{i[m]} \\ 0 & \text{otherwise} \end{cases}$.

2.5 Signaling

The uncertainty about the opponent prevents a player from taking the suitable actions; however, the defender and the attacker will receive signals to reduce the uncertainty with the progress of the MSADS. The defender receives IDS alerts and understands the target list gradually. Then, the defender chooses and deploys defenses based on $P(\mathbb{T})$, C_i and L_i. On the same time, the attacker learns the result (success or failure) of each attack action. The attack result is related to the deployed defenses, helping to deduce L_i.

Reducing the uncertainty helps to find the suitable actions. If the defender knows more about the target list, he won't deploy unnecessary defenses protecting a fact node not in the list. On the other hand, when L_i is greater, the defender prefers to disable the causality relations to N_i. The knowledge about L_i helps the attacker to judge whether a causality relation is disabled or not. So, he can avoid wasting time and effort on a well-protected target, or quitting too early when the causality relations are not disabled.

The defender calculates the initial $P(\mathbb{T})$ based on statistics, experience and the price of the confidential information. During the MSADS, $P(\mathbb{T})$ is updated as the defender receives IDS alerts (i.e., signals). Similarly, the attacker will update $F(L_i)$ after he learns the result of each attack action. Note that the initial $P(\mathbb{T})$ is also known to the attacker, so is the initial $F(L_i)$ to the defender.

The signaling mechanism introduces additional uncertainty. Firstly, the IDS produces false positive and false negative alerts. Secondly, an attack action A_j succeeds only if there is no defense disabling the causality relation R_j. But when

A_j fails, the attacker can not distinguish whether it fails due to the deployed defenses or the complexity of exploiting the vulnerability.

3 Signaling Game Model for the Multi-step Attack-Defense Scenarios on Confidentiality

In this section, we firstly introduce the basic signaling game, and then extend it to handle the two-way signaling mechanism in the MSADSs on confidentiality, focusing on how each player to gradually reduce the uncertainty about his opponent and find the action maximizing his expected payoff.

3.1 Basic Signaling Game

A basic signaling game [8] is a two-step dynamic game with incomplete information between two players, called the *sender* and the *receiver*. The sender belongs to a type t_s from the space $\{t_1, t_2, \ldots, t_T\}$. The sender knows its type, while the receiver only knows the probability $P(t_i)$ that the sender belongs to t_i $(i = 1, 2, \cdots, T)$ and $\sum_i P(t_i) = 1$. The game is performed as follows:

1. The sender chooses a signal m from the set of feasible signals.
2. The receiver observes m and calculate a new belief about which types could send m, denote as the conditional probability $P(t_i|m)$ and $\sum_i P(t_i|m) = 1$. Then, the receiver chooses an action a from the set of feasible actions.

The payoffs of the sender and the receiver are determined by t_s, m and a, denoted as $U_s(t_s, m, a)$ and $U_r(t_s, m, a)$, respectively. Different types of senders have different payoff functions. Each player knows all information except that the receiver doesn't know t_s.

A pure-strategy perfect Bayesian equilibrium of a signaling game is a pair of strategies $(m^*(t_s), a^*(m))$ and a probability distribution $P(t_i|m)$, satisfying:

– Given $P(t_i|m)$ and m, the receiver's action $a^*(m)$ maximizes its expected payoff; i.e., $a^*(m)$ solves

$$\max_{a_k} \sum_i P(t_i|m) U_r(t_i, m, a_k)$$

– Given $a^*(m)$, the sender's signal $m^*(t_s)$ maximizes its payoff; i.e., $m^*(t_s)$ solves

$$\max_{m_j} U_s(t_s, m_j, a^*(m_j))$$

– For each m, if there exists t_i such that $m^*(t_i) = m$, $P(t_i|m)$ follows Bayes' rule and the sender's strategy:

$$P(t_i|m) = \begin{cases} \frac{P(t_i)}{\sum_{j, m^*(t_j)=m} P(t_j)} & \text{if } m^*(t_i) = m \\ 0 & \text{otherwise} \end{cases}$$

3.2 Analogy of MSADSs and Basic Signaling Games

In the the equilibrium of a basic signaling game, the receiver uses signal m to reduce his uncertainty about the sender (i.e., update $P(t_i)$ to $P(t_i|m)$), and then chooses the action maximizing his expected payoff. From the defender point of view, the MSADS is composed of several phases, each of which can be viewed as a basic signaling game. In each phase or game, the defender receives an IDS alert and may deploy some defenses. Note that the defender doesn't know the precise target list (i.e., the type of the sender in these games) and only has the probability distribution. At the same time, the attacker views the MSADS as another sequence of phases, each of which is also a basic signaling game. In these games, the attacker doesn't know the attack loss L_i (i.e., the type of the sender) and acts as another receiver: he learns the result of the last attack action (i.e., a signal) and takes the next action.

There are two notes on this analogy as below. Firstly, although the attacker or the defender doesn't send signals intentionally, the IDS alerts are triggered by the attack actions and the result of attack actions are affected by the deployed defenses[1]. Then, the defender regards his opponent as the sender sending signals, so does the attacker. Secondly, the payoffs of the defender and the attacker are determined by the target list, the attack losses, the attack actions and the deployed defenses, which can be considered as the sender's type, the signal and the receiver's action in basic signaling games. Finally, both the defender and the attacker act as the receivers in different basic signaling games, and each of them follows an equilibrium to reduce the uncertainty and maximize his payoff.

3.3 Repeated Two-Way Signaling Game Model

In the proposed game model, the MSADS on confidentiality makes progress as follows, shown in Figure 3(a):

- On receiving an IDS alert $m_{[d]}$, the defender calculates the equilibrium of a basic signaling game as the receiver, i.e., updates $P(\mathbb{T}_k)$ to $P(\mathbb{T}_k|m_{[d]})$ and finds the defenses minimizing the expected defense costs and attack losses.
- On learning the result of his last attack action $m_{[a]}$, the attacker calculates the equilibrium of another basic signaling game as the receiver, i.e., updates $F(L_i)$ to $F(L_i|m_{[a]})$, estimates the deployed defenses and takes the action maximizing the expected benefits and minimizing the expected attack costs.

As the attacker and the defender do in the real world, in the proposed game model, each player takes actions in his own way and there is no synchronization. Therefore, in the attacker's and the defender's views, the game model has different elements. We define *attack phases* and *defense phases* as follows.

Definition 3 (Attack Phase). *An attack phase is composed of three sequential steps by the attacker: learn the last attack result, find the next attack action, and take the action.*

[1] As mentioned in Section 2.5, the IDS cannot accurately detect every attack action, and the results of attack actions are also related to the complexity of exploiting vulnerabilities. They are considered as the uncertainty of the signaling mechanism.

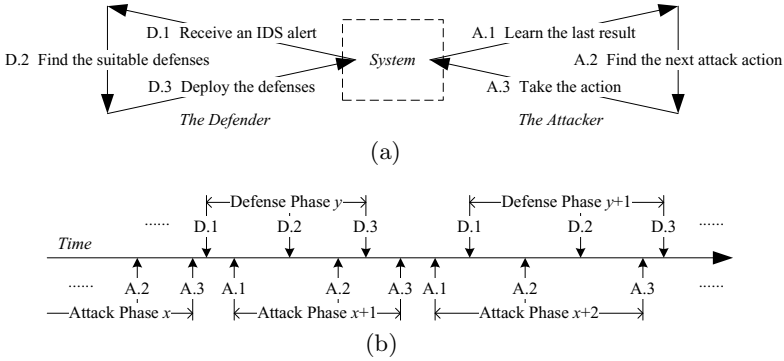


Fig. 3. Progress of the game

Definition 4 (Defense Phase). *A defense phase is composed of three sequential steps by the defender: receive an IDS alert, find the suitable defenses, and deploy the defenses.*

As shown in Figure 3(b), the MSADS is composed of a finite number of attack phases in the attacker's view or defense phases in the defender's view. Due to IDS false positive or negative alerts, the amount of attack phases may be unequal to that of defense phases; and attack phases and defense phases don't start by turns due to IDS alert delays.

3.4 Features of the Repeated Two-Way Signaling Game

In the proposed game model, the following essential features of the MSADSs on confidentiality are emphasized:

Two-Way Signaling. Each player plays the roles of both the sender and the receiver of basic signaling games. In the MSADSs, the attacker receives signals (i.e., attack results) and triggers signals to the defender (i.e., IDS alerts), while the defender receives IDS alerts and deploys defenses affecting attack results.

Uncertainty. Two types of uncertainty are reflected. The uncertainty about opponents is handled as the belief about the sender's type, and the uncertainty of signals is considered when a player updates the belief about his opponent.

Irregular Repetition. The basic signaling game is irregularly repeated with changing parameters. Either the attacker or the defender acts as the sender in his own way, and they don't take actions cooperatively or by turns. As the scenario goes on, the parameters of the basic signaling games change. For example, when the attacker reaches a fact node, his action space and the set of feasible signals change; the set of available preventions also changes after the defender deploys some defenses. Moreover, $P(\mathbb{T}_k)$ and $F(L_i)$ are updated in each phase.

3.5 Equilibrium of Basic Signaling Games

In each phase, a player follows the equilibrium of the basic signaling game to reduce the uncertainty about his opponent. The initial probability distribution of the target list, denoted as $P_0(\mathbb{T})$, is updated to $P_y(\mathbb{T}) = P_{y-1}(\mathbb{T}|m_y[d])$ step by step after the defender receives $m_y[d]$ in the y^{th} defense phase. Similarly, the attacker calculates $F_x(L_i) = F_{x-1}(L_i|m_x[a])$ in the x^{th} attack phase.

Attack Phase. The attacker's next attack action depends on the chosen *attack path*. An attack path is a part of the attack graph, in which the attacker wants to activate all causality relations and reach all fact nodes. The attacker doesn't change his targets, but may change his attack graph in the scenario.

To find the attack path maximizing his expected payoff, the attacker uses the received signals to estimate whether each causality relation is disabled or not. In the x^{th} attack phase, the attacker updates the probability that R_j is disabled (i.e., D_j is deployed) from $P_{x-1}(R_j)$ to $P_x(R_j)$ as follows:

- The attacker knows the current *stage* (denoted as G_{x-1}), representing the causality relations activated and the fact nodes reached. The signal received in the x^{th} phase is the result of the last attack action $A_{j_{x-1}}$.
- If $A_{j_{x-1}}$ fails, the stage keeps unchanged (i.e., $G_x = G_{x-1}$) and $P_x(R_{j_{x-1}}) = P_{x-1}(R_{j_{x-1}}|\hat{A}_{j_{x-1}}) = \frac{P(\hat{A}_{j_{x-1}}|R_{j_{x-1}})P_{x-1}(R_{j_{x-1}})}{P(\hat{A}_{j_{x-1}}|\hat{R}_{j_{x-1}})P_{x-1}(\hat{R}_{j_{x-1}})+P(\hat{A}_{j_{x-1}}|R_{j_{x-1}})P_{x-1}(R_{j_{x-1}})}$ $= \frac{P_{x-1}(R_{j_{x-1}})}{(1-S_{j_{x-1}})(1-P_{x-1}(R_{j_{x-1}}))+P_{x-1}(R_{j_{x-1}})}$. Here, $P(\hat{\Omega})$ represents the probability that Ω is false. For other $j \neq j_{x-1}$, $P_x(R_j) = P_{x-1}(R_j)$.
- If $A_{j_{x-1}}$ succeeds, $R_{j_{x-1}}$ is not disabled and $P_x(R_{j_{x-1}}) = 0$. The attacker activates $R_{j_{x-1}}$ on G_{x-1} to obtain G_x, and updates $F_{x-1}(L_i)$ to $F_x(L_i)$:
  - Based on $P_0(\mathbb{T})$, $F_{x-1}(L_i)$ and G_{x-1}, the attacker analyzes the possible defense-sets, denoted as $\bar{\mathbb{D}}_u \subseteq \mathbb{D}$ $(u = 1, 2, \cdots, U)$. The probability of $\bar{\mathbb{D}}_u$ is $P_{x-1}(\bar{\mathbb{D}}_u) = \int\cdots\int_{Max_u(L)} \prod_i F_{x-1}(L_i)\mathrm{d}L_1\cdots\mathrm{d}L_I$; here, $Max_u(L)$ represents the space of L_i where $\bar{\mathbb{D}}_u$ maximizes the defender's expected payoff (i.e., minimizes the defense costs and attack losses; the detailed calculation of the defender's payoff is described in the remainder).
  - If any $\bar{\mathbb{D}}_u$ violates G_x (i.e., $\bar{\mathbb{D}}_u$ disables a causality relation that has been activated), the space $Max_u(L)$ is asserted to be impossible and the attacker follows Bayes' rule to update $F(L_i)$. Then, the attacker re-analyzes $P_x(\bar{\mathbb{D}}_u)$ based on $F_x(L_i)$, and $P_x(R_j) = \sum_{D_j \in \bar{\mathbb{D}}_u} P_x(\bar{\mathbb{D}}_u)$.
  - If there is not any violation, $F_x(L_i) = F_{x-1}(L_i)$ and $P_x(R_j) = P_{x-1}(R_j)$ for $j \neq j_{x-1}$.

Additional, in the 1^{st} attack phase, a *virtual* successful action is assumed, thus $G_1 = G_0 = \emptyset$ and $F_1(L_i) = F_0(L_i)$. And $P_1(R_j)$ is calculated based on $P_0(\mathbb{T})$, $F_1(L_i)$ and G_1.

The attacker compares his expected payoff for each path, and chooses the best one. For example, if the target is N_3, there are two alternative paths: $R_1 \rightarrow R_2 \rightarrow R_3$ and $R_1 \rightarrow R_4$. Then, the expected benefits of the two paths in the *whole* scenario (not only in the x^{th} attack phase) are $B_3 \prod_{1,2,3}(1 - P_x(R_j))$ and

$B_3 \prod_{1,4}(1 - P_x(R_j))$, respectively. The attack costs are $AC_x + \sum_{j\in\{1,2\}, R_j \notin G_x} \frac{E_j}{S_j}$ and $AC_x + \sum_{j\in\{1,4\}, j\notin G_x} \frac{E_j}{S_j}$. Here, AC_x is the total attack costs before the x^{th} phase, E_j and S_j are the cost and the success rate of A_j, respectively; and note that R_3 is an automatic causality relations not involving any attack action.

Defense Phase. On receiving an IDS alert of A_{j_y} in the y^{th} defense phase, the defender firstly calculates the probability $P(G_v^*)$ of each possible stage (denoted as G_v^*, $v = 1, 2, \cdots, V$) based on the IDS alert history, and then updates $P_{y-1}(\mathbb{T})$ to $P_y(\mathbb{T})$ as follows:

- For each $P_{y-1}(\mathbb{T}_k) > 0$, find the attack path maximizing the expected payoff for the attacker with $\mathbb{T}_k$, denoted as H_k.
- If any H_k violates the alert history (i.e., H_k doesn't cover any G_v^*), it is impossible for an attacker with $\mathbb{T}_k$ to trigger these IDS alerts and $P_y(\mathbb{T}_k) = 0$.
- Follow Bayes' rule to update $P_y(\mathbb{T}_k) = \frac{P_{y-1}(\mathbb{T}_k)}{\sum_{P_y(\mathbb{T}_k)\neq 0} P_{y-1}(\mathbb{T}_k)}$ for other $\mathbb{T}_k$.

The defender knows the deployed defense-set $\mathbb{D}_{y-1} \subseteq \mathbb{D}$. From the available defense-sets $\mathbb{D}_w^\star \subseteq \mathbb{D} \setminus \mathbb{D}_{y-1}$ $(w = 1, \cdots, W)$, the defender finds the best one (denoted as $\mathbb{D}_y^\Delta$) in the *whole* scenario as follows:

- Let $\mathbb{D}_y^\star$ be $\mathbb{D}_w^\star \cup \mathbb{D}_{y-1}$, and the total costs are the sum of every defense cost C_j; i.e., $TC = \sum_{D_j \in \mathbb{D}_y^\star} C_j$.
- The expected losses consist of two parts: the fact nodes not protected by $\mathbb{D}_y^\star$, i.e., $LN = \sum_{N_i \in \mathbb{T}_k, N_i \in \widetilde{G}_{\mathbb{D}_y^\star}} P(\mathbb{T}_k) L_i$, and the ones that will be protected but might have been reached, i.e., $LR = \sum_{N_i \in \mathbb{T}_k, N_i \notin \widetilde{G}_{\mathbb{D}_y^\star}, N_i \in G_v^*} P(G_v^*) P(\mathbb{T}_k) L_i$. Here, $\widetilde{G}_{\mathbb{D}_y^\star}$ is the pruned attack graph after $\mathbb{D}_y^\star$ is deployed.

$\mathbb{D}_y^\Delta$ is the defense-set $\mathbb{D}_w^\star$ with the minimal sum of $TC + LN + LR$. If $\mathbb{D}_y^\Delta = \emptyset$, then $\mathbb{D}_y = \mathbb{D}_{y-1}$ and this defense phase ends; otherwise, the defender deploys $\mathbb{D}_y^\Delta$ when it is imperative enough.

3.6 Opportunity to Deploy Defenses

Even when $\mathbb{D}_y^\Delta \neq \emptyset$, the defender may not deploy $\mathbb{D}_y^\Delta$ for the following reasons. Firstly, different from the receiver in a basic signaling game, the defender still has the chance to take actions if he deploys nothing in the current defense phase, because there are still games (or phases) in the future. Secondly, $\mathbb{D}_y^\Delta$ is the best action, only relative to the signals received in the past y defense phases. As more signals are received and the uncertainty about the target list is gradually reduced, $\mathbb{D}_y^\Delta$ might be found to contain unnecessary defenses.

However, if the defender waits for more signals to calculate a more suitable defense-set, the risk increases that the attacker will reach more targets before he deploys defenses. Therefore, the defender shall balance (a) the costs of $\mathbb{D}_y^\Delta$ and (b) the addition possible losses if these defenses are put off to be deployed in the future. In addition, whether to deploy $\mathbb{D}_y^\Delta$ in the y^{th} defense phase or not, also depends on the IDS alert delays and the time for deploying them. How to find the suitable opportunity to deploy the defenses is beyond the scope of this paper. In the case study in Section 4, a simple and intuitive method is used.

Table 1. Parameters in the case study

Related to Fact Node N_i						Related to Causality Relation R_j						
i	1	2	3	4	5	j	1	2	3	4	5	6
L_i	10	-	70	-	100	C_j	70	5	-	60	40	-
B_i	50	-	150	-	200	S_j	0.05	0.05	-	0.25	-	0.25
$L_{i[m]}, L_{i[M]}$	0, 50	-	30, 80	-	50, 100	E_j	1	1	-	1	-	1

4 Case Study

In this section, we show a MSADS performed on the system in Figure 1. In this case study, the defender's uncertainty about the attack target list is reduced gradually as the scenario goes on, and the suitable defense-set is finally deployed.

4.1 Parameter

Table 1 give the estimated parameters. In the scenario, `webServer` provides WWW services for clients, `fileServer` stores some sensitive data and the most files of the WWW services, and the most confidential information is only on `workStation`. If the attacker reaches N_1, N_3 and N_5, he can steal the confidential information on `webServer`, `fileServer` and `workStation`, respectively. Accordingly, $0 < L_1 < L_3 < L_5$ and $0 < B_1 < B_3 < B_5$. The upper and lower limits of L_i known to the attacker, are also shown. For other fact nodes, $L_i = 0$ and $B_i = 0$, because the attacker doesn't obtain any extra confidential information and the defender doesn't suffer any extra loss when the attacker reaches these fact nodes. For example, when the attacker has reached N_1, he doesn't find any other confidential information by reaching N_2.

Table 1 also shows the cost C_j of defense D_j that disables causality relation R_j, the success rate S_j and cost E_j of attack action A_j involved in R_j. D_1 filters all packets from the Internet to `webServer` and stops the WWW services, so C_1 is the greatest. D_2 prevents the communications from `webServer` to `fileServer` via RPC-100005 and causes unavailability only when `webServer` is mounting directories, while D_4 causes unavailability when `webServer` accesses the files on `fileServer`. If D_5 is deployed, `workStation` cannot conveniently share files with other users but the WWW services are not affected. Therefore, $C_2 < C_5 < C_4 < C_1$ and the example values are listed in Table 1. These values are estimated, assuming that D_1 and D_2 are implemented on the firewalls. The results are similar if they are implemented by patching vulnerabilities.

S_j is estimated based on the complexity to take attack actions. Considering that the attacker doesn't know the detailed vulnerabilities, S_j is 1%, 2% and 5% when the complexity is high, medium and low, respectively. According to the national vulnerability database [16], the complexity of `CVE-2002-0392` and `CVE-2003-0252` is low; so, S_1 and S_2 are 5%. A_4 and A_6 are rather simple and several vulnerabilities can be exploited; then, S_4 and S_6 are much greater. To simplify this case study, we assume that the effort and time of each attack

action is uniform; i.e., $E_j = 1$. Note that L_i and C_j are used to calculate only the defender's payoff, so are B_i and E_j to calculate the attacker's. Thus, it is meaningless and unreasonable to compare L_i and B_i (or C_j and E_i), because they are never used in one equation and have different units.

Assume that the IDS doesn't produce false negative alerts, because the game plays based on an attack graph known to the defender and a signature-based IDS can detect all attack packets. Besides, the false positive rate (denoted as P_p) is assumed to be 0.03 and it is impossible to produce two consecutive false positive alerts for a certain attack action.

In the y^{th} defense phase, the defender uses the following method to decide whether to deploy $\mathbb{D}_y^{\Delta}$ or not: he firstly calculates the expected *additional* losses supposing that the defenses are deployed in the *next* phase, and deploys them only if the additional losses are large enough; in particular, the ratio of the additional losses to the defense costs of $\mathbb{D}_y^{\Delta}$, is greater than 0.1500.

4.2 Progress

In this case study, assume that each attacker has only one target and $P_0(\mathbb{T} = \{N_i\}) = \frac{B_i}{\sum_k B_k}$. The MSADS makes progress with the attacker interested in the confidential information on `workStation` (i.e., $\mathbb{T} = \{N_5\}$). Then, we will show that, $P(\{N_5\})$ is 0.5000 in the beginning and updated to 0.5714 in the 17^{th} defense phase. The defender finds the suitable defense-set $\{D_5\}$ in the 16^{th} defense phase and deploys it in the 17^{th} phase.

Attack Phase 1. Based on $P_0(\mathbb{T})$, the defender's expected defense costs and attack losses (denoted as ED^a) of each rational defense-set is listed as follows. Not all defense-sets are rational; e.g., $\{D_1, D_2\}$ is irrational, because D_2 is useless when D_1 has been deployed to prevent attackers from intruding `webServer`.

- $\emptyset_D$ (i.e., no defense): $ED^a_{\emptyset} = P_0(\{N_1\})L_1 + P_0(\{N_3\})L_3 + P_0(\{N_5\})L_5$
- $\{D_1\}$: $ED^a_{\{1\}} = C_1$
- $\{D_5\}$: $ED^a_{\{5\}} = P_0(\{N_1\})L_1 + P_0(\{N_3\})L_3 + C_5$
- $\{D_2, D_4\}$: $ED^a_{\{2,4\}} = P_0(\{N_1\})L_1 + C_2 + C_4$

The defender deploys nothing if and only if $ED^a_{\emptyset}$ is the minimum. So,

$$P_1(\emptyset_D) = \iiint_{L_{i[m]}, ED^a_{\emptyset} = \min(\cdot)}^{L_{i[M]}} F_0(L_1)F_0(L_3)F_0(L_5)\mathrm{d}L_1\mathrm{d}L_3\mathrm{d}L_5 = 0.5719$$

Similarly, it can be calculated that $P_1(\{D_1\}) = 0.0308$, $P_1(\{D_5\}) = 0.2907$ and $P_1(\{D_2, D_4\}) = 0.1067$. The probabilities that R_j are disabled: $P_1(R_1) = P_1(\{D_1\}) = 0.0308$, $P_1(R_2) = P_1(R_4) = 0.1067$, and $P_1(R_5) = 0.2907$. Note that $P(R_3) = P(R_6) = 0$, because R_3 and R_6 cannot be disabled.

There are two alternative attack paths to reach N_5: $R_1 \to R_2 \to R_3 \to R_5 \to R_6$ and $R_1 \to R_4 \to R_5 \to R_6$. The attacker's expected payoffs are listed:

- $EA_{\{1,2,3,5,6\}} = B_5 \prod_{1,2,5}(1 - P_1(R_j)) - \sum_{1,2,6} \frac{E_j}{S_j} = 78.8336$
- $EA_{\{1,4,5,6\}} = B_5 \prod_{1,4,5}(1 - P_1(R_j)) - \sum_{1,4,6} \frac{E_j}{S_j} = 94.8336$

So, the attacker chooses $R_1 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6$ and takes A_1.

Defense Phase 1. The defender receives an alert of A_1, and analyzes the possible stages: (a) no fact node is reached if the alert is false positive or the attack action fails, and (b) N_1 is reached if the attack action succeeds. Then,

- $P(G_1^* = \emptyset) = P_p + (1 - P_p)(1 - S_1) = 0.9515$
- $P(G_2^* = \{R_1 N_1\}) = (1 - P_p)S_1 = 0.0485$

Given the current pruned attack graph (i.e., no defense is deployed), the best attack path for each type of attacker is:

- $\mathbb{T} = \{N_1\}$: R_1
- $\mathbb{T} = \{N_3\}$: $R_1 \rightarrow R_4$
- $\mathbb{T} = \{N_5\}$: $R_1 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6$

None of them violates the current possible stages, so $P_1(\mathbb{T}) = P_0(\mathbb{T})$. Based on $P_1(\mathbb{T})$ and $P(G_v^*)$, the expected defense costs and attack losses of each rational defense-set is calculated:

- $ED^d_{\emptyset} = P_1(\{N_1\})L_1 + P_1(\{N_3\})L_3 + P_1(\{N_5\})L_5 = 77.5000$
- $ED^d_{\{1\}} = P(G_2^*)P_1(\{N_1\})L_1 + C_1 = 70.0606$
- $ED^d_{\{5\}} = P_1(\{N_1\})L_1 + P_1(\{N_3\})L_3 + C_5 = 67.5000$
- $ED^d_{\{2,4\}} = P_1(\{N_1\})L_1 + C_2 + C_4 = 66.2500$

The best defense-set is $\{D_2, D_4\}$. Note that ED^d calculated based on L_i, $P_y(\mathbb{T}_k)$ and $P(G_v^*)$ by the defender, differs from ED^a based on $F_x(L_i)$, $P_0(\mathbb{T}_k)$ and G_x by the attacker. The defender analyzes the expected additional losses, supposing that $\{D_2, D_4\}$ is deployed in the next phase:

- If $\mathbb{T}$ is $\{N_1\}$ or $\{N_5\}$, there is no additional loss. It makes no difference to deploy $\{D_2, D_4\}$ in this phase or the next one.
- If $\mathbb{T}$ is $\{N_3\}$, there is an additional loss when (a) the current stage is G_2^* and (b) A_4 in the next attack phase succeeds. The loss is $P_1(\{N_3\})P(G_2^*)S_4L_3$. Finally, no defense is deployed, because $\frac{P_1(\{N_3\})P(G_2^*)S_4L_3}{C_2+C_4} = 0.0049 < 0.1500$.

Attack Phase 2. A_1 fails and the attacker learns it. Then, $P_2(R_1)$ is updated to $\frac{P_1(R_1)}{(1-S_1)(1-P_1(R_1))+P_1(R_1)} = 0.0323$, and $P_2(R_j) = P_1(R_j)$ for other R_j. The stage G_x and the probability function $F_x(L_i)$ are kept unchanged; i.e., $G_2 = G_1 = \emptyset$ and $F_2(L_i) = F_1(L_i)$.

The best attack path is still $R_1 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6$, and A_1 is taken again. If A_1 fails always, $P_x(R_1)$ will increase gradually and the attacker will quit in the 56^{th} attack phase. After 55 times of failure, $P_{56}(R_1) = 0.3478$ and the expected payoff of the best path is $B_5 \prod_{1,4,5}(1-P_{56}(R_j)) - \sum_{1,4,6} \frac{E_j}{S_j} - (56-1)E_1 = -0.3484 < 0$.

Defense Phase 2. On receiving the second alert of A_1, the defender doesn't consider it as a false positive one. The possible stages are listed as follows:

- $P(G_1^* = \emptyset) = (1 - S_1)^2 = 0.9025$
- $P(G_2^* = \{R_1 N_1\}) = 1 - (1 - S_1)^2 = 0.0975$

There is no violation between the possible target lists and the IDS alert history, and then $P_2(\mathbb{T}_k) = P_1(\mathbb{T}_k)$. Again, the best defense-set is $\{D_2, D_4\}$ and the defender doesn't deploy them in this phase, either.

However, if the attacker always takes A_1 (and doesn't quit after 55 times of failures) when it fails always, more alerts of A_1 are received and finally $P(G_2^* = \{R_1N_1\}) \approx 1$. The best defense-set is $\{D_2, D_4\}$, and the defender never deploys them because $\frac{P(\{N_3\})S_4L_3}{C2+C4} = 0.1010 < 0.1500$. It can be explained that the defender doesn't deploy anything if the attack threatens no important machines.

Attack Phase 16. A_1 fails always from the 2^{nd} attack phase to the 15^{th}, so the stage and the probability function of L_i are kept unchanged. In the past 15 phases, based on $P_0(\mathbb{T}_k)$, G_x and $F_x(L_i)$, the attacker analyzes the possible defense-sets, which turn to be the same as those in the 1^{st} attack phase.

After the attacker tries it for 15 times, A_1 succeeds. The attacker receives a signal that R_1 is not disabled. Thus, D_1 is not deployed and $P_{16}(D_{\{1\}}) = 0$. It is asserted that $ED^a_{\{1\}} \neq \min(ED^a_{\emptyset}, ED^a_{\{1\}}, ED^a_{\{5\}}, ED^a_{\{2,4\}})$. The probability function $F_{16}(L_i)$ of L_1, L_3 and L_5 are uniformly distributed with the constraints: $L_{i[m]} \leq L_i \leq L_{i[M]}$ and $ED^a_{\{1\}} \neq \min(\cdot)$. And $P_{16}(\mathbb{D}_u)$ is calculated based on the updated $F_{16}(L_i)$: $P_{16}(D_{\emptyset}) = 0.5900$, $P_{16}(D_{\{5\}}) = 0.2999$, and $P_1(D_{\{2,4\}}) = 0.1101$. The attacker chooses $R_1 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6$ and takes A_4, after comparing the expected payoffs of the two attack paths.

Defense Phase 16. The defender receives an alert of A_4. This alert of A_4 may be false positive. The possible stages are listed:

- $P(G_1^* = \emptyset) = P_p(1 - S_1)^{15} = 0.0139$, if the alert of A_4 is false positive, and A_1 fails for 15 times.
- $P(G_2^* = \{R_1N_1\}) = P_p(1 - (1 - S_1)^{15}) + (1 - P_p)(1 - S_4) = 0.7436$, if the alert of A_4 is false positive and one of A_1 succeeds, or A_4 fails.
- $P(G_3^* = \{R_1N_1, R_4N3\}) = (1 - P_p)S_4 = 0.2425$, if A_4 succeeds and one of A_1 succeeds.

There is no violation because the alert of A_4 may be false positive. Thus, $P_{16}(\mathbb{T}_k) = P_0(\mathbb{T}_k)$. After comparing the expected payoffs, the defender finds the best defense-set $\{D_5\}$:

- $ED^d_{\emptyset} = P_{16}(\{N_1\})L_1 + P_{16}(\{N_3\})L_3 + P_{16}(\{N_5\})L_5 = 77.5000$
- $ED^d_{\{1\}} = P(G_2^*)P_{16}(\{N_1\})L_1 + P(G_3^*)P_{16}(\{N_3\})L_3 + C_1 = 77.2951$
- $ED^d_{\{5\}} = P_{16}(\{N_1\})L_1 + P_{16}(\{N_3\})L_3 + C_5 = 67.5000$
- $ED^d_{\{2,4\}} = P_{16}(\{N_1\})L_1 + P(G_3^*)P_{16}(\{N_3\})L_3 + C_2 + C_4 = 72.6156$

The additional loss is $P_{16}(\{N_5\})P(G_3^*)S_6L_5$ if the defender deploys $\{D_5\}$ in the next defense phase. And $\{D_5\}$ is not deployed because $\frac{P_{16}(\{N_5\})P(G_3^*)S_6L_5}{C_5} = 0.0758 < 0.1500$.

Attack Phase 17. The attacker learns that A_4 fails, and $P_{17}(R_4)$ is updated to $\frac{P_{16}(R_4)}{(1-S_4)(1-P_{16}(R_4))+P_{16}(R_4)} = 0.1415$. For other R_j, $P_{17}(R_j) = P_{16}(R_j)$. The attacker calculates the expected payoffs of two attack paths, chooses $R_1 \rightarrow R_4 \rightarrow R_5 \rightarrow R_6$ and takes A_4. In fact, if A_4 fails for 2 more times, then $P(R_4) = 0.2267$

and $EA_{\{1,2,3,5,6\}} > EA_{\{1,4,5,6\}}$. Thus, the attacker would choose $R_1 \rightarrow R_2 \rightarrow R_3 \rightarrow R_5 \rightarrow R_6$ and take A_2 instead.

Defense Phase 17. The defender receives another alert of A_4. So, it is not a false positive alert. The defender calculates $P(G_v^*)$ as follows:

- $P(G_1^* = \{R_1N_1\}) = (1 - S_4)^2 = 0.5625$, if A_4 fails for 2 times; otherwise,
- $P(G_2^* = \{R_1N_1, R_4N3\}) = 1 - (1 - S_4)^2 = 0.4375$

Because the alert of A_4 is not false positive, there is violation: an attacker with $\mathbb{T} = \{N_1\}$ never takes A_4. Thus, $P_{17}(\{N_1\}) = 0$. The defender updates $P(\mathbb{T})$: $P_{17}(\{N_3\}) = \frac{P_{16}(\{N_3\})}{P_{16}(\{N_3\})+P_{16}(\{N_5\})} = 0.4286$ and $P_{17}(\{N_5\}) = 0.5714$. The defender calculates ED^d of each defense-set, and the best one is $\{D_5\}$:

- $ED^d_{\emptyset} = P_{17}(\{N_3\})L_3 + P_{17}(\{N_5\})L_5 = 87.1429$
- $ED^d_{\{1\}} = P(G_2^*)P_{17}(\{N_3\})L_3 + C_1 = 83.1250$
- $ED^d_{\{5\}} = P_{17}(\{N_3\})L_3 + C_5 = 70.0000$
- $ED^d_{\{2,4\}} = P(G_2^*)P_{17}(\{N_3\})L_3 + C_2 + C_4 = 78.1250$

When the attack target is $\{N_1\}$ or $\{N_3\}$, there is no additional possible loss (if that $\{D_5\}$ is deployed in the next defense phase). If $\mathbb{T} = \{N_5\}$, there is an additional loss when (a) the current stage is G_2^* and (b) the next A_6 succeeds. The additional loss is $P_{17}(\{N_5\})P(G_2^*)S_6L_5$ and $\frac{P_{17}(\{N_5\})P(G_2^*)S_6L_5}{C_5} = 0.1563 > 0.1500$. The defender deploys D_5, and the attacker will quit eventually.

5 Related Work

The attacker and the distributed IDS are modeled as two players of a finite game with dynamic information [1], and the equilibrium helps to analyze the relationship between the players' expected payoffs and the false alert rate. Game theory is applied to analyze the network with independent defenders, each of which invests in self-insurances and protections [7]. The defender's attack loss is related to both his self-insurance and the overall protections. The Nash equilibria of this attack-defense game are discussed in [7], for loosely and tightly coupled networks with different attack targets. Different games are designed to model the attackers and the defender of virtual coordinate systems [3], worm propagation [9] and distributed denial-of-service attacks [11], respectively. Signaling games are used in wireless networks to detect and prevent attacks [6, 10, 12, 21, 27], where the defender acts as the receiver that receives alerts and updates the belief about the sender's type (i.e., an attacker or a regular node). In an iterated attack-defense game [4], the defender's uncertainty about the attack costs is reduced as the attacker targets the weakest link. Therefore, an adaptive defense strategy is proposed, and the defenses are optimized repeatedly. The fictitious game model of attackers and IDSs [17], focuses on the uncertainty about the opponent's payoff function and action history. Compared with the work above, firstly, we consider attackers with different targets (or payoff functions), while the existing approaches assume only one type of attackers. Secondly, our model focuses on multi-step attacks but not one-step attacks in the above models, which

is completed by one successful attack action. Finally, our model supports the two-way signaling mechanism (i.e., both the defender and the attacker receive signals to reduce the uncertainty about his opponent), while some of them capture only the defender's uncertain [4, 6, 10–12, 21, 27].

Stochastic games are used to model the interactions of the attacker and the defender [14, 22, 24, 30, 31], where the players' actions result in state transitions of the system. Although the multi-step scenarios can be depicted as multiple state transitions in such stochastic games, they assume attackers with a uniform payoff function, while attackers with different targets are emphasized in our paper. A game tree is proposed to model the MSADSs, where the attacker and the defender take actions by turns [13]. Therefore, our irregularly-repeated model is closer to the real-world scenarios, because the players are not assumed to take actions by turns or synchronously. The analytical model of multi-step attacks [29] tries to find the state transitions connecting vulnerabilities and the cost-saving defenses. It does not consider the dynamic effect from the defenses on the attack actions, which is modeled as the signals to the attacker in our work.

A Bayesian-network (BN) tool [28] is proposed to analyze the uncertainty whether a fact node is reached by attackers, based on the attack graph and IDS alerts. In our case study, a straightforward deduction is applied to analyze the possible stages, and this BN tool can be extended to analyze the stages of the MSADS with our work. The correlated attack modeling language (CAML) specifies a multi-step attack as an attack pattern composed of reusable modules, each of which corresponds to an attack action [5]; then, these patterns are used to identity multi-step attacks from IDS alerts. Based on the prerequisites and consequences of multi-step attacks, IDS alerts are correlated to construct the attack scenarios [18]; and a comprehensive alert correlation framework is proposed [26]. These correlation approaches can be integrated in our game model to analyze the current stages of the MSADS and handle simultaneous attacks by independent attackers, which is a part of our future work.

6 Conclusion and Future Work

A repeated two-way signaling game is proposed to model the MSADSs on confidentiality. It describes the procedure that each player receives signals to gradually reduce the uncertainty about his opponent in the attack-defense scenarios. On receiving an IDS alert, the defender analyzes the attack targets through the equilibrium of basic signaling games. Then, the defender understands the attack list step by step as the scenario goes on and more IDS alerts are received, helping him deploy the suitable defense-set minimizing the attack losses and the defense costs. It is confirmed by the case study, where the optimal defenses are deployed finally. At the same time, the attacker also receives signals and his uncertainty about the defender is gradually reduced.

This work is the first attempt to apply signaling games to model the multi-step attacks, and some simplifications are assumed. We will improve it to handle the MSADSs on other properties such as availability and integrity, and the scenarios

with simultaneous non-cooperative attackers. In the future, we will also discuss the game model when the attack graph becomes large.

Acknowledgement. Jingqiang Lin and Jiwu Jing were partially supported by National Natural Science Foundation of China grant 70890084/G021102, 61003273 and 61003274, and Strategy Pilot Project of Chinese Academy of Sciences sub-project XDA06010702. Peng Liu was partially supported by AFOSR FA9550-07-1-0527 (MURI), ARO W911NF-09-1-0525 (MURI), ARO W911NF1210055, NSF CNS-0905131, NSF CNS- 0916469, and AFRL FA8750-08-C-0137.

References

1. Alpcan, T., Basar, T.: A game theoretic approach to decision and analysis in network intrusion detection. In: IEEE Conference on Decision and Control (CDC), pp. 2595–2600 (2003)
2. Ammann, P., Wijesekera, D., Kaushik, S.: Scalable, graph-based network vulnerability analysis. In: ACM Conference on Computer Communications Security (CCS), pp. 217–224 (2002)
3. Beckery, S., Seibert, J., et al.: Applying game theory to analyze attacks and defenses in virtual coordinate systems. In: IEEE/IFIP Conference on Dependable Systems and Networks (DSN), pp. 133–144 (2011)
4. Bohme, R., Moore, T.: The iterated weakest link: A model of adaptive security investment. In: Workshop on Economics of Information Security (WEIS) (2009)
5. Cheung, S., Lindqvist, U., Fong, M.: Modeling multistep cyber attacks for scenario recognition. In: DARPA Information Survivability Conference and Exposition (DISCEX), pp. 284–292 (2003)
6. Estiri, M., Khademzadeh, A.: A theoretical signaling game model for intrusion detection in wireless sensor networks. In: International Telecommunications Network Strategy and Planning Symposium (Networks), pp. 1–6 (2010)
7. Fultz, N., Grossklags, J.: Blue versus Red: Towards a Model of Distributed Security Attacks. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 167–183. Springer, Heidelberg (2009)
8. Gibbons, R.: Game Theory for Applied Economists. Princeton Press (1992)
9. Khouzani, M., Sarkar, S., Altman, E.: A dynamic game solution to malware attack. In: IEEE INFOCOM, pp. 2138–2146 (2011)
10. Li, F., Yang, Y., Wu, J.: Attack and flee: Game-theory-based analysis on interactions among nodes in MANETs. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics 40(3), 612–622 (2010)
11. Liu, P., Zang, W.: Incentive-based modeling and inference of attacker intent, objectives, and strategies. In: ACM Conference on Computer Communications Security (CCS), pp. 179–189 (2003)
12. Liu, Y., Comaniciu, C., Man, H.: A Bayesian game approach for intrusion detection in wireless ad hoc networks. In: International Workshop on Game Theory for Communications and Networks (GameNets), pp. 3–14 (2006)
13. Luo, Y., Szidarovszky, F., et al.: Game theory based network security. Journal of Information Security 1(1), 41–44 (2010)
14. Lye, K., Wing, J.: Game strategies in network security (extended abstract). In: IEEE Computer Security Foundations Workshop (CSFW), pp. 2–11 (2002)

15. Mell, P., Scarfone, K., Romanosky, S.: A complete guide to the common vulnerability scoring system (version 2.0). Forum of Incident Response and Security Teams (2007)
16. National Institute of Standards and Technology, USA. National vulnerability database (2010), http://nvd.nist.gov/home.cfm
17. Nguyen, K., Alpcan, T., Basar, T.: Security games with incomplete information. In: IEEE International Conference on Communications (ICC), pp. 714–719 (2009)
18. Ning, P., Cui, Y., Reeves, D.: Constructing attack scenarios through correlation of intrusion alerts. In: ACM Conference on Computer Communications Security (CCS), pp. 245–254 (2002)
19. Noel, S., Jajodia, S., et al: Efficient minimum-cost network hardening via exploit dependency graphs. In: Annual Computer Security Applications Conference (ACSAC), pp. 86–95 (2003)
20. Ou, X., Boyer, W., McQueen, M.: A scalable approach to attack graph generation. In: ACM Conference on Computer Communications Security (CCS), pp. 336–345 (2006)
21. Patcha, A., Park, J.-M.: A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. In: IEEE Workshop on Information Assurance and Security, pp. 1555–1559 (2004)
22. Sallhammar, K., Helvik, B., Knapskog, S.: On stochastic modeling for integrated security and dependability evaluation. Journal of Networks 1(5), 31–42 (2006)
23. Schiffman, M., Eschelbeck, G., et al.: CVSS: A common vulnerability scoring system. National Infrastructure Advisory Council (2004)
24. Shen, D., Chen, G., et al.: Adaptive Markov game theoretic data fusion approach for cyber network defense. In: IEEE Military Communications Conference (MILCOM), pp. 1–7 (2007)
25. Sheyner, O., Haines, J., et al.: Automated generation and analysis of attack graphs. In: IEEE Symposium on Security and Privacy (S&P), pp. 254–265 (2002)
26. Valeur, F., Vigna, G., et al.: A comprehensive approach to intrusion detection alert correlation. IEEE Transactions on Dependable and Secure Computing 1(3), 146–169 (2004)
27. Wang, W., Chatterjee, M., Kwiat, K.: Coexistence with malicious nodes: A game theoretic approach. In: ICST International Conference on Game Theory for Networks (GameNets), pp. 277–286 (2009)
28. Xie, P., Li, J., et al.: Using Bayesian networks for cyber security analysis. In: IEEE/IFIP Conference on Dependable Systems and Networks (DSN), pp. 211–220 (2010)
29. Zhang, Z., Ho, P.-H.: Janus: A dual-purpose analytical model for understanding, characterizing and countermining multi-stage collusive attacks in enterprise networks. Journal of Network and Computer Applications 32(3), 710–720 (2009)
30. Zhu, Q., Basar, T.: Dynamic policy-based IDS configuration. In: IEEE Conference on Decision and Control (CDC), pp. 8600–8605 (2009)
31. Zonouz, S., Khurana, H., et al.: RRE: A game-theoretic intrusion response and recovery engine. In: IEEE/IFIP Conference on Dependable Systems and Networks (DSN), pp. 439–448 (2009)

Simulation and Game-Theoretic Analysis of an Attacker-Defender Game

Alan Nochenson[1,⋆] and C.F. Larry Heimann[2,⋆⋆]

[1] College of Information Sciences and Technology, Penn State University
[2] Information Systems Department, Carnegie Mellon University
anochenson@psu.edu, profh@cmu.edu

Abstract. This paper uses agent-based simulation to determine appropriate strategies for attackers and defenders in a simple network security game, using a method which is generalizable to many other security games. In this game, both sides are modeled as strategic entities. The attacker is trying to maximize the amount of damage he causes, and the defender is trying to minimize her loss subject to cost constraints. Through simulation, we derive Nash equilibrium strategies for each side under a variety of cost conditions in order to better inform network administrators about attacker behaviors and possible mitigations.

Keywords: security, game theory, agent-based modeling and simulation.

1 Introduction

Defending a network of computers is becoming increasingly difficult. A network administrator needs to juggle server configurations, access controls, data storage and more. There are many magazines and articles about how to be a better Chief Information Security Officer (CISO). However, one of the key criteria of being a good CISO is simply knowing how much effort and time to spend securing your system and where to focus that effort and time. There has been a good deal of research in this area, covering everything from surveys of security professionals [4] to looking into misaligned incentives [2] and even behavioral and organizational factors [20]. This paper aims to augment the existing literature by using agent-based modeling and simulation applied to game-theoretic conceptions of information security (such as [11] and [15]) to identify optimal strategies for network administrators defending their networks in the presence of strategic attackers.

⋆ This work was partially created while Alan Nochenson was at Carnegie Mellon University.

⋆⋆ We would like to thank Jens Grossklags and Naomi Borrebach for their thoughtful editing and reviews. We would also like to thank the five anonymous referees for providing comments and suggestions. Any remaining errors or omissions are completely our own.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 138–151, 2012.

2 Background

2.1 Agent-Based Modeling and Simulation

Agent-based modeling is a fairly new approach to modeling complex systems. The key feature of an agent-based modeling scenario is the presence of a set of agents (with attributes and behaviors), relationships between and among the agents, and the environment that the agents "live" in. Macal et al. [18] describe autonomy as the defining characteristic of an ABMS. This autonomy is bounded by some normative model that describe the behavior of a given agent, which can be heterogeneous. This paper uses an agent-based model to simulate security scenarios to identify Nash Equilibria strategies for both attackers and defenders.

Stocco and Cybenko [22] used simulation to model various strategies in a game of High Card, which they used as a stand-in for a security investment game. This paper takes a similar strategy with a very different game and explicitly models a network security scenario to best generalize strategies to other related scenarios. The game of High Card simulates a single round of betting in a game of poker. Stocco and Cybenko pit bots with various utility functions against each other, and concluded that the behavior of a bot using a prospect theory-based function [16] performs better than using utility functions based on linear, sublinear and superlinear functions. They then developed a modeling bot that learned from the past behavior of its opponents, which was superior to the prospect-utility bot. They propose applications of this game to diplomacy and computer security.

2.2 Interdependent Security

Kunreuther and Heal [17] were some of the first researchers to look at interdependent problems in security. Interdependence refers to the idea that decisions are not made in a vacuum. The choices one agent in a network makes affects the well-being of others. This paper does not have multiple strategic defenders like in [17], but it does have multiple machines which are all interconnected. In our model, there is one defender responsible for providing security to these interdependent machines. Varian [23] looks at a variety of security games and acknowledges that security is a public good. Adding security to one machine in a network increases the security of the overall system. Later, we will see how a strategic attacker can mount an attack on one machine through another in the interdependent nature of security.

More recent literature ([8] [9] [12] [13]) has focused on modeling defenders in network security scenarios with ever-increasing complexities. These works mainly model attackers as non-strategic entities. This approach works well to emphasize the various concepts espoused in those papers. However, acknowledging that attackers are strategic entities can add another dimension to the understanding of security decision-making.

Fultz and Grossklags [7] and Hausken [10] have looked at games with strategic attackers. The former looks at the effects strategic attackers and network couplings (loosely or tightly coupled networks) have on security games such as in [8].

The latter [10] looks at network security with strategic attackers, merging operations research, reliability theory, and game theory. It does not make many specific recommendations which are understandable to the average defender. This paper is aimed at simulating attacker and defender behavior, instead of modeling a situation to obtain a closed-form result. We hope that, through this approach, we can give concrete recommendations to security administrators grounded in simulated results. While closed-form results are more rigorous, they are generally not as accessible to the non-academic security professional. We believe that our simulation-based approach is more straightforward.

3 Model

In this paper we propose an approach for agent-based decision-making in interdependent network security. The goal of this approach is to identify optimal strategies for a defender given the presence of a strategic attacker. The agents in our model are a defender and an attacker who are both strategic individuals aimed at maximizing their own utility[1]. The attacker in this game is restricted to attacking the machines in the defender's network with a single type of attack (e.g., exploiting a well-known vulnerability in an outdated version of a web server)[2]. This specific attack is not always going to be applicable to each machine in the network (e.g. if the attack is against a Windows machine and the target is running a Linux distribution). The defender possesses knowledge of which machines are vulnerable, and the attacker does not.

The defender's utility is percent loss incurred minus the cost of protecting each individual[3]. The attacker's utility is the amount of loss he causes out of the total he could possibly cause.

$$U_{defender} = -l - \sum_{i=1}^{n} c_d(e_i) \tag{1}$$

$$U_{attacker} = l \tag{2}$$

In the above equations, l is the percent loss incurred by the defender (or equivalently, inflicted by the attacker), n is the number of machines in the defender's

[1] While it has been shown is various experiments and theories (such as the allais paradox [1], Kahneman and Tversky's Prospect Theory [16], and Ariely's work on predictable irrationality [3]) that cardinal utility is not always accurate in decision-making, it is nonetheless still informative to postulate and analyze utilities. For a discussion and analysis of other factors that affect security decisions, see [20].

[2] In a real scenario, an attacker would have many attacks. For the purposes of this analysis, we restrict the scenario to look at a single class of attack. This encapsulates a scenario with a short time horizon. It is reasonable to assume the attacker will only attack with one type of attack when restricting the time period to a small one (such as 1 minute).

[3] While it is true that defenders are not always risk-neutral, we assume that is the case here for simplicity. For an analysis of non-risk neutral defenders, see [14].

network, $c_d : [0, 1) \rightarrow \Re^+$ is the defender's cost function, and e_i is the minimum effort needed to penetrate machine $i \in \{1, 2, \ldots, n\}$ in the defender's network.

The game is played in the following order:

1. Nature determines which machines are vulnerable to the attack.
2. The defender allocates resources towards defending each machine, based on her strategy.
3. The attacker attacks machines in the defender's network, allocating his efforts towards breaking into the machines in accordance with his strategy.
4. If the attacker succeeds in compromising at least one machine, he gets another chance to reallocate his resources and attack the remaining machines.
5. Utilities are calculated for both players.

The word "strategy" in the above context refers to the game-theoretical concept which maps an action to every situation in which the player is called upon to act. Specifically, the defender's strategies differ in how they prescribe resource allocation among the machines in her network. And, the attacker's strategies refer to how he should allocate his effort towards attacking each machine in the network (Steps 3 and 4 above)[4].

At this point, one may ask why the attacker gets another chance to attack in Step 4? Step 4 is the result of a successful attack – if Step 4 is executed, the attacker has already successfully compromised one or more machines in the network. Using the same attack, he reallocates his resources and attacks the remaining machines in the network. This captures the idea of an attacker using a well-known exploit on an outdated software version, or that of a zero-day attack[5].

3.1 Utilities

There are many possible strategies that the attacker and defender can possibly choose when allocating their resources. We model a short-term utility function for the attacker. In our opinion it is reasonable to assume that the attacker faces no significant cost associated with an attack. We make this assumption for

[4] While it is possible for the attacker to change his strategy after successfully attacking in Step 3, we are not analyzing this possibility here for sake of simplicity. The same analysis can be performed with new strategies that do not prescribe a single way of acting in Steps 3 and 4, but prescribe one way in Step 3 and another (possibly different) way of acting in Step 4.

[5] A zero-day attack is an attack that has not been widely seen prior to the attack being launched (e.g. the attack on Adobe Flash content embedded in Microsoft Excel spreadsheets in March 2011 [19]). These types of attacks are the types that attackers are able exploit for a period of time before the attack is detected and able to be effectively defended against. Step 4, above, accounts for the attacker using an attack that is not effectively being defended against, through the defender's lack of maintaining updated software or through the attack being previously unknown (i.e. the attack is a zero-day attack).

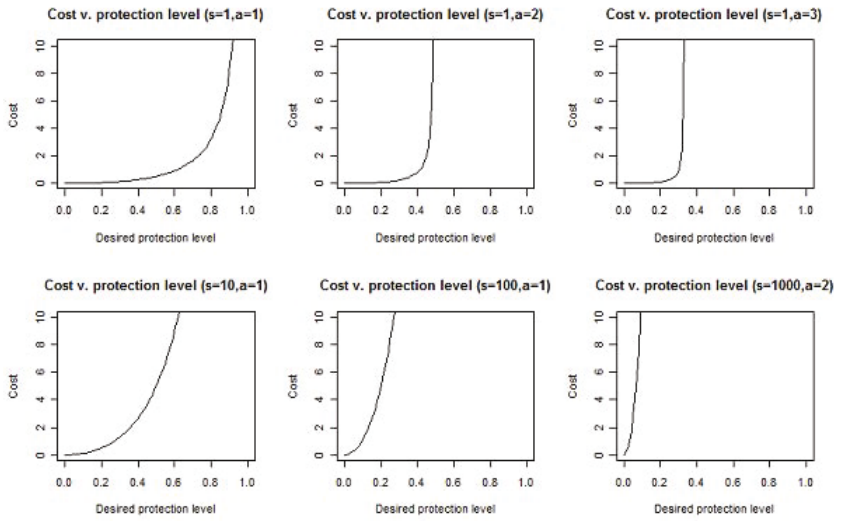


Fig. 1. Some example defender cost functions

the following reasons: the attacker is limited to a single attack and only a few discrete attacks (so technical costs are minimal), and the attacker is limited by a total amount of effort. In a game with a horizon that is farther off, the attacker would need to worry about being apprehended by law enforcement officials in addition to worrying about increasing technical costs to mounting attacks[6]. For this reason, we define the utility function of the attacker as directly proportional to the amount of damage he is able to cause.

The defender, unlike the attacker, is limited by cost. To see why the defender must have a cost function, imagine the opposite – if the defender had no cost function in the point-in-time model proposed here, she would protect the machines in her network until they are virtually impenetrable. There are no additional constraints put upon the defender that can make up for cost, as with the attacker. Since each side must have reasonable constraints in order to create a realistic scenario, the defender needs to have a cost function.

The defender's cost function must meet some key criteria for it to be valid. As previously stated, the cost function $c_d : [0, 1) \rightarrow \Re^+$ maps protection levels to their respective costs (e.g. $c_d(0.5)$ is the cost the defender must pay per machine to ensure a successful defense against attacks with half of an attacker's efforts or less allocated towards penetrating this machine)[7]. The cost function must be

[6] The main costs come at time used on the attack that could have been spent elsewhere (which is limited due to the short time frame), and costs to obtaining materials necessary to perform the attack (which are small due to this being a single type of attack).

[7] For this paper, we restrict our analysis to a single cost function for each machine in the network. While it is certainly possible to have distinct cost functions for each machine, we save that possibility for future research.

monotonically increasing and convex over its domain. It must approach infinity as it approaches some maximum possible protection level less than or equal to 1. The following as a general form for some of the functions of this type:[8]

$$f(x) = -\frac{sx^2}{ax - 1}$$

and

$$c_d(x) = \begin{cases} f(x), & \text{if } f(x) \geq 0 \\ \infty, & \text{if } f(x) < 0 \end{cases}$$

where $s > 0$ and $a \geq 1$. For the purpose of discussion, the parameter s will henceforth be referred to as the scaling parameter, and a as the asymptote parameter. We will soon see how variations in these parameters affect optimal strategies. First, we need to look at possible strategies for the attacker and defender.

3.2 Strategies

Strategies for the attacker and defender are surely numerous in real-world applications. For the purposes of the simulation, we have come up with some reasonable classes of strategies for each side.

Attacker Strategies. The strategies we identified for the attacker are called HIGH, PROPORTIONAL, LOWHIGH, UNIFORM, and RANDOM. Each of these prescribes a way the attacker should allocate his efforts among network machines, given that he knows the values of the machines (and does not know which ones are vulnerable to attack)[9]. In each strategy, the attacker will attack some $m \in \{1, 2, ..., n\}$ machines. If he successfully compromises $k > 0$ machines, he will use his same strategy in attacking the $n - k$ uncompromised machines in the next step.

The HIGH strategy prescribes the attacker to allocate 100% of his effort in each round towards attacking the machine with the highest value. The PROPORTIONAL strategy prescribes the attacker to allocate a portion of his efforts

[8] This general form is not exhaustive, and only provides functions of the correct type in the interval $[0, 1]$. Additionally, functions of this type are "curved" the way they are to emphasize the diminishing marginal returns of security investment. There will be some initial investment to set up software, then a steadily increasing (near linear) cost for additional security, until a very high level, where protection costs will increase at a quadratic rate (e.g. it may cost the same to get a machine from 10% to 20% as 20% to 30%, but it will be surely more expensive to get a machine from 80% to 90%).

[9] Value is not only monetary. We use value here loosely to refer to anything that makes the machine valuable to the defender or attacker. This includes but is not limited to the monetary value of a machine, time to repair the machine, cost of replacing the machine, and possible damage caused by a successful attack including loss of competitive advantage and decreased privacy.

towards each machine which is equal to its relative network value. For example, if a given machine represents 30% of the value in the network, the attacker under this strategy will attack this machine with 30% of his effort[10]. The LOWHIGH strategy tells the attacker to attack the lowest value machine in the first attack round in order to attack the highest value machine in the second attack round. This strategy works on the assumption that the defender will be allocating his resources in some manner related to machine value, which is reasonable. It also assumes that by penetrating the network in one spot (the lowered-valued machine), it will make mounting an attack against other points in the network simpler[11]. The UNIFORM strategy tells the attacker to spread his resources evenly among all of the network machines, and the RANDOM strategy tells the attacker to randomly allocate his resources.

Defender Strategies. Much like the attacker, the defender has many strategies at her disposal. They are named HIGH, PROPORTIONAL, LOW, UNIFORM, and NOTHING. The defender does *not* get a chance to reallocate her resources before the attacker (possibly) launches a second attack. Because the defender is unaware of how the attacker will attack, his attacks will come so close together in time that reallocation is not possible.

The HIGH strategy tells the defender to focus on securing her highest-valued machine. This strategy is often used in real situations where defenders believe that an attacker will only go after the highest valued machine (i.e. use the HIGH attacking strategy). The PROPORTIONAL strategy tells the defender to allocate resources proportionally based on machine value. The LOW strategy tells the defender to allocate exclusively to the lowest-valued machine (i.e. using "reverse psychology" and assuming the attacker will use the LOWHIGH strategy). The UNIFORM strategy tells the defender to allocate resources equally to each machine. The NOTHING strategy tells the defender to *not secure* her machines at all. With this strategy, she saves on protection costs and only incurs losses.

Since the defender has a cost function associated with her utility, she cannot simply allocate "all of her resources" to the highest machine, like the attacker is able to (see the discussion on the shape of the cost function above). For this reason, each defender strategy is not complete without a *critical value*, e'_i. This critical value is defined as the amount of effort that the defender is willing to ensure her machines are secured, in accordance with the strategy. For example, the strategy HIGH with $e'_i = 0.5$ means the defender will ensure her highest-valued machine is protected to a level of 0.5. In the PROPORTIONAL strategy,

[10] In the second attack stage, the "network value" that the attacker weighs machine value against is the sum total of the values of uncompromised machines in the network only.

[11] This notion of not directly attacking your primary target goes all the way back to the prominent example of the Maginot line in France. This line was a fortification between France and Germany that successfully dissuaded the German army from attacking that spot directly - they instead went around through the Ardennes forest and mounted a successful attack. The LOWHIGH strategy is roughly the same strategy the German army used to get around the Maginot line.

Algorithm 1. Simulation procedure

```
for s ∈ {0.1, 0.2, ..., 1} do
    for a ∈ {1, 2, ..., 10} do
        for strategy_a ∈ {PROPORTIONAL, HIGH} do
            for strategy_d ∈ {NOTHING, PROPORTIONAL, HIGH} do
                for e_i' ∈ {0.1, 0.2, ..., 0.9} do
                    U_a ← ∅
                    U_d ← ∅
                    loop 50,000 times
                        machineValues ← 3 random values in 1 to 100
                        Defender secures machines according to strategy_d
                        Attacker attacks machine(s) according to strategy_a
                        if Attacker successfully penetrates a machine then
                            Attacker attacks remaining machine(s)
                        end if
                        U_a ← U_a ∪ attacker utility for this run
                        U_d ← U_d ∪ defender utility for this run
                    end loop
                    Record avg(U_d), avg(U_a) in cell (strategy_d e_i', strategy_a)
                end for
            end for
        end for
        Find Nash Equilibrium for 19x2 normal-form game created by recording
        and record this as a point on a − s graph
    end for
end for
```

$e_i' = 0.5$ means that the defender will pay for a protection level of 0.5, which she will spread among the network machines in proportions according to their values (a machine with 20% of network value will get 0.5*0.2=10% protection level).

3.3 Simplification

For the purposes of simulation, we have decided to restrict our analysis such that the defender only has the options HIGH and PROPORTIONAL. The defender only has the choices of HIGH, PROPORTIONAL, and NOTHING[12].

3.4 Simulation Procedure

We have created a framework for agent-based simulations of various security games. In this paper, we are simulating the specific game that has been described

[12] From preliminary simulation results, for the attacker, LOWHIGH, UNIFORM, and RANDOM did not perform very well. For the defender, LOW and UNIFORM did not perform very well. To limit the scope of our simulation, we exclude these strategies from in-depth analysis and only include more-informed strategies for both the attacker and defender.

up until this point (for details on the simulation, see Algorithm 1). For each parameter configuration, this game is run 50,000 times[13]. The average utility for each player is calculated over the 50,000 trials, and recorded in a cell of a normal-form game. Each combination of defender utility function parameters (a and s) has its own normal-form game, which is 19x2[14]. For each of these normal-form games, we found Nash Equilibrium strategies for both players. Each game contributed one point to each of Figures 3 and 4.

This framework can be used to simulate a game with even more complexity than the one described here. To accomplish this, lines 9-16 in Algorithm 1 need to be replaced by the specifics of a different game. Strategies, cost functions, etc. can easily be exchanged or expanded upon. While it is sometimes possible to obtain closed-form results for these individual games (through expected value calculations or other means), in very large or very complex games, this might be cumbersome or impossible. This framework, and the analysis presented below, can serve as an example for others who wish to investigate security games of ever-increasing complexity.

4 Results

In this section, we will discuss some of the effects that we observed from running these simulations. Most of these results generalize to other scenarios, and the lessons learned here can be applied to them as well. First, we look at the example from above. Then, we look at how equilibria change when the value of s and a change.

4.1 Defender Strategies in Equilibrium

We used the simulation results to calculate equilibrium strategies for defenders given different levels of s and a in the cost function. The results of this can be seen in Figure 3. Note that in this figure we limit the scaling parameter to $s \leq 1$. We found that for $s \geq 1$, the defender should play the NOTHING strategy, regardless of the protection levels available (limited by a), because no level is cost-effective. At this point, buying even a 10% level of protection is so costly due to the scaling that it is not worth its cost.

In the same way, we limit the range of asymptote parameter a to 10 because at greater values the defender should always play the NOTHING strategy and lose approximately 70% of her network's value. There is an inverse relationship between a and the highest level of protection available. That is, the larger a

[13] The number of trials, 50,000, is not arbitrarily chosen. When using some number of trials less than this, there were conflicting results among the average utilities calculated across all runs. Simulation values are rounded to nearest integers in [-100,100], so standard deviations larger than 0.2 among runs are prone to change the results in a meaningful way. Increasing the number of trials above 50,000 did not meaningfully decrease the standard deviations among runs.

[14] See 3.4 for an example of one normal-form game generated in this method.

Defender's strategy \ Attacker's strategy	HIGH	PROPORTIONAL
NOTHING	-70 , 70	-70 , 70
HIGH 0.1	-63 , 63	-70 , 70
HIGH 0.2	-63 , 63	-70 , 69
HIGH 0.3	-65 , 62	-71 , 69
HIGH 0.4	-70 , 63	-70 , 63
HIGH 0.5	OUT OF RANGE	OUT OF RANGE
HIGH 0.6	OUT OF RANGE	OUT OF RANGE
HIGH 0.7	OUT OF RANGE	OUT OF RANGE
HIGH 0.8	OUT OF RANGE	OUT OF RANGE
HIGH 0.9	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.1	-51 , 50	-70 , 70
PROPORTIONAL 0.2	-51 , 51	-69 , 69
PROPORTIONAL 0.3	-52 , 51	-70 , 69
PROPORTIONAL 0.4	-53 , 51	-70 , 67
PROPORTIONAL 0.5	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.6	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.7	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.8	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.9	OUT OF RANGE	OUT OF RANGE

Fig. 2. Simulation results for $a = 2, s = 0.1$. Nash Equilibrium is in yellow and best responses are underlined. Combinations that are impossible to reach due to the defender's cost function are indicated by the words OUT OF RANGE.

becomes, the smaller the highest level available is. When $a \geq 10$, there isn't a level of protection available that is worth its cost, regardless of the scale (and the scaling parameter s).

Within these ranges for s and a we still find that the pure strategy of NOTHING covers over 40% of the area in the graph. At the same time, a pure strategy of PROPORTIONAL is far less viable, covering less than 10% of the area. Essentially, only when costs are at their lowest – both s and a are low in these cases – is a pure proportional strategy an equilibrium.

Of course, a mixed strategy of PROPORTIONAL and NOTHING is an equilibrium under a far greater range of s and a. More interesting is the fact that the mixed strategy of PROPORTIONAL, NOTHING and HIGH is viable when s stays low, but a grows. In this area, the unit cost of protection (which is directly proportional to s) is so small, that it is viable for the defender to protect all of her machines. However, since a is large, the greatest level of protection available is small (in $[0.10 - 0.33]$). Therefore, protecting only the highest-valued machine is a reasonable strategy. The rationale is this, if you can't buy much protection for each machine, the likelihood of each machine becoming compromised is high

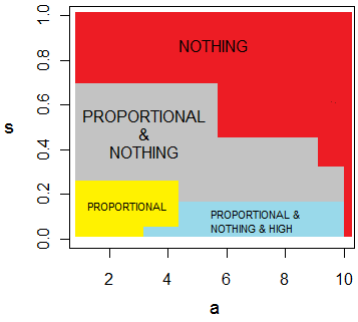


Fig. 3. Equilibrium strategies for defender at values of s and a

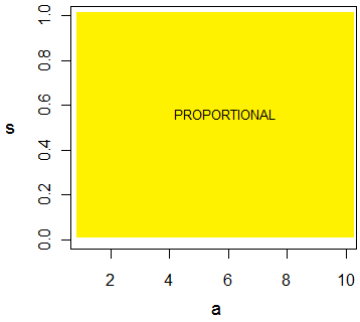


Fig. 4. Equilibrium strategies for attacker at values of s and a

with or without protection. If the defender does not buy protection, she loses an average of 70% of her value. If she does buy protection, she cannot buy much, for either the highest-valued or all machines. So, the protection that she does buy will most likely not protect her machine(s) (since protection will be at a low level). Therefore, the costs of buying protection with the minimal increased benefit is about equal to the expected loss without buying protection. So, protection may not help, and may even be an unnecessary cost. That is why, in a certain area, a mix of all three strategies is the defender's side of a Nash Equilibrium.

However, the only place where the HIGH strategy is ever played is in that area, and even there it is only played as part of a mixed strategy. From this area, if a becomes smaller, more protection is available and playing PROPORTIONAL alone is best. If s only becomes greater, the cost of protection either completely outweighs the benefit (and the NOTHING strategy is played), or the protection is worth the cost for every machine (and PROPORTIONAL is played). If both s grows and a becomes smaller, there is more protection available, but at a higher price - either worth investing in every machine or no machines. Only in that one area is the HIGH strategy even part of a mix, for the reasons discussed in the proceeding paragraph.

4.2 Attacker Strategies in Equilibrium

Knowing that the defender has the above reservations about her cost function, the attacker is enticed to play the PROPORTIONAL strategy. His logic is thus: since the defender will play NOTHING when $s \geq 1$ or $a \geq 10$, he should spread out his efforts, hoping the defender will not even protect. This reasoning is more decision-theoretic than game-theoretic – we haven't dealt with equilibra or dominated strategies at all yet.

The previous paragraph may convince you that the attacker has a dominant strategy to play PROPORTIONAL. This is not, however, the case. The attacker would do better to attack only the highest-valued machine in certain scenarios (e.g. when $a = 1, s = 0.1$ and the defender plays a proportional strategy with $e_i \geq 0.8$).

Regardless of the fact that PROPORTIONAL is not a dominant strategy, it is the only attacker strategy that is part of a Nash Equilibrium, for any values of s and a. As Fig. 3 shows, the defender has many strategies that can be part of a Nash Equilibrium depending on the values of a and s. For the attacker, every Nash Equilibrium has him playing the PROPORTIONAL strategy.

5 Discussion

Through agent-based modeling and simulation, we have reached a number of important conclusions about this attacker-defender security game. These conclusions are recommendations to defenders and probable behaviors for attackers. The key recommendation is to not invest all efforts towards protecting the highest-valued asset under your control. By doing that, a defender makes it easy for an attacker to target the other assets and still reap in a significant gain.

There is no Nash Equilibrium where an attacker directs all of his efforts towards compromising the highest valued asset. This means that the attacker will spread his efforts out, and in many scenarios so should the defender. Since the defender is limited by costs, she may not always be able to buy enough protection to protect each machine proportionally, though our simulation shows that in a Nash Equilibrium, the PROPORTIONAL strategy is always assigned some positive probability in the mixed strategy a defender plays, when she secures.

Additionally, network administrators can adapt their own systems to this model by better understanding the parameters in the defender's cost function. The scaling parameter s is representative of the relative costs of security compared to machine value. For example, if s is high, then each unit of protection is as costly as a large percentage of network value. The asymptote parameter a captures the technological uncertainty associated with securing a given machine. Namely, even if a defender thinks a machine is 100% secure, attackers in the future may develop a method to circumvent this protection. Therefore, a is forward-looking and attempts to account for the unknowns that can occur in the future. For example, a defender can use parameterized statements to avoid SQL injection attacks, but cannot protect against vulnerabilities introduced in the underlying database framework after the security system is implemented. When uncertainty is high, the defender's cost function is "squished" such that a is high and the available protection amount is low (e.g. when securing a system against attacks on quantum cryptography, it is difficult to encapsulate all variables because there are simply too many unknowns). When uncertainty is low, the function is "stretched out" such that a is low and a defender can account for nearly all of the risk to a system.

These results are applicable to other scenarios outside of network security. When defending airlines from terrorist threats, one can look to these results and realize it is not prudent to allocate all of your efforts towards protecting the area with the most value. When defending a home against invaders, it is important to secure not only the front door, but the windows as well. If attackers are very likely to spread their efforts among available targets, it is important to identify and protect against all threats in proportion with their impact.

There are numerous avenues where this work can be expanded. We would like to apply the concept of loss profiles from our previous work [12] to this game. Loss profiles capture the idea of variable loss when infected. Applying the concept of loss profiles, a compromised machine would not necessarily lose all of its value, but would lose some percentage drawn from a characteristic distribution. Another avenue for future research is to apply the concept of risk aversion (such as in [14] and [21]) to the defender's decisions.

Another future refinement to this model would be in regard to attacker resource allocations. In the real world a defender is fending off attacks from a large number of attackers, each with a limited resource pool. For modeling purposes we have only looked at a single point in time, where one attacker is performing an attack and is limited to a single attack, both in number and in type, making allocation decisions without cost considerations. Another means of modeling attackers would be to use the framework of a Colonel Blotto game. In a Blotto game, a player captures an asset by allocating more of his/her limited resources than an opponent in an attack; thus in Blotto games an under-resourced attacker can still capture assets by strategically mismatching the actions of his/her opponent and creating a favorable resource imbalance. The Colonel Blotto game has been successfully applied to web security (briefly in [6]) and phishing attacks [5] as well as to other aspects of security in general. Using the Blotto game framework in future research could allow us to examine cost allocations of multiple small attackers in more detail.

References

1. Allais, M.: Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'école Américaine. Econometrica 21, 503–546 (1953)
2. Anderson, R., Moore, T.: The economics of information security. Science 314(5799), 610–613 (2006)
3. Ariely, D.: Predictably Irrational: The Hidden Forces That Shape Our Decisions. HarperCollins (February 2008)
4. Baldwin, A., Beres, Y., Duggan, G.B., Mont, M.C., Johnson, H., Middup, C., Shiu, S.: Economic methods and decision making by security professionals. In: The Tenth Workshop on Economics and Information Security, WEIS 2011 (2011)
5. Chia, P.H., Chuang, J.: Colonel Blotto in the Phishing War. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 201–218. Springer, Heidelberg (2011)
6. Chia, P.H.: Colonel Blotto in web security. In: The Eleventh Workshop on Economics and Information Security, WEIS Rump Session (2012)
7. Fultz, N., Grossklags, J.: Blue versus Red: Towards a Model of Distributed Security Attacks. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 167–183. Springer, Heidelberg (2009)
8. Grossklags, J., Christin, N., Chuang, J.: Security and insurance management in networks with heterogeneous agents. In: Proceedings of the 9th ACM Conference on Electronic Commerce, EC 2008, pp. 160–169. ACM, New York (2008)
9. Grossklags, J., Johnson, B.: Uncertainty in the weakest-link security game. In: Proceedings of the First ICST International Conference on Game Theory for Networks, GameNets 2009, pp. 673–682. IEEE Press, Piscataway (2009)

10. Hausken, K.: Protecting complex infrastructures against multiple strategic attackers. Intern. J. Syst. Sci. 42(1), 11–29 (2011)
11. Heal, G., Kunreuther, H.: You only die once: Managing discrete interdependent risks. In: Columbia Business School and Wharton Risk Management and Decision Processes (2002)
12. Heimann, C.F.L., Nochenson, A.: The effects of loss profiles in interdependent network security. In: The World Congress on Internet Security, WorldCIS (2012)
13. Heimann, C.F.L., Nochenson, A.: Identifying Tipping Points in a Decision-Theoretic Model of Network Security. ArXiv e-prints (March 2012)
14. Johnson, B., Böhme, R., Grossklags, J.: Security Games with Market Insurance. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 117–130. Springer, Heidelberg (2011)
15. Johnson, B., Grossklags, J., Christin, N., Chuang, J.: Uncertainty in Interdependent Security Games. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 234–244. Springer, Heidelberg (2010)
16. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica 47(2), 263–291 (1979)
17. Kunreuther, H., Heal, G.: Interdependent security. Journal of Risk and Uncertainty 26, 231–249 (2003)
18. Macal, C.M., North, M.J.: Tutorial on agent-based modeling and simulation. In: 2005 Winter Simulation Conference (2005)
19. Naraine, R.: Adobe warns of flash player zero-day attack (2011), `http://www.zdnet.com/blog/security/adobe-warns-of-flash-player-zero-day-attack/8438`
20. Nochenson, A., Heimann, C.F.L.: Optimal security investments in networks of varying size and topology. In: International Workshop on Socio-Technical Aspects in Security and Trust (2012)
21. Pratt, J.W.: Risk Aversion in the Small and in the Large. Econometrica 32 (1964)
22. Stocco, G.F., Cybenko, G.: Exploiting Adversary's Risk Profiles in Imperfect Information Security Games. In: Baras, J.S., Katz, J., Altman, E. (eds.) GameSec 2011. LNCS, vol. 7037, pp. 22–33. Springer, Heidelberg (2011)
23. Varian, H.R.: System reliability and free riding. In: Economics of Information Security, Kluwer 2004, pp. 1–15. Kluwer Academic Publishers (2004)

Linear Loss Function for the Network Blocking Game: An Efficient Model for Measuring Network Robustness and Link Criticality

Aron Laszka[1], Dávid Szeszlér[2], and Levente Buttyán[1]

[1] Laboratory of Cryptography and System Security,
Department of Telecommunications,
Budapest University of Technology and Economics
{laszka,buttyan}@crysys.hu
[2] Department of Computer Science,
Budapest University of Technology and Economics
szeszler@cs.bme.hu

Abstract. In order to design robust networks, first, one has to be able to measure robustness of network topologies. In [1], a game-theoretic model, the network blocking game, was proposed for this purpose, where a network operator and an attacker interact in a zero-sum game played on a network topology, and the value of the equilibrium payoff in this game is interpreted as a measure of robustness of that topology. The payoff for a given pair of pure strategies is based on a loss-in-value function. Besides measuring the robustness of network topologies, the model can be also used to identify critical edges that are likely to be attacked. Unfortunately, previously proposed loss-in-value functions are either too simplistic or lead to a game whose equilibrium is not known to be computable in polynomial time. In this paper, we propose a new, linear loss-in-value function, which is meaningful and leads to a game whose equilibrium is efficiently computable. Furthermore, we show that the resulting game-theoretic robustness metric is related to the Cheeger constant of the topology graph, which is a well-known metric in graph theory.

Keywords: game theory, adversarial games, network robustness, computational complexity, blocking games, Cheeger constant.

1 Introduction

In order to be able to design networks that resist malicious attacks and accidental failures, one must be able, first of all, to measure the robustness of network topologies. A number of graph-theoretic robustness metrics, such as node and edge connectivity, graph strength [2], toughness [3], and persistence [4], can be used for this purpose. Recently, however, another approach for measuring the robustness of network topologies has been proposed by Gueye *et al.* in a series of papers [5,6,1]. In their approach, the robustness of a network topology is

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 152–170, 2012.

characterized by the equilibrium payoff in a two-player zero-sum game played by a network operator and an attacker.

In this model, the network operator chooses a spanning tree of the topology graph to be used for routing messages in the network, and simultaneously, the attacker chooses an edge of the topology graph to be removed. If the edge chosen by the attacker happens to be in the spanning tree chosen by the operator, then the attacker's payoff is positive and the operator's payoff is negative, otherwise they both receive 0 as payoff. In the former case, the actual value of the payoff depends on the loss-in-value function that is used for characterizing the effect of an edge being removed from the spanning tree. For instance, in [5], a simple indicator function is used: if the edge removed by the attacker is in the spanning tree, then the attacker's payoff is 1, otherwise it is 0. Another example is given in [7], where the payoff for the attacker is equal to the number of nodes that the attacker separates from a designated node in the spanning tree (e.g., a gateway in an access network) by removing the chosen edge.

The optimal strategies of such an attacker-defender game can be used to *identify critical edges* that are likely to be attacked, and the equilibrium payoff can be interpreted as a *measure of robustness* of the network topology at hand. In some cases, the game-theoretic robustness of a network may even be directly related to a graph-theoretic robustness metric. For instance, in [7], it is shown that the equilibrium payoff of the game where the loss-in-value function is defined as the number of nodes that the attacker separates from a designated node in the network is equal to the reciprocal of the persistence of the network as defined in [4]. Hence, the game-theoretic model can provide additional insights into the understanding of the graph-theoretic robustness metrics.

Unfortunately, the loss-in-value functions of [5] and [7] can not be used generally: The former is too simplistic as it does not take into account the magnitude of the damage caused by the attack. The latter is concerned with only those types of networks, where the nodes have to communicate only with a designated node, such as access and sensor networks. In [1], a number of loss-in-value functions, derived from previously proposed network value functions, are introduced and studied. However, to the best of our knowledge, finding efficient algorithms to compute the equilibrium payoff or the optimal strategies in case of these loss-in-value functions is still an open question.

Our contributions in this paper are the following: We propose a new, linear loss-in-value function, for which we also provide strongly polynomial-time algorithms to compute optimal adversarial and operator strategies and, thus, the payoff in the Nash equilibria of the game. This means that we can compute the game-theoretic robustness efficiently in case of this linear loss-in-value function. Moreover, our proposed linear loss-in-value function is meaningful in the sense that it is lower- and upper-bounded by loss-in-value functions previously proposed in [1]. In addition, we prove that the payoff in the Nash equilibria is closely related to the Cheeger constant of a graph (also called minimum edge expansion or isoperimetric number), a well-known metric in graph theory. Thus,

we can relate the game-theoretic robustness metric resulting from the proposed loss-in-value function to a graph-theoretic robustness notion.

The organization of this paper is the following: In Section 2, we briefly summarize previous related results. In Section 3, we describe the game model and introduce our linear loss-in-value function. In Section 4 and 5, we propose optimal operator and adversarial strategies and show how to compute them very efficiently. In Section 6, we combine the results of the previous sections to study the Nash equilibria of the game. In Section 7, we discuss properties of the optimal adversarial strategies. In Section 8, we show how the equilibrium payoff is related to the Cheeger constant. In Section 9, we generalize the game model to allow nodes with non-uniform weights. Finally, in Section 10, we conclude the paper.

2 Related Work

In [5], the strategic interactions between a network operator, whose goal is to keep a network connected, and an attacker, whose goal is to disconnect the network, were modeled as a two-player, one-shot, zero-sum game: The operator chooses a spanning tree T of the network G as the communication infrastructure while the adversary chooses a link e as the target of her attack. The payoff of the adversary (or the loss of the operator) is $1_{e \in T}$, i.e., it is 1 if the targeted link is part of the chosen spanning tree, and 0 otherwise[1] . It was shown that the payoff in every Nash equilibrium of the game is equal to the reciprocal of the *(undirected) strength* of the network $\sigma(G)$, which can be computed efficiently. Furthermore, an efficient algorithm was provided to compute an optimal adversarial strategy.

In [6], the model was generalized to include link attack costs, which can vary based on the targeted links, resulting in a non-zero-sum game. Efficient algorithms were provided to compute the payoff in the Nash equilibria and to obtain an optimal adversarial strategy.

In [8], the model was generalized to incorporate link faults, which are random malfunctions independent of the adversary. In this model of interdependent reliability and security, the operator knows the distribution of link faults and the relative frequencies of faults and attacks, while the role of the adversary remains the same as in the basic model. Efficient algorithms were provided for two particular link fault distributions, the uniform distribution and a special distribution with a critical link being more vulnerable.

In [1], the indicator function $1_{e \in T}$ was replaced with a general *loss-in-value* function $\lambda(T, e)$, which quantifies the gain of the adversary and the loss of the operator when link e is targeted and communication is carried over tree T. Previously proposed models for the value of a network were used to derive various loss-in-value functions. Some of the proposed loss-in-value functions (Metcalfe, Reed, BOT), as well as the indicator loss-in-value function $1_{e \in T}$ (termed GWA) are plotted in Figure 1.

[1] The details of the model are described in Section 3. Here, we only introduce the basic concepts that are necessary to the discussion of related work.

In [7], the interactions in a many-to-one network, such as an access network or sensor network, were studied using a special loss-in-value function. The proposed function measures the number (or total value) of the nodes that are separated from a designated node when an attack occurs. It was shown that the payoff in every Nash equilibrium of the game is equal to the reciprocal of the *persistence* (or *directed strength*) of the network $\pi(G)$, which can be computed efficiently. Furthermore, efficient algorithms were provided to compute optimal operator and adversarial strategies.

3 Game Model

The network topology is represented by a connected undirected simple graph $G = (V, E)$. The goal of the network operator is to keep the nodes of the network connected to each other, while the goal of the adversary is to separate the nodes from each other.

The interaction between the network operator and the adversary is modeled as a two-player, one-shot, zero-sum game. The network operator chooses a spanning tree to be used for communications. The mixed strategy of the network operator is a distribution on the set of spanning trees $\mathcal{T}(G)$, i.e., $\mathcal{A} := \{\alpha \in \mathbb{R}_{\geq 0}^{|\mathcal{T}(G)|} | \sum_{T \in \mathcal{T}(G)} \alpha_T = 1\}$. The adversary chooses an edge to be attacked. The mixed strategy of the adversary is a distribution on the set of edges $E(G)$, i.e., $\mathcal{B} := \{\beta \in \mathbb{R}_{\geq 0}^{|E(G)|} | \sum_{e \in E(G)} \beta_e = 1\}$.

The payoff of the adversary (or the loss of the operator) is given by the loss-in-value function $\lambda(T, e)$. Thus, the expected payoff of the adversary is

$$\sum_{e \in E(G)} \sum_{T \in \mathcal{T}(G)} \alpha_T \beta_e \lambda(T, e) \ , \tag{1}$$

which the adversary tries to maximize and the operator tries to minimize.

3.1 Our Proposed Loss Function

In this paper, we propose a "linear" loss-in-value function, denoted by $\lambda(T, e)$, where T and e are the spanning tree and edge chosen by the operator and the adversary, respectively. If $e \in T$, then let $\lambda(T, e)$ be the number of nodes in the smaller component of $G[T \setminus \{e\}]$, where $G[F]$ denotes the graph $G' = (V(G), F)$, i.e., $\lambda(T, e)$ is the number of nodes that are separated from the larger connected component after the attack. If $e \notin T$, then let $\lambda(T, e) = 0$, i.e., there is no loss if the spanning tree remains intact. More formally:

Definition 1 (Linear loss-in-value function)

$$\lambda(T, e) := \begin{cases} \min_{C \in \text{ components of } G[T \setminus \{e\}]} |C|, & \text{if } e \in T \\ 0, & \text{if } e \notin T \ . \end{cases} \tag{2}$$

Figure 1 compares our linear loss-in-value function to some of the functions proposed in [1]. The comparison is performed in a network consisting of 60 nodes.

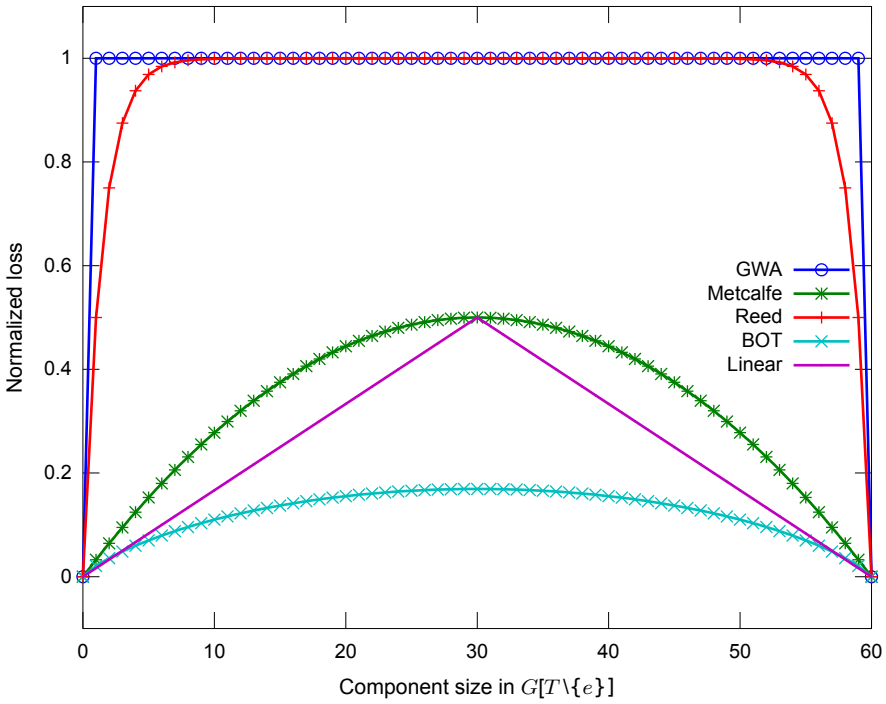


Fig. 1. Comparison of different loss-in-value functions

The horizontal axis shows the sizes of the components of the graph resulting from the attack. Extreme values 0 and 60 correspond to intact networks, i.e., when $e \notin T$. Values $0 < n < 60$ correspond to a damaged network consisting of two components of size n and $60 - n$. The vertical axis measures the payoff of the adversary or, equivalently, the loss of the operator. The previously proposed loss functions are "normalized" in the manner described in [1]: each loss function is divided by the value of the intact network, which is determined by the corresponding network value function. Our loss function is not based on a network value function, but, since each node being separated causes a loss of value 1, we can establish that the value of the intact network is the number of nodes $|V(G)|$. Therefore, our linear loss function is "normalized" by dividing it with the constant coefficient $|V(G)|$. The normalization allows us to make an unbiased comparison between the different loss-in-value functions.

The figure shows that our linear loss function is bounded by the previously proposed loss functions Metcalfe and BOT. The Metcalfe function measures a special quadratic loss and is an upper bound of our function. The BOT function measures a special logarithmic loss and is a lower bound of our function. For the exact definitions and a discussion of these functions we refer the reader to [1]. We can conclude that our linear loss function is at least as realistic as the previously proposed functions.

Please recall that, to the best of our knowledge, there are no polynomial-time algorithms known to compute the equilibria or the optimal strategies of the games based on the above loss functions, except for the GWA function. For further comparison between different loss functions, see Section 6.1.

4 Operator Strategy

In this section, we propose an operator strategy. Later, in Section 5, we show that this strategy is optimal by proving that it attains the lowest possible expected loss for the operator if the adversary is rational.

Definition 2 (Expected loss). *The* expected loss *(or* importance*) of an edge $e \in E(G)$ in a given operator strategy α is the expected payoff of the pure adversarial strategy targeting exclusively e, i.e., $\sum_{T \in \mathcal{T}} \alpha_T \cdot \lambda(T, e)$.*

To obtain an optimal operator strategy, consider the following linear program:
Variables:

$$\forall r \in V(G): \ \alpha_r \in \mathbb{R}_{\geq 0}$$
$$\forall r \in V(G)\ \forall \{u, v\} \in E(G): \ f_r(u, v), f_r(v, u) \in \mathbb{R}_{\geq 0}$$

Objective:

$$\text{maximize} \sum_{r \in V(G)} \alpha_r \tag{3}$$

Constraints:

$$\forall \{u, v\} \in E(G): \ \sum_{r \in V(G)} f_r(u, v) + f_r(v, u) \leq 1 \tag{4}$$

$$\forall r \in V(G)\ \forall v \in V(G) \setminus \{r\}: \ \sum_{\{u,v\} \in E(G)} f_r(v, u) - f_r(u, v) \geq \alpha_r \ . \tag{5}$$

Let $h'(G)$ denote the optimal value of the above linear program.

The above linear program can be viewed as a special multi-commodity flow problem: There is a commodity for every $r \in V(G)$. For every commodity, r is a sink and every other node is a source producing α_r. For the commodity consumed by r, the amount of flow from u to v is given by $f_r(u, v)$. Finally, every edge has a capacity of 1.

Theorem 1. *There is an operator strategy that achieves at most $\frac{1}{h'(G)}$ loss for the operator (regardless of the strategy of the adversary).*

Proof. Our goal is to find a distribution $\overline{\boldsymbol{\alpha}}$ such that

$$\forall e \in E: \ \sum_{T \in \mathcal{T}} \overline{\alpha}_T \lambda(T, e) \leq \frac{1}{h'(G)} \ , \tag{6}$$

i.e., the expected loss of every edge is at most $\frac{1}{h'(G)}$. Equivalently, we have to find weights $\boldsymbol{\alpha} \geq \mathbf{0}$ such that

$$\forall e \in E : \sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq 1 \ , \tag{7}$$

i.e., the expected loss of every edge is at most one, and

$$\sum_{T \in \mathcal{T}} \alpha_T = h'(G) \ . \tag{8}$$

Our proof is constructive and it is based on the following algorithm:

1. Solve the above LP.
2. For each $r \in V(G)$,
   - find a set of weighted spanning trees $T_r^1, \ldots, T_r^{m_r}$ with a total weight of α_r that satisfies the constraint that the expected loss of each edge $\{u, v\}$ is less than or equal to $f_r(u, v) + f_r(v, u)$ using the flow decomposition algorithm proposed in [7]. The details of this algorithm can be found in Appendix A.

We claim that the resulting set of spanning trees $T_r^1, \ldots, T_r^{m_r}$ and corresponding coefficients $\alpha_r^1, \ldots, \alpha_r^{m_r}$ as a strategy satisfy that the expected loss of every edge is at most $\frac{1}{h'(G)}$.

Firstly, we have

$$\forall r \in V : \sum_{i=1}^{m_r} \alpha_r^i = \alpha_r \tag{9}$$

from the flow decomposition algorithm and

$$\sum_{r \in V} \alpha_r = h'(G) \tag{10}$$

by definition. Therefore, the total weight of the spanning trees is equal to $h'(G)$.

Let $\lambda_r(T, e)$ denote the number of nodes that are separated from $r \in V(G)$ in $G[T \setminus e]$, i.e., the number of nodes that are not in the same component as r in $G[T \setminus e]$. Then, we have

$$\forall \{u, v\} \in E, r \in V : \sum_{i=1}^{m_r} \alpha_r^i \lambda_r(T_r^i, \{u, v\}) \leq f_r(u, v) + f_r(v, u) \tag{11}$$

from the flow decomposition algorithm.

By definition, $\lambda(T, e) \leq \lambda_r(T, e)$ as $\lambda(T, e)$ is the number of nodes in the not larger component (or 0, if there is only one component). Since $\lambda(T_r^i, e) \leq \lambda_r(T_r^i, e)$,

$$\forall \{u, v\} \in E : \sum_{r \in V} \sum_{i=1}^{m_r} \alpha_r^i \lambda(T_r^i, \{u, v\}) \leq \sum_{r \in V} f_r(u, v) + f_r(v, u) \ . \tag{12}$$

We also have

$$\forall \{u, v\} \in E : \sum_{r \in V} f_r(u, v) + f_r(v, u) \leq 1 \tag{13}$$

from the definition of the LP. Therefore, the expected loss of every edge (LHS of Equation 12)is at most 1. □

The following theorem shows how efficient the above algorithm is:

Theorem 2. *The operator strategy described in Theorem 1 can be computed in strongly polynomial time and its support, i.e., the set of spanning trees that have nonzero probability, consists of at most* $|V(G)| \cdot |E(G)|$ *spanning trees.*

Proof. We show that both steps of the algorithm presented in the proof of Theorem 1 can be performed in strongly polynomial time. In [9], a polynomial linear programming algorithm was presented whose number of arithmetic steps depends only on the size of the numbers in the constraint matrix. Since, in our case, the constraint matrix consists of only values of -1, 0 and 1, the linear programming problem can be solved in strongly polynomial time. It can be easily verified that the flow decomposition algorithm also runs in strongly polynomial time: Clearly, the number of arithmetic steps in every iteration (see Appendix A) of the algorithm is polynomial. In [7], it was shown that there are at most $|E(G)|$ iterations. Since the algorithm has to be run once for every node, the total number of iterations is at most $|V(G)| \cdot |E(G)|$.

In [7], it was also shown shown that the support of the distribution produced by the flow decomposition algorithm consists of at most $|E(G)|$ spanning trees. Since at most $|V(G)|$ flows have to be decomposed, the support of the resulting strategy has at most $|V(G)| \cdot |E(G)|$ spanning trees. □

The first part of the theorem states that our algorithm is very efficient as it is not only polynomial-time, but strongly polynomial-time, while the second part of the theorem states that resulting strategy is surprisingly simple.

5 Adversarial Strategy

In this section, we propose an optimal adversarial strategy, which attains $\frac{1}{h'(G)}$ expected payoff, regardless of the strategy of the operator. We have already shown in the previous section that this is the best attainable payoff for the adversary if the operator is rational.

Lemma 1. *For every spanning tree* T*, there exists a spanning reverse arborescence*[2] *such that*

- *the arborescence consists of the edges of* T *and*
- *each arc* $e \in T$ *is directed such that its target is in the larger component of* $G[T \setminus \{e\}]$[3] .

Proof. Direct each edge e of T such that its target is not in the smaller component of $G[T \setminus \{e\}]$. We have to prove that the result is indeed an arborescence, i.e., there is no pair of arcs $(u, v), (u, w) : u, v, w \in V(G), v \neq w$. Assume that this is not true: Let W denote the node set of the not smaller component of $G[T \setminus \{(u, w)\}]$.

[2] A directed, rooted spanning tree in which all edges point to the root.
[3] If the two components are of equal size, then the direction is arbitrary.

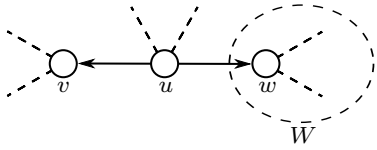


Fig. 2. Illustration for the proof of Lemma 1

Since W consists of the nodes of the larger component, $|W \cup \{u\}| > \frac{|V(G)|}{2}$. But this leads to a contradiction as $W \cup \{u\}$ is a subset of the smaller component of $G[T \setminus \{(u,v)\}]$. See Figure 2 for an illustration. □

Now, consider the dual of the linear program introduced in the previous section:

Variables:

$$\forall e \in E(G): \ \beta_e \in \mathbb{R}_{\geq 0}$$
$$\forall r, v \in V(G), r \neq v: \ \pi_r(v) \in \mathbb{R}_{\geq 0}$$

Objective:

$$\text{minimize} \sum_{e \in E(G)} \beta_e \tag{14}$$

Constraints:

$$\forall r \in V(G): \sum_{v \in V(G) \setminus \{r\}} \pi_r(v) \geq 1 \tag{15}$$
$$\forall r \in V(G)\ \forall \{u,v\} \in E(G): \ \beta_{\{u,v\}} - \pi_r(u) + \pi_r(v) \geq 0 \tag{16}$$
$$\forall r \in V(G)\ \forall \{u,v\} \in E(G): \ \beta_{\{u,v\}} + \pi_r(u) - \pi_r(v) \geq 0\ , \tag{17}$$

where $\pi_r(r) \equiv 0$ to simplify the equations.

Variable π can be viewed as a set of $|V(G)|$ potential functions, one for each node $r \in V(G)$. For every potential function π_r, the potential difference between two adjacent nodes connected by edge e is bounded by the edge weight β_e.

The last two constraints can be written as

$$|\pi_r(u) - \pi_r(v)| \leq \beta_{\{u,v\}}\ . \tag{18}$$

Clearly, the optimal value of the dual program is equal to the optimal value $h'(G)$ of the primal program.

Theorem 3. *There is an adversarial strategy that achieves at least $\frac{1}{h'(G)}$ payoff for the adversary (regardless of the strategy of the operator).*

Proof. Our goal is to find a distribution $\overline{\boldsymbol{\beta}}$ such that

$$\forall T \in \mathcal{T}: \sum_{e \in E(T)} \overline{\beta}_e \lambda(T,e) \geq \frac{1}{h'(G)}\ . \tag{19}$$

Equivalently, we have to find weights $\boldsymbol{\beta} \geq \mathbf{0}$ such that

$$\forall T \in \mathcal{T} : \sum_{e \in E(T)} \beta_e \lambda(T, e) \geq 1 \tag{20}$$

and

$$\sum_{e \in E} \beta_e = h'(G) \ . \tag{21}$$

We claim that the weights β_e of the optimal solution of the above dual linear program are such. To prove this, let T be an arbitrary spanning tree and r be the root of a reverse arborescence defined in Lemma 1. Then,

$$\sum_{e \in E(T)} \beta_e \lambda(T, e) = \sum_{e \in E(T)} \beta_e \lambda_r(T, e) \tag{22}$$

$$= \sum_{v \in V(G) \setminus \{r\}} \left(\sum_{e \in \{\text{edges of the } (v,r) \text{ path in } T\}} \beta_e \right) \tag{23}$$

$$\geq \sum_{v \in V(G) \setminus \{r\}} \pi_r(v) \tag{24}$$

$$\geq 1 \ , \tag{25}$$

where (22) holds by definition (see proof of Theorem 1). In (23), we used the observation that $\lambda_r(T, e)$ is the number of nodes v from which the path to r in T contains e. (24) follows from Constraint 18 by applying it to every edge along the path. Finally, (25) follows from Constraint 15. □

Furthermore, the dual linear programming problem can be also solved in strongly polynomial time as the constraint matrix consists of only values of -1, 0 and 1. Thus, an optimal adversarial strategy can be obtained in strongly polynomial time.

6 Nash Equilibria and Sets of Critical Edges

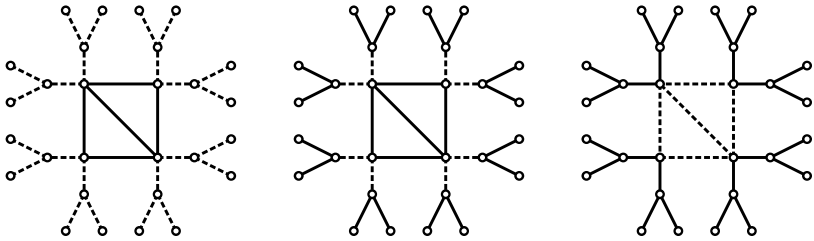
From Theorem 1 and Theorem 3, the following corollary directly follows:

Corollary 1. *In every Nash equilibrium, the expected payoff for the adversary (or the expected loss of the operator) is* $\frac{1}{h'(G)}$*. The optimal operator and optimal adversarial strategies form Nash equilibira of the game.*

The higher the value of $\frac{1}{h'(G)}$ is, the more vulnerable the network is. Consequently, $h'(G)$ and $\frac{1}{h'(G)}$ can be used as measures of network robustness and network vulnerability. From Theorem 2, it readily follows that these metrics can be computed efficiently.

6.1 Sets of Critical Edges

Besides measuring the robustness of network topologies, the network blocking game can be also used to identify critical edges that are likely to be attacked. Formally, an edge is called *critical* if it is in the support of an optimal adversarial strategy [5]. In this subsection, we make a comparison between the sets of critical edges resulting from different loss-in-value functions.



(a) GWA **(b)** Reed **(c)** BOT, Metcalfe and Linear

Fig. 3. The set of critical edges for different loss-in-value functions. Critical edges are represented by dashed lines.

In [1], the proposed loss-in-value functions were compared using the example network shown in Figure 3. For the sake of consistency, we use the same network for our comparison. The sets of critical edges identified using the previously proposed loss functions are taken from [1].

Figure 3a shows the set of critical edges identified using the simple indicator function $1_{e \in T}$ (termed GWA). The critical set consists exclusively of bridges, edges whose removal disconnects the network. This can be explained by the fact that the indicator function does not take into account the magnitude of the damage; therefore, the adversary maximizes solely the probability of hitting the spanning tree, regardless of the expected number of nodes cut off.

Figure 3b shows the set of critical edges identified using the Reed loss function. The set is similar to the one identified using the indicator function, but contains only those bridges that cut off more than one node. This result is consistent with Figure 1, which also shows that the Reed and the indicator functions are similar, but that the Reed function also takes the magnitude of the damage into account to some extent.

Finally, Figure 3c shows the set of critical edges identified using the BOT, Metcalfe and our linear loss function. Again, the fact that these three functions result in the same set is consistent with our earlier comparison based on Figure 1, which also showed that these functions are similar.

7 Properties of the Optimal Adversarial Strategies

In this section, we discuss properties of the optimal solutions of the dual problem. These properties allow us to formulate the dual problem as a graph partitioning

problem at the end of this section. Please note that, since optimal strategies are normalized optimal solutions, Lemma 3 and Lemma 5 can be applied to optimal adversarial strategies as well.

For any $\beta \in \mathbb{R}_{\geq 0}^{|E(G)|}$, let $E(\beta) = \{e \in E(G) : \beta_e > 0\}$. Let the partitioning defined by β be such that two nodes belong to the same partition iff there is a path between them consisting exclusively of edges in $E(G) \setminus E(\beta)$.

Lemma 2. *Let $V_1, \ldots, V_k$ be the partitioning defined by an optimal solution β^* of the dual problem. If $u, v \in V_i$, then $\pi_r(u) = \pi_r(v)$ for every r.*

Proof. If u and v are connected by an edge $e \in E(G) \setminus E(\beta^*)$, then $\pi_r(u) = \pi_r(v)$ as $|\pi_r(u) - \pi_r(v)| \leq \beta_e^* = 0$. If u and v are not connected by an edge, then there has to be a path $(u, w_1), (w_1, w_2), \ldots, (w_l, v)$ consisting of edges in $E(G) \setminus E(\beta)$. Then, $\pi_r(u) = \pi_r(w_1) = \pi_r(w_2) = \ldots = \pi_r(w_l) = \pi_r(v)$ using the same argument. □

Lemma 3. *If β^* is an optimal solution of the dual problem, then every edge in $E(\beta^*)$ connects nodes from different partitions defined by β^*.*

Proof. Assume that the claim of this lemma does not hold for a graph G and an optimal solution β^*. Let $e^* \in E(\beta^*)$ be an edge that connects two nodes u, v from the same partition. From Lemma 2, we have that $\pi_r(u) = \pi_r(v)$ for every r. Let $\beta' \in \mathbb{R}_{\geq 0}^{|E(G)|}$ be the following vector: $\beta'_e = \beta^*_e$ if $e \neq e^*$, and $\beta'_e = 0$ if $e = e^*$. Then,

- Constraint 18 of the dual problem is satisfied by β', since $0 = |\pi_r(u) - \pi_r(v)| \leq \beta'_{e^*} = 0$, and no other constraint depends on the value of β'_{e^*}. Thus, β' is a solution of the dual problem.
- The total weight of β' is less than the total weight of β^* as $\sum_{e \in E(G)} \beta'_e = \left(\sum_{e \in E(G)} \beta^*_e\right) - \beta^*_{e^*}$.

Therefore, β^* cannot be an optimal solution. □

Lemma 4. *Let $V_1, \ldots, V_k$ be the partitioning defined by an optimal solution β^* of the dual problem. If $r, v \in V_i$, then $\pi_r(v) = 0$.*

Proof. We have $\pi_r(r) = 0$ by definition. From Lemma 2, we also have that $\pi_r(v) = \pi_r(r)$ as r and v are in the same partition. □

Lemma 5. *Let $V_1, \ldots, V_k$ be the partitioning defined by an optimal solution β^* of the dual problem and let $E(V_i, V_j)$ denote the set of edges between V_i and V_j. For every V_i and V_j, if $e', e'' \in E(V_i, V_j)$, then $\beta^*_{e'} = \beta^*_{e''}$.*

Proof. Assume that the claim of this lemma does not hold for a graph G, an optimal solution β^* and a pair of edges $e' = (v'_i, v'_j)$, $e'' = (v''_i, v''_j)$, i.e., $\beta^*_{e'} > \beta^*_{e''}$. From Lemma 2, we have that $\pi_r(v'_i) = \pi_r(v''_i)$ and $\pi_r(v'_j) = \pi_r(v''_j)$ for every r. Therefore, $|\pi_r(v'_i) - \pi_r(v'_j)| = |\pi_r(v''_i) - \pi_r(v''_j)| \leq \beta^*_{e''}$. Let $\beta' \in \mathbb{R}_{\geq 0}^{|E(G)|}$ be the following vector: $\beta'_e = \beta^*_e$ if $e \neq e'$, and $\beta'_e = \beta^*_{e''}$ if $e = e'$. Then,

– Constraint 18 of the dual problem is satisfied by β', since $|\pi_r(v'_i) - \pi_r(v'_j)| \leq \beta^*_{e''} = \beta'_{e'}$, and no other constraint depends on the value of $\beta'_{e'}$. Thus, β' is a solution of the dual problem.
– The total weight of β' is less than the total weight of β^* as it was decreased by $\beta^*_{e'} - \beta^*_{e''} > 0$.

Therefore, β^* cannot be an optimal solution. □

Lemma 6. *There is an optimal solution of the dual problem such that if $r, s \in V_i$ then $\pi_r(v) = \pi_s(v)$ for every v.*

Proof. Let β^* and π^* be an optimal solution such that there exists a pair of nodes $r, s \in V_i$ that do not satisfy the constraint of the lemma, i.e, there exists an $v \in V(G)$ such that $\pi^*_r(v) \neq \pi^*_s(v)$. Let π' be the following potential function: $\pi'_r = \pi^*_s$ and $\pi'_u = \pi^*_u$ if $u \neq r$. Since π^*_s satisfies every constraint of the dual problem, so does π'_r; thus, π' is also an optimal solution. By repeatedly applying the above step, we can construct a solution that satisfies the constraint of the lemma. □

From the above lemmas, the following corollary directly follows:

Corollary 2. *The dual problem is equivalent to the following optimization problem:*

On every partition $V_1, \ldots, V_k$ ($k \geq 2$) of $V(G)$ and every potential function $\pi_{V_i}(V_j) \geq 0$, $\forall 1 \leq i, j \leq k$ such that

$$\forall i : \ \pi_{V_i}(V_i) = 0 \ , \tag{26}$$

$$\forall i : \ \sum_{1 \leq j \leq k} |V_j| \cdot \pi_{V_i}(V_j) \geq 1 \ , \tag{27}$$

minimize the objective function

$$\sum_{1 \leq i < j \leq k} |E(V_i, V_j)| \cdot \max_{1 \leq r \leq k} |\pi_{V_r}(V_i) - \pi_{V_r}(V_j)| \ , \tag{28}$$

where $E(V_i, V_j)$ is the set of edges between V_i and V_j.

Formulating the dual problem as a graph partitioning problem allows us to prove Theorem 4 in Section 8, which shows that $h'(G)$ is related to a graph-theoretic metric.

8 Relation to the Cheeger Constant

In [5], it was shown that, in case of the simple loss-in-value function $1_{e \in T}$, the payoff in every Nash equilibrium of the game is the reciprocal of the *strength* of the network $\sigma(G)$. In [7], it was shown that, in case of a natural loss-in-value function for many-to-one networks, the payoff in every Nash equilibrium is the reciprocal of the *persistence* of the network $\pi(G)$. These results link the

graph-theoretic robustness of a network to game theory, which gives a better understanding of network robustness. The question naturally arises: can the above computed equilibrium payoff $\frac{1}{h'(G)}$ be linked to an elementary graph metric? In this section, we show that this is possible indeed by studying the relationship between the equilibrium payoff $\frac{1}{h'(G)}$ and the Cheeger constant $h(G)$.

In graph theory, the *Cheeger constant* [10,11] (also called *edge expansion coefficient* [12,13] or *isoperimetric number* [14,15]) of a graph is a measure of "bottleneckedness". It is related to the spectral (or eigenvalue) gap of graph by the Cheeger inequalities and also has interesting applications, such as spectral clustering [16].

Definition 3 (Cheeger constant). *The* Cheeger constant *of a graph G, denoted by $h(G)$, is*

$$h(G) = \min \left\{ \frac{|\partial U|}{|U|} : U \subset V(G), 0 < |U| \leq \frac{|V(G)|}{2} \right\} \ , \tag{29}$$

where ∂U is the collection of all edges between U and $V(G) \setminus U$.

If $h(G)$ is low, then there is a relatively small set of edges A that partitions the graph into two connected components which are both relatively large, i.e., A is a "bottleneck". The intuition is that these bottlenecks correspond to the optimal attacks against a network. We will see that this is indeed true for many graphs[4].

Theorem 4. *For every graph G,*

$$h'(G) \leq h(G) \ . \tag{30}$$

Proof. We show that the value of the optimization problem in Corollary 2, which is equal to $h'(G)$, is upper bounded by $h(G)$. Consider a restricted optimization problem, where the search space is restricted to partitions into two parts, denoted by V_1 and V_2. Since this is a minimization problem, the value of the restricted problem is an upper bound of the value of the original problem. The optimal values of the potential function are determined by the sizes of V_1 and V_2: $\pi_{V_1}(V_2) \geq \frac{1}{|V_2|}$ and $\pi_{V_2}(V_1) \geq \frac{1}{|V_1|}$. Without loss of generality, let $|V_1| \leq |V_2|$. Then, the value of the restricted optimization problem is

$$\min_{V_1 \subset V(G)} |E(V_1, V_2)| \cdot \max\{\pi_{V_1}(V_2), \pi_{V_2}(V_1)\} \tag{31}$$

$$= \min_{V_1 \subset V(G)} |E(V_1, V_2)| \cdot \max\{\frac{1}{|V_2|}, \frac{1}{|V_1|}\} \tag{32}$$

$$= \min_{V_1 \subset V(G)} |E(V_1, V_2)| \cdot \frac{1}{|V_1|} \tag{33}$$

$$= \min_{V_1 \subset V(G)} \frac{|\delta V_1|}{|V_1|} \tag{34}$$

$$= h(G) \ . \tag{35}$$

[4] As a first example, we note that it is true for the network shown in Figure 3.

Therefore, we have that

$$h'(G) \leq h(G) \ . \tag{36}$$

□

The proof of the above theorem shows that the robustness metric $h'(G)$ can be interpreted as a possible "generalization" of the Cheeger constant to arbitrary partitionings.

Theorem 5. *There is a graph G such that $h'(G) < h(G)$.*

Proof. Consider the complete graph K_3. It is easy to see, that the Cheeger constant of K_3 is $h(K_3) = 2$. We now show that the adversary can achieve a higher payoff than $\frac{1}{h(K_3)} = \frac{1}{2}$. Let the strategy of the adversary be $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. In any pure strategy of the operator, two edges are used and the expected loss of both edges is 1, since one node is cut off by the removal of each edge; therefore, the expected payoff is $\frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 0 = \frac{2}{3}$. Since this is true for every pure strategy of the operator, it is also true for every mixed strategy. □

The following theorem shows that the bound is tight:

Theorem 6. *There are an infinite number of graphs such that $h'(G) = h(G)$.*

Proof. Consider a complete graph K_{2n}, $n \in \mathbb{Z}_+$. It is well-known that the Cheeger constant of a complete graph K_{2n} is $h(K_{2n}) = \lceil \frac{2n}{2} \rceil = n$. We now show that $h'(G) \geq h(G)$ by describing an operator strategy that achieves $\frac{1}{n}$ expected payoff. Let the strategy of the operator be the uniform distribution on the set consisting of every star subgraph S_{2n} of K_{2n}. There are $2n$ such stars; therefore, the probability of each star is $\frac{1}{2n}$. Each edge of the graph is contained by two stars and the loss of an edge is 1 in both stars, since one node is cut off by the removal of the edge in both stars. Thus, its expected loss is $2 \cdot \frac{1}{2n} \cdot 1 = \frac{1}{n}$. Since the expected loss every edge is $\frac{1}{n}$, so is the expected payoff. □

9 Generalization to Non-uniform Node Weights

By measuring the *number of nodes* that are cut off from the larger connected component, we assume that each node is equally valuable or important. In practice, however, this assumption does not always hold. To relax this assumption, in this section, we generalize our results to the case where nodes have non-uniform value or importance, which can be represented by assigning a d_v weight to each node.

Let $\lambda(T, e)$ measure the *total weight of nodes* that are separated from the larger connected component of the network after the attack, i.e.,

$$\lambda(T, e) := \begin{cases} \sum_{v \in \text{ smaller component of } G[T \setminus \{e\}]} d_v, & \text{if } e \in T \\ 0, & \text{if } e \notin T \ . \end{cases} \tag{37}$$

In this model, an optimal operator strategy is given by the following linear program:

Variables:

$$\forall r \in V(G): \ \alpha_r \in \mathbb{R}_{\geq 0}$$
$$\forall r \in V(G)\ \forall \{u,v\} \in E(G): \ f_r(u,v), f_r(v,u) \in \mathbb{R}_{\geq 0}$$

Objective:

$$\text{maximize} \sum_{r \in V(G)} \alpha_r \tag{38}$$

Constraints:

$$\forall \{u,v\} \in E(G): \sum_{r \in V(G)} f_r(u,v) + f_r(v,u) \leq 1 \tag{39}$$

$$\forall r \in V(G)\ \forall v \in V(G) \setminus \{r\}: \sum_{\{u,v\} \in E(G)} f_r(v,u) - f_r(u,v) \geq \alpha_r \cdot d_v \tag{40}$$

and an appropriately modified flow decomposition algorithm. The details of this modified algorithm can be found in Appendix A.1. Please note that the definition of the function $\lambda_r(T,e)$ is also modified appropriately: it measures the total weight of nodes that are separated from r in $G[T \setminus \{e\}]$.

An optimal adversarial strategy is given by the dual problem:

Variables:

$$\forall e \in E(G): \ \beta_e \in \mathbb{R}_{\geq 0}$$
$$\forall r, v \in V(G), r \neq v: \ \pi_r(v) \in \mathbb{R}_{\geq 0}$$

Objective:

$$\text{minimize} \sum_{e \in E(G)} \beta_e \tag{41}$$

Constraints:

$$\forall r \in V(G): \sum_{v \in V(G) \setminus \{r\}} \pi_r(v) \cdot d_v \geq 1 \tag{42}$$

$$\forall r \in V(G)\ \forall \{u,v\} \in E(G): \ |\pi_r(u) - \pi_r(v)| \leq \beta_{\{u,v\}} \ . \tag{43}$$

Otherwise, everything is the same. Since the proofs for this model can be obtained by appropriately modifying the original proofs in a very straightforward way, we omit them here.

10 Conclusions

In this paper, we introduced a linear loss-in-value function for the network blocking game. As one of our main contributions, we provided strongly polynomial-time algorithms to compute optimal adversarial and operator strategies and, thus, the payoff in the Nash equilibria of the game. To the best of our knowledge, these are the first efficient algorithms for the network blocking game with a

loss-in-value function that is not too simplistic. The efficiency of these algorithms allows us to measure the game-theoretic robustness of networks in practice. Furthermore, the optimal strategies can be also used to identify critical edges that are likely to be attacked. We also generalized our model to non-uniform node weights, which allows nodes to have varying importance or value.

In addition, we proved that the payoff in the Nash equilibria of the game is closely related to the Cheeger constant of a graph (also called minimum edge expansion or isoperimetric number), a well-known metric in graph theory. Therefore, the game-theoretic robustness metric resulting from the linear loss-in-value function can be related to a graph-theoretic robustness notion.

Acknowledgements. This paper has been supported by HSN Lab, Budapest University of Technology and Economics, http://www.hsnlab.hu. Levente Buttyán is supported by the Hungarian Academy of Sciences through the funding of the Academic Research Group on Information Systems (ID: 04-130). The work is also related to the internal project of the authors' hosting institution on "Talent care and cultivation in the scientific workshops of BME", which is supported by the grant TÁMOP - 4.2.2.B-10/1–2010-0009.

References

1. Gueye, A., Marbukh, V., Walrand, J.C.: Toward a metric for communication network vulnerability to attacks: A game theoretic approach. In: Proc. of the 3rd International ICST Conference on Game Theory for Networks, GameNets 2012, Vancouver, Canada (May 2012)
2. Cunningham, W.: Optimal attack and reinforcement of a network. Journal of the ACM 32(3), 549–561 (1985)
3. Bauer, D., Broersma, H., Schmeichel, E.: Toughness in graphs–a survey. Graphs and Combinatorics 22(1), 1–35 (2006)
4. Laszka, A., Buttyán, L., Szeszlér, D.: Optimal selection of sink nodes in wireless sensor networks in adversarial environments. In: Proc. of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia, WoWMoM 2011, Lucca, Italy, pp. 1–6 (June 2011)
5. Gueye, A., Walrand, J.C., Anantharam, V.: Design of Network Topology in an Adversarial Environment. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 1–20. Springer, Heidelberg (2010)
6. Gueye, A., Walrand, J.C., Anantharam, V.: A network topology design game: How to choose communication links in an adversarial environment? In: Proc. of the 2nd International ICST Conference on Game Theory for Networks, GameNets 2011, Shanghai, China (April 2011)
7. Laszka, A., Szeszlér, D., Buttyán, L.: Game-theoretic robustness of many-to-one networks. In: Proc. of the 3rd International ICST Conference on Game Theory for Networks, GameNets 2012, Vancouver, Canada (May 2012)
8. Schwartz, G.A., Amin, S., Gueye, A., Walrand, J.: Network design game with both reliability and security failures. In: Proc. of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 675–681. IEEE (September 2011)

9. Tardos, E.: A strongly polynomial algorithm to solve combinatorial linear programs. Operations Research 34(2), 250–256 (1986)
10. Chung, F.R.K.: Spectral graph theory. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence (1997)
11. Chung, F.R.K.: Laplacians and the Cheeger inequality for directed graphs. Annals of Combinatorics 9(1), 1–19 (2005)
12. Alon, N.: On the edge-expansion of graphs. Combinatorics, Probability and Computing 6(2), 145–152 (1997)
13. Alon, N.: Spectral techniques in graph algorithms. In: Proc. of the 3rd Latin American Symposium on Theoretical Informatics, Campinas, Brazil, pp. 206–215 (April 1998)
14. Mohar, B.: Isoperimetric numbers of graphs. Journal of Combinatorial Theory, Series B 47(3), 274–291 (1989)
15. Mohar, B.: Isoperimetric inequalities, growth, and the spectrum of graphs. Linear Algebra and Its Applications 103, 119–131 (1988)
16. Bühler, T., Hein, M.: Spectral clustering based on the graph p-Laplacian. In: Proc. of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Canada, pp. 81–88 (June 2009)

A Flow-Decomposition Algorithm

We introduced our flow decomposition algorithm in [7]. To make this paper self-contained, in this section, we give an overview of the algorithm. For a detailed proof, see [7].

In Section 4, we used two non-negative flow variables for each undirected edge to allow the summation of the absolute values of multiple flows in the linear program. For simplicity, in the following description, we use a more classical approach and replace each edge with two arcs facing opposite directions, each arc having a single non-negative flow value.

Theorem 7. *Given a graph G with a sink node r and a multi-source flow f*[5] *such that each $v \in V(G) \setminus \{r\}$ node is a source producing α, there exist weights α_T, $T \in \mathcal{T}(G)$ such that $\sum_{T \in \mathcal{T}(G)} \alpha_T = \alpha$ and $\forall e \in E(G) : \sum_{T \in \mathcal{T}(G)} \alpha_T \cdot \lambda_r(T, e) \leq f(e)$*[6] *.*

Proof. Our proof is constructive and it is based on the following algorithm:

1. Find a spanning reverse arborescence T rooted at r in G such that
   - T only includes edges to which f assigns a positive flow amount and
   - every edge is directed in the same way as the flow.
2. Calculate $\lambda_r(T, e)$ for every $e \in T$.
3. Let $\alpha_T := \min_{e \in T} \frac{f(e)}{\lambda_r(T,e)}$.
4. For every $e \in E(G)$, let $f(e) := f(e) - \alpha_T \cdot \lambda_r(T, e)$.

[5] A multi-source flow is a network flow with a set of sources instead of only one source.
[6] Please recall the definition of $\lambda_r(T, e)$ from Section 4: it is the number of nodes that are separated from r in $G[T \setminus \{e\}]$.

5. If the incoming flow assigned by f to r is greater than zero, then continue from Step 1.
6. Let $\alpha_T := 0$ for every other spanning tree.

Before proving the correctness of the algorithm, we have to prove that Step 1 can be executed in each iteration, otherwise the algorithm would terminate with an error. Obviously, if f is a network flow and the amount of outgoing flow from every $v \in V(G) \setminus \{r\}$ is positive, there has to be a directed path from every $v \in V(G) \setminus \{r\}$ to r consisting of edges with positive flow amounts. Thus, we have to show that the outgoing flow from every $v \in V(G) \setminus \{r\}$ remains positive as long as the incoming flow to r is positive.

For a $v \in V(G) \setminus \{r\}$, let Λ_v denote $\lambda_r(T, e_{out})$, where e_{out} is the outgoing edge of v in T. Clearly, the sum of $\lambda_r(T, e_{in})$ over all incoming edges $e_{in} \in E(G)$ of v is $\Lambda_v - 1$. Since the flow along every edge e is decreased by $\alpha_T \cdot \lambda_r(T, e)$, the sum of outgoing flows is decreased by $\alpha_T \cdot \Lambda_v$. Similarly, the sum of incoming flows is decreased by $\alpha_T \cdot (\Lambda_v - 1)$. Therefore, the net outgoing flow of v is decreased by α_T. Since the outgoing flow of every v is the same at the beginning and it is decreased by the same amount in every iteration, they are decreased to zero simultaneously.

Now, we can prove the correctness of the algorithm. First, we have to prove that α is indeed a distribution. This is evident, as the amount of incoming flow to r is decreased by $\alpha_T(|V(G)| - 1)$ at every assignment, and the amount is $|V(G)| - 1$ at the beginning and zero after the algorithm has finished; therefore, $\sum_{T \in \mathcal{T}} \alpha_T = 1$.

Second, we have to prove that $\forall e \in E(G) : \sum_{T \in \mathcal{T}(G)} \alpha_T \cdot \lambda_r(T, e) \leq f(e)$. At every α_T assignment, the flow along every edge is decreased by $\alpha_T \cdot \lambda_r(T, e)$ and it is never decreased to a negative value. Therefore $\sum_{T \in \mathcal{T}} \alpha_T \cdot \lambda_r(T, e) \leq f(e)$.

Finally, we show that the algorithm terminates after at most $|E(G)|$ iterations. In every iteration, the flow along at least on edge (i.e., along every edge for which $\frac{f(e)}{\lambda_r(T,e)}$ is minimal) is decreased from a positive amount to zero. Since there are $|E(G)|$ edges, there can be at most $|E(G)|$ iterations. □

From the last paragraph of the proof, it also follows that the support of the resulting distributions consists of at most $|E(G)|$ spanning trees.

A.1 Flow-Decomposition Algorithm with Non-uniform Node Weights

The algorithm is fundamentally the same as in the case of uniform node weights. The following modifications have to be made:

- Each $v \in V(G) \setminus \{r\}$ node is a source producing $\alpha \cdot d_v$, instead of α.
- Consequently,
  - the sum of $\lambda_r(T, e_{in})$ over all incoming edges $e_{in} \in E(G)$ of v is $\Lambda_v - d_v$, instead of $\Lambda_v - 1$,
  - the net outgoing flow of v is decreased by $\alpha_T \cdot d_v$, instead of α_T,
  - the incoming flow to r is decreased by $\alpha_T \sum_{v \in V(G) \setminus \{r\}} d_v$, instead of $\alpha_T(|V(G)| - 1)$.

Deceptive Routing in Relay Networks

Andrew Clark[1], Quanyan Zhu[2], Radha Poovendran[1], and Tamer Başar[2,⋆]

[1] Department of Electrical Engineering,
University of Washington,
Seattle, WA 98195 USA
{awclark,rp3}@u.washington.edu
[2] Coordinated Science Laboratory and
Department of Electrical and Computer Engineering,
University of Illinois at Urbana Champaign,
1308 W. Main St., Urbana, IL, USA, 61801
{zhu31,basar1}@illinois.edu

Abstract. Physical-layer and MAC-layer defense mechanisms against jamming attacks are often inherently reactive to experienced delay and loss of throughput after being attacked. In this paper, we study a proactive defense mechanism against jamming in multi-hop relay networks, in which one or more network sources introduce a deceptive network flow along a disjoint routing path. The deceptive mechanism leverages strategic jamming behaviors, causing the attacker to expend resources on targeting deceptive flows and thereby reducing the impact on real network traffic. We use a two-stage game model to obtain deception strategies at Stackelberg equilibrium for selfish and altruistic nodes. The equilibrium solutions are illustrated and corroborated through a simulation study.

Keywords: Game Theory, Stackelberg Equilibrium, Routing Algorithms, Jamming and Security, Relay Networks.

1 Introduction

Wireless networks play a crucial role in many military and commercial applications. The open wireless medium, however, leaves such networks vulnerable to jamming attacks, in which an adversary broadcasts an interfering signal in the vicinity of a node, preventing any incoming packets from being correctly decoded. Jamming attacks are particularly harmful when the adversary can exploit weaknesses in the physical or MAC layer protocols used by the nodes [4], or target intermediate relay nodes in a multi-hop network to reduce the end-to-end-throughput [11]. Different classes of jamming adversary have been studied, including constant jammers that emit a constant interfering signal, random jammers that broadcast an interfering signal at random intervals, and intelligent

⋆ The research was partially supported by the AFOSR MURI Grant FA9550-10-1-0573, ARO MURI Grant W911NF-07-1-0287, and also by an NSA Grant through the Information Trust Institute at the University of Illinois.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 171–185, 2012.

jammers that can selectively target packets from different flows to maximize the damage of the attack [5].

Defense mechanisms against jamming are based on physical-layer techniques, such as beamforming, spread-spectrum, and directional antennas [7], or MAC-layer protocols such as channel surfing [12]. When multi-hop routing is used, the source nodes can also decrease the flow rate on paths that experience high packet-loss due to jamming, while increasing the rate on routes experiencing lower packet-loss [10]. This, however, is an inherently reactive defense that cannot be employed until the network has already been targeted by the adversary and experienced loss of throughput.

In this paper, we study a proactive defense mechanism against jamming for multi-hop wireless networks, in which one or more network sources introduce a deceptive network flow, consisting of randomly generated dummy packets, along a disjoint routing path. When the real and deceptive packets are encrypted, the adversary will be unable to distinguish between them, and will expend limited resources, such as jamming power, on targeting a false flow. This leaves fewer jamming resources available for targeting real packets, allowing those packets to escape jamming. The goal of this approach is to use the intelligent attributes of the adversary, such as the ability to target individual packets from specific flows, to create deception and thus mitigate the impact of the attack.

While this approach is promising, several challenges must first be addressed. First, the deceptive packets will traverse the same links as real packets, leading to increased congestion and delays. Second, each source node may have limited capacity to generate, encrypt, and transmit packets, and this scarce capacity must be divided between real and fake flows. Third, if the fake packets are not introduced according to an optimal strategy that leverages information on the adversary's capabilities and goals, then the deception may be ineffective in increasing the throughput of real nodes, and may be counterproductive due to the increase in congestion.

To address these issues, we introduce a game-theoretic framework for thwarting jamming attacks through deceptive flows. Our framework is based on a two-stage game between a set of sources and an adversary mounting the jamming attack. In the first stage of the game, the sources play a noncooperative game in order to select the real and deceptive flow allocations. In the second stage, the adversary observes the total flow allocation of each source and selects a jamming strategy accordingly in order to maximize the decrease in throughput. We study the deceptive jamming game under two types of source behavior, namely a selfish source that maximizes its own throughput while disregarding the delays experienced by other sources, and an altruistic source that incorporates the delays of other sources when choosing flow rates. We derive the equilibria of the game for each case, and provide efficient algorithms for allocating real and deceptive flows at each source based on the equilibria. Our results are illustrated through a simulation study.

The paper is organized as follows. In Section 2, we review related work on jamming attacks and defenses. In Section 3, the system and adversary models

are introduced. Section 4 contains the game formulation and solution algorithms for each player. Section 5 presents our simulation results. Section 6 concludes the paper.

2 Related Work

The vulnerability of wireless networks to jamming attacks has been extensively explored [7]. In particular, the use of commodity wireless devices has led to efficient jamming attacks that target specific network protocols, such as 802.11 [4]. Jamming defenses at the physical layer are based on spread-spectrum communication [3], such as frequency hopping, in which jammed receivers change frequency in order to prevent the attacker from discovering the channel [8]. Spatial retreat, in which nodes that detect a jammer move away from the jammed region, was discussed in [12]. These lower-layer defenses are not affected by our proposed approach, and can be employed alongside our methods to further increase the robustness to jamming.

The impact of jamming attacks on multi-hop wireless networks, in which the jammer targets intermediate relay nodes in order to disrupt the end-to-end throughput, was studied in [11]. This work focuses on quantifying the impact of jamming for a given set of network flows and not on responding to jamming. In [10], a flow allocation approach to mitigating jamming was presented, in which each source responds to an increase in packet-loss rate, corresponding to increased jamming activity, by shifting flow to an alternative path with lower loss rate. The work of [10], however, does not explicitly model the goals and constraints of the adversary, and therefore does not enable a strategic approach to flow allocation, let alone introducing deception.

In [13], we proposed thwarting jamming attacks by introducing a deceptive flow, causing the adversary to waste resources and allowing valid packets to avoid being jammed. That work, however, focused on a single source selecting routing paths for real and deceptive flows. Multiple sources introducing deceptive flows leads to several challenges. First, the added deceptive flows may increase congestion and delay in the network. Second, the effect of the deceptive flow will depend on the flow allocations of other sources, resulting in a coupling between sources. For example, by introducing a deceptive flow that is jammed by an adversary, a source will not only improve its own throughput, but also the throughput of nearby sources, since the adversary will have fewer resources available to target those flows.

3 Model and Preliminaries

In this section, we introduce the network and adversary models along with relevant notations.

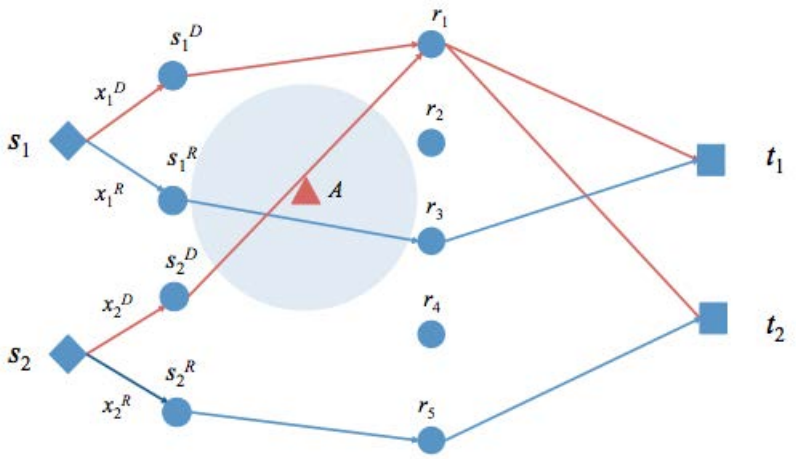


Fig. 1. Illustration of the network model with two source nodes s_1 and s_2, which transmit data to destination t_1 and t_2, respectively, via the relay network consisting of five relay nodes $r_1, r_2, \cdots, r_5$

3.1 Network Model

We consider N source nodes, indexed in the set $\mathcal{S} = \{s_1, \ldots, s_N\}$. Source $s_i, i = 1, 2, \cdots, N$, has a corresponding destination node $d_i \in \mathcal{T}$, where $\mathcal{T} := \{t_1, t_2, \cdots, t_T\}$ denotes the set of T destinations. Each source s_i maintains a real flow to d_i, consisting of data packets, with rate x_i^R, as well as a deceptive flow consisting of randomly-generated fake packets at rate x_i^D, with $x_i^D + x_i^R \leq m_i, x_i^D \geq 0, x_i^R \geq 0$. The deceptive flow aims to deceive the attackers along the routing path between the source and destination pair in order to protect the real flow[1]. Since each source maintains two flows, we can equivalently represent each source node $s_i, i = 1, 2, \cdots, N$, with two virtual source nodes s_i^D and s_i^R, where s_i^D is the virtual node that transmits deceptive flows while s_i^R is the virtual node that sends real data. Let $\mathcal{S}^D := \{s_1^D, s_2^D, \cdots, s_N^D\}$ be the set of N deceptive source nodes , and likewise, let $\mathcal{S}^R := \{s_1^R, s_2^R, \cdots, s_N^R\}$ be the set of N real source nodes.

We consider a multi-hop relay network where sources have to send their data via intermediate nodes. Let $\mathcal{R} := \{r_1, r_2, \cdots, r_R\}$ be the set of R relay nodes deployed between sources and destinations. In general, the relay nodes can form a hierarchical structure for multi-hop routing. In this work, without loss of generality, we consider a two-hop routing scenario, where source nodes first route data to relay nodes and then relay nodes to the destinations. We assume that the routing paths have been predetermined by standard routing protocols [2,6]. Let $a_i^e \in \mathcal{R}$

[1] In principle, the deceptive flow could also contain duplicate copies of real packets. We assume, however, that the sources will not choose to send real packets along routes that are likely to be jammed.

be the relay node chosen by the source node $s_i^e, i = 1, 2, \cdots, N, e \in \{R, D\}$, with $a_i^R \neq a_i^D$ for all $s_i \in \mathcal{S}$, and $B_j \subset \mathcal{T}$ be the set of destinations that receive packets from relay r_j. Let $\mathcal{L}_S^R$ be the set of links between sources and relays, and $\mathcal{L}_R^T$ be the set of links between relays and destinations. We can represent the routing network by the graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices consisting of deceptive and real virtual source nodes, relay nodes and destinations, i.e., $\mathcal{V} := \mathcal{S}^D \cup \mathcal{S}^R \cup \mathcal{R} \cup \mathcal{T}$, and $\mathcal{E}$ is a set of data links between nodes, i.e., $\mathcal{E} := \mathcal{L}_S^R \cup \mathcal{L}_R^T$.

Each source node maintains two routes to its destination, one route for the real flow and one for the deceptive flow. The real flow of source s_i in the relay network can be represented by the set $f_i^R := \{(s_i^R, a_i^R), (a_i^R, d_i)\}, i \in \mathcal{S}$, and the deceptive flow of source s_i can be represented by the set $f_i^D := \{(s_i^D, a_i^D), (a_i^D, d_i)\}, i \in \mathcal{S}$. We let $\mathcal{F}^R := \{f_i^R, i = 1, 2, \cdots, N\}$ be the set of real flows, $\mathcal{F}^D := \{f_i^D, i = 1, 2, \cdots, N\}$ be the set of deceptive flows, and $\mathcal{F} := \mathcal{F}^D \cup \mathcal{F}^R$ be the set of flows in the relay network.

To conserve notation, we let f denote a particular flow in $\mathcal{F}$, and write x_f to denote the data rate for flow $f \in \mathcal{F}$ at the transmission rate, which may be real or deceptive. For example, the real flow $f_i^R, i \in \mathcal{S}$, has its transmission data rate $x_{f_i^R} = x_i^R$ and the data rates of deceptive flows $f_i^D, i \in \mathcal{S}$ are given by x_i^D. We let $\mathcal{L}$ denote the set of L bottleneck links in the network, and they consist of links directed to relay nodes and destination nodes. Hence $L = R + T$. We use $\mathcal{L}_f \subset \mathcal{L}$ to denote the set of those links traversed by flow $f \in \mathcal{F}$, with $\mathcal{L}_i^R \subset \mathcal{L}$ denoting the set of links traversed by real flow f_i^R and $\mathcal{L}_i^D \subset \mathcal{L}$ denoting the set of links traversed by deceptive flow f_i^D. The routes for real and deceptive flows for source i are assumed to be link-disjoint, with $\mathcal{L}_i^R \cap \mathcal{L}_i^D = \emptyset$. Each link l is assumed to have a finite capacity μ_l. Letting $\mathcal{F}_l$ denote the set of flows traversing link l, the capacity constraint can be expressed as $\sum_{f \in \mathcal{F}_l} x_f \leq \mu_l$. We assume that the delays $\tau_l \in \mathbb{R}_+$ experienced by each link follow an independent M/M/1 queueing model [9], with the delay on link l given by

$$\tau_l = \frac{1}{\bar{\mu}_l - \sum_{f \in \mathcal{F}^l} x_f}, \quad l \in \mathcal{L}, \tag{1}$$

where $\bar{\mu}_l = \mu_l - \epsilon$, for $\epsilon > 0$ sufficiently small. In addition, each source i has a capacity constraint m_i, so that $x_i^R + x_i^D \leq m_i$. The routing path of each source is represented by the routing matrix W, which is a $|\mathcal{L}| \times |\mathcal{F}|$ real matrix with a 1 in the (l, f) entry if flow f traverses link l and a 0 otherwise. The capacity constraint can be expressed in a more compact form as

$$W\mathbf{x} \leq \mu, \tag{2}$$

where $\mu = [\mu_1, \mu_2, \cdots, \mu_{|\mathcal{L}|}]$. We use $W_R \in \mathbb{R}^{|\mathcal{L}|} \times \mathbb{R}^{|\mathcal{F}^R|}$ to denote the routing matrix restricted to the set of real flows.

We illustrate the network model in Fig. 3. Sources s_1 and s_2 transmit data to destinations t_1 and t_2, respectively. Both sources split their traffic into two flows: one is the deceptive flow containing randomly generated packets at rates x_1^D and x_2^D for sources s_1 and s_2 respectively; the other one is the legitimate flow

containing the real data at rates x_1^R and x_2^R for sources s_1 and s_2 respectively. A relay network consisting of a set of relay nodes $\mathcal{R} = \{r_i, i = 1, 2, \cdots, 5\}$ is used to transmit data. The topology of the routing network can be represented by the graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \{s_1^D, s_1^R\} \cup \{s_2^D, s_2^R\} \cup \{t_1, t_2\} \cup \mathcal{R}$, and $\mathcal{E} = \{(s_1^D, r_1), (s_2^D, r_1), (s_1^R, r_3), (s_2^R, r_5), (r_1, t_1), (r_1, t_2), (r_3, t_1), (r_5, t_2)\}$. An attacker A can jam the flows within its range of influence. The network consists of four flows: $f_1^R = \{(s_1^R, r_3), (r_3, t_1)\}$, $f_1^D = \{(s_1^D, r_1), (r_1, t_1)\}$, $f_2^R = \{(s_2^R, r_5), (r_5, t_2)\}$ and $f_2^D = \{(s_2^D, r_1), (r_1, t_2)\}$, among which f_1^R and f_2^D are jammed by the attacker A. The relay network has 5 links associated with 5 relay nodes. Note that flows f_1^D and f_2^D share the same link and hence their rates are constrained by $x_{f_1^D} + x_{f_2^D} \leq \mu_1$, where μ_1 is the capacity constraint on link 1 associated with r_1.

3.2 Adversary Model

The network is deployed in the presence of an adversary mounting a jamming attack on a set of flows $\mathcal{F}_A \subseteq \mathcal{F}$. The adversary has knowledge of the routing topology for the flows in $\mathcal{F}_A$ as well as the flow rate x_f for all $f \in \mathcal{F}_A$. The adversary is capable of differentiating between packets from different flows and targeting individual packets for attack [11]. Since packets are encrypted, however, the adversary cannot differentiate between real and deceptive flows.

The adversary chooses a fraction of flow $f \in \mathcal{F}_A$ to target, denoted p_f. The cost to jam flow f is equal to $c_f p_f$, where c_f is a nonnegative constant determined by the jamming power, the distance between the jammer and the jammed receiver, and the channel characteristics. The total jamming power budget is equal to J, resulting in a jamming power constraint $\sum_{f \in \mathcal{F}_A} c_f p_f \leq J$.

We assume that the adversary does not attempt to differentiate between the real and deceptive flows by observing the flow rates or network topology, and instead assumes that all packets have an equal likelihood of being real. Otherwise, the sources could gain an advantage by choosing the rate or routing path of the deceptive flow in order to convince the adversary that it is real.

4 Game Formulation and Equilibria

In this section, the interaction between the adversary and network sources is described. We first describe the actions of the adversary, who observes the flow rates and routing topology and chooses a jamming strategy accordingly. We then discuss the actions of the sources, who determine the flow rates x_f.

4.1 Game Formulation

The deceptive jamming game consists of two stages. In the first stage, each source s_i selects real and deceptive flows x_i^R and x_i^D simultaneously. In the second stage, the adversary observes the flow rates x_f for all $f \in \mathcal{F}_A$ and chooses the jamming rates p_f. When f is a real flow with source s_i, we write $p_f := p_i^R$, while p_i^D is

the probability of jamming for a deceptive flow with source s_i. The adversary's goal is to find the optimal jamming strategy p_f^*, $f \in \mathcal{F}_A$, which is the solution to the optimization problem

$$\begin{array}{c} \text{maximize} \ \sum_{f\in\mathcal{F}_A} U_A(p_f, x_f) \\ p_f, f \in \mathcal{F}_A \\ \text{s.t.} \quad \sum_{f\in\mathcal{F}_A} c_f p_f \le J \end{array} \tag{3}$$

The constant J is the adversary's total power budget. We select $U_A(p_f, x_f) = \ln p_f x_f$ for the analysis later in Section 4.3. At each source s_i, the goal is to optimize a utility function $U_i(x_i^R, x_i^D, x_{-i})$, where x_{-i} is the flow rates of the other sources. We consider two types of utility functions, selfish and altruistic. In the selfish case, source s_i's only goal is to maximize its own throughput while limiting the delay of real packets, leading to utility function

$$U_i^S(x_i^R, x_i^D, x_{-i}) = (1 - p_i^R(x_i^R, x_i^D, x_{-i}))x_i^R - \beta \sum_{l\in L_i^R} \frac{1}{\mu_l - \sum_{f\in\mathcal{F}_l} x_f}. \tag{4}$$

The second term of (4) quantifies the delay resulting from flow rates x_i^R and x_i^D, based on the M/M/1 model described in Section 3.1. In Section 4.3, a closed-form expression for the dependence of p_i^R on the x_i^R and x_i^D values will be derived. The formulation is illustrated in Figure 2(a).

While introducing a deceptive flow on a separate path may increase the achieved throughput and reduce the error rate of a source, it will also increase the congestion, and hence the delays, experienced by the remaining sources. We denote sources that attempt to minimize the delay experienced by other sources,

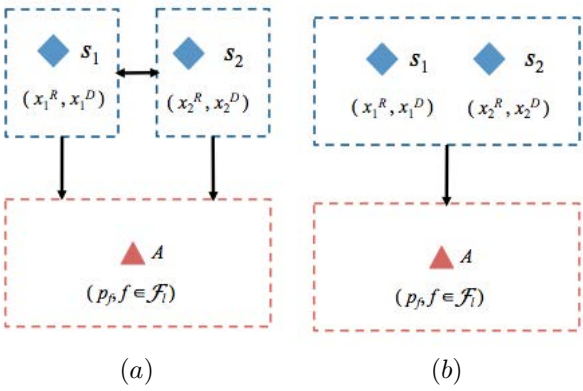


Fig. 2. Illustration of two-stage games and Stackelberg equilibrium is used as solution concept (a) Selfish source nodes: each source first decides on deceptive and real flows in a noncooperative way. (b) Cooperative source nodes: source nodes jointly optimize their data rates to achieve the best total utility. The attacker A sniffs the traffic of the network after source nodes decide on their data rates, and launches a jamming attack by choosing the power levels to affect the flows within its range of influence.

in addition to maximizing their own utility, as *altruistic*. An altruistic source has utility function defined by

$$U_i^T(x_i^R, x_i^D, x_{-i}) = (1 - p_i^R(x_i^R, x_i^D, x_{-i}))x_i^R - \beta \sum_{l \in L_i^R \cup L_i^D} \frac{1}{\mu_l - \sum_{f \in \mathcal{F}_l} x_f}. \quad (5)$$

The second (delay) term of (5) incorporates the delays experienced by both real and deceptive flows, as opposed to the delay term of (4) which only measures delay of real flows. The rationale is that increases in delay in fake packets are due to increased congestion, which will affect the real flows of other sources as well. Hence each source is penalized for the delays caused by fake packets. The altruistic user case is illustrated in Figure 2(b).

4.2 Equilibrium Concepts

The equilibrium concept for the game is dependent on the amount of information available to each player. For the game between the sources and the adversary, the adversary observes the sources' actions, equal to the source rates $x_1^R, x_1^D, \ldots$, before selecting a jamming strategy $p_1^R, p_1^D, \ldots$. Hence the adversary will select the jamming strategy p_f^* according to the optimization problem (3) after observing the actions of the sources.

In the case of selfish sources (see Fig. 2(a)), there are also strategic interactions among the sources. Since the sources cannot observe each others' actions before selecting real and deceptive flow rates, their interactions can be described by a normal-form game by fixing the behavior of the adversary. Hence, *Stackelberg equilibrium* solutions can be used to characterize the outcome for this $(n+1)$-person hierarchical game. Let $\tilde{\mathbf{p}}$ be a feasible action of the attacker as his response to the sources based on the attacker utility function, i.e.,

$$\tilde{p}_f = p_f^*(\mathbf{x}^R, \mathbf{x}^D), f \in \mathcal{F}_A,$$

where $\mathbf{x}^R = [x_i^R]_{i \in \{1,\ldots,N\}}, \mathbf{x}^D = [x_i^D]_{i \in \{1,\ldots,N\}}$, and $\tilde{p}_f : \mathbb{R}^{2N} \to \mathbb{R}^{|\mathcal{F}_A|}, f \in \mathcal{F}_A$, is the reaction map of the attacker.

Definition 1 (Stackelberg Equilibrium). *An action profile* $(\mathbf{x}^{R*}, \mathbf{x}^{D*}, \tilde{\mathbf{p}}) \in \mathbb{R}^{2S} \times \mathbb{R}^{|\mathcal{F}_A|}$ *is a Stackelberg equilibrium if*

$$\tilde{p}_f = p_f^*(\mathbf{x}^{R*}, \mathbf{x}^{D*}), f \in \mathcal{F}_A$$

and the source rates x_i^{R*}, x_i^{D*} *satisfy, for all* $i \in \mathcal{S}$,

$$U_i(x_i^{R*}, x_i^{D*}, \mathbf{x}_{-i}, p_f^*(x_i^{R*}, x_i^{D*}, \mathbf{x}_{-i})) \geq U_i(x_i^R, x_i^D, \mathbf{x}_{-i}, p_f^*(x_i^R, x_i^D, \mathbf{x}_{-i})), \quad (6)$$

for all feasible flow rates x_i^R, x_i^D.

4.3 Solution for Adversary

We consider an attacker with utility function $U_A = \exp\left\{\gamma \sum_{f\in\mathcal{F}_A} \alpha_f \ln x_f p_f\right\}$, where the log function reflects the fact that the adversary attempts to distribute the jamming impact among multiple flows and γ is a risk parameter. The coefficient α_f represents the relative importance of flow f, which is normalized so that $\sum_{f\in\mathcal{F}_A} \alpha_f = 1$. We define $\alpha_f = \frac{x_f}{\sum_{f'\in\mathcal{F}_A} x_{f'}}$, modeling an adversary who places a higher priority on flows that carry more network traffic.

The attacker then solves the optimization problem

$$\begin{aligned} \max_{p_f, f\in\mathcal{F}_A} \ & \exp\left\{\gamma \textstyle\sum_{x\in\mathcal{F}_A} \alpha_f \ln x_f p_f\right\}, \\ \text{s.t.} \ & \textstyle\sum_{f\in\mathcal{F}_A} c_f p_f \le J, \end{aligned} \tag{7}$$

Let $\bar{c}_f = c_f/J$. The solution to this optimization problem is given by

$$p_f = \alpha_f/\bar{c}_f. \tag{8}$$

4.4 Solution for Selfish Sources

We first consider the behavior of source nodes when each source attempts to maximize its own utility, represented by its throughput and delay. In this case, the utility of source i is given by

$$U_i^S(x_i^R, x_i^D) = (1 - p_i^R)x_i^R - \beta \sum_{l\in L_i^R} \frac{1}{\mu_l - \sum_{f\in\mathcal{F}_l} x_f}.$$

Substituting the adversary's strategy for p_i^R yields

$$\begin{aligned} U_i^S(x_i^R, x_i^D) &= \left(1 - \frac{\alpha_i^R}{\bar{c}_i^R}\right) x_i^R - \beta \sum_{l\in L_i^R} \frac{1}{\mu_l - \sum_{f\in\mathcal{F}_l} x_f} \\ &= \left(1 - \frac{x_i^R}{\bar{c}_i^R \sum_{f\in\mathcal{F}_A} x_f}\right) x_i^R - \beta \sum_{l\in L_i^R} \frac{1}{\mu_l - \sum_{f\in\mathcal{F}_l} x_f} \end{aligned}$$

Furthermore, since the total flow originating at source i cannot exceed m_i, we have that each source's optimization problem, given the behavior of the other sources, is

$$\begin{aligned} \text{maximize}_{x_i^R} \ & \left(1 - \tfrac{x_i^R}{\bar{c}_i^R \sum_{f\in\mathcal{F}_A} x_f}\right) x_i^R - \beta \textstyle\sum_{l\in L_i^R} \frac{1}{\mu_l - \sum_{f\in\mathcal{F}_l} x_f} \\ \text{s.t.} \ & x_i^R + x_i^D \le m_i \end{aligned} \tag{9}$$

We observe that the objective function of (9) is strictly increasing in x_i^D, since the routes used by real and deceptive flows are link-disjoint. As a result, the

constraint $x_i^R + x_i^D \leq m_i$ will hold with equality. Moreover, in equilibrium, $x_i^R + x_i^D = m_i$ for all sources i. Letting $M = \sum_{i=1}^{N} m_i$, (9) can be rewritten as

$$\underset{x_i^R}{\text{maximize}} \left(1 - \frac{x_i^R}{M\bar{c}_i^R}\right) x_i^R - \beta \sum_{l \in L_i^R} \frac{1}{\mu_l - \sum_{f \in \mathcal{F}_l} x_f} \tag{10}$$

Taking a linear approximation around the origin to the second term, which models the case where the total data flow through each link is significantly less than the link capacity (as in sensor networks where the nodes themselves face energy constraints that prevent full utilization of the channel), yields

$$U_i^S(x_i^R) = \left(1 - \frac{x_i^R}{M\bar{c}_i^R}\right) x_i^R - \beta \sum_{l \in L_i^R} \frac{1}{\mu_l} \left(1 + \sum_{f \in \mathcal{F}_l} x_f\right) \tag{11}$$

The value of x_i^R that maximizes (11) is $x_i^R = \frac{\bar{c}_i M}{2}\left(1 - \beta \sum_{l \in L_i^R} \frac{1}{\mu_l}\right)$. Furthermore, a quadratic approximation yields

$$U_i^S(x_i^R) = \left(1 - \frac{x_i^R}{M\bar{c}_i^R}\right) x_i^R - \beta \sum_{l \in L_i^R} \frac{1}{\mu_l} \left(1 + \sum_{f \in \mathcal{F}_l} x_f + \left(\sum_{f \in \mathcal{F}_l} x_f\right)^2\right) \tag{12}$$

which attains its maximum at

$$x_i^R = \left(\frac{2}{\bar{c}_i M} + 2\beta \sum_{l \in L_i^R} \frac{1}{\mu_l^2}\right)^{-1} \left[1 - \beta \sum_{l \in L_i^R} \left(\frac{1}{\mu_l} + \frac{2\sum_{f \in \mathcal{F}_l} x_f}{\mu_l^2}\right)\right]. \tag{13}$$

Obtaining the equilibria for the games with responses (11) and (12) is equivalent to solving a system of linear equations. Define D to be a diagonal matrix with entries $D_{ii} = \left(\frac{2}{c_i M} + 2\beta \sum_{l \in L_i^R} \frac{1}{\mu_l^2}\right)^{-1}$. Then (13) can be rewritten as

$$x_i^R = D_{ii}\left(1 - \beta \sum_{l \in \mathcal{L}} \frac{W_{il}}{\mu_l} - 2\beta \sum_{l \in \mathcal{L}} \left[\frac{W_{il}}{\mu_l^2} \sum_{f \in \mathcal{F}} W_{fl} x_f\right]\right).$$

Multiplying by W_{il} and W_{fl} allows us to sum over all entries in $\mathcal{F}$ and $\mathcal{L}$. Let U be a diagonal matrix with entries $U_{ll} = \mu_l$. Since the flow rates satisfy $x_i^R + x_i^D = m_i$, define

$$Z = \begin{pmatrix} 1 & 0 \cdots & 0 \\ -1 & 0 \cdots & 0 \\ & \vdots & \\ 0 & 0 \cdots & 1 \\ 0 & 0 \cdots & -1 \end{pmatrix}, \qquad \mathbf{m} = \begin{pmatrix} 0 \\ m_1 \\ \vdots \\ 0 \\ m_n \end{pmatrix},$$

where $Z \in \mathbf{R}^{2n \times 2n}$ and $\mathbf{m} \in \mathbf{R}^{2n}$, so that $\mathbf{x} = Z\mathbf{x}^R + \mathbf{m}$. Finally let ν be a vector with $\nu_l = 1/\mu_l$. Then the vector of real flow rates can be obtained by solving the matrix equation

$$(I + 2\beta D W_R^T (U^2)^{-1} W Z)\mathbf{x}_R = D(\mathbf{1} - \beta W_R^T \nu - 2\beta W_R^T (U^2)^{-1} W \mathbf{m}). \quad (14)$$

In the case where the quadratic approximation does not hold, the Stackelberg equilibrium of the game with selfish sources can be computed by observing that the best-response optimization problems (10) for each source define a potential game, with potential function

$$\Phi(x_1^R, \ldots, x_N^R) = \sum_{i=1}^{N} x_i^R \left(1 - \frac{x_i^R}{M\bar{c}_i^R}\right) - \beta \sum_{l \in \mathcal{L}} \frac{1}{\mu_l - \sum_{f \in \mathcal{F}_l} x_f} \quad (15)$$

Computing the equilibrium of the selfish-user game is equivalent to solving the optimization problem

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{N} x_i^R \left(1 - \frac{x_i^R}{M\bar{c}_i^R}\right) - \beta \sum_{l \in \mathcal{L}} \frac{1}{\mu_l - \sum_{f \in \mathcal{F}_l} x_f} \\ x_1^R, \ldots, x_N^R & \\ \text{s.t.} & 0 \le x_i^R \le m_i \end{array} \quad (16)$$

Since the potential function $\Phi(\cdot)$ is a strictly concave function of $x_1^R, \ldots, x_N^R$, the optimization problem (16) has a unique solution that can be computed efficiently.

4.5 Solution for Altruistic Sources

When the sources behave altruistically, the utility function for source s_i is given by (5). In this case, we observe that the utility of s_i is no longer increasing in x_i^D, and hence we may have $x_i^R + x_i^D < m_i$. The best response of s_i to the other sources and the attacker is then given by the optimization problem

$$\begin{array}{ll} \text{maximize} & \left(1 - \frac{x_i^R}{\sum_{f \in \mathcal{F}_A} x_f \bar{c}_i^R}\right) x_i^R - \beta \sum_{l \in L_i^R \cup L_i^D} \frac{1}{\mu_l - \sum_{j \in \mathcal{F}_l} x_f} \\ x_i^R, x_i^D & \\ \text{s.t.} & Wx \le \mu \\ & x_i^R + x_i^D \le m_i \;\; i = 1, \ldots, N \end{array} \quad (17)$$

Lemma 1. *The utility function U_i^T is a strictly concave function of x_i^R and x_i^D.*

Proof. The function $(x_i^R)^2 / \sum_{f \in \mathcal{F}} x_f$ is a quadratic-over-linear function, and hence is strictly convex, implying that the first term of U_i^T is strictly concave. The concavity of the second term can be verified by computing its second derivative.

Lemma 1 yields the following theorem.

Theorem 1. *The simultaneous-move game, in which player s_i has utility function U_i^T, has a pure-strategy Stackelberg equilibrium.*

Proof. By [1, Theorem 4.4], an equilibrium in pure strategies exists if the set of feasible flow allocations $(x_1^R, x_2^R, \ldots)$ is compact and convex and the utility function U_i^T is a strictly concave function of x_i^R and x_i^D. The second condition holds by Lemma 1. The set of feasible flow allocations is convex and closed due to the convexity of the constraints $W\mathbf{x} \leq \mu$ and $0 \leq x_i^R + x_i^D \leq m_i$. Furthermore, since $0 \leq x_i^R, x_i^D \leq m_i$ for all i, the set of feasible flow allocations is bounded, and hence compact.

A heuristic algorithm for approximating a solution to the altruistic sources game is as follows. Each source initializes its flow rate to a feasible value, such as 0. At each iteration, each source computes its best-response to the observed flows of the other sources, based on (17). The algorithm terminates when no source can improve its utility by changing its strategy, or after a fixed number of iterations. The algorithm is summarized in Figure 3.

Approximate-Equilibrium: Algorithm for approximating a Stackelberg equilibrium when sources are altruistic.
Input: Link capacities μ, source capacities m_i, $i = 1 \ldots, N$
Routing matrix W, number of iterations K
Output: Real flow rate x_i^R and deceptive flow rate x_i^D for each $s_i \in \mathcal{S}$

$x_i^R, x_i^D \leftarrow 0,\ \forall i = 1, \ldots, N,\ k \leftarrow 0$
while $k < K$
 $b \leftarrow 0$
 for $i = 1, \ldots, N$
 $x_i^{R,old} \leftarrow x_i^R,\ x_i^{D,old} \leftarrow x_i^D$
 $x_i^R,\ x_i^D \leftarrow$ solution to (17) with $x_{-i}^R,\ x_{-i}^D$ as input
 $b \leftarrow 1$ **if** $x_i^R \neq x_i^{R,old}$ or $x_i^D \neq x_i^{D,old}$
 end for
 if $b == 0$
 exit while loop; **return** $x_1^R, x_1^D, \ldots, x_N^R, x_N^D$
end while
return $x_1^R, x_1^D, \ldots, x_N^R, x_N^D$

Fig. 3. Algorithm for approximating a Stackelberg equilibrium of the altruistic sources game

We observe that the sources can update their rates in an arbitrary order (i.e., source 1 does not have to update first, as in Figure 3).

5 Simulation Results

We illustrate our proposed approach through a Matlab simulation study. We consider a network with four sources, four relays, and one destination. Each source has a capacity of 1. All network links have equal capacity, which we varied from $\mu = 2$ to $\mu = 3$. We simulated both selfish and altruistic sources,

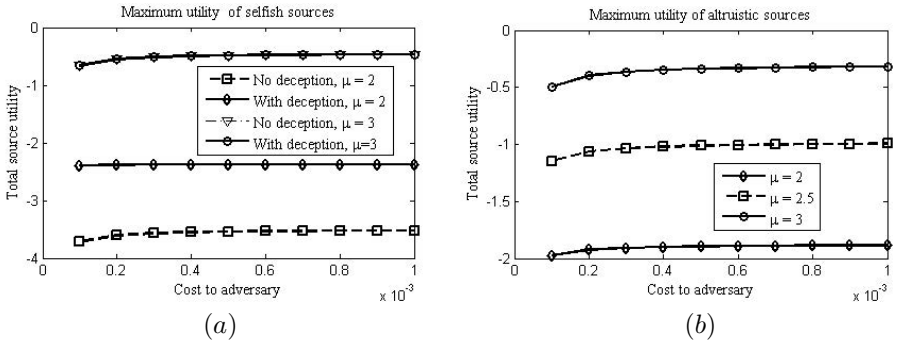


Fig. 4. Illustration of our proposed approach for a network of four sources, four relays, and one destination, with $\beta = 1$ and adversary cost proportional to the distance to each relay. (a) Case where sources are selfish. The utility achieved by the sources increases as the capacities of links and the adversary's cost increase. Deception increases the source utilities. (b) Case of altruistic sources. The utility achieved is higher on the average than the utility of selfish sources.

with trade-off parameter $\beta = 1$. An adversary is assumed to be active in the presence of the relay nodes, with jamming cost proportional to the square of its distance to each relay. The adversary's jamming budget is normalized to 1.

Figure 4(a) shows the utility achieved by selfish sources. The benefit of deception is reflected by an increase in utility. Each source's utility increases as the adversary's cost of jamming increases, since the adversary requires more power to jam the deceptive flows. Furthermore, a higher capacity results in lower delays, further increasing the utility. In the case where the capacities are low, the use of deceptive flows increases the utility of the sources. Increasing the capacity reduces the benefit of deception.

We also observe that altruistic sources yield higher overall utility (Figure 4(b)), since these sources minimize the congestion and delays caused by deceptive flows. As in the selfish source case, an increase in the adversary's cost results in higher utility for the sources. An increase in link capacity will also result in lower delays and higher throughput, increasing the source utility.

6 Conclusion

In this paper, we have studied the problem of mitigating jamming attacks through deception. We considered a defense mechanism in which each source generates a false traffic flow, causing the attacker to expend resources targeting a deceptive flow and enabling real packets to avoid jamming. We formulated deceptive jamming as a two-stage game between the sources and the jammer. In the first stage, the sources simultaneously choose both real and deceptive flow rates to maximize throughput and minimize delay. In the second stage, the

attacker observes the real and deceptive flow rates and selects a jamming strategy, represented by the fraction of each flow to jam. We derived a closed-form expression for the attacker's optimal strategy, which shows the fraction of the adversary's jamming resources that will be used to target deceptive flows, as well as the additional throughput of the real flows resulting from using deception. For the sources, we proved the existence of pure-strategy Stackelberg equilibria for two cases, namely the case where each source allocates flow in order to maximize its own utility (selfish users) and the case where each source incorporates the congestion of other sources when choosing a flow rate (altruistic users). We proposed algorithms for computing the equilibria for both cases, resulting in efficient methods for allocating real and deceptive flows at each source in order to maximize throughput and minimize delay. We illustrated our approach through a simulation study. Our simulations show that altruistic behavior improves the overall utility of the sources. In future work, we intend to analyze the loss of efficiency caused by selfish source behavior, and develop metrics for quantifying the value of deception. We will also study the case where the sources have imperfect information regarding the adversary's utility function and cost.

References

1. Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory, vol. 23. Society for Industrial and Applied Mathematics (SIAM) (1999)
2. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L.: Optimized link state routing protocol for ad hoc networks. In: Proceedings of IEEE International Multi-Topic Conference, INMIC (2001)
3. Liu, Y., Ning, P., Dai, H., Liu, A.: Randomized differential dsss: Jamming-resistant wireless broadcast communication. In: Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM), pp. 1–9 (2010)
4. Noubir, G., Rajaraman, R., Sheng, B., Thapa, B.: On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming. In: Proceedings of the 4th ACM Conference on Wireless Network Security (WiSec), pp. 97–108. ACM (2011)
5. Pelechrinis, K., Iliofotou, M., Krishnamurthy, S.: Denial of service attacks in wireless networks: The case of jammers. IEEE Communications Surveys and Tutorials 13(2), 245–257 (2011)
6. Perkins, C., Royer, E.: Ad-hoc on-demand distance vector routing. In: Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90–100 (1999)
7. Poisel, R.: Modern Communications Jamming Principles and Techniques. Artech House Publishers (2011)
8. Pöpper, C., Strasser, M., Čapkun, S.: Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. IEEE Journal on Selected Areas in Communications (JSAC) 28(5), 703–715 (2010)
9. Ross, S.: Introduction to Probability Models. Academic Press (2009)
10. Tague, P., Nabar, S., Ritcey, J., Poovendran, R.: Jamming-aware traffic allocation for multiple-path routing using portfolio selection. IEEE/ACM Transactions on Networking 19(1), 184–194 (2011)

11. Tague, P., Slater, D., Poovendran, R., Noubir, G.: Linear programming models for jamming attacks on network traffic flows. In: Proceedings of the 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 207–216. IEEE (2008)
12. Xu, W., Wood, T., Trappe, W., Zhang, Y.: Channel surfing and spatial retreats: defenses against wireless denial of service. In: Proceedings of the 3rd ACM Workshop on Wireless Security (WiSE), pp. 80–89. ACM (2004)
13. Zhu, Q., Clark, A., Poovendran, R., Başar, T.: Deceptive routing games. To appear in Proceedings of IEEE Conference on Decision and Control (CDC), Maui, Hawaii (2012)

A Game-Theoretic Framework for Network Security Vulnerability Assessment and Mitigation

Assane Gueye and Vladimir Marbukh

National Institute of Standards and Technology (NIST)

Abstract. In this paper we propose and discuss a game-theoretic framework for (a) evaluating security vulnerability, (b) quantifying the corresponding Pareto optimal vulnerability/cost tradeoff, and (c) identifying the optimal operating point on this Pareto optimal frontier. We discuss our framework in the context of a flow-level model of Supply-Demand (S-D) network where we assume a sophisticated attacker attempting to disrupt the network flow. The vulnerability metric is determined by the Nash equilibrium payoff of the corresponding game. The vulnerability/cost tradeoff is derived by assuming that "the network" can reduce the security vulnerability at the cost of using more expensive flows and the optimal operating point is determined by "the network" preferences with respect to vulnerability and cost. We illustrate the proposed framework on examples through numerical investigations.

1 Introduction

Since achieving complete security is typically an unattainable task, a realistic approach to survival is effective security vulnerability (risk) management. Effective security vulnerability management schemes should be able to (a) quantify security vulnerability and cost of security, (b) determine the set of feasible (vulnerability, cost) operating points and the corresponding (Pareto) optimal frontier representing the best achievable vulnerability/cost tradeoff, and (c) identify, given specific user security and cost preferences, the optimal operating point on this tradeoff curve.

The challenge in determining such schemes resides in the difficulty of estimating the security risk posed by a strategic adversary attempting to exploit system vulnerabilities as opposed to conventional risk management situations of reliability or fault tolerance models which are based on assumption of random failures with predetermined probabilities. This paper attempts to address this issue by proposing and discussing a game theoretic framework. Employing game theory allows us to capture the strategic nature of all parties (attackers and defenders). We illustrate our approach by considering the security vulnerability problem in a flow-level model of Supply-Demand (S-D) network.

In the proposed approach, we model "conceptual" game(s) between a network manager/operator (defender) and a strategic attacker. The network manager's

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 186–200, 2012.

goal is to insure uninterrupted transport of goods by choosing a feasible flow and the adversary attempts to disrupt the flow by attacking a link. We model this situation as a 2-player game and use the attacker's Nash equilibrium payoff to define a vulnerability metric.

We combine this vulnerability metric with the fact that each feasible flow has a (different) cost to derive the vulnerability/cost tradeoff. For that we assume that the network manager can reduce vulnerability at the cost of using more expensive flows. The maximum vulnerability corresponds to the case where the manager can choose only minimum cost flows (MCF). The minimum vulnerability is achieved when the network operator can choose among all feasible flows (i.e., even the most expensive ones). We derive the vulnerability/cost tradeoff by considering all 'costs' in between.

The vulnerability/cost of security tradeoff curve is the frontier separating the feasible region of (vulnerability, cost) pairs to the infeasible region. Once this frontier is drawn, the next question is finding the optimal operating point. The optimal operating point depends on the "network" utility function which specifies the "network" preferences with respect to vulnerability and cost. Using an illustrative example, we show how this optimal point can be computed for a given S-D network.

Related Work. The problem of security cost/benefit tradeoff has previously been considered in the literature. Gordon *et.al.* [4] use a version of *ALE* (annual loss expectation) to propose an economic model that determines the optimal amount to invest in security. The paper [12] by Tiwari and Karlapalem studies cost/benefit tradeoffs for information security assurance in terms of the defender's *investment* as well as the attacker's *opportunity*. The paper by Alexander J. McNeil [10] discusses a risk measurement model based on *extreme value theory* (EVT). Extreme events occur when a risk takes values from the tail of its probability distribution: i.e., rare events. All these approaches assume that failures are due to random events (faults) and according to a predetermined probability distribution. This assumption is justified in situations where failures occur because of natural disaster, machine breakdown, human error etc. However, when failures are due to the action of a strategic adversary, this assumption is no longer appropriate. In the present paper, we use game theory to model the strategic nature of both the attacker and the defender. In our framework, failure probabilities are derived from the attacker's Nash equilibrium strategy.

Attempts to quantify security vulnerability also include the NIST Common Vulnerability Scoring System (CVSS)[11]. The CVSS is an *expert's opinion*-based system that gathers scores for different aspects of security, quantifies the scores, and combines them in an equation that outputs a metric for vulnerability. Other attempts to measure vulnerability are by Symantec, McAfee, IBM, and Microsoft. Although all these reports provide some ideas about security vulnerability, they are all subjective and often lack solid (first principle-based) ground. The game theoretic approach proposed in this paper provides a principled and analytical way to analyze vulnerability.

In a very recent paper, Anderson *et. al.* [1] have presented a framework for a systematic study and analysis of the costs of cybercrime. They classify the costs of cybercrime into *direct losses, indirect losses, and defense cost*. Direct losses quantify the losses, damage, or user suffering felt by the victim as a consequence of an attack. Direct losses also include the attack reward obtained by the criminal. Indirect losses measure the effects of attacks on reputation, consumer trust, missed business opportunity etc. Defense cost is the monetary equivalent of prevention efforts. Put in their framework, our quantification of vulnerability reflects both direct losses (the loss seen by the network manager when a link is successfully attacked) as well as criminal's revenue (the willingness of an attacker to attack a link).

This paper is organized as follow. The next section presents our game theoretic framework to analyze vulnerability. We discuss our framework in the context of supply-demand (S-D) network which we introduce in subsection 2.1. Then, we present our assessment of vulnerability and our derivation of the vulnerability/cost tradeoff in subsection 2.2. The game theoretic model and the analysis of its Nash equilibrium are respectively introduced in subsections 2.3 and 2.4. We discuss the implications of our framework in section 3. The paper ends with concluding remarks in section 4.

2 Game Theoretic Framework

It is widely known that security is not free. A minimal effort in security results in an unacceptably high vulnerability. This is very well understood and it explains the billions of dollars spent every year on prevention and protection of systems. On the other hand, there is no such thing as absolute security. "*We have to build our systems on the assumption that adversaries will get in*" as put by Debora Plunkett, head of the NSA's Information Assurance Directorate. Furthermore, independently of the amount of effort spent, one can never guarantee complete security. In this situation, the real challenge is to determine *how much effort is needed to achieve an adequate level of security?*

To answer to this question, security experts must derive effective security vulnerability/risk management schemes that are able to quantify security vulnerability and the cost of security and determine the interplay between the two: i.e., the vulnerability/cost of security tradeoff. Once this tradeoff curve is drawn, and given the vulnerability/cost preferences of the system under consideration, one can compute the optimal operating point on that curve.

Next, we propose and discuss a game theoretic framework for security vulnerability assessment and mitigation. We first propose a quantification of the cost of security (or direct losses using the terminology defined in [1]), then, solving an *imaginary* 2-player between the defender of the system and the attacker, we derive a metric for security vulnerability, finally, by combining the two, we derive the vulnerability/cost of security tradeoff. We then use an illustrative example to show how to compute the optimal operating point.

The framework considered here applies to the generic security/availability problem discussed in [6] under the notion of *Blocking Games*. The notion of

blocking games has been used in [8], [7] and [9] in a situation where the defender chooses a spanning tree and the attacker picks a link. In this paper we use the results of blocking games to develop a framework for analyzing security vulnerability/cost tradeoff in the particular context of supply-demand (S-D) networks. The next subsection is an introduction on S-D networks.

2.1 Supply-Demand Networks [2]

We assume that the topology of a supply-demand network is given by a directed graph $G = (\mathcal{V}, \mathcal{A})$, with $|\mathcal{A}| = m$ is the cardinality of $\mathcal{A}$. Links (edges) are considered to be able to carry goods. We use the notation $a = (x, y)$ to designate the directed link (x, y). When the end nodes x and y need to be specified, we use (x, y) for the link, otherwise, we use the notation $'a'$ to designate the link.

Let some nonempty subset $S \subseteq \mathcal{V}$ be the "source" nodes, and some nonempty subset $T \subseteq \mathcal{V}$ be considered as "terminal" nodes, where $S \cap T = \emptyset$. With each node $x \in S$ we associate a nonnegative number $s(x)$, the "supply" at x, and with each node $x \in T$ we associate a nonnegative number $d(x)$, the "demand" at x. Throughout the paper, we assume, without any loss of generality, that the total demand is equal to the total supply

$$\sum_{x \in S} s(x) = \sum_{x \in T} d(x) = \Delta. \tag{1}$$

In general, each link a is associated with some *capacity* $c(a)$ which corresponds to the maximum amount of goods that can be carried through a. By *un-capacitated* network, we mean one for which $c(a) = \infty$ for all links a. A *capacitated* network is one where links have finite capacity.

Definition 1. *A* feasible flow *for this network is a function $f : \mathcal{A} \to \mathbb{R}_+$ that associates to each edge $a = (x, y) \in \mathcal{A}$ a nonnegative number $f(x, y) \geq 0$ verifying the following:*

$$f(x, \mathcal{V}) - f(\mathcal{V}, x) = s(x) \qquad \text{for all} \quad x \in S \tag{2}$$
$$f(\mathcal{V}, x) - f(x, \mathcal{V}) = d(x) \qquad \text{for all} \quad x \in T \tag{3}$$
$$f(x, \mathcal{V}) - f(\mathcal{V}, x) = 0 \qquad \text{for all} \quad x \notin S \cup T \tag{4}$$
$$f(x, y) \leq c(x, y) \qquad \text{for all} \quad (x, y) \in \mathcal{A}, \tag{5}$$

In other terms, a feasible flow is an assignment of values to the links that satisfies the conservation of flows at each node and the capacity constraint at each link.

Throughout, we use the following notations for arbitrary $X \subseteq \mathcal{V}$ and $Y \subseteq \mathcal{V}$:

$$f(x, \mathcal{V}) = \sum_{\{y \in N | (x,y) \in \mathcal{A}\}} f(x, y), \qquad (X, Y) = \{(x, y) \in \mathcal{A} | x \in X,\ y \in Y\}$$
$$g(X, Y) = \sum_{(x,y) \in (X,Y)} g(x, y), \quad \text{and} \qquad h(X) = \sum_{x \in X} h(x).$$

Remark 1. In this paper, all data (i.e. supplies and demands) are assumed to be integers. We are interested in the finite list of all integral feasible flows which we denote $\mathcal{F}$. We use $f = [f(a_1), f(a_2) \ldots, f(a_m)]$ to denote a generic feasible flow. In general, there is an exponential number of flows; and in most cases an exhaustive search is needed to list all feasible flows. Later we will see that to compute the minimum vulnerability (metric) introduced in this paper, one does not need to list all feasible flows.

In this paper, we assuming that all feasible flows are computed and we (abusively) use the same $\mathcal{F}$ to denote the flow-link matrix whose rows are indexed by feasible flows f and whose columns are indexed by the links a of the network, with $\mathcal{F}[f, a] = f(a)$: the amount that flow f assigns to link a. This matrix will serve as a payoff matrix for the *quasi-zero-sum* game defined later.

2.2 Security Cost and Vulnerability/Cost Tradeoff

In general, each link $a = (x, y) \in \mathcal{A}$ of the network is associated with a given cost that the network manager incurs by sending a unit of goods through $a = (x, y)$. This cost can be thought of as the delay associated with the link, the distance between the two ends, the operation/maintenance cost, or in general the total effort needed to move a unit of good from node $x \in \mathcal{V}$ to node $y \in \mathcal{V}$.

Letting $w(a)$ be the cost of sending a unit of goods through link a and $f(a)$ the amount of goods that flow f carries over a, $f(a)w(a)$ is the total cost of flow f associated with link a. The total cost of flow f can then be written as

$$w(f) = \sum_{a \in \mathcal{A}} f(a)w(a). \tag{6}$$

We assume throughout this paper that the costs $w(a)$ are fixed and given.

In a non-adversarial environment, the network operator/manager would choose a feasible flow of minimum cost to operate the network. In an adversarial environment where an attacker strategically chooses the edge to attack, it is no longer obvious how the network manager should choose a feasible flow. Indeed, if the network manager were to always choose the minimum cost feasible flow (MCF) (assuming that it is unique[⋆]), the attacker could target one link of this MCF to disrupt the transport. Hence, such choice could result to maximally vulnerable transport infrastructure. On the other hand, if the manager chooses randomly among a set of feasible flows, an attack becomes less likely to succeed: i.e., the network is less vulnerable to attacks. However, choosing in a bigger set of feasible flows implies additional cost to the network manager. We set this cost as a proxy for the cost of security and use it to quantify the vulnerability/cost of security tradeoff.

To quantify such tradeoff, we proceed as follows. We assume that the network manager has a "maximum cost" b that he can afford: i.e the network operator can choose any feasible flow with total cost $w(f) \leq b$; where $\min_f (w(f)) \leq b \leq \max_f (w(f))$. For instance, if $b = \min_f (w(f))$ (the minimum cost of a feasible flow), the network manager can only choose a minimum cost feasible flow (MCF)

[⋆] If there are more than one MCF, the attacker can still launch a very targeted attack.

and $b = \max_f (w(f))$ corresponds to the case where the operator can randomly choose among all feasible flows. We let $\mathcal{F}^{(b)} = \{f \ : \ w(f) \leq b\}$ and we (abusively) use $\mathcal{F}^{(b)}$ to also denote the matrix whose rows correspond to $f \in \mathcal{F}^{(b)}$.

For each maximum cost b, we setup a (conceptual) 2-player game between the network manager and a strategic adversary, where the manager chooses a feasible flow from $\mathcal{F}^{(b)}$ to operate the network, while the attacker targets a link. The details of the game are described in the next subsection. We use the "value" of the game to define a metric for *vulnerability to attack (VtA)* associated with b (in Section 2.4) and (numerically) analyze the VtA as a function of b.

When $b = \max_f (w(f))$, a closed-form characterization of the (minimum achievable) VtA exists and is provided in Section 3.2 for both un-capacitated and capacitated networks. For general value of the maximum cost b, such closed-form characterization is difficult to obtain. In this case, one can use tools such as the Gambit solver [3] in order to solve the game and compute the VtA.

2.3 Game Model

For each value of the maximum cost b, we setup an *imaginary* game between a "defender" (the network manager) and an attacker. The network manager chooses a feasible flow from the collection $\mathcal{F}^{(b)} = \{f \ : \ w(f) \leq b\}$ to move a total of Δ units of goods from set S to set T. The attacker wants to prevent the maximum amount of goods to reach the terminals by selecting an link to attack. When link a is successfully attacked, the amount of goods it carries ($f(a)$) is lost (by the defender). The attacker pays a cost $\boldsymbol{\mu}(a)$ to successfully disrupt the flow on link a. She also has the option of not attacking. Hence, if flow $f \in \mathcal{F}^{(b)}$ is selected by the defender and link a is attacked, the defender loses $f(a)$ and the attacker gets a net attack gain of $f(a) - \boldsymbol{\mu}(a)$. If the attacker decides to not launch an attack, there is no gain to her and no loss to the defender.

We model this interaction as a 2-player game and assume the *idealized*** case where all the information about the game is known to all players– the network topology, the amount of goods to be moved Δ, the costs of operation/maintenance $w(f)$, and the costs of attack $c(a)$. We are mainly interested in mixed strategy equilibria where the defender chooses a distribution $\{\boldsymbol{\alpha} \in \mathbb{R}_+^N \mid \sum_{f \in \mathcal{F}^{(b)}} \boldsymbol{\alpha}(f) = 1\}$ over the collection of feasible flows $\mathcal{F}^{(b)}$, while the attacker picks a distribution $\{\boldsymbol{\beta} \in \mathbb{R}_+^m \mid \sum_{a \in \mathcal{A}} \boldsymbol{\beta}(a) = 1\}$ over the set of links $\mathcal{A}$. The defender wants to minimize the expected loss $L^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and the attacker wants to maximize $\max(0, R^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta}))$, where $R^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is her expected net gain. $L^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $R^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ are defined below.

$$L^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{f \in \mathcal{F}^{(b)}} \boldsymbol{\alpha}(f) \sum_{f \in \mathcal{A}} \boldsymbol{\beta}(a) f(a), \tag{7}$$

$$R^{(b)}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{f \in \mathcal{A}} \boldsymbol{\beta}(a) \left(\sum_{f \in \mathcal{F}^{(b)}} \boldsymbol{\alpha}(f) f(a) - \boldsymbol{\mu}(a) \right). \tag{8}$$

** A more realistic model assumes limited knowledge for both players. Although the analysis will be more involved, the same framework can be applied.

We assume that if the attacker decides to not launch an attack, she chooses an *imaginary* link $a_\emptyset$ with probability $\boldsymbol{\beta}(a_\emptyset) = 1$, and any other *real* link with probability $\boldsymbol{\beta}(a) = 0$.

Remark 2. – Notice that the maximum cost 'b' is used to "parameterize" the games: for each b, there is a different game. We are interested in analyzing the network's *vulnerability to attack (VtA)* (introduced in the next section and denoted as $\theta^*(b)$) as a function of b. We particularly discuss the case $b = \max_f(w(f))$ (when all feasible flows can be chosen) which corresponds to the minimum achievable VtA.

- The operation/maintenance costs ($w(a)$) are chosen *once* and *fixed* in the entire paper. As a consequence, the costs of flows ($w(f)$) are fixed. With this, the collections $\mathcal{F}^{(b)}$ are well defined and form an increasing sequence (as b increases).
- The reader should be advised that the use of Game Theory in this paper is not meant to capture the actual *active* interaction between a defender who "dynamically" chooses a feasible flow and an attacker who "dynamically" tries to disrupt the transport of goods. Game Theory is rather used here as a modeling tool to study network vulnerability in *adversarial* environment.

2.4 Nash Equilibrium Theorem

The Nash equilibrium theorem was established in Gueye *et. al.* [5, Chap. 4] using the theory of Blocking Pairs of Polyhedra. In this paper, we consider polyhedra (introduced shortly) associated with integer flows and, hence, reduce the discussion of the Nash equilibrium theorem below to the context of feasible flows.

Recall that $\mathcal{F}^{(b)}$ is used to denote both set $\mathcal{F}^{(b)} = \{f_1, f_2, ..., f_{k^{(b)}}\}$ as well as the matrix whose rows correspond to $f_i,\ i = 1, \ldots, k^{(b)}$, $f_i = [f_i(a_1), \ldots, f_i(a_m)]$. Here, $k^{(b)}$ denote the cardinality of $\mathcal{F}^{(b)}$. From now on, we mainly consider the matrix interpretation. The *flow polyhedron* $P_{\mathcal{F}^{(b)}}$ associated with $\mathcal{F}^{(b)}$ is defined as the vector sum of the convex hull of the rows $(f_1, f_2, \ldots, f_{k^{(b)}})$ of $\mathcal{F}^{(b)}$ and the nonnegative orthant:

$$P_{\mathcal{F}^{(b)}} = \text{conv.hull}\,(f_1, f_2, \ldots, f_{k^{(b)}}) + \mathbb{R}^m_+. \tag{9}$$

The *blocker* $bl(P_{\mathcal{F}^{(b)}})$ of the flow polyhedron $P_{\mathcal{F}^{(b)}}$ is the polyhedron defined as:

$$bl(P_{\mathcal{F}^{(b)}}) = \left\{ \mathbf{y} \in \mathbb{R}^m_+ : \sum_{f \in \mathcal{A}} \mathbf{x}(a)\mathbf{y}(a) \geq 1 \ \forall\ \mathbf{x} \in P_{\mathcal{F}^{(b)}} \right\}. \tag{10}$$

Now, let $\boldsymbol{\omega}$ be a vertex (i.e., an extreme point) of $bl(P_{\mathcal{F}^{(b)}})$. We write $\boldsymbol{\omega} = (\boldsymbol{\omega}(a), a \in \mathcal{A})$ and let $\boldsymbol{\omega}(\mathcal{A}) = \sum_{f \in \mathcal{A}} \boldsymbol{\omega}(a)$. Note that $\boldsymbol{\omega}(a) \geq 0$ for all $a \in \mathcal{A}$ and $\boldsymbol{\omega}(\mathcal{A}) > 0$[***]; so that $\boldsymbol{\beta}_{\boldsymbol{\omega}} = (\frac{\boldsymbol{\omega}(a)}{\boldsymbol{\omega}(\mathcal{A})}, a \in \mathcal{A})$ is a probability distribution on $\mathcal{A}$. We call it the probability distribution associated to $\boldsymbol{\omega}$. Finally, let us define $\theta^{(b)}(\boldsymbol{\omega})$ as

[***] This is because the blocker $bl(P_{\mathcal{F}^{(b)}})$ is not empty, and does not contain the all-zero vector–the origin ($P_{\mathcal{F}^{(b)}}$ is not empty).

$$\theta^{(b)}(\boldsymbol{\omega}) := \frac{1}{\boldsymbol{\omega}(\mathcal{A})}\left(1 - \sum_{f\in\mathcal{A}} \boldsymbol{\omega}(a)\boldsymbol{\mu}(a)\right). \tag{11}$$

$\theta^{(b)}(\boldsymbol{\omega})$ is the expected attack reward associated with $\boldsymbol{\omega}$ if the attacker were to choose a link to attack according to the distribution $\boldsymbol{\beta} = (\frac{\boldsymbol{\omega}(a)}{\boldsymbol{\omega}(\mathcal{A})}, a \in \mathcal{A})$. $\frac{1}{\boldsymbol{\omega}(\mathcal{A})}$ is the loss seen by the defender.

We call the vertex $\boldsymbol{\omega}$ *critical* if

$$\theta^{(b)}(\boldsymbol{\omega}) = \underline{\theta^*(b)} := \max_{\tilde{\boldsymbol{\omega}}\in bl(P_{\mathcal{F}^{(b)}})} \left(\theta^{(b)}(\tilde{\boldsymbol{\omega}})\right) . \tag{12}$$

We call $\theta^*(b)$ the network's *vulnerability to attack (VtA)* associated with the maximum cost b. We discuss this choice of vulnerability metric in Section 3.1. In the context of the S-D network considered in this paper, the entries of a vextex $\boldsymbol{\omega}$ are indexed by the links of the network. The support of a vector $\boldsymbol{\omega}$ is the set of indices (i.e., links) a for which $\boldsymbol{\omega}(a) > 0$. The support of critical vertex is said to form a *critical subset* of links.

The Nash equilibrium theorem [5, Chap. 4] gives a characterization of the players' strategies and the attacker's maximum net attack gain $\theta^*(b)$ in any Nash equilibrium.

Theorem 1 (Gueye *et. al.* 2011).

1. *If the maximum gain is negative ($\theta^*(b) < 0$), the attacker will not launch an attack and the defender randomly chooses a feasible flow according to a distribution $\boldsymbol{\alpha}^{(b)*}$ that satisfies*

$$\bar{\boldsymbol{\alpha}}^{(b)*}(a) := \sum_{f\in\mathcal{F}^{(b)}} f(a)\boldsymbol{\alpha}^{(b)*}(f) \leq \boldsymbol{\mu}(a). \tag{13}$$

2. *If the gain is nonnegative ($\theta^*(b) \geq 0$), an equilibrium strategy for the attacker is to always launch an attack that focuses only on edges belonging to critical subsets. Her randomized strategy is a convex combination of the probability distributions induced by the critical vertices as*

$$\boldsymbol{\beta}^{(b)*}(a) = \sum_{\boldsymbol{\omega}\in\mathcal{C}} \pi_{\boldsymbol{\omega}}\boldsymbol{\beta}_{\boldsymbol{\omega}}(a); \tag{14}$$

where each $\boldsymbol{\omega} \in \mathcal{C}$ is a critical vertex, $\pi_{\boldsymbol{\omega}} \geq 0$ and $\sum_{\boldsymbol{\omega}\in\mathcal{C}} \pi_{\boldsymbol{\omega}} = 1$. The defender's equilibrium is such that:

$$\begin{cases} \bar{\boldsymbol{\alpha}}^{(b)*}(a) - \boldsymbol{\mu}(a) = \theta^*(b) & \text{for all } \ a \in \mathcal{A} \ \text{ such that } \ \boldsymbol{\beta}^{(b)*}(a) > 0. \\ \bar{\boldsymbol{\alpha}}^{(b)*}(a) - \boldsymbol{\mu}(a) \leq \theta^*(b) & \text{for all } \ a \in \mathcal{A}. \end{cases} \tag{15}$$

In every Nash equilibrium of the game, the attacker's expected net attack gain achieves the maximum of $\theta^*(b)$, and the defender's expected loss has the form $\sum_{\boldsymbol{\omega}\in\mathcal{C}} \pi_{\boldsymbol{\omega}}/\boldsymbol{\omega}(\mathcal{A})$, for the same π introduced above.

3. *If the attack cost $\boldsymbol{\mu} = \mathbf{0}$, any equilibrium strategy for the attacker can be written as a convex combination of some $\boldsymbol{\beta}_{\boldsymbol{\omega}}$'s where each $\boldsymbol{\omega} \in \mathcal{C}$ is a critical vertex and the defender's equilibrium strategies verify (15) (with $\boldsymbol{\mu} = \mathbf{0}$).*

3 Discussions

The implications of the NE theorem are discussed in this section. The vulnerability to attack (VtA) as well as the attacker and defender's strategies are analyzed in subsection 3.1 then we discuss the minimum achievable vulnerability of the network by considering the particular case of $b = \max_f (w(f))$ in subsection 3.2. In subsection 3.3 we use the VtA metric to study the vulnerability vs cost of security tradeoff.

3.1 Vulnerability to Attack (VtA) and Critical Subsets of Links

The vulnerability metric $(\theta^*(b))$ proposed in this paper *reflects both the loss seen by the network manager when a link fails (due to attack) as well as the willingness of an attacker to attack a link (i.e., the cost of attacking a link)* (see equation (11)). This is a desirable feature for a vulnerability metric because no rational adversary will launch an attack if the expected net attack reward is less than zero. On the other hand, links with high loss (i.e., high volume of traffic) and low cost of attack are very attractive to adversaries.

Also, $\theta^*(b)$ is maximized (and is the same) at any equilibrium of the game (in general different Nash equilibria might have different payoffs for a given player). This implies that the vulnerability metric is *uniquely* defined once the parameters of the games are set. Furthermore $\theta^*(b)$ is closely dependent to the parameters of the network. $\theta^*(b)$ is derived from a vertex of the blocker polyhedron $(bl(P_{\mathcal{F}^b}))$, which is solely dependent on the topology of the network and the amount of goods to move from the sources to the terminals (and of course on the maximum cost b and the costs of attack $\boldsymbol{\mu}$).

It is interesting to make the "distinction" between the loss seen by the defender when a link is attacked $(\boldsymbol{\alpha}^{(b)*}(a))$ and the link's *criticality* $(\bar{\boldsymbol{\alpha}}^{(b)*}(a) - \boldsymbol{\mu}(a))$. Once the defender chooses a particular flow f, the loss he sees whenever a link a fails is equal to the amount of goods that flow f carries over the path containing a. The defender chooses a flow such that the amount of goods carried over any critical link is minimized (as we will see later). The *criticality* of a link indicates the net gain an attacker receives by attacking the link; hence, how much the link is attractive to the attacker. It depends not only on the *loss* of a link, but also on the cost of attacking the link. *The vulnerability metric* $\theta^*(b)$ *corresponds to the* criticality *of the most critical links.*

In order to achieve such maximum vulnerability, the attacker has to *focus only on links that are critical*, according to the strategies given by equation (14). Notice that, as for the vulnerability metric, the attacker's strategy is closely dependent to the parameters of the network. *This indicates that a sophisticated attacker would analyze the topology of the network to decide which links to attack.* This contrasts with conventional reliability models where the failure probability of a link is chosen without any consideration of the structure of the graph.

The defender's equilibrium strategy $\boldsymbol{\alpha}^{(b)}$ can be interpreted as the *best way to choose a feasible flow in the presence of a strategic adversary.* In fact, as a best response to the attacker's strategy, $\boldsymbol{\alpha}^{(b)}$ minimizes the *overall* expected loss. Each

entry $\boldsymbol{\alpha}^{(b)}(f)$ of the distribution vector is an indication about the potential loss associated to using flow f– whenever $\boldsymbol{\alpha}^{(b)}(f) = 0$ choosing feasible flow f implies high expected loss due to an attack. Since $\boldsymbol{\alpha}^{(b)}$ is a best response to the attacker's strategy, all flows f with $\boldsymbol{\alpha}^{(b)}(f) > 0$ have the same (minimum) expected loss.

When there is no attack cost, the probability distribution $\boldsymbol{\alpha}$ is such that the *links with highest overall expected loss correspond to the most critical ones.* When attacking requires a relatively substantial effort the maximum expected net attack reward can be negative $\theta^*(b) < 0$. In this case the defender chooses the distribution $\boldsymbol{\alpha}$ such that the attacker has no incentive to attack. *Such a choice can be seen as a deterrence tactic for the defender.*

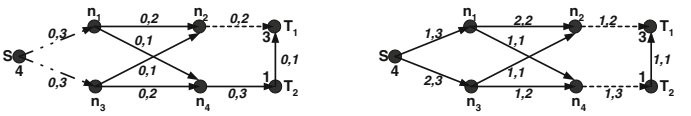
3.2 Minimum Vulnerability

In this section we assume that the defender's maximum cost $b = \max_f (w(f))$ (so that he can choose among all feasible flows) and illustrate the NE theorem for both un-capacitated and capacitated networks. In this case, we can give closed-form characterizations for $\boldsymbol{\omega}$, $\boldsymbol{\beta_\omega}$, $\theta(\boldsymbol{\omega})$, and $\theta^*(\max_f (w(f)))$ (which we just denote θ^*). Notice that $b = \max_f (w(f))$ corresponds to the minimum achievable VtA of the network (the network operator can use all resources available to him).

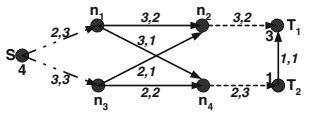
The following theorem by Fulkerson and Weinberger [2] describes the flow polyhedron $P_\mathcal{F}$ and characterizes the vertices of its blocker $bl(P_\mathcal{F})$ in the case when $b = \max_f (w(f))$.

Theorem 2 (Fulkerson and Weinberger [2]). *Let $\mathcal{F}$ be the matrix of integral feasible flows in a capacitated S-D network $G = (\mathcal{V}, \mathcal{A})$ with integral-valued supply, demand and capacity functions, respectively $s(\cdot)$, $d(\cdot)$, and $c(\cdot)$. Then the polyhedron $P_\mathcal{F}$ is described by*

$$P_\mathcal{F} = \Bigg\{ \mathbf{x} \in \mathbb{R}_+^{|\mathcal{A}|} \mid \sum_{a \in F \subseteq (X, \bar{X})} \mathbf{x}(a) \geq d(\bar{X}) - s(\bar{X}) - c(\bar{F}), \text{ for all } X \subseteq \mathcal{V} \text{ and any } F \subseteq (X, \bar{X}) \text{ such that } d(\bar{X}) - s(\bar{X}) - c(\bar{F}) > 0 \Bigg\}. \tag{16}$$



(a) $c < \infty$, $\boldsymbol{\mu} = \mathbf{0}$ (b) $c < \infty$, "low" $\boldsymbol{\mu} > \mathbf{0}$



(c) $c < \infty$, "high" cost $\boldsymbol{\mu} > \mathbf{0}$

Fig. 1. Example of S-D network with different attack cost

$\bar{F}$ is the complement of F in $(X, \bar{X})$ (the set of edges from X to $\bar{X}$).

The vertices of the blocker $\mathrm{bl}(P_{\mathcal{F}})$ *are given by the essential vectors (i.e., vectors that do not dominate a convex combination of the others) of the set of* $\{\boldsymbol{\omega}_{X,F}\}_{X\subseteq\mathcal{V}, F\subseteq(X,\bar{X})}$ *defined by the pairs* $\big((X,\bar{X}),F\big)$*, for every* $X \subseteq \mathcal{V}$ *and every* $F \subseteq (X,\bar{X})$ *verifying* $d(\bar{X}) - s(\bar{X}) - c(\bar{F}) > 0$*, as follow:*

$$\boldsymbol{\omega}_{X,F}(a) = \frac{1}{d(\bar{X}) - s(\bar{X}) - c(\bar{F})}\mathbf{1}_{a\in F}. \tag{17}$$

The theorem indicates that vertices of the blocker polyhedron correspond to pairs $((X,\bar{X}),F)$ where $X \subseteq \mathcal{V}$ is a cut-set of the graph of the network, and $F \subseteq (X,\bar{X})$. More precisely, they correspond to pairs that verify the "excess demand property": $d(\bar{X}) - s(\bar{X}) - c(\bar{F}) > 0$.

The quantity $d(\bar{X}) - s(\bar{X}) - c(\bar{F})$ can be interpreted as follow. $d(\bar{X}) - s(\bar{X})$ is the excess demand in $\bar{X}$ that every feasible flow has to compensate. This compensation can be done using links in F and in $\bar{F}$, for any $F \in (X,\bar{X})$. If each link $a \in \bar{F}$ carries its maximum possible flow $(c(a))$ and there is still a remaining deficit $(d(\bar{X}) - s(\bar{X}) - c(\bar{F}))$, then links in F have to be used to compensate this remaining deficit. Any feasible flow should send over the links in F an amount of flow at least equal to the deficit $d(\bar{X}) - s(\bar{X}) - c(\bar{F})$.

Remark 3. Notice that the theorem describes the flow polyhedron and its blocker for general capacitated network. When the network is un-capacitated (i.e., $c(a) = \infty$ for all links) the condition $d(\bar{X}) - s(\bar{X}) - c(\bar{F}) > 0$ is satisfied only when $F = (X,\bar{X})$, implying $\bar{F} = \emptyset$ and $c(\bar{F}) = 0$. The excess demand property also becomes $d(\bar{X}) - s(\bar{X}) \geq 1$ (because we have integer flows). In this case the discussion below can be repeated for $F = (X,\bar{X})$.

From (17), we have that

$$\boldsymbol{\omega}_{X,F}(\mathcal{A}) = \sum_{a\in\mathcal{A}} \boldsymbol{\omega}_{X,F}(a) = \frac{|F|}{d(\bar{X}) - s(\bar{X}) - c(\bar{F})}. \tag{18}$$

The distribution associated with the pair $((X,\bar{X}),F)$ (via $\boldsymbol{\omega}_{X,F}$) is given by

$$\beta_{X,F}(a) = \frac{1}{|F|}\mathbf{1}_{a\in F}; \tag{19}$$

which is uniform over F. This implies that all links belonging to the same critical subset are attacked with the same probability (independently of the attack cost on each link). The expected attack reward $\theta(X,F)$ associated with X and F (defined in (11)) is equal to

$$\theta(X,F) = \frac{d(\bar{X}) - s(\bar{X}) - c(\bar{F}) - \boldsymbol{\mu}(F)}{|F|}. \tag{20}$$

The equation above is quite intuitive. In fact, each feasible flow has to compensate the excess demand in $\bar{X}$ by sending a total amount of $d(\bar{X}) - s(\bar{X}) - c(\bar{F})$

over the edges in $F \subseteq (X, \bar{X})$. By randomly attacking one of these links with the uniform probability $\boldsymbol{\beta}_{X,F}$ in (19), the expected reward for the attacker is $\left(d(\bar{X}) - s(\bar{X}) - c(\bar{F})\right)/|F|$ and the expected attack cost is equal to $\boldsymbol{\mu}(F)/|F|$. Hence, the quantity above represents the average net attack reward that the attacker gets *per link* of F.

A critical subset of links has the form $F \subseteq (X, \bar{X})$ where the pair $((X, \bar{X}), F)$ satisfies the excess demand property and achieves the maximum vulnerability to attack (VtA) given by

$$\theta^* = \max_{\substack{X \subseteq \mathcal{V},\ F \subseteq (X,\bar{X}):\\ d(\bar{X}) - s(\bar{X}) - c(\bar{F}) > 0}} \left(\frac{d(\bar{X}) - s(\bar{X}) - c(\bar{F}) - \boldsymbol{\mu}(F)}{|F|} \right). \tag{21}$$

Remark 4. For un-capacitated S-D network, the vulnerability to attack (VtA) can be simplified to

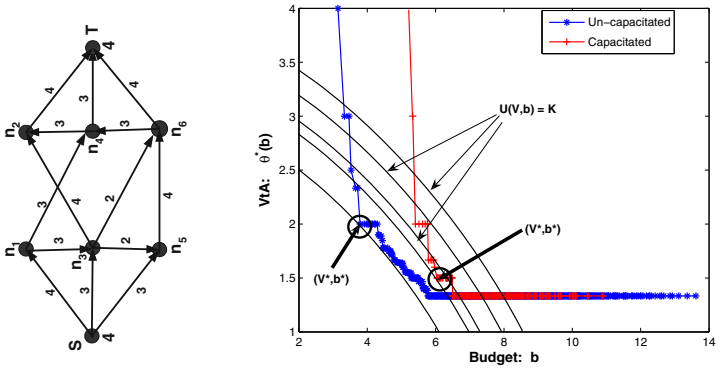
$$\theta^* = \max_{\substack{X \subseteq \mathcal{V},\\ d(\bar{X}) - s(\bar{X}) \geq 1}} \left(\frac{d(\bar{X}) - s(\bar{X}) - \boldsymbol{\mu}(X, \bar{X})}{|(X, \bar{X})|} \right). \tag{22}$$

Computing this VtA can be shown to be equivalent to a minimization of the form $\min_{X \subseteq \mathcal{V}} \left(\rho|(X, \bar{X})| + \boldsymbol{\mu}(X, \bar{X}) + g(X)\right)$, where $g(X) := d(X) - s(X)$. The function $g(\cdot)$ is a *modular*. Hence, using techniques of (sub)modular function minimization (as in [8, Section 4]), one can derive a polynomial algorithm to compute a critical subset. For the general capacitated network, the reduction to a (sub)modular function minimization is less obvious because the maximization is over the pairs $((X, \bar{X}), F)$. Authors are studying a generalization of the definition of submodular functions that can be applied to pairs.

Figures (1) show examples of networks with their minimum achievable vulnerability θ^* and the corresponding critical subsets (shown in dotted and dashed-dotted lines) for different attack cost vectors. The cost of attack and the capacity are shown by the number next to the link: the first number (left) is the attack cost and the second (right) the capacity. In example (1(a)), the costs of attacking the links are all equal to zero. There are two critical subsets of links. The first one (dashed-dotted line) corresponds links $\{(S, n_1), (S, n_2)\}$. The second one is the singleton (n_2, T_1). The corresponding VtA is $\theta^* = 2$. When the attacker targets critical subset (n_2, T_1), the attack is deterministic while an attack to the critical subset $\{(S, n_1), S, n_2)\}$ is randomized and uniform. In example (1(b)), there is a positive attack cost $\boldsymbol{\mu}$ that is *relatively* low. The VtA $\theta^* = 1$ is still positive and the attacker will uniformly target at random one of the critical links (n_2, T_1) or (n_4, T_2). Example (1(c)) is one where the attack costs are high enough to result to a negative VtA ($\theta^* = -0.5$). The figure shows the (critical) subsets that achieve this maximum (but negative) VtA. In this case, an attack will not be launched.

3.3 Vulnerability to Attack Cost of Security Tradeoff

In this subsection, we study the vulnerability/cost of security tradeoff. For that, we compute the vulnerability to attack (VtA) $\theta^*(b)$ for each value of the maximum



(a) S-D Network (b) VtA/Cost Tradeoff

Fig. 2. Vulnerability/Cost of security tradeoff and optimal operating points (V^*, b^*) for un-capacitated and capacitated S-D networks

cost b, $\min_f(w(f)) \leq b \leq \min_f(w(f))$. When $b = \max_f(w(f)))$, we have shown that the VtA ($\theta^* := \theta^*(\max_f(w(f)))$ can be characterized in closed-form. For a general $\mathcal{F}^{(b)}$, such characterization is very involved. In fact, a concise description of the polyhedron $P_{\mathcal{F}^{(b)}}$ and of the vertices of its blocker does not exist for an arbitrary collection $\mathcal{F}^{(b)}$ (to the best knowledge of the authors) . One can use techniques described in [13, Chap. II.1] to characterize $P_{\mathcal{F}^{(b)}}$ and identify the vertices of its blocker, or directly solve the game (using numerical methods). In this paper, we use the Gambit [3] solver to compute the value of $\theta^*(b)$ as a given b.

We illustrate our approach using the example of the S-D network depicted in Figure (2(a)). The amount of goods to be moved from the single source to the single destination is assumed to be equal to 4 (units of goods). We consider both an un-capacitated and a capacitated network (the links' capacities are given by the numbers next to the links). We consider the case of the most powerful attacker whose cost of attack is equal to zero ($\boldsymbol{\mu} = \mathbf{0}$). Figure (2) shows the vulnerability/cost of security tradeoff curves for the un-capacitated (star '*' curve) and the capacitated network (plus '+' curve).

The tradeoff curves show two distinct regions. Initially, the vulnerability to attack (VtA) $\theta^*(b)$ rapidly decreases as a maximum cost b increases. From this, we can infer that in this region *a small investment in randomness (i.e., security) has very high returns for the network manager.* This first region corresponds to a small interval of values of b; hence a small subset of feasible flows. Then, the curve settles at the minimum possible vulnerability: *once in this region security investment has very low returns.* This second region corresponds to a large interval of values of the maximum cost b (hence a large subset of feasible flows). These two observations imply that *to achieve the minimum possible vulnerability, the network manager has to randomly choose from a relatively small subset of feasible flows.* This is a very desirable feature because choosing from the set of all feasible flows–which is of exponential size–can be very demanding (both in computational time and in storage).

The vulnerability/cost of security tradeoff curve is the frontier that separates, for a given network, the feasible region $\mathcal{R}$ from the infeasible region $\bar{\mathcal{R}}$. Once it is determined, the next question is finding the optimal operating point on this frontier. Apparently, the optimal operating point depends on the specific "network" preferences with respect to the vulnerability $V = \theta^*(b)$ and maximum cost b. These preferences can be quantified by the "network" utility function $U(V, b)$. In general, the optimal operating point is determined by solving a 2-dimensional $\max\{U(V, b) : (V, b) \in \mathcal{R}\}$ optimization problem which, in this case, can be reduced to a one-dimensional optimization (because of $V = \theta^*(b)$), and can be written as $(V^*, b^*) = (\theta^*(b^*), b^*)$ where

$$b^* \in \underset{b:(\theta^*(b),b)\in\mathcal{R}}{\arg\min} U(\theta^*(b), b). \tag{23}$$

Figure (2(b)) shows the optimal operating points for the un-capacitated and capacitated networks assuming a network utility function of the form $U(V, b) = 2.6V^{1.4} + 0.01b^{1.4}$.

4 Conclusion

In this paper, we use a Game Theoretic approach to derive a vulnerability to attack metric for (un-capacitated and capacitated) supply-demand networks and use this metric to compute the vulnerability/cost of security tradeoff. The metric reflects both the loss seen by the network when a link fails (due to attack) as well as the willingness of an attacker to attack a link (i.e., the cost of attacking a link). It also can be used to determine the most critical links in the network. The vulnerability/cost of security tradeoff curve shows a first (relatively small) region with high returns in security investment, followed by a (relatively large) region where investment in security has very low returns. This curve is the frontier that separates the feasible region of (vulnerability,cost) pairs from the infeasible region. Once it is determined, the optimal operating point can be computed by considering the "network" utility function. In this paper, we illustrate this process using a numerical example.

References

1. Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M.J.G., Levi, M., Moore, T., Savage, S.: Measuring the Cost of Cybercrime. In: 11th Workshop on the Economics of Information Security (June 2012)
2. Fulkerson, D.R., Weinberger, D.B.: Blocking Pairs of Polyhedra Arising from Network Flows. Journal of Combinatorial Theory, Series B 18(3), 265–283 (1975)
3. Gambit. Game theory analysis software and tools @ONLINE (2002), `http://www.gambit-project.org/doc/index.html`
4. Gordon, L.A., Loeb, M.P.: The Economics of Information Security Investment. ACM Trans. Inf. Syst. Secur. 5(4), 438–457 (2002)

5. Gueye, A.: A Game Theoretical Approach to Communication Security. PhD dissertation, University of California, Berkeley, Electrical Engineering and Computer Sciences (March 2011)
6. Gueye, A., Lazska, A., Walrand, J., Anantharam, V.: A Polyhedral-Based Analysis of Nash Equilibrium of Quasi-Zero-Sum Games and its Applications to Communication Network Security. Symmetry – Special Issue: Polyhedra (submitted)
7. Gueye, A., Marbukh, V., Walrand, J.C.: Towards a Quantification of Communication Network Vulnerability to Attacks: A Game Theoretic Approach. In: 3rd International ICST Conference on Game Theory for Networks, Vancouver, Canada (May 2012)
8. Gueye, A., Walrand, J.C., Anantharam, V.: Design of Network Topology in an Adversarial Environment. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 1–20. Springer, Heidelberg (2010)
9. Laszka, A., Szeszlér, D., Buttyán, L.: Game-theoretic Robustness of Many-to-one Networks. In: 3rd International ICST Conference on Game Theory for Networks, Vancouver, Canada (May 2012)
10. Mcneil, E.J.: Extreme Value Theory for Risk Managers, pp. 93–113. RISK Books (1999)
11. Mell, P., Scarfone, K., Romanosky, S.: A Complete Guide to the Common Vulnerability Scoring System. In: NIST CVSS. National Institute of Standards and Technology (June 2007)
12. Tiwari, R.K., Karlapalem, K.: Cost Tradeoffs for Information Security Assurance. In: 4th Annual Workshop on the Economics of Information Security, WEIS, June 1-3. Harvard University, Cambridge (2005)
13. Wolsey, L.A., Nemhauser, G.L.: Integer and Combinatorial Optimization. Wiley-Interscience (November 1999)

Game Theoretic Model of Strategic Honeypot Selection in Computer Networks

Radek Píbil[1], Viliam Lisý[1], Christopher Kiekintveld[2], Branislav Bošanský[1], and Michal Pěchouček[1]

[1] Agent Technology Center,
Department of Computer Science and Engineering,
Faculty of Electrical Engineering, Czech Technical University in Prague,
Czech Republic
{radek.pibil,viliam.lisy,
branislav.bosansky,michal.pechoucek}@agents.fel.cvut.cz
[2] Department of Computer Science,
University of Texas at El Paso (UTEP), United States of America
cdkiekintveld@utep.edu

Abstract. A honeypot is a decoy computer system used in network security to waste the time and resources of attackers and to analyze their behaviors. While there has been significant research on how to design honeypot systems, less is known about how to use honeypots strategically in network defense. Based on formal deception games, we develop two game-theoretic models that provide insight into how valuable should honeypots look like to maximize the probability that a rational attacker will attack a honeypot. The first model captures a static situation and the second allows attackers to imperfectly probe some of the systems on the network to determine which ones are likely to be real systems (and not honeypots) before launching an attack. We formally analyze the properties of the optimal strategies in the games and provide linear programs for their computation. Finally, we present the optimal solutions for a set of instances of the games and evaluate their quality in comparison to several baselines.

Keywords: honeypots, game theory, network security, deception.

1 Introduction

Society increasingly depends on information technology and computer networks to deliver vital information and services. Protecting these systems and the information they contain is a growing priority, even as they become more attractive targets for criminal activity. Cybercriminals are highly motivated and devote large efforts to launching sophisticated attacks, requiring network administrators to adopt increasingly sophisticated countermeasures to protect their networks. Honeypots are one of these countermeasures that provide a unique set of benefits for network defense. Falling costs for deploying honeypots and improved virtualization technologies are likely to lead to increased use of honeypots, including systems with many honeypots on a single network.

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 201–220, 2012.

A honeypot is a computer system placed on a network explicitly in order to attract the attention of an attacker. It does not store any valuable data and it thoroughly logs everything that happens in the system. Honeypots help to increase the security of computer systems in two ways [1]: (1) The presence of honeypots wastes the attacker's time and resources. The effort an attacker spends to compromise a honeypot and learn that it does not contain any useful information directly takes away time and resources that could be used to compromise valuable servers. (2) Moreover, once the attacker compromises a honeypot, the network administrator can analyze all of the attacker's actions in great detail, and use the information obtained to better protect the network. For example, specific security holes used in an attack can be patched, and new attack signatures added to antivirus and intrusion detection systems. Attacks on honeypots can also serve as an "early warning" system for administrators, providing more time to react to attacks in progress.

For these reasons, the network administrators using honeypots try to maximize the probability that the attacker attacks a honeypot and not a real server. However, with an increasing use of this technology, attackers have started to consider the existence of honeypots during their attacks and take steps to avoid attacking them. For example, once they gain access to a system, they can use multitude of methods to probe the system and rule out the possibility that they are in a honeypot before they continue with their attack (e.g., [2]). To be effective against more sophisticated attackers, honeypots must be sufficiently disguised that they are not obvious (i.e., they cannot simply present the most vulnerable possible target). This leads us to analyze using honeypots from an adversarial perspective, where network administrators reason about the strategies of the attackers and vice versa.

Game theory is a formal framework developed to analyze interactions between multiple decision makers. In this paper, we present two novel game-theoretic models of adding honeypots to a network and the following target selection by the attacker. The first model combines a resource allocation game and a deception game, and is designed to answer basic question about how many honeypots a defender should use, and how they should be configured. In particular, we consider the possibility that honeypots can be configured to look like real targets of varying importance, offering new ways to deceive an attacker. The second model extends the first one to add the capability for an attacker to strategically probe targets before launching an attack to determine whether they are likely to be honeypots or real servers. Both models are formulated as zero-sum extensive-form imperfect-information games, and we provide linear programs for computing the optimal strategies of the players (i.e., the network administrator and the attacker) in both cases.

We solve the linear programs using a state-of-the-art optimization toolkit (*CPLEX*). This provides greater scalability than previous models [3] that were solved using Gambit [4], allowing us to analyze the models in greater detail. These previous models found simple uniform randomization strategies to be optimal for honeypot placement. However, our models show richer and more complex

strategies are necessary when we generalize the assumptions to include non-uniform server values and sophisticated attackers with probing capabilities. Our empirical evaluation shows that the game-theoretic strategies are significantly better in reducing the expected harm of the attacks and they allow using a larger numbers of honeypots more efficiently than two heuristic approaches. We also test our strategies against simple heuristic attackers, in addition to optimal ones. Based on the analysis of the optimal game-theoretic strategies, we provide recommendations to the network administrators applying honeypots in their networks.

The next section explains the relation of the presented research to the previous work. In Section 3, we introduce the basic model without probing, we analyze its properties and present the solution LP. In Section 4, we introduce the possibility of probing. The evaluation and analysis of the optimal strategies for a set of instances of the models is presented in Section 5 and we conclude the paper in Section 6.

2 Related Work

Many software packages for creating honeypots and analyzing attackers' behavior are available through the honeynet project website[1]. This paper does not focus on the technical aspects of creating honeypots, so we do not review this line of research here. An extensive introduction to the practices and technological challenges of applying honeypots is available in [1]. We focus our review on more closely related work that applies game theory to honeypots.

2.1 Honeypots and Game Theory

There are relatively few papers that explore using game theory for creating and deploying honeypots. They can be divided into two categories. One models the interaction within a honeypot during an ongoing attack. The other models the situation before the actual attack, when the attacker selects a target.

In the first category, game theory is used to optimize the information learned about the attacker's strategies by modeling the progress of the attack. In [5] the authors give the defender the option to block the action, or let it be executed, while the attacker can either retry, continue, or stop the attack. In [6] the defender models the attack as a movement on a graph and tries learn the attacker's strategy by making some of the graph nodes more desirable using simulated user activity.

The approach presented in this paper belongs to the second category, in which the game theory is used to optimize the probability that the attacker will attack a honeypot and not a real system. In [3], the authors model situations similar to the ones we model in this paper. However, their model is simpler and results in simple, uniform strategies. They analyze the problem of allocating the real servers and honeypots to the space of IP addresses. However, the attacker cannot distinguish between individual servers and honeypots, so the only meaningful

[1] www.honeynet.org

strategy the attacker can use is to attack a random server. Only if the defender gives the attacker some hint based on the address of the servers, e.g., by assigning the honeypots to the lowest IP addresses, a rational attacker can deviate from a random strategy. Therefore, a rational defender also allocates addresses randomly. In reality, however, not all computers in the network are identical to the attacker. In our model, we consider the importance of the computers, which make the optimal strategies non-trivial and much harder to compute.

In the second part of [3] as well as in [7], the authors give the attacker the option of probing the servers before the attack. The result of a probe is whether the server is real or a honeypot, but the authors assume that the result is fully determined by the defender and not the reality. This implies that the probe results are only useful if the defender voluntarily discloses some information to the attacker. A rational defender uses uniform random probe results and the attacker ignores them. A more realistic assumption is that the defender can successfully deceive the attacker only with certain probability. Otherwise, his probe will identify the real nature of the server. In this paper, we consider this generalization and it results to non-trivial strategies for both players.

2.2 Related Game Theoretic Models

A similar task to the honeypot selection is the deployment of false targets in warfare as studied in [8] (among many others). Targets are identically valued as in [3]. The defender selects the number of fake targets to deploy, for which he has to pay from a resource pool that he also has to use for the protection of genuine targets. The attacker chooses the number of targets to attack. However, the attacker is also limited by his own resource pool, and may possibly be able to attack only a single target as in our case. The paper focuses on a proper resource allocation between protection and defense, not the protection strategy, which is uniform. Our focus is on protection strategies taking values of targets into account.

The game theoretic models presented in this paper are a special case of imperfect-information extensive-form games (EFG) with chance nodes. The state-of-the-art algorithm for solving these games optimally is the mathematical program for sequence-form representation of the games [9]. More efficient algorithms can be found for sub-classes of EFGs with special structure. Two such subclasses are the Bayesian Stackelberg games [10] and signaling games [7]. As in our game models, these games include hidden information available only to one of the players, however, this information modifies only the payoffs of the players and not the applicable actions. In our games, the hidden information defines the applicable actions as well, which makes the techniques developed for Bayesian Stackelberg games inapplicable.

A less studied class of games that are most closely related to our models are deception games. A formal deception game was first formulated as an open problem in [11]. One player is given a vector of three random numbers from uniform distribution on unit interval. It changes one of the numbers to an arbitrary number from the interval and presents the modified vector to the second player. The second player chooses one position in the vector and receives as its reward

the number that was originally on that position. The open question stated in the paper is whether there is a better strategy than randomly choosing one of the positions. This question was answered in [12] and a few similar questions about various modifications of the model were published in the next years, but the results generally apply only to the specific game formulations and they do not present the complete strategies to play the game.

3 Honeypot Selection Game

The *Honeypot Selection Game* models a situation where an attacker is deciding which server in a computer network to attack. However, the administrator has added a set of honeypots to the network, and wants to configure them to maximize the probability that the attacker chooses to attack a honeypot rather than a real computer. There are two basic kinds of honeypots. A *low interaction* honeypot is relatively simple, and therefore it can be added to the network at a low cost [13], but even a simple probing by the attacker will reveal it is not a real system. A *high interaction* honeypot is much more expensive to create and maintain. In order to make it believable, authentic user activity and network traffic has to be simulated. Therefore, high interaction honeypots are a limited resource and it is important to optimize their deployment. We focus on the latter category in this paper.

One of the important features of real-world networks is that they have many different types of servers with different configurations (available services, hardware, etc.). Some categories of servers are more important than others, both to the owner of the network and as targets for the attacker. For example, a database server containing valuable customer information would be of a very high value, while a standard desktop computer acting as a server may have a relatively low value. To model this, we assume that each server in the network can be classified into one of a few categories of *importance*, which can be assigned a numeric value that represents the gain/loss associated with a successful attack. One of the decisions that the defender makes when deploying honeypots on a diverse network is how to disguise the honeypots – in other words, which category of server should each honeypot be designed to look like?

We represent a *configuration* of the network by a vector of values representing the apparent importance of each server. The defender knows the values of each of the real servers in the network, and is able to extend the vector of values by adding honeypots. For each honeypot, the defender is able to select the value of the server that will be observed by the attacker (by configuring the honeypot to emulate servers of that category). We assume that both players have knowledge of the typical configurations of the network, so both players know the distribution of values in the network. For any configuration, the players can calculate the probability that the configuration is the actual configuration of the network. We also assume that the defender uses a fixed number of honeypots to add to the network, and that the attacker knows the number of honeypots (but not their assigned values). This is a worst case assumption about the attacker, and the model could be generalized to allow for imperfect information about the number of honeypots, though it makes the problem more difficult to solve.

Consider the following example. The network has two servers, which have importance values 4 and 3. The administrator has one honeypot to deploy, and needs to decide how to configure it, which corresponds to assigning a value in our model. He could assign it a value of 5 to make it appear very attractive (e.g., by making it appear that contains valuable data and exposing obvious vulnerabilities). The attacker observes the unordered vector of values by doing a scan of the network, including the value of the honeypot: (5,4,3). A naïve attacker might attack the server with the highest value (5), therefore attacking the honeypot. However, a sophisticated attacker might reason that this is "too good to be true" and choose instead to attack the next best server, with a value of 4. If the attacker chooses a real server to attack, he obtains a reward and the network administrator is penalized. If the attacker chooses to attack a honeypot, he does not obtain any reward and possibly is penalized for disclosing his attack strategy. We model the game as a zero-sum game, so a gain for one player is a loss for the other. While this may not always be the case, it allows for faster solution methods and can provide a solution with guaranteed quality against any (not necessarily rational) opponent. From this example, we can see that the defender's goal is to somehow convince the attacker to selecting a honeypot, and that assigning all honeypots the maximal value may not be the optimal strategy.

3.1 Formal Definition of the Honeypot Selection Game

The Honeypot Selection Game (HSG) is a two-player zero-sum extensive-form game with imperfect and incomplete information.

Definition 1. *The Honeypot Selection Game (HSG) is defined by the tuple* $G = (\boldsymbol{d}, \boldsymbol{a}, \boldsymbol{n}, \boldsymbol{k}, \boldsymbol{D}, p, \mathcal{I}, \chi, \boldsymbol{A}, u)$*:*

- $\mathbf{d}, \mathbf{a}$ *are the players in the game called the defender and the attacker;*
- $\mathbf{n}$ *is the number of real servers;*
- $\mathbf{k}$ *is the number of honeypots;*
- D *is a set of importance values;*
- $p : D^n \to [0, 1]$ *is the probability of each configuration of real servers;*
- $\mathcal{I}$ *is a set of all attacker information sets* ($I \in \mathcal{I}$, $I \subseteq D^{n+k} = D^s$)*;*
- $\chi : D^n \to \mathcal{P}(\mathcal{I})$ *is a function that provides a set of possible actions for the defender, which appends a set of honeypot values to the observed* $\boldsymbol{x} \in D^n$*;*
- $\mathbf{A}$ *is a union of all possible attacker actions for all* $\boldsymbol{y} \in \mathcal{I}$*;*
- $u : D^n \times \mathcal{I} \times \mathbf{A} \to \mathbb{R}^+$ *is the expected utility function for the attacker* ($-u$ *is the utility function for the defender), defined if the second parameter is in* $\chi(\boldsymbol{x})$ *with* $\boldsymbol{x}$ *being first parameter.*

Nature starts by randomly choosing the network configuration $\boldsymbol{x} \in D^n$ according to a known probability distribution p. The defender learns the value $\boldsymbol{x}$ and chooses values for the k honeypots to apply. The defender can insert honeypots anywhere in vector $\boldsymbol{x}$, creating a vector $\boldsymbol{y}$ of length $s = n + k$, which is presented to the attacker. The attacker then chooses one server to attack from $\boldsymbol{y}$. If he

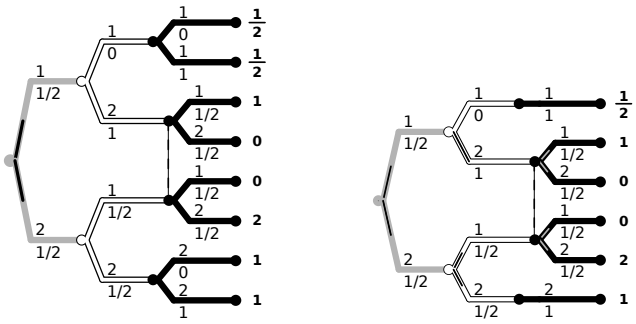


Fig. 1. (a) The game tree of a Honeypot Selection Game rendered by Gambit [4] with one real server, one honeypot and a domain $\{1, 2\}$. Light gray edges are random choices, white edges are defender's actions and black edges the attacker's actions. Server values corresponding to actions are above the branches, while probabilities are under them. (b) The same game with grouped attacker's actions.

attacks a real server i, he obtains the reward y_i from $\boldsymbol{y} = (y_1, \ldots, y_i, \ldots, y_s)$. If he attacks a honeypot the attacker obtains a reward of 0.

Extensive form games are usually represented as game trees. An example of a small HSG with one real server ($n = 1$), one honeypot ($k = 1$) and two importance values ($D = \{1, 2\}$) is shown in Figure 1(a). The root node of the game is a chance node with two outcomes representing the nature. In the example, two configurations are possible and the distribution p depicted below the branches is uniform. In each possible real configuration (i.e., child of the root), the defender chooses to add a set of honeypots to the network and defines the information set the attacker will be in (χ). In the example, the defender can add one honeypot with a value 1 or 2. The possible attacker information sets are $\mathcal{I} = \{11, 12, 22\}$.

We assume that the ordering of the vector is arbitrary and contains no information, so (12) and (21) are equivalent. The tree nodes in the same information set are connected by dashed line. In each information set, the attacker can attack one of the servers, which is the set of actions $\mathbf{A}$. In the example, in the top information set (11), the attacker can choose the first server with importance 1 or the second server with importance 1. One of them is real, but they are indistinguishable, so the expected payoffs for either is $u(1, 11, *) = \frac{1}{2}$. In the middle information set, the attacker can choose to attack the server with a value 1 or 2. In the top node of the information set, he can gain 0 or 1, in the bottom, he can gain 0 or 2, but he cannot distinguish between these two nodes, and must use the same strategy in both.

3.2 Solution of the Game

A *strategy* of a player in a game defines what action the player performs in any situation that can occur in the game. A *solution* of a game is a set of strategies, one for each player, that satisfies some notion of optimality. We will search for

the solution in form of a *behavioral strategy* [9], i.e., a strategy that prescribes a probability distribution over all possible actions in each possible situation. For the defender in our game, this means determining the probability of using each combination of available honeypot values for each possible configuration of the real part of the network. We allow *mixed strategies* (i.e., randomized strategies), since they generalize pure strategies and allow for strategic deception in adversarial games. The goal of the defender is to maximize his expected payoff, which in a zero-sum game corresponds to minimizing the attacker's expected payoff. A pair of strategies that achieves these maximal/minimal expected payoffs is a Nash equilibrium of the game.

3.3 Properties of the Honeypot Selection Game

We present some useful properties of the HSG game. Our analysis provides intuition about meaningful strategies by identifying sets of dominated actions. Removing these dominated strategies also allows us to reduce the size of the game and improve scalability of solution methods.

Lemma 1. *If the attacker sees a vector of values* $\boldsymbol{y} \in D^s$*, then he has a strategy that guarantees payoff*

$$m(\boldsymbol{y}) = \max_{S \subseteq \{1,\ldots,s\}} U(S, \boldsymbol{y}), \text{ where } U(S, \boldsymbol{y}) = \frac{\sum_{i \in S} y_i - \sum max(k, \boldsymbol{y})}{|S|}. \quad (1)$$

The function $max(k, \boldsymbol{y})$ *takes* k *maximum values from* $\boldsymbol{y}$*.*

Proof. $U(S, \boldsymbol{y})$ is the lower bound on the value of the attacker's strategy that uniformly randomizes over the targets in S, which is met if the most important targets in the set are honeypots. The attacker can choose the best S with the information he has available and obtain $m(\boldsymbol{y})$. □

Lemma 2. *The maximizing* S *from Lemma 1 does not contain any index of a server with a value lower than* $m(\boldsymbol{y})$*.*

Proof. If $m(\boldsymbol{y}) \leq 0$ the lemma holds trivially so WLOG $|S| > k$. For contradiction assume the maximizing S contains index j, such that $y_j < m(\boldsymbol{y})$. Then $m(\boldsymbol{y}) = U(S, \boldsymbol{y}) \Rightarrow (|S| - 1)m(\boldsymbol{y}) = (\sum_{i \in S \setminus \{j\}} y_i - \sum max(k, \boldsymbol{y})) + y_j - m(\boldsymbol{y})$ $\Rightarrow m(\boldsymbol{y}) - U(S \setminus \{j\}, \boldsymbol{y}) = (y_j - m(\boldsymbol{y}))/(|S| - 1) < 0$. Hence $m(\boldsymbol{y}) < U(S \setminus \{j\}, \boldsymbol{y})$ which contradicts with S being maximizing. □

Corollary 1. *Attacking a target with a value lower than* $m(\boldsymbol{y})$ *can never appear with non-zero probability in any attacker's optimal strategies.*

Proof. If any attacker's strategy attacks a server j with $y_j < m(\boldsymbol{y})$ with positive probability then the strategy can be modified to attack the set S from the Lemma 1 with uniform probability anytime is it supposed to attack j. This increases the expected payoff of the strategy, which contradicts its optimality. □

Corollary 2. *If the defender receives a vector* $\boldsymbol{x} \in D^n$ *of real targets, it does not have to consider honeypots with value lower than* $m(\boldsymbol{x})$.

Proof. If the defender uses honeypots that do not make it to the set S in computing $m(\boldsymbol{x} \cup \boldsymbol{h})$ then $m(\boldsymbol{x} \cup \boldsymbol{h}) = m(\boldsymbol{x})$. If some of them are present in S, $m(\boldsymbol{x} \cup \boldsymbol{h}) \geq m(\boldsymbol{x})$. Either way, the attacker presented by $y = \boldsymbol{x} \cup \boldsymbol{h}$ would not consider attacking a target with value below $m(\boldsymbol{x})$ by Corollary 1. □

Grouping of Server Values. We also suggest a more compact representation of the game. Since we assume that the attacker cannot distinguish between the servers of the same value, we reduce the number of the actions available to the attacker in each information set $I \in \mathcal{I}$ to the number of different values in the observed configuration $\boldsymbol{y}$. To do this we create *groups* of servers that have identical importance values (and are therefore indistinguishable). The expected value for choosing any server from that group is computed by assuming that the attacker actually chooses uniformly among members of the group, some of which may be real and some honeypots. Recall the example from Figure 1(a), where the attacker could not distinguish between the real server and the honeypot, both valued 1. We limit the attacker to one action for this information set, with the expected value of $\frac{1}{2}$. The reduced game tree is in Figure 1(b).

3.4 Solution Using Linear Programming

We compute a Nash equilibrium of the game in behavioral strategies using a linear program (LP) based on the state-of-the-art method for imperfect-information extensive-form games – a sequence-form LP (e.g., see [9]). The sequence-form utilizes a compact representation of imperfect-information extensive-form games with perfect recall termed *sequences* [14,15], where one sequence for a player represents an ordered list of actions for the player from the root to some node in the game tree. In the following we use the term *compatibility* of sequences – we say that two sequences (one for each player) are compatible, if a step-by-step execution of all the actions in the sequences is a valid course of play. The behavioral strategies can be represented as a probability of executing some sequence conditioned on the opponent playing a compatible sequence. We present two different LP formulations for finding the optimal strategies for the attacker and the defender, assuming in each case that the opponent plays a best response.

Defender's Linear Program. The LP for computing the defender's strategy is as follows. There are two types of variables: (1) $v_I \in \mathbb{R}^+$ represents an expected value of a subgame assigned to each information set of the attacker $I \in \mathcal{I}$, and (2) variables $p_{d_I^{\boldsymbol{x}}} \in [0, 1]$ represent the probability of the defender choosing set I (adding a specific honeypots) for each possible real configuration of the network $\boldsymbol{x} \in D^n$. Furthermore, u denotes the utility function of the attacker that defender minimizes, and $\chi^{-1}(I) : \mathcal{I} \mapsto \mathcal{P}(D^n)$ denotes an inverse function that maps an information set to a set of possible configurations of the real part of the network. Finally, $p_{\boldsymbol{x}}$ denotes the probability of network configuration $\boldsymbol{x}$.

$$\min_{v,d} \sum_{I \in \mathcal{I}} v_I \tag{2a}$$

$$v_I \geq \sum_{\boldsymbol{x} \in \chi^{-1}(I)} u(\boldsymbol{x}, I, a_i^I) p_{d_I^{\boldsymbol{x}}} \quad \forall I \in \mathcal{I},\ \forall a_i^I \text{ action applicable in } I \tag{2b}$$

$$\sum_{I \in \chi(\boldsymbol{x})} p_{d_I^{\boldsymbol{x}}} = p_{\boldsymbol{x}} \quad \forall \boldsymbol{x} \in D^n \tag{2c}$$

The program minimizes the utility of the attacker by searching for the optimal strategy of the defender $p_{d_I^{\boldsymbol{x}}}$. These variables are constrained by (2c) in order to represent valid probabilities of sequences played by the defender, conditioned on the other players playing compatible sequences (both nature and the attacker). Finally, the attacker chooses the optimal solution in each information set I. Hence, the expected value v_I is maximized for all possible configurations and actions of the attacker in constraints (2b).

Attacker's Linear Program The LP for computing the optimal strategy for the attacker is similar – the attacker is maximizing its utility value through probabilities for each action $p_{a_i^I} \in [0, 1]$ in each information set I, while the defender selects an optimal action minimizing the expected utility value at each information set corresponding to each network configuration $\boldsymbol{x}$ in constraints (3b).

$$\max_{v,a} \sum_{\boldsymbol{x} \in D^n} p_{\boldsymbol{x}} v_{\boldsymbol{x}} \tag{3a}$$

$\forall I \in \mathcal{I}$ assume the attacker can perform actions $\{a_1^I, \ldots, a_m^I\}$:

$$\sum_{i \in \{1,\ldots,m_I\}} u(\boldsymbol{x}, I, a_i^I) p_{a_i^I} \geq v_{\boldsymbol{x}} \quad \forall I \in \mathcal{I},\ \forall \boldsymbol{x} \in \chi^{-1}(I) \tag{3b}$$

$$\sum_{i \in \{1,\ldots,m_I\}} p_{a_i^I} = 1 \tag{3c}$$

Size of the Linear Programs The size is exponential with $|\boldsymbol{y}| = s$ in both constraints and variables. This follows from the upper bound of the number of attacker's information sets, $|\mathcal{I}|$, which is at most equal to $|D|^s$. The exponential size of the programs currently limits the applicability of this approach to large computer networks. In this paper, we focus on the validation of the proposed model and we leave further solution computation optimization to future research. Moreover, if the instance is too large, good strategies can be computed using approximation algorithms, like CFR, instead of LP. The optimal solutions, however, provide better grounds for our initial analysis.

4 Honeypot Selection Game with Probes

In this section we extend the basic model from the previous section by allowing the attacker to analyze the observed servers to learn, whether they are real servers or honeypots. The main idea of the model with probes is that the attacker, prior to the actual attack, can use *probes* to try to discover the true nature of servers, whether a probed server is real (denoted R), or a honeypot (HP).

We assume that the attacker can use a limited number of probes, and that the results of the probes are stochastic. The first assumption reflects the limited time and resources the attacker typically has for the attack before being exposed. The second assumption models the fact that the attacker cannot be perfectly sure if the server is a honeypot or not, even after gathering some information through probing.

4.1 Formal Definition of Honeypot Selection Game with Probes

The formal definition of Honeypot Selection Game with Probes (HSGp) follows:

Definition 2. *The HSGp is defined by the tuple* $G = (\Gamma, q, \mathcal{I}_\mathcal{E}, \mathbf{A_p}, \mathbf{A_a}, \psi, u)$*:*

- $\Gamma = (\boldsymbol{d}, \boldsymbol{a}, \boldsymbol{n}, \boldsymbol{k}, \boldsymbol{D}, p, \mathcal{I}, \chi, \boldsymbol{A}, u)$ *is a basic HSG;*
- q *is the number of probes to be performed by the attacker;*
- $\mathcal{I}_\mathcal{E}$ *is a set of all attacker information sets,* $\mathcal{I} \subseteq \mathcal{I}_\mathcal{E}$*;*
- $\mathbf{A_p}$ *is a set of all possible attacker probing actions;*
- $\mathbf{A_a}$ *is a set of all possible attacker attacking actions (not the same as* $\mathbf{A}$*, because of probed servers, explained in this section);*
- $\psi : \{R, HP\}^i \times {\mathbf{A_p}}^{i+1} \to [0,1]; \forall i \in \{0, \ldots, q-1\}$ *is a function that assigns the probability of a probe result being R, based on the history of probing decisions and observations;*
- $u' : D^n \times \mathcal{I} \times ({\mathbf{A_p}}^q, \mathbf{A_a}) \times \{R, HP\}^q \to \mathbb{R}^+$ *is the expected utility function for the attacker (*$-u$ *for the defender). The observations are necessary, because the servers of the same value are indistinguishable as explained in this section.*

In order to define ψ, we assume that results of probing a single fixed server are independent and identically distributed to simplify the mathematical expression (though in principle the model is not restricted to this). The probability that a probe to a server that is either R or HP returns either result R or HP is fixed and does not change with repeated attempts. We denote these probabilities $\alpha(R|R)$ – the probability of R when probing a real server – and $\alpha(HP|HP)$ – the probability of HP when probing a honeypot. The complementary probabilities for false positives and false negatives (misidentification) follow from these.

Figure 2 shows part of a game tree for an instance of honeypot selection game with probes. The attacker chooses a server to probe in its information set (12), followed by chance nodes representing the uncertain results of the probes. The probability values for the chance nodes that determine the results of the probes are given according to the function ψ. Although we assume that the probe results

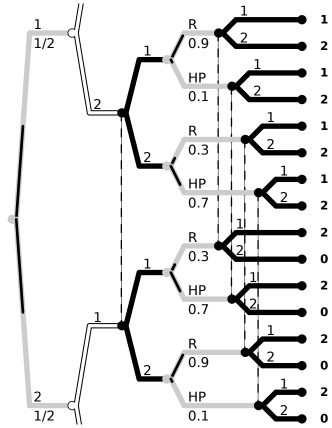


Fig. 2. One root information set with observed values 1 and 2 for the attacker including subtrees for the node from the set. R and HP are the outcomes of probes.

from a fixed server are independent from each other and they are determined by the parameters α, the probabilities in the game tree depend on the path in the tree that lead to them. In the following section we describe a methodology for computing ψ based on α and observations.

4.2 Probabilities of the Chance Nodes after Probing Outcomes

The probing model also handles a set of servers of a single value as indistinguishable. Probing a server distinguishes it from, but the rest remains indistinguishable. Each of these servers has a probability of being real, at first depending only on the number of honeypots among them. The probing modifies these probabilities and directly affects ψ. Let us describe the methodology for defining the ψ function more formally. We focus on a single set of servers sharing the same importance value ϕ. We base our notation on previous definitions: k_ϕ is the number of honeypots, n_ϕ is the number of real servers, $s_\phi = k_\phi + n_\phi$ is the total.

The prior probability of the i-th server with value ϕ being real is $p(i)$. The ordering is drawn at random uniformly to make sure that it cannot be exploited. We denote $p(i|\boldsymbol{o}, \boldsymbol{b})$ as the posterior probability of the i-th server being real after a sequence of observations $\boldsymbol{o} = (o_1, \ldots, o_l)$ and probing actions $\boldsymbol{b} = (b_1, \ldots, b_l)$ with l as the l-th probe. The ψ function value, the probability of an outcome, for the first probe of the attacker (examining server i) can be calculated as $\psi(\emptyset, i) = p(R) = p(R|i)p(i) + p(R|\neg i)p(\neg i)$.

Based on the outcome we can update the probabilities $p(i)$ for servers in ϕ. We can use the Bayes rule to calculate $p(i|R)$ for the probed server. For any other server $j \neq i$, the probability of being real after the first probing can be calculated as $p(j|o_1, b_1) = p(j|i)p(i|o_1, b_1) + p(j|\neg i)p(\neg i|o_1, b_1)$; where $p(j|i)$ represents the probability of server j being real if server i is real without any observations calculated as $p(j|i) = \frac{n_\phi - 1}{s_\phi - 1}$, and $p(j|\neg i)$ representing the case, where i is not

R. However, this rule becomes difficult to express concisely, with the increasing amount of probes, because calculating $p(j|i, o_{l+1}\boldsymbol{o}, b_{l+1}\boldsymbol{b})$ becomes very difficult.

To see why this is the case, let us denote each of the possible assignments of real servers and honeypots for ϕ by *characteristic vectors* $c \in \{R, HP\}^{s_\phi}$. Let us put each of the vectors into groups that have honeypots and real servers in the same places for all probed locations. For example, the first server was probed and yields two groups of characteristic vectors, one with a honeypot as the first server, and one with a real server as the first server. Each newly probed server subdivides the groups further. Each subdivided group requires a separate Bayesian update. There will be at most 2^{s_ϕ} groups, each representing a single characteristic vector per group.

To exactly calculate all the probabilities $p(i|o_{l+1}\boldsymbol{o}, b_{l+1}\boldsymbol{b})$ after $(l+1)$-th probe, we consider all characteristic vectors that are compatible with the current information set in the game tree. Each game situation has a list of probabilities of being true assigned to each of the characteristic vectors for each of the importance values. The probability $p(i|\boldsymbol{o}, \boldsymbol{b})$ can be calculated by summing over probabilities of characteristic vectors with a real server at the i-th position:

$$p(i|\boldsymbol{o}, \boldsymbol{b}) = \sum_{c \in S} p(c|\boldsymbol{o}, \boldsymbol{b}), \quad S = \{c | \forall c \in \{R, HP\}^{s_\phi}; c_i = R\} \tag{4}$$

With the *i.i.d.* assumption, the updates are based on Bayes' Rule. Vector c is the characteristic vector, whose probability is being updated after probing b_{l+1}.

$$p(c|o_{l+1}\boldsymbol{o}, b_{l+1}\boldsymbol{b}) = \begin{cases} \frac{\alpha(o_{l+1}|R)p(c|\boldsymbol{o},\boldsymbol{b})}{p(o_{l+1}\boldsymbol{o}|b_{l+1}\boldsymbol{b},\boldsymbol{o})}, & \text{iff } c_{b_{l+1}} = R \\ \frac{\alpha(o_{l+1}|HP)(1-p(c|\boldsymbol{o},\boldsymbol{b}))}{p(o_{l+1}\boldsymbol{o}|b_{l+1}\boldsymbol{b},\boldsymbol{o})}, & \text{iff } c_{b_{l+1}} = HP \end{cases} \tag{5}$$

The updated vector of probabilities is used in the subtree of the node.

Grouping with Probes. We can reduce the number of actions for the attacker by grouping all servers of the same importance value that have not been probed yet. These are treated identically as the *"next server to be probed"*. They have the same outcomes and same probabilities of being real, so we do not break the interpretation of the game. Every time a new server is probed, it is differentiated from the rest of the servers in the group. This approach keeps a fixed ordering, which the defender still cannot influence.

Properties of HSGp. There is an opportunity for further pruning in the HSGp besides creating groups. In the final decision node of the attacker, we can replace a set of attacks on the servers of a same importance with a single attack that represents an attack on the server with the largest probability of being real. Among all the servers of the same importance value, the one with the highest probability being real (in the node) has the highest expected utility and the observations from probes would also lead the attacker to see it as such; hence, this strategy is dominant and will be selected by a rational attacker.

4.3 Solution Using Linear Programming.

The linear program calculating the solution is an extension of the linear program presented in Section 3.4. The extension treats the chance nodes as defender's choice nodes with a fixed strategy. However, it is still necessary to provide constraints for the weighted values for the attacker's choice nodes for probes.

In order to improve readability we denote $fin(\mathcal{I}_\mathcal{E})$ to be a set of all information sets where the attacker chooses the server to attack. $\Sigma_{a,c,I}$ refers to compatible sequences of attacker's actions and chance node outcomes for I, one of the starting information sets for the attacker. Function $orgn(I)$ returns the first information set of the attacker encountered on the path in the game tree to I. $Ext_a(\sigma_a)$ returns the shortest extension to sequence $\sigma_a \in \Sigma_a$, where Σ_a is the set of all possible sequences of attacker's actions. By the shortest extension we mean sequence σ_a with a single, valid attacker action appended to its end. In the program, we also use $\mathcal{I}_\mathcal{E}(\sigma_a)$ as a function that returns a set of information sets reached by the attacker after executing sequence of actions σ_a.

$$\min_{v,d} \sum_{I \in \mathcal{I}} v_I \tag{6a}$$

$$v_I \geq \sum_{\boldsymbol{x} \in \chi^{-1}(orgn(I))} -u'(\boldsymbol{x}, I, \sigma_a, \sigma_e) p_{d_I^{\boldsymbol{x}}} \quad \forall I \in fin(\mathcal{I}_\mathcal{E}), \forall (\sigma_a, \sigma_e) \in \Sigma_{a,e,I} \tag{6b}$$

$$v_{I(\sigma_a)} \geq \sum_{I' \in \mathcal{I}(Ext_a(\sigma_a))} v_{I'} \quad \forall \sigma_a \in \Sigma_a \tag{6c}$$

$$\sum_{I \in \chi(\boldsymbol{x})} p_{d_I^{\boldsymbol{x}}} = p_{\boldsymbol{x}} \quad \forall \boldsymbol{x} \in D^n \tag{6d}$$

The defender aims to minimize the expected utilities of the attacker's best response. We define u' as $u'(\boldsymbol{x}, I, \sigma_a, \sigma_e) = \phi p_e(\sigma_e) p_t(\phi_i)$, where $p_t(\phi_i)$ is the probability of the i-th server in the ϕ-valued set being real in the final decision node t, while $p_e(\sigma_e)$ is the probability of the outcomes of the observations that led to the final information set.

Inequality (6b) provides constraints that maximize the attacker's expected utility in the level just above the one with terminal nodes. The second inequality (6c) provides constraints that maximize over the expected value of the subtrees of attacker's probing decisions by summing over the expected value of each possible probing. The final inequality (6d) makes sure that the probabilities of defender's actions form a valid probabilistic distribution.

Due to space limits, we omit the attacker's LP. The difference between the HSG program and HSGp is in the addition of new constraints that make sure that the probabilities of attacker's sequence are valid in each node, including the chance nodes. The "variables" for chance node probabilities are fixed in each of the chance nodes for each probing outcome. Due to the sequence of q decisions of the attacker, the size of the linear program is exponential in q (and also in s as is the basic HSG).

5 Evaluation of Computed Strategies

In this section we provide experiments and analysis of the behavior of the models with varying parameters. The goal is to identify key characteristics of the game, and compare the quality of the game-theoretic solution to baseline strategies. Finally, we want to derive general principles from the results in order to give the network administrators some rules of thumb for placing the honeypots in a computer network. All of the results are computed using the LP formulations described earlier with *CPLEX 12.1*.

5.1 Experimental Settings

In our experiments we fix the number of real servers, $n = 5$, and the importance values $D = \{1, \ldots, 4\}$. The number of honeypots is $k \in \{0, \ldots, 5\}$. The size of these games is plausible for a small computer network. We use two different probability distributions over the possible network configurations $\boldsymbol{x}$, a uniform distribution and a power-law Yule-Simon distribution with parameter $\rho = 1$. The Yule-Simon distribution reflects a common situation in computer networks with relatively few high-valued targets, and a larger number of less significant targets. For our domain with four values the probabilities from this distribution are in increasing importance order: $(0.6250, 0.2083, 0.1042, 0.0625)$.

As baselines for comparison with the game-theoretic strategies we introduce two methods for each player: (1) *Random* strategy, in which the player always selects a uniform random action in each information set, and (2) *Maximum* strategy, which uses a greedy heuristic always attacking/adding targets with the maximal observed value[2].

Our first set of results compares the payoffs of our baseline strategies and the game-theoretic strategies from two different perspectives. First, we evaluate the guaranteed utility of the different defender's strategies, and then we compare the quality of the attacker's strategies against the defender's Nash equilibrium (*NE*) strategy. The guaranteed utility of a strategy σ is the payoff for the strategy when the opponent plays a best response to σ. We calculate the guaranteed utility using the linear program for the game-theoretic solution, but with fixed probabilities for the defender's actions (and vice versa for the attacker).

Our second set of results shows the details of the optimal defender's strategies, i.e., the probability that a honeypot with a specific value will be actually deployed in a computer network. We present the results about honeypot likelihood in two different ways: (1) the probability that *at least one* honeypot of the given value is used by the defender, and (2) the portion of honeypots assigned to each value. In the first case we marginalize over the probabilities of defender's actions that add a honeypot of this value, weighted by the network configuration probabilities. For the second case we marginalize over all defender's actions and weight each component by the number of honeypots of this value added by the action, as well as by the network configuration probabilities. We then renormalize

[2] Maximal expected value in the case of HSGp, with the same greedy rule for probing.

the probabilities (divide by k) so the proportions sum to 1. For example, if an action $d_I^{\boldsymbol{x}}$ adds three honeypots of value 4, its component in the sum is $\frac{3p(d_I^{\boldsymbol{x}})}{k}$.

5.2 Basic HSG

Game Values. The results for the game values are presented in Figure 3, first row. In Figure 3(a) we show the guaranteed utility of the defender's *NE*, *Random*, and *Maximum* strategies. The results show that the *Maximum* strategy gains almost no benefit from more than one honeypot, while the *Random* strategy shows a small gain. The *NE* is clearly stronger than the two baselines and significantly increases the defender's utility as the number of honeypots increases.

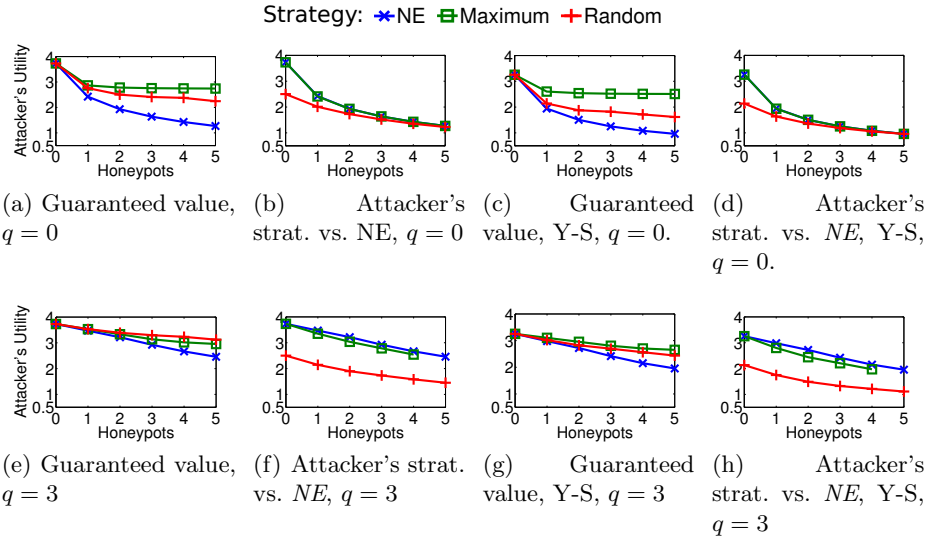
Figure 3(b) shows the quality of attacker's strategies against the defender's *NE* strategy (attacker prefers higher values). The *Random* strategy performs better with more honeypots, and almost matches the other two strategies when $n = k$. This suggests that the defender's strategy is effectively making it impossible for the attacker to distinguish between the servers based on value. From $k = 0$, *Maximum* strategy has exactly the same payoff as the *NE* strategy, implying that it is part of the support set that the *NE* strategy randomizes over.

The second pair of subfigures (3(c), (d)) shows the game values for the Yule-Simon distribution. For both the guaranteed utility of defender's strategies and the payoffs of the attacker's strategies against the *NE*, the progression is nearly identical with an increasing number of honeypots. The values are smaller overall, which reflects the lower frequency of high-valued servers. The only exception is the *Random* strategy, which improves slightly more than the other strategies, though the *NE* strategy is still better. The overall similarity of results indicates that the choice of distribution does not have a strong effect on the results.

Defender's Strategy Analysis. The plots in Figure 4, first row, show how the defender chooses to assign values $D \in \{1, \ldots, 4\}$ to the honeypots. Each line represents one of the four possible values.

Figure 4(a) shows the probability that *at least* one honeypot of the given value is used by the *NE* strategy. We see that it is very rare to use any honeypot 1, as there is little gain from protecting these servers. With few honeypots this is also the case for value 2, but with increased number of honeypots the number 2s becomes more significant. In Figure 4(b), we present the expected proportion of honeypots in the network that have each value. The proportions tend to slightly converge as the number increases, with probabilities of lower valued honeypots increasing, while the probabilities of the higher values decrease. The stability is interesting, as it suggests that network administrators can use the same basic selection ratio over a range of possible amounts of honeypots.

In Figures 4(c) and (d) we show the results under Yule-Simon distribution. The increase in probability of using at least one lower-valued server, primarily at the expense of value 4, is caused by the higher probability they have and therefore the defender protects them more. A slight convergence can be seen here as well, but the portions show less difference than in Figure 4(b).



(a) Guaranteed value, $q = 0$ (b) Attacker's strat. vs. NE, $q = 0$ (c) Guaranteed value, Y-S, $q = 0$. (d) Attacker's strat. vs. *NE*, Y-S, $q = 0$.

(e) Guaranteed value, $q = 3$ (f) Attacker's strat. vs. *NE*, $q = 3$ (g) Guaranteed value, Y-S, $q = 3$ (h) Attacker's strat. vs. *NE*, Y-S, $q = 3$

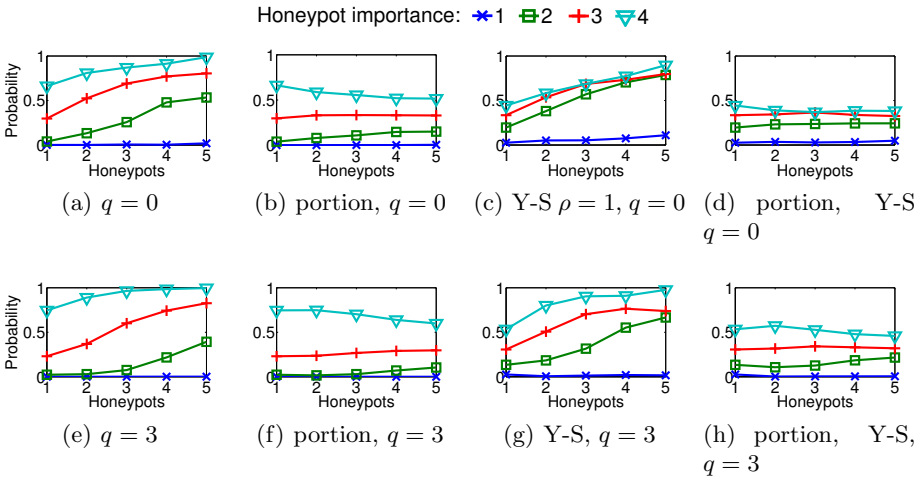
Fig. 3. For all figures $n = 5$, $h = \{1, \ldots, 5\}$, $D = \{1, \ldots, 4\}$. For (a), (b), (c), (d) $q = 0$ (HSG). (a) Guaranteed values of defender's strategies. (b) Values of the attacker's strat. against Nash strategy *NE*. (c) Game values under the Yule-Simon distribution of the R server values with $\rho = 1$. (d) Values of the attacker's strat. vs. *NE* under Y-S. – For (e), (f), (g), (h) $q = 3$ (HSGp). (e) Guaranteed values of defender's strategies. (f) Values of the attacker's strat. vs. *NE*. (g) Game values for the under Y-S. (h) Values of the attacker's strat. vs. *NE* under Y-S.

5.3 HSG with Probes

Next, we present results for the model with probes HSGp. Our set of parameters is $\alpha(R|R) = 0.9$, $\alpha(HP|HP) = 0.7$, with $q = 3$. The probability $\alpha(R|R)$ models the assumption that it is unlikely for an R server to behave as an *HP*. The $\alpha(HP|HP)$ assumes that it is slightly less likely to correctly identify an *HP*.

The attacker's *Random* strategy uniformly probes the servers and then uniformly chooses a server to attack. The attacker's *Maximum* strategy assumes a uniform prior probability of any server being real, regardless of the defender's strategy. The strategy probes according to the current highest expected value, conditioned by the observations and probings. After all probes have been used, the server with the maximal expected value is attacked. While this strategy is strategically simple, it has high memory requirements for evaluation because it needs to keep a separate probability vector for each possible plan. The missing data point for $k = 5$ honeypots in Figures 3(f) and (h) is the result.

Game Values. The results for the game values are presented in Figure 3, second row. There is an almost linear decrease in attacker's utility in Figure 3(e), which contrasts with the results for $q = 0$, especially for the *NE* strategy (see Section 5.2). The almost-linearity is present also in the attacker's strategies



(a) $q = 0$ (b) portion, $q = 0$ (c) Y-S $\rho = 1$, $q = 0$ (d) portion, Y-S $q = 0$

(e) $q = 3$ (f) portion, $q = 3$ (g) Y-S, $q = 3$ (h) portion, Y-S, $q = 3$

Fig. 4. For all figures $n = 5$, $h = \{1, \ldots, 5\}$, $D = \{1, \ldots, 4\}$. For (a), (b), (c), (d) $q = 0$ (HSG). (a) Probability of use of *HP* values under uniform distribution of R values. (b) The expected portion *HP* values, uniform. (c) Probability of use of *HP* values under Yule-Simon distribution of the R values with $\rho = 1$. (d) The expected portion of *HP* values, Y-S. – For (e), (f), (g), (h) $q = 3$ (HSGp). (e) Probability of use of *HP* values,, uniform. (f) The expected portion of *HP* values, uniform. (g) Probability of use of *HP* values, Y-S. (h) The expected portion of *HP* values, Y-S.

in Figure 3(f). The *Maximum* strategy compares reasonably well with the *NE* strategy for the attacker. The *Random* strategy performs much worse than the other two. These two observations support the use of *NE* attacker's strategy.

Results under both distributions (Figures 3(e), (f) and 3(g), (h)) are very similar. The only difference, apart from the shift towards 0 for the same reason as in HSG, is that *Random* and *Maximum* strategies have exchanged places in Figures 3(e) and 3(g). With $q = 0$, the *Random* strategy is better than *Maximum*, not with $q = 3$. Intuitively higher values need to be protected more, because a probe result gives R the attacker a high confidence that the server is real.

Defender's Strategy Analysis Most of the observations for $q = 0$ hold for $q = 3$ as well. One exception is that the leveling out of value 3 in Figure 4(b) is not present in Figure 4(f). Comparing figures from the first row of Figure 4 ($q = 0$) with the second row ($q = 3$), we can see that with the increased amount of probes, the highest valued 4 is more preferred. We speculate that the reason for this might be the increased chance of the attacker of discerning honeypots from R servers. The selected values for $\alpha(\bullet|\bullet)$ give high probability of a server being observed as R, if it is R ($\alpha(R|R)$), while a slightly lower probability for a *HP* observed as a *HP* ($\alpha(HP|HP)$). This could explain why probabilities for 3s do not level out (Figure 4(f)), as opposed to the $q = 0$ case (Figure 4(b)).

6 Conclusion

We introduce new game-theoretic models for analyzing honeypot configuration problems in network security. These models significantly extend previous work in this area, and provide new insights into non-trivial strategies for using honeypots effectively in network security. Our model shows that honeypots should not always be configured to look like the most or least valuable servers in a network, but instead the optimal strategy is randomized and distributes honeypots that look like different types of servers on the network. This becomes increasingly important as networks move towards using a larger number of honeypots as ways to deceive and attract the attention of attackers. This is shown in our empirical results as we see that the Nash equilibrium strategies have a stronger performance relative to baselines as the number of available honeypots increases.

The first model we present is a type of deception game, where the defender tries to disguise honeypots in a network so that the attacker will choose to attack honeypots instead of real servers. Our second model extends this by including probing actions for the attackers, who can try to distinguish honeypots from real servers before actually launching an attack. The probes are noisy, so the attacker still needs to act with imperfect information in these models. We present linear programming models for solving both of these classes of games.

We study the behavior of both of our models empirically, using heuristic baseline strategies for both players. We also vary the assumption about the distribution of importance values on the network. The Nash equilibrium strategies in our models significantly outperform the baseline strategies, regardless of the distribution of values of real servers in the network. We also studied the structure of the equilibrium strategies in these games, which shows that honeypot values in both cases should be distributed across the space of possible configurations. As the number of honeypots increases, there is a change in the strategies, with the optimal strategies placing greater weight on lower values.

Our analysis shows that there are important strategic issues that must be investigated to maximize the efficiency of honeypots in network security, particularly as the purpose of honeypots evolves from learning about attackers to actively deceiving and delaying attackers. It is not sufficient to consider only the technical issues involved in honeypot design, but also the strategic issues about how they should be used.

Acknowledgments. This research was supported by the Office of Naval Research Global (grant no. N62909-12-1-7019), and the Czech Science Foundation (grant no. P202/12/2054).

References

1. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)

2. Dornseif, M., Holz, T., Klein, C.N.: NoSEBrEaK - attacking honeynets. In: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, pp. 123–129 (June 2004)
3. Garg, N., Grosu, D.: Deception in Honeynets: A Game-Theoretic Analysis. In: IEEE Information Assurance Workshop, pp. 107–113 (2007)
4. McKelvey, R.D., McLennan, A.M., Turocy, T.L.: Gambit: Software Tools for Game Theory. Technical report, Version 0.2010.09.01 (2010)
5. Wagener, G., State, R., Dulaunoy, A., Engel, T.: Self Adaptive High Interaction Honeypots Driven by Game Theory. In: Guerraoui, R., Petit, F. (eds.) SSS 2009. LNCS, vol. 5873, pp. 741–755. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-05118-0_51
6. Williamson, S.A., Varakantham, P., Hui, O.C., Gao, D.: Active Malware Analysis Using Stochastic Games. In: Proceedings of AAMAS, pp. 29–36 (2012)
7. Carroll, T.E., Grosu, D.: A game theoretic investigation of deception in network security. Security and Communication Networks 4(10), 1162–1172 (2011)
8. Hausken, K., Levitin, G.: Protection vs. false targets in series systems. Reliability Engineering & System Safety 94(5), 973–981 (2009)
9. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, pp. 130–144. Cambridge University Press (2009)
10. Paruchuri, P., Pearce, J., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. In: Proceedings of AAMAS, pp. 895–902 (2008)
11. Spencer, J.: A deception game. American Mathematical Monthly, 416–417 (1973)
12. Lee, K.: On a deception game with three boxes. Int. Jour. of Game Theory 22(2), 89–95 (1993)
13. Cohen, F.: A Mathematical Structure of Simple Defensive Network Deception. Computers & Security 19(6), 520–528 (2000)
14. von Stengel, B.: Efficient Computation of Behavior Strategies. Games and Economic Behavior 14(2), 220–246 (1996)
15. Koller, D., Megiddo, N., von Stengel, B.: Efficient Computation of Equilibria for Extensive Two-Person Games. Games and Economic Behavior 14(2), 247–259 (1996)

A Cost-Based Mechanism for Evaluating the Effectiveness of Moving Target Defenses

M. Patrick Collins

RedJack LLC, 8484 Georgia Avenue, Ste. 940, Silver Spring, MD 20910
michael.collins@redjack.com

Abstract. We propose a means for evaluating the strength of network-based moving target defenses using a general model of *tag switching*. Tag switching breaks the network into tags (labels for entities on the network) and assets (hosts present on the network) whose relationshps are moderated by lookup protocols, such as DNS, ARP or BGP. Lookup protocols hide the relationship between tags and assets, and are already used to provide dynamic asset allocation for scaling and defense. Our model provides a generalize means for describing tags and assets within *tag spaces* defined by the defender and then quantifies the attacker's ability to manipulate a network within a tag space. Defenders manipulate the tag/asset relationship over time using one of a number of moving target defenses. The impact of these defenses is quantifiable and can be used to determine how effective different defensive postures will be.

1 Introduction

A *moving target defense* is any of a family of defenses where a defender constantly shifts the configuration or placement of assets on a network in order to confuse and constrain an attacker. A simple example of a moving target defense is Antonatos *et al.*'s NASR system [1], where on fixed intervals, hosts are randomly assigned new IP addresses. NASR, as with many moving target defenses, effectively imposes a lifetime on attacker intelligence - if the attacker acts on information acquired before the defender changed the system's configuration, then his attack will likely fail and he risks exposing his own assets and methodology.

Moving target strategies are currently used in computer networks both for performance and for defense. Since at least the common adoption of *content distribution networks* (CDNs) [10], moving targets have played an important part in Internet infrastructure, and they have been used to implement a variety of defenses in the research literature. These defenses include NASR's straight randomization [1], DYNAT's cryptographic permutation of networks [9], and Mailinator's temporary addressing [14]. Attackers have also adopted strategies using moving targets, such as fast-flux networks [3].

While a variety of moving target strategies have been proposed, implemented and discussed, these defenses are almost invariably discussed within the context of a single addressing scheme. For example, DYNAT and NASR discuss the permutation of IP addresses within networks, while CDN's and fast-flux work

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 221–233, 2012.

within DNS. We describe these disparate defenses under a general framework of *tag-switching*. A tag-switching system consists of a set of *tags* which serve as labels for a set of *assets*; the relationship between tags and assets is moderated by a *lookup protocol*, which dynamically associates tags and assets as defined by the defender.

For a tag-switching defense to operate, attackers must reach a target through the agency of a lookup protocol. A lookup protocol is an internet protocol or service which a host contacts in order to find the location of a target via a label assigned to it. Examples of lookup protocols include DNS (which associates names to IP addresses), and ARP (which associates IP addresses with hardware addresses). Lookup protocols are a key feature of Internet architecture, as they provide a means to dynamically allocate resources in response to demand – the IP address associated with a DNS name, for example, can be changed rapidly and regularly for load balancing, maintenance or to evade blacklisting.

Our tag-switching approach describes a tag as a label in a larger *tag space* owned by a network defender. The size and composition of a tag space is a function of both the protocol and the defender's resources. For example, a defender who owns a /24 can create a defense with a tag space consisting of all 256 addresses he owns. The tags themselves point to *assets*, which are the resources the defender is actually protecting. Given the structure of the internet, it is possible that an asset for one lookup protocol may be the tag for another protocol (*e.g.*, DNS uses names as tags and IP addresses as assets, while ARP uses IP addresses as tags and hardware addresses as assets).

Moving target defenses change the relationships between tags and assets without the attacker's knowledge. As an attacker conducts reconnaissance on a network, he builds a map of that network – this intelligence includes such features as where hosts are located and what vulnerabilities they present, but is also moderated by the lookup protocol. Attackers do *not* directly communicate with targets, they work through the agency of the lookup protocol. Each defense we describe in this paper represents a different mechanism for manipulating the relationships between tags and assets in the lookup protocol.

It is important to note that the agency of the lookup protocol is a unique feature of network based defenses. Network based moving target defenses rely on a number of decentralized features which are expressed using the lookup protocols. These systems are harder for an attacker to globally subvert than resources resident on a single host.

The ultimate goal of this work is to develop a toolkit for quantifiably evaluating the defensibility of various network configurations. The model developed in this paper quantifies the attacker's ability to strike at the network by comparing the attacker's knowledge of the network to its current configuration. Based on the type of attack the attacker conducts, they have a different probability of success, a function of the type of attack, the moving target defense adopted by the defender and the time at which the attack takes place. The strength of these attacks can then be compared in real world terms: the number of hosts occupied, the size of the network defended, the expected time before a host is recovered.

The remainder of this paper is structured as follows. §2 describes the attack and defense model used in this paper. §3 shows how to use the model to evaluate several forms of attack and defense. §4 discusses previous work on network-based defenses. §5 concludes the work.

2 Methodology

We now discuss our method for evaluating the relative strength of a network-based moving target defense. Recall from §1 that, in addition to prototypes in the research literature, several moving target systems are already implemented in in contemporary network engineering. The goal of this section is to unify these disparate defensive techniques under a common model that enables an engineer to evaluate the relative strength of a defensive strategy.

Our model is based around four components: a *lookup protocol* that defines the relationship between tags and assets, an *attack model* which is based around the intelligence an attacker has on a network, *defenses* which rely on the lookup protocol to move host around a network, and a *damage model* which shows how well the attacker can achieve his goals. Combined, these components allow a network engineer to specify a defensive strategy and based on the configuration and hosts on the network, estimate the damage an attacker will cause when attacking.

The lookup protocol and associated values define the potential tag space a tag can move through, while the attack strategies defines the attacker's success. The defensive strategies are applied to the tag space to determine how well a network defends against a particular attack.

The remainder of this section is structured as follows. §2.1 defines the roles of tags, assets and lookup protocols, explaining how to map a real-world criterion such as a class B network to a tag space. §2.2 defines the attack model and its relationship to the lookup protocol, as well as defining attacker goals. §2.3 describes how moving target defenses are implemented in this model.

2.1 Tags, Assets and Lookup Protocols

A *lookup protocol* is a network protocol which takes a *tag* (t) and returns connection to an *asset* (a). Examples of lookup protocols include DNS A records (where the tag is a DNS name and the asset is an IP address), ARP (where the tag is an IP address and the asset a link-level address), and DNS MX records (where the tag is a domain name and the asset the IP address of a mail server). The lookup protocol serves as a mechanism for hiding the actual relationship between tags and assets, as such it does not need to maintain a fixed relationship between them. For example, DNS servers can use a feature referred to as *round robin* load balancing[1], which allows the DNS server to randomly return one of several IP addresses for the same name.

[1] Brisco, "RFC 1794: DNS Support For Load Balancing", `http://tools.ietf.org/html/rfc1794`, April 1995.

The actual configuration of a lookup protocol is managed by a network's defender. In our model, the defender has access to an *asset set* (A). The asset set comprises the targets the defender protects, and which the attacker wants to investigate, subvert, or control. An asset set is assumed to be an ordered set of assets of the form $a_1 \dots a_k \in A$ where $k = |A|$ is the cardinality of the asset set. For the purposes of the model, the position of the assets within the set is constant (*i.e.*, a_1 remains a_1 consistently).

The lookup protocol relates these assets to tags selected from a *tag space* (T). As with the asset set, the tag space is an ordered set of tags of the form $t_1 \dots t_n \in T$, where $n = |T|$ is the cardinality of the tag space. The size of the tag space is a function of the lookup protocol. For example, if the defender has access to a class C network (a /24), then he has 256 potential tags and $|T| = 256$. Access to a class B network (a /16), would set $|T| = 65536$. Even for protocols such as SMTP for DNS, the tag space is enumerable as 64 and 253 octets respectively[2,3].

During configuration (and reconfiguration, in the moving target case), the defender defines the relationship between the assets in the asset set and tags in the tag space. We refer to this process of association as *linking*, if a tag t is linked to an asset a, then consulting the lookup protocol for tag t will return a. We will represent this formally using a lookup function, ℓ:

$$\ell(t, \tau) \equiv \begin{cases} a & \text{if } a \text{ is linked to } t \\ \varnothing & \text{otherwise} \end{cases} \tag{1}$$

2.2 Attack

We now describe our attack model in this method. Attacks are based around gathering intelligence about a network from lookup protocols. Lookup protocols effectively control an attacker's access to the network; from the attacker's perspective, the network is composed of the *tags*, not the assets themselves.

Assets in the network exist in one of three states, which represents the knowledge the attacker has about an asset at any time. These states are *unexplored* (unx), *identified* (id), and *subverted* (sub). These states represent what an attacker believes about the asset pointed to by the tag, but not the true state of the associated asset. We represent this model as a set of (t, a, σ) tuples, where $\sigma \in \{\text{unx}, \text{id}, \text{sub}\}$.

Figure 1 is a graphical representation of the attacker's potential knowledge about a host. As this figure shows, the attacker's knowledge progresses through several states, and can potentially be frustrated by different defenses. An *unexplored* tag is one that the attacker has no information about. Such a tag may have no asset behind it, it may have an asset that is of no value to the attacker,

[2] Klensin, J. "RFC 5321: Simple Mail Transfer Protocol", `http://tools.ietf.org/html/rfc5321`, October 2008.

[3] Elz, R.*et al.*, "RFC 2181: Clarifications to the DNS Specification", `http://tools.ietf.org/html/rfc2181`, July 1997.

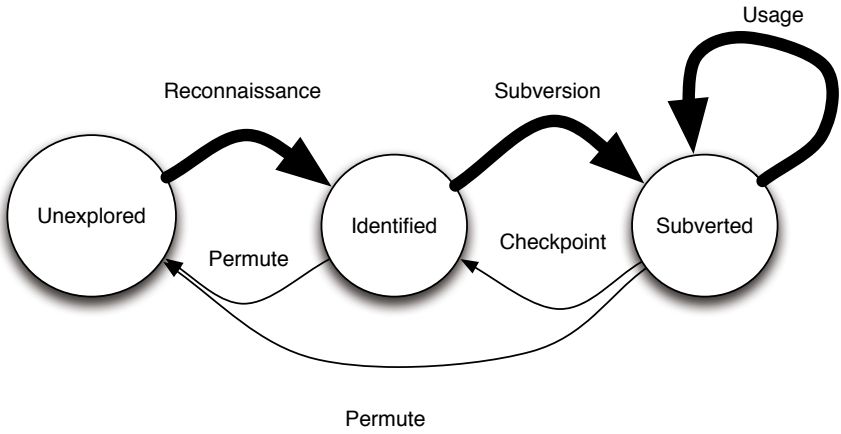


Fig. 1. Model of attacker knowledge, boldfaced arrows indicate the attacker's transitions

or it amy have a critical asset. An *identified* tag is one whose existence has been verified by the attacker, possibly through defender actions (such as publishing a server address), or by an attacker successfully identifying them (such as by scanning). Once an attacker identifies a tag, they believe that they have access to a unique resource. The attacker's ultimate goal is to *subvert* assets; an asset is subverted once an attacker takes control of that asset through any attack that offers it control. Once subverted, an attacker can exploit the asset as long as he can contact it – the state of the asset remains subverted.

To represent the true state of the network, and the attacker's perception thereof, we introduce two state functions: σ_{blv} and σ_{tru}. $\sigma_{\text{blv}}(t, \tau)$ represents the attacker's *belief* about an asset on the network, it takes a tag and returns the attacker's understanding about the asset pointed by the tag. $\sigma_{\text{tru}}(a, \tau)$ returns the true state of the asset at time τ. $\sigma_{\text{blv}}(t) = \sigma_{\text{tru}}(a, \tau)$ iff $\ell(t, \tau) = a$ *and* the state the attacker believes t is in is the same as the state a is in.

We represent the attacker's chance of success using a base probability. This probability $\mathcal{P}_{\sigma}^{\text{atk}}$ is the attacker's chance of successfully advancing the state of an asset from σ to its following state if the attacker is able to successfully connect to the target. The probability of success is exclusively a function of σ: $\mathcal{P}_{\text{unx}}^{\text{atk}} = 1$, as the attacker will always found out something about the tag. $\mathcal{P}_{\text{subverted}}^{\text{atk}} = 1$, as the attack will always be able to control a previously subverted host. $\mathcal{P}_{\text{id}}^{\text{atk}}(\tau)$ will vary based on the type of attack; in §3, we will show several examples.

Attacks take time. We will use the notation $\lambda_{\sigma}^{\text{exp}}$ to refer to the time required for $\mathcal{P}_{\sigma}^{\text{atk}}$ to execute, regardless of its success. The probability that an attacker can execute $\mathcal{P}_{\sigma}^{\text{atk}}$ is expressed independently of whether the attacker can successfully contact the target of the attack. Determining that probability is a function of the defense, and will be described in §2.3.

2.3 Defenses

We now address the issue of defense. As discussed in §1, network-based moving target defenses obfuscate the relationship between tags and assets. For this paper we discuss four types of moving target defense: *permutation*, *ephemeralization*, *checkpoint*, and *replication*. For the sake of simplicity in our methodology, the defense is expressed in terms of a lifetime λ. Each defense modifies the lookup function $\ell(t, a, \tau)$ over time; within a lifetime (*e.g.*, $\tau \in [0, \lambda]$), ℓ will return a consistent result. After each λ, however, the values returned by ℓ will change.

We express the attacker's success rate in an attack as $\mathcal{P}_\sigma^{\text{con}}(t, a, \tau)$. $\mathcal{P}_\sigma^{\text{con}}$ is the probability of the attacker using the tag t connecting to the asset a. This $\mathcal{P}^{\text{con}}$ probability represents the intersection between the attacker's model of the network (derived from intelligence he has gathered) and the target network's true configuration, which changes due to the defense. Recall from §2.2 that the attacker's probability of successfully carrying out an attack is $\mathcal{P}_\sigma^{\text{atk}}$, this probability is independent, yielding an aggregate probability of success of $\mathcal{P}_\sigma^{\text{con}}(t, a, \tau) \cdot \mathcal{P}_\sigma^{\text{atk}}$.

A *permutation* defense involves moving assets around the tag space over time. At any time, each asset is A is assigned a tag in T, and at the end of every lifetime, each asset is assigned to a new tag randomly. An exemplar form of permutation defense is Antonatos*et al.*'s NASR [1], which randomly shuffles IP addresses within a constrained netblock at fixed intervals.

A permutative defense means that any knowledge an attacker has on the network is reduced in value after λ. This change does not impact $\mathcal{P}_{\text{unx}}^{\text{con}}$; the attacker's chance of contacting an asset remains effectively random at $\frac{|A|}{|T|}$. However, the likelihood that a particular asset will remain at the same tag after λ is $\frac{1}{|T|}$. Consequently, attacks relying on this knowledge (idand sub), will have their probability of success reduced.

An *ephemeralization* defense involves using short-lived tags for an asset. In order for ephemeralization to be effective, $|T|$ must be considerably larger than $|A|$, large enough that the likelihood of recycling tags is low. If both sets are close in size, then permutation is a better model of the behavior. In an ephemeralization defense, a tag is assigned to an asset for the lifetime λ, and the asset is only accessible during that lifetime.

An ephemeralization defense means that any knowledge an attacker has on the network is *destroyed* after λ, as the tag is removed from use and new tags are put in place. As with permutation, the probability of discovery remains the same: $\frac{|A|}{|T}$, with the caveat that for a practical ephemeralization defense, $|T| \gg |A$. For other attacks, the probability of success is zero after λ because the targeted tag is no longer used.

In comparison to the permutation and ephemeralization defenses, the *checkpoint* defense does not change the relationship between assets and tags. Instead, the checkpoint defense changes the assets themselves. After every λ, each asset is returned to a checkpointed state. Mechanically, this impacts the system by

changing the state of any asset a where $\sigma(a) = \text{sub}$ to id. The attacker loses control of any target, and must take them over again.

Checkpointing does not affect the attacker's intelligence gathering capabilities. Any maps of the network he developed remain the same after λ as before. Furthermore, if an attacker has identified the host as subvertible, it remains so – the checkpointing does not change the potential for the system to be taken over, just the actual state. Consequently, the only attack the attacker disrupts is $\mathcal{P}^{\text{atk}}_{\text{sub}}$, as the attacker finds that he must retake the system after each lifetime.

A *replication* defense involves hiding multiple assets behind a single tag. Examples of replication appear repeatedly in load-balancing literature, and have been modified as DDoS defenses [7]. We model replication defenses by creating a subset of A referred to as a pool, $R \subset A$. For the purposes of this paper, we will divide A into a number of equally sized pools – all assets will belong to one pool, and each pool will consist of the same number of assets. Replication defenses will return a specific member of the pool with each request, if $R_i \equiv \{a_1, a_2, a_3\}$ and t_i is linked to R_i, then $\ell(pool_i)$ will always be either a_1, a_2 or a_3. The linking will be assigned randomly and changes each request.

Replication defenses are the only form of defense that adversely affect's an attacker's ability to scan a network, by reducing the number of tags used by the defender. As a result of this, the attacker's ability to find a host is reduced by the size of the pools in the defended network. Replication also impacts the attacker's ability to take over an asset, as the asset pointed to by the tag changes randomly among the pool. This reduces the attacker's chance of continuity to $\frac{1}{|R|}$. The same probability affects the attacker's ability to exploit an already controlled host, and is unique among the defenses in that it is largely independent of λ.

Table 1 summarizes the impact of moving target defenses on the different types of attacks. The values in Table 1 describe $\mathcal{P}^{\text{con}}_{\sigma}$ where $\sigma \in \{\text{unx}, \text{id}, \text{sub}\}$.

Table 1. Probability of Success For Each Attack Model Against Defense

Defense	$\mathcal{P}^{\text{con}}_{\text{unx}}$		$\mathcal{P}^{\text{con}}_{\text{id}}$		$\mathcal{P}^{\text{con}}_{\text{sub}}$	
	$t \leq \lambda$	$t > \lambda$	$t \leq \lambda$	$t > \lambda$	$t \leq \lambda$	$t > \lambda$
Nothing	$\frac{\lvert A\rvert}{\lvert T\rvert}$	$\frac{\lvert A\rvert}{\lvert T\rvert}$	1	1	1	1
Permutation	$\frac{\lvert A\rvert}{\lvert T\rvert}$	$\frac{\lvert A\rvert}{\lvert T\rvert}$	1	$\frac{1}{\lvert T\rvert}$	1	$\frac{1}{\lvert T\rvert}$
Ephemeralization	$\frac{\lvert A\rvert}{\lvert T\rvert}$	0	1	0	1	0
Checkpoint	$\frac{\lvert A\rvert}{\lvert T\rvert}$	$\frac{\lvert A\rvert}{\lvert T\rvert}$	1	1	1	0
Replication	$\frac{\lvert A\rvert}{\lvert T\rvert}$	$\frac{\lvert A\rvert}{\lvert T\rvert}$	$\frac{1}{\lvert R\rvert}$	$\frac{1}{\lvert R\rvert}$	$\frac{1}{\lvert R\rvert}$	$\frac{1}{\lvert R\rvert}$

Table 1 and the probabilities of attack success together define the likelihood that an attacker will succeed or fail to communicate with a target. The final step in the model is to now determine how the attacker's actions impact their ability to achieve their goals.

3 Evaluation

We now consider the capability of moving target defenses to constrain attacker behavior. In order to do so, we construct a zero-sum game. For each game, the payoff function is attacker driven – the attacker has a goal to achieve, and the defender's success is evaluated in his ability to frustrate that goal. To demonstrate the flexibility of our approach, we consider two attack scenarios. In the first, the attacker attempts to subvert hosts to use them for spamming or other attacks, in the second, the attacker attempts to DoS a target.

The remainder of this section is structured as follows. §3.1 looks at a model for a compromise-based attack. §3.2 looks at Denial of Service.

3.1 First Scenario: Subversion and Control

We now consider a general model for a subversion and control attack. Here, the attacker's goal is to subvert as many hosts on the targeted network as possible. To model this, we construct an inventory of the damage the attacker has done to the network of the form (a, τ, σ), where a is an asset in the asset set, $\tau$is the time, and *state* is the true state of the asset linked to the tag. Note that the true state of the assets is not known to the attacker, and may not be accessible (such as in replication defenses, where multiple assets share a tag).

The attacker's goal is to control the network. This is accomplished by converting the state of an asset from unexplored to controlled via progressive attacks. At any time, the attacker's control over the network can be expressed as a sum over all the assets:

$$\mathcal{C}(A, \tau) = \sum_{a \in A} \begin{cases} 1 & \text{if } \sigma_{\text{tru}}(a, \tau) = \text{sub} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Given a network with identical A and T, one moving target strategy is preferable to another one if over time, $\mathcal{C}$ is smaller for the one strategy than the other. $\mathcal{C}$ is best evaluated over multiple lifetimes, as an attacker's impact on the network will change every λ in accordance with the different defensive strategies. We will now use the methodology described in §2 to examine various forms of moving target defense under specific network configurations. For this exercise, we consider a simple buffer overflow. In the first stage of the attack, the attacker scans hosts on the network, and based on intelligence gathered, crafts a specific attack for each target. During the second phase, he strikes and subverts the hosts.

Evaluating a defense is a three step process: the first step requires describing the network and determining values for A and T. The second step involves describing the attack, determining attacker behavior and building a model for $\mathcal{P}^{\text{atk}}(\text{id})$. The final step involves choosing a moving target defense and corresponding λ. Once these steps are done, we can use Table 1 to calculate the various $\mathcal{P}_{\sigma}^{\text{con}}$ probabilities, and finally $\mathcal{C}$.

We assume that the defender is protecting a /24 and that the attacker is exclusively interested in SSH traffic. This produces a tag space consisting of

every IP address in the /24 at port 22. A consists of all hosts on the network which are running SSH servers, which we will set at 16 for this exercise. Note that purposes of this model, a host that is not running SSH is *not* an asset, and ℓ will return $\varnothing$ in that case. $|T|$ is therefore 256, and $|A|$ is 16.

For the attack proper, we assume that the attacker is perfectly successful, *i.e.*, $\mathcal{P}_{\text{id}}^{\text{con}} = 1$, and that $\lambda_{\text{id}}^{\text{exp}} = 0$. If the attacker can find the same target he scanned, he can subvert that target instantly.

Using our different defensive strategies, we can now calculate $\mathcal{C}$. We first note that ephemeralization is not valid for this network, as the asset space and target set are close in size. For permutation and virtualization, we will set λ to 8 hours, the course of a single workday, and we assume the attacker will conduct the targeted attack after the blind scan.

In the permutative scenario, the attacker engages in a targeted attack against the hosts he identified in the blind scan. However, based on Table 1, the attacker will find the chance of successfully finding the same target to be $1/|T|$. We assume that if the attacker does find the target, he takes it over instantly. If he does not find the target, he rescans and tries again after the attack. Over time, the attacker will slowly subvert the network.

In the checkpoint scenario, the attacker instantly subverts his targets at the beginning of the attack. However, after every λ, the hosts will revert to their uncontrolled state and the attacker will have to subvert them again.

Finally, in the replication scenario, the attacker's chance of subverting a host is *increased* relative to the permutative scenario, as the targets no longer migrate across the entire tag space but in their constrained pool. The net result is that the attacker's ability to subvert the network is greatly increased.

Figure 2 shows the impact of these different defensive mechanisms on the defender's rate of success over time. As this figure shows, the most consistent results come from the permutation and replication defenses, which result in a steadily degrading network capability. The checkpointing defense, in comparison, rapidly oscillates between states of complete attacker control and no attacker control.

The evaluation provided in this section is intended as a proof of concept. In order to implement an analytical solution, we have necessarily simplified attacker and defender behavior. However, we have been able to demonstrate that we can compare different classes of defense and translate real-world values (the number of IP addresses in the network) into the evaluation.

3.2 Second Scenario: Denial of Service

We now consider a Denial of Service (DoS) attack. For the purposes of this model, we treat denial of service as focused on the target rather than the network (*i.e.*, the attack is not strong enough to disrupt network service as a secondary effect). A historical example of a moving target defense applied to Denial of Service is the defense against the original Code Red Worm[4]. In the case of the original

[4] Lemos, R. "Web Worm Targets White House", July 19, 2001. `http://news.cnet.com/2100-1001-270272.html`

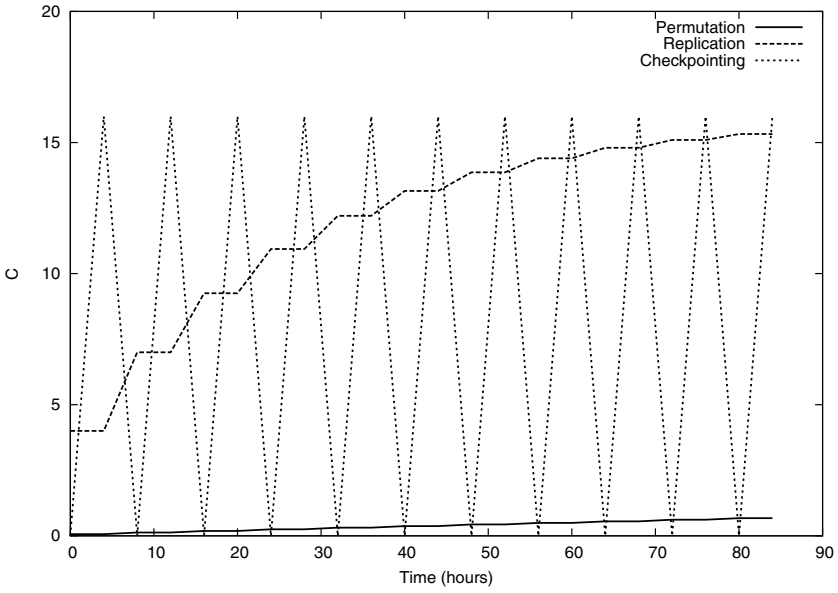


Fig. 2. Attacker success against different forms of moving target defense

Code Red worm, the attack was evaded by moving the White House's website to a new IP address, therefore invalidating the original hard-coded address used by the attackers.

DoS, in comparison to the subversion attack discussed in §3.1 does not require the use of a particular vulnerability or exploit. Instead, a DDoS attack will simply rely on marshalling a sufficient volume of bots to take out a target sending bogus requests or other hostile traffic. Because it does not subvert hosts, the attacker's success is a function of $\mathcal{P}^{\text{con}}\text{id}(a)$ – the probability of connecting to a known host.

Equation 3 represents the effectiveness of a single host attempting to DoS one or more targets given a set of target tags $G \subset T$.

$$\mathcal{C}(G,\tau) = \sum_{g \in G} \begin{cases} 1 & \text{if } \sigma_{\text{tru}}(a,\tau) = \text{id} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Based on Table 1, we can calculate the effective impact of different defenses. The relative simplicity of a DoS attack, in comparison to the subversion and control scenario in §3.1, enables us to develop purely symbolic answers.

We consider permutative and replication defenses, ephemeralization is impractical for a person network, and replication does not address the resource exhaustion caused by a DDoS. For a replication defense, the attacker's effective impact is $|G|$ if $\tau \leq \lambda$ and $\frac{|G|}{|T|}$ afterwards, when the probability of still having a target at the same asset is $\frac{1}{|T|}$. For a replication defense, the effetiveness is

always $\frac{|G|}{|R|}$, as the pool effectively reduces the impact of the attack by spreading it across multiple hosts.

4 Previous Work

Several forms of moving target defenses have been proposed in the past decade as a mechanism for mitigating both intelligence gathering and scanning techniques, as well as a means for evading DDoS and resource exhaustion attacks. The most notable defenses include NASR [1] and TAO [2], as well as the BBN DYNAT [9]. All of these approaches use permutative defenses on IP addresses; DYNAT is particularly notable for the author's use of pseudorandom address permutation – addresses are predictable using a shared key, enabling members of the community to predict network addresses and communicate reliably. These approaches are all constrained, however, by focusing exclusively on IP addresses.

Attackers use moving target defenses more aggressively, and several forms of attacks have used various moving target strategies to evade detection. Conficker [12] used short-lived DNS addresses, also with cryptographic sharing, to manage download sites. A similar example of ephemeralization is used by fast-flux networks [11] which use short-lived DNS addresses as a mechanism for evading blacklists.

Our evaluation strategy builds on previous work [6, 8, 5, 13, 4] on game based intrusion evaluation. These works provided a general framework for evaluating attack and defense, but are not focused on the specific issue of moving target defense. Our work differs by applying a new family of defensive strategies and trying to develop a common framework for them.

5 Conclusions and Future Work

In this paper we have developed an devaluated a simple game-based model for evaluating moving target defensive strategies. Moving target approaches to network engineering have been used practically for a variety of performance and defense-related reasons, but these approaches are generally treated as singular artifacts. Our work is an effort to develop a common framework for evaluating these systems and provide a means to determine how well, based on network configuration and structure, a particular defense will work.

This work is very much an early effort. The models in place in this paper were developed in order to provide an analytically viable solution, and in the course of doing so many details of attack, defense and timing were intentionally simplified. We believe that we have reached the upper limit of a meaningful analytic solution, however. Further work will need to implemented using a simulation-based solution, using the same language for success rates (some form of Equation 2, but accommodating more complex network configurations. Ultimately, our goal is to develop a simulator where we can input a current network inventory and evaluate each strategy in turn.

Problems which will be viable to address in the simulation solution include the need for heterogeneity. The current model assumes that members of A are distinguishable, however many operational environments use homogenous assets, or multiple groups of identical assets such as all apache servers or all IIS servers. Similarly, defender response must be expanded; in the current model, defenders do not react to attackers at all. However, one of the advantages of a moving target defense is that it not only moves defenders, it provides more opportunities for an attacker to identify themselves. In future releases of the model, we expect to include reactive defenses such as blocking and honeypots which will require more sophisticated attacker response.

References

1. Antonatos, S., Akritidis, P., Markatos, E.P.: Defending against hitlist worms using network address space randomization. In: Proceedings of the 3rd ACM Workshop on Rapid Malcode (WORM) (2005)
2. Antonatos, S., Anagnostakis, K.G.: TAO: Protecting against Hitlist Worms Using Transparent Address Obfuscation. In: Leitold, H., Markatos, E.P. (eds.) CMS 2006. LNCS, vol. 4237, pp. 12–21. Springer, Heidelberg (2006)
3. Caglayan, A., Toothaker, M., Drapaeau, D., Burke, D., Eaton, G.: Behavioral analysis of fast flux service networks. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (2009)
4. Cai, J.-Y., Yegneswaran, V., Alfeld, C., Barford, P.: An Attacker-Defender Game for Honeynets. In: Ngo, H.Q. (ed.) COCOON 2009. LNCS, vol. 5609, pp. 7–16. Springer, Heidelberg (2009)
5. Cárdenas, A., Baras, J., Seamon, K.: A framework for evaluation of intrusion detection systems. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy (2006)
6. Collins, M.: Payoff based ids evaluation. In: Proceedings of the 2nd Annual CSET Workshop on Computer Security Experimentation and Test (2009)
7. Davis, B.: Leveraging the load balancer to fight DDoS. In: SANS GIAC Gold Certification Report (2009)
8. Gaffney, J., Ulvila, J.: Evaluation of intrusion detectors: A decision theory approach. In: Proceedings of the 2001 IEEE Symposium on Security and Privacy (2001)
9. Kewley, D., Fink, R., Lowry, J., Dean, M.: Dynamic approaches to thwart adversary intelligence gathering. In: DARPA Information Survivability Conference and Exposition, vol. 1 (2001)
10. Krishnamurthy, B., Wills, C., Zhang, Y.: On the use and performance of content distribution networks. In: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (2001)
11. Moore, T., Clayton, R.: Examining the impact of website take-down on phishing. In: Proceedings of the Anti-phishing Working Groups 2nd Annual eCrime Researchers Summit (2007)

12. Shin, S., Gu, G.: Conficker and beyond: a large-scale empirical study. In: Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC 2010 (2010)
13. Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P.: Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In: Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (2000)
14. Tyma, P.: The architecture of mailinator

Are We Compromised? Modelling Security Assessment Games

Viet Pham and Carlos Cid

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, United Kingdom
{viet.pham.2010,carlos.cid}@rhul.ac.uk

Abstract. Security assessments are an integral part of organisations' strategies for protecting their digital assets and critical IT infrastructure. In this paper we propose a game-theoretic modelling of a particular form of security assessment – one which addresses the question "are we compromised?". We do so by extending the recently proposed game "FlipIt", which itself can be used to model the interaction between defenders and attackers under the Advanced Persistent Threat (APT) scenario. Our extension gives players the option to "test" the state of the game before making a move. This allows one to study the scenario in which organisations have the option to perform periodic security assessments of such nature, and the benefits they may bring.

1 Introduction

The protection of digital assets and critical IT infrastructure is an ever-growing concern for individuals, companies and nations. Information security is now a priority area for investment, given the growing threats from hackers, competitors, organised criminal gangs and enemy nation-states, and the potential for loss of privacy and revenue, negative reputational impact and effects in public welfare. In addition to direct investment in suitable and robust IT infrastructure, the performance of frequent security assessments is also considered an important component of the defense strategy against cyber-attacks. A security assessment is the process of determining how effectively an entity being assessed meets specific security objectives [11]. A common method of assessment is a *penetration testing*, where security professionals target the network and other IT resources, to try to identify and verify any vulnerabilities found. Popular penetration testing methodologies and frameworks work by essentially *mimicking* the popular forms of attack used by hackers.

The nature of cyber attacks has however been steadily changing in recent years. While previously the typical threats were *script kiddies*, more interested in defacing websites for fun and pride, attacks motivated by financial gains are increasingly becoming more prevalent. Particularly in the corporate and government spheres, the threat of espionage and theft of intellectual property and state secrets are growing causes of concern. With these goals in mind, the methods

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 234–247, 2012.

used by attackers have also evolved. A form of attack that has received much attention recently are the so-called *Advanced Persistent Threats* (APT), which can often be seen as a signal of international cyber warfare [2]. The premises in this form of attack are that IT networks and systems *are* vulnerable, and therefore can be compromised by adversaries with enough resources and motivation; furthermore, attacks are stealthy in nature [15,5], and adversaries can remain in control of the network and systems for long periods without detection. Recent examples of cyber attacks that fit this profile are the security breach at RSA Data Security [6], and the Stuxnet [9] worm infection of Iranian systems.

These developments should in turn motivate a reflection on whether current methods of security assessment remain sound under the changing nature of attacks. A security assessment is typically seen to be trying to answer the question "are we vulnerable?" (and if so, how can we fix it?). Under APT's premise, the answer for this question is certainly "yes". Thus a security assessment needs also to address the question "are we compromised?", and organisations need to consider cost-effective ways in which they can can regain control of their IT assets if the answer is positive. This current gap should certainly be the cause of concern for professionals involved in the security of highly-targeted organisations.[1]

In this paper we propose a simple game-theoretic modelling of this form of security assessment, and study its application in 2-player security games. Game modelling has been shown to be useful in studying strategic decisions toward a wide range of security problems, from technical [10] to managerial [8,13]. Our model extends the recently proposed game "FlipIt" [14], which itself can be used to model the interaction between defenders and attackers under the APT scenario. Our extension gives players the option to "test" the state of the game (i.e. answer the question "are we compromised"). This allows one to study the scenario in which organisations have the option of performing periodic security assessments of such nature, and the benefits they may bring. In particular, how these assessments can fit into an organisation's security investment strategy. Proposals of models for security investment and security testing have appeared before in the literature (e.g. [7,4,3]); here we leverage on the elegance of FlipIt to investigate strategies for the application of this form of security assessment.

This paper is organised as follows. In Section 2 we describe the game "FlipIt". In Section 3 we propose our extension to the game, by introducing the option of a security assessment which discloses the state of the game. We study further extensions in Sections 4 and 5. We finish with our conclusions in Section 6.

2 FlipIt: The Game

The original FlipIt games [14] capture the battle between a defender and an advanced persistent threat (APT) attacker for the control of a resource. The

[1] In fact these points were emphatically argued in a recent testimony before the U.S.-China Economic and Security Review Commission Hearing on "Developments in China's Cyber and Nuclear Capabilities", where one of the participants stressed the need of periodic security assessments of the latter nature [1].

game is modelled over infinite time, in which a player makes a move to gain control of the resource; it remains in this state until the opponent makes its own move to take over. This control-alternating process repeats infinitely as time passes, and the utility of each player is determined by the total/average amount of time it controls the resource, as well as the cost required to take over the resource from its opponent.

Formally, the defender and the attacker are denoted by player 0 and 1, respectively. The game timeline starts from some moment $t = 0$ and is continuously indefinite, so that the amount of control time for each player can be conveniently computed in $\mathbb{R}$. Let $C_i(t)$ be 1 if player i controls the resource at time t, and 0 otherwise. For example, if the defender moves at time t, then $C_0(t) = 1$; similarly, we have $C_0(t') = 0$ if the attacker moves at time t'. This allows the total control time of player i until time t to be computed as

$$G_i(t) = \int_0^t C_i(t)dt.$$

Denote player i's number of moves until time t by $n_i(t)$, and the constant cost for each move by k_i; then the *net benefit* of player i is given by

$$B_i(t) = G_i(t) - n_i(t)k_i.$$

Alternatively, since the game continues indefinitely, a player's utility can be represented by its *average benefit* per unit time:

$$\beta_i(t) = \frac{B_i(t)}{t} = \frac{G_i(t)}{t} - \frac{n_i(t)}{t}k_i = \gamma_i(t) - \alpha_i(t)k_i.$$

We call $\gamma_i(t)$ and $\alpha_i(t)$ the *average gain rate* and the *average move rate* of player i up to time t, respectively. One may further assume that the functions $\gamma_i(t)$ and $\alpha_i(t)$ converge to the values γ_i and α_i, respectively, as $t \to \infty$. We can then conveniently represent player i's utility without the time dimension as simply

$$\beta_i = \lim_{t\to\infty} \beta_i(t) = \gamma_i - \alpha_i k_i.$$

What remains to be modelled are γ_i and α_i, which strongly depend on how the players strategically act in the game. While the authors in [14] discuss several types of strategies for each player, in this paper we focus only on the so-called *periodic strategies with random phase*, which is the main tool for our work. In periodic games, we assume that before start, each player chooses a rate $\alpha_i > 0$ so that as the game progresses, player i moves at rate α_i, i.e., after every $\delta_i = 1/\alpha_i$ units of time. Furthermore, player i does not start moving immediately at $t = 0$, but selects uniformly at random a starting point in the interval $[0, \delta_i]$; this is called *phase*. While player i cannot control its phase, its game action is determined by the chosen move rate α_i. For convenience, we denote the action space for periodic moving strategies for both players as

$$P = \{P_\alpha | \alpha > 0\}.$$

Since players move periodically, their expected average control time γ_i, or average gain, can be computed in the following two cases:

- $\alpha_0 \geq \alpha_1$: let $r = \alpha_1/\alpha_0 = \delta_0/\delta_1$; we note that for every attacker's period interval $[t^*, t^* + \delta_1]$, the defender moves at time t uniformly random within $[t^*, t^* + \delta_0]$, yielding a gain $t^* + \delta_1 - t$, which can be expectedly computed as

$$G_0^* = \int_{t^*}^{t^*+\delta_0} \frac{t^* + \delta_1 - t}{\delta_0} dt = \delta_1 - \frac{\delta_0}{2} = \delta_1(1 - \frac{r}{2}).$$

  This implies that the defender's average gain is $\gamma_0 = G_0^*/\delta_1 = 1 - r/2$; it also means that the attacker's average gain is $\gamma_1 = 1 - \gamma_0 = r/2$. Therefore, we have the players' utilities as

$$\beta_0(\alpha_0, \alpha_1) = 1 - \frac{r}{2} - \alpha_0 k_0 = 1 - \frac{\alpha_1}{2\alpha_0} - \alpha_0 k_0,$$
$$\beta_1(\alpha_0, \alpha_1) = \frac{r}{2} - \alpha_1 k_0 = \frac{\alpha_1}{2\alpha_0} - \alpha_1 k_1.$$

- $\alpha_0 \leq \alpha_1$: similar analysis gives the following

$$\beta_0(\alpha_0, \alpha_1) = \frac{r}{2} - \alpha_0 k_0 = 1 - \frac{\alpha_0}{2\alpha_1} - \alpha_0 k_0,$$
$$\beta_1(\alpha_0, \alpha_1) = 1 - \frac{r}{2} - \alpha_1 k_0 = \frac{\alpha_0}{2\alpha_1} - \alpha_1 k_1.$$

We note that when a player has lost the control due to the opponent's move, it does not immediately move to regain it but rather needs to wait for its periodic move. This is because moves are presumably "stealthy", and neither player knows at any time who is controlling the resource. In addition to the periodic move scenario, [14] also studies strategies involving randomised moves, as well as adaptive strategies based on the opponent's past moves. Although we do not consider these here, we note that the modelling presented in this paper may be similarly applied to other scenarios discussed in [14].

The main reason for choosing FlipIt to base our work on is its simple, though elegant, modelling of real-world IT security defender-attacker interaction. Indeed, strategies for organisational security are often determined in the very early phase of the business, and they are normally deterministic (quarterly assessments, periodic guard patrolling, etc.) rather than being oblivious and temporary [12]. In addition, as information systems become more sophisticated in size and structure, and the motivation and nature of attacks change, it is becoming more difficult at any moment to be certain whether resources are secure, hence allowing "stealthy" moves to be realistic. In the next sections, we propose an extension to the above model as an attempt for the defender to more efficiently counter such moves.

3 Test It Before Flipping It

The original FlipIt game models types of strategies for a player to regain control of a resource (i.e. to move) based on some pre-defined or on-the-fly tactics,

which however possess some limitations. In particular, a player may waste many moves if they happen while it is still controlling the resource. This becomes more serious if its periodic movement is significantly faster than the opponent's. Even if a move really serves its purpose, i.e., to regain control, it may still be an "almost" waste. This happens, for example, when the opponent's move is immediately (but coincidently) after such a move, rendering it ineffective.

Rather than blindly moving, an interesting question is whether knowing the state of control would be more beneficial to a player. In terms of information security assessment, this can be represented by the question "are we compromised?". The intuition behind this addition is rather simple. Knowing the state of control would prevent a waste move while the resource is still at hand. Also, even though it may not prevent an "almost" waste, it may suggest a timely response to a lost of control. This, of course, depends on how regularly the knowledge of the control state is updated.

To model such situations, we introduce a new class of strategies to FlipIt, namely the *state checking* strategies. As opposed to the ability to move/flip, a player is now able to check the game state, and then move/flip if necessary. In particular, we consider a strategy class $S = \{S_\alpha | \alpha > 0\}$ such that, given a strategy $S_\alpha \in S$, with $\delta = 1/\alpha$, player i may:

- perform a periodic state checking with period δ and cost u_i, with the first check occurring at a uniformly random time phase, i.e., within $[0, \delta]$;
- if a state check indicates a loss of control, immediately perform a move/flip (at cost k_i) to regain its control.

In addition to the original game FlipIt(P, P), several games might be introduced given S, for example FlipIt(S, P), FlipIt$(S \cup P, P)$, and FlipIt$(S \cup P, S \cup P)$. To study such games, it is important to notice that in all cases, the expected control time for each player can be formulated in the same way as that in $\{P, P\}$, using only δ_0 (or α_0) and δ_1 (or α_1). Indeed, at a time t, if a player is occupying the resource, a blind move action and a check-then-move action would yield the same effect, i.e., allowing it to regain control. Likewise, while it is in control of the resource, neither of the moves would bring any change. As this happens independently of the opponent's strategy, the same expected control time can be used for any game with strategies restricted to S and P.

Since a player's utility depends only on its expected control time and the cost of moving and/or checking, it is also independent of the opponent's type of strategy. Indeed, player 0 with strategy P_{α_0} would for example have a benefit as mentioned in Section 2

$$\beta_0(\alpha_0, \alpha_1) = \begin{cases} 1 - \frac{\alpha_1}{2\alpha_0} - k_0\alpha_0 & \text{if } \alpha_0 \geq \alpha_1 \\ \frac{\alpha_0}{2\alpha_1} - k_0\alpha_0 & \text{if } \alpha_0 < \alpha_1 \end{cases}.$$

With a strategy S_{α_0}, the average state checking cost for player 0 is $\alpha_0 u_0$. For moving cost, since S_{α_0} is employed, no move is wasted, thus player 0's number of moves is at most player 1's number of moves, i.e., $\min(\alpha_0, \alpha_1)$. This allows the construction of its utility to become

$$\beta_0(\alpha_0, \alpha_1) = \begin{cases} 1 - \frac{\alpha_1}{2\alpha_0} - u_0\alpha_0 - k_0\alpha_1 & \text{if } \alpha_0 \geq \alpha_1 \\ \frac{\alpha_0}{2\alpha_1} - u_0\alpha_0 - k_0\alpha_0 & \text{if } \alpha_0 < \alpha_1 \end{cases}. \quad (1)$$

Given this new type of strategies S, a natural approach is to compare between S and P, that is, in which situations one is preferred over the other. The following theorem provides such comparison based on the relation between the costs of moving and state checking[2].

Theorem 1. *In the game FlipIt($P \cup S$, $P \cup S$), if $u_i \leq k_i/4$, player i does not prefer periodic moving. Otherwise, when $u_i \geq k_i$ player i does not prefer state checking.*

Proof. This theorem can be proved as a special case of Theorem 5, when $p = 1$.

Corollary 1. *Consider the game FlipIt($P \cup S$, $P \cup S$) with $k_i/4 < u_i < k_i$. Player i prefers a state checking strategy if and only if $\alpha_{1-i} \leq \frac{2(\sqrt{k_i}-\sqrt{u_i})^2}{k_i^2}$.*

The above results point out that when the cost of checking is sufficiently low, i.e., at most a quarter of the moving cost, it is always worth performing a check-then-move strategy. Indeed, as a low checking cost suggests a frequent checking schedule, a player is more closely up-to-date with its state of control of the resource. This helps the player to improve its expected control time, while keeping the moving cost at a reasonable level by eliminating wasted moves. Conversely, it is also intuitively clear that when the cost of checking exceeds that of moving, it is unreasonable to perform checking-then-moving. Furthermore, Corollary 1 indicates that, when the two cost are comparable, the best response for the opponent playing too fast is to either simply move at every step or not play at all, because at every step it is likely that without state checking the player is aware of its loss of control of the resource.

In the realm of information security, many situations may suggest that state checking strategies indeed outperform their moving counterparts. Consider an information system as the resource; the defender's act of moving/flipping is often expensive, as it might involve resets and restores of the system. This becomes more serious for large organisations, or those that require uninterrupted, real-time system availability and reliability, such as e-commerce, large computing facilities. On the other hand, checking for successful take-over of the system might be significantly cheaper and thus can be performed frequently, using intrusion detection systems (IDSs), auditing schemes, logging, etc. In such cases, it is recommended that funds are allocated for more frequent auditing of the system security to maximise the organisational benefit from the information system.

In another aspect, we recall from [14] that the game FlipIt(P, P) has a Nash equilibrium. As this game behaves similarly to infinitely repeated games, the equilibrium indicates the stage to which the game would eventually converge

[2] All proofs can be found in the full version of this paper at `http://eprint.iacr.org`

if both players kept adjusting their actions upon realisation of the opponent's action. In the game FlipIt(S, S) however, such stable stage does not exist, as we show in Theorem 2. The intuition behind is reasonably simple. We notice that the total moving cost, i.e., $k_i \min(\alpha_i, \alpha_{1-i})$ for each player does not just depend on that player's rate, but also on its opponent. Thus, if a player keeps increasing its rate until it is faster than the opponent's, then its total moving cost will stop rising. This in turn results on a better chance for that player to come across a rate (possibly faster than the opponent) yielding higher benefit. This fact emphasises that when such situation occurs, the players' strategies are unstable, and it is best for a player to always monitor its opponent's state checking frequency and adjust its accordingly. In real life, this lack of stability suggests that the defender must keep consulting the statistics on how often attacks occur and adapt its strategy accordingly.

Theorem 2. *The Game FlipIt(S, S) has no pure strategy Nash equilibrium.*

4 Hardening Control over Time

Besides reactive measures such as state checking and moving, a proactive concern is on how to prevent losses of control from happening. In many cases this is more desirable because it is possible that consequences from attacks might have been overlooked, and thus it is better that attacks are prevented given the current realisation of potential losses. In the context of FlipIt, it may mean, for example, preventing a player from participating in the game, or to stop it after the game has run for some time. Following the analysis of the original FlipIt game, as well as those involving state checking strategies, it is not difficult to see that in order for a player to stop its opponent from participating in the game, it needs to play quick enough. Based on the best response functions for periodic moving and periodic state checking players, the rate limit above which player i should play so as to prevent its opponent from engaging on the game is

$$\alpha_i^{threshold} = \max\left(\frac{1}{2k_{1-i}}, \frac{k_{1-i} + u_{1-i} - \sqrt{u_{1-i}(2k_{1-i} + u_{1-i})}}{k_{1-i}^2}\right).$$

While this is desirable, it is sometimes infeasible to play fast enough if the state checking cost is high. A different preventive approach for a player is to somehow make it increasingly more difficult for its opponent to take over the resource over time. When the level of difficulty reaches some threshold, its opponent will automatically cease playing, and thus resulting in a long-term benefit for the player. In FlipIt type of games, this can be modelled by having a player spending an additional *periodic hardening cost* h_i every time it regains control, so that the opponent would have to spend more and more whenever trying to take over the resource. This cost could feature, for example, some penetration testing process that results in vulnerabilities being patched, similar to that modelled in [3]. It modifies the net utility of player i who performs state checking with hardening as follows, with $m_i(t)$ being the number of state checks occurred prior to t:

$$B_i(t) = G_i(t) - (k_i + h_i)n_i(t) - u_i m_i(t).$$

In this section we aim to study how the defender selects its strategy based on the observed attacker's period. For the game analysis, we note that the utility of a player is represented by its average benefit since the game is infinite. The game in this section is however finite, and thus it is more reasonable to represent player i's utility as its net benefit over the whole game, i.e., $B_i(t_{end})$ where t_{end} is the moment in which the game ends. Assume that given a hardening cost h_0, the game ends after s state-changing attacks (i.e. flipping the state from the defender to the attacker). Since it is not difficult to see that such an attack occurs for every $\max(\delta_0, \delta_1)$ period, we may assume for simplicity that

$$t_{end} = s \cdot \max(\delta_0, \delta_1).$$

To analyse this game, we first model the utility function for each player. This can be done with two cases similar to the previous games.

- $\alpha_0 \geq \alpha_1$: similar to other periodic FlipIt games, the expected control time for the defender (i.e. player 0), is $(1 - r/2)\delta_1$ per δ_1, for $r = \alpha_1/\alpha_0$. We thus have the defender's utility as:

$$\begin{aligned} B_0(s\delta_1) &= (1 - \frac{r}{2})t - (k_0 + h_0)n_0(t) - u_0 m_0(t) \\ &= (1 - \frac{\delta_0}{2\delta_1})s\delta_1 - (k_0 + h_0)s - u_0 s\frac{\delta_1}{\delta_0} \\ &= s\left[\left(1 - \frac{\delta_0}{2\delta_1}\right)\delta_1 - k_0 - h_0 - u_0\frac{\delta_1}{\delta_0}\right]. \end{aligned}$$

  However, since different choices of h_0 yield different end times $t_{end} = s\delta_1$, it would be unreasonable to consider utility as the net benefit only until t_{end}. Indeed, consider h_0 and h_0' that yield ending time t_{end} and t'_{end} with net benefit B_0 and B_0', respectively, such that $t_{end} < t'_{end}$ and $B_0 < B_0'$. Even though $B_0 < B_0'$, this does not mean that the defender would prefer h_0' over h_0 since within the interval $[0, t'_{end}]$, the defender's net benefit would be $B_0 + (t'_{end} - t_{end})$, which might still be greater than B_0'.
  To resolve this issue, consider two choices of hardening costs h_0 and h_0' yielding different attack times s and s', with $s' > s$. The defender's net benefit within $[0, s'\delta_1]$ in these cases are respectively

$$B_0^* = B_0 + (s'\delta_1 - s\delta_1) = s\left[\left(1 - \frac{\delta_0}{2\delta_1}\right)\delta_1 - k_0 - h_0 - u_0\frac{\delta_1}{\delta_0}\right] + \delta_1(s' - s)$$

$$\text{and} \quad B_0' = s'\left[\left(1 - \frac{\delta_0}{2\delta_1}\right)\delta_1 - k_0 - h_0' - u_0\frac{\delta_1}{\delta_0}\right].$$

  By subtracting the latter to the former we get:

$$B_0' - B_0^* = s\left[\frac{\delta_0}{2\delta_1}\delta_1 + k_0 + h_0 + u_0\frac{\delta_1}{\delta_0}\right] - s'\left[\frac{\delta_0}{2\delta_1}\delta_1 + k_0 + h_0' + u_0\frac{\delta_1}{\delta_0}\right].$$

This implies that h_0' is preferred over h_0 if and only if

$$s\left[\frac{\delta_0}{2\delta_1}\delta_1 + k_0 + h_0 + u_0\frac{\delta_1}{\delta_0}\right] \geq s'\left[\frac{\delta_0}{2\delta_1}\delta_1 + k_0 + h_0' + u_0\frac{\delta_1}{\delta_0}\right].$$

As a result, we may effectively represent the defender's utility function in the following form:

$$U_0(\delta_0, h_0) = -s\left[\frac{\delta_0}{2\delta_1}\delta_1 + k_0 + h_0 + u_0\frac{\delta_1}{\delta_0}\right], \tag{2}$$

where the defender's action is a pair $(\delta_0, h_0) \in H_0$ implying the chosen state checking frequency (period) and hardening cost.

- $\alpha_0 \leq \alpha_1$: let $r = \delta_1/\delta_0 = \alpha_0/\alpha_1$. With similar reasoning as in the previous case, together with $n_0(t) = n_1(t) = s$ (due to alternating control) and $n_0(t) = m_0(t)$ (since $\alpha_0 \leq \alpha_1$) we have that the defender's net benefit is

$$\begin{aligned} B_0(s\delta_0) &= \frac{r}{2}t - (k_0 + h_0)n_0(t) - u_0 m_0(t) \\ &= s\left(\frac{\delta_1}{2\delta_0}\delta_0 - k_0 - h_0 - u_0\right). \end{aligned}$$

This leads to the defender's actual utility function as:

$$U_0(\delta_0, h_0) = -s\left[\left(1 - \frac{\delta_1}{2\delta_0}\right)\delta_0 + k_0 + h_0 + u_0\right]. \tag{3}$$

To complete the defender's utility function, it is important to compute s, the number of attacks, from the hardening cost h_0 and the original attack cost k_1. This can be generally modelled with a function f, such that at the s-th attack, the attack cost becomes $f_{h_0}^{s-1}(k_1)$, where $f_{h_0}(k_1) = f(k_1, h_0)$ gives the new cost of an attack due to h_0. The attacks stop at the $(s+1)$-th attempt if the cost involved is greater than the attacker's expected control, i.e.,

$$u_1 + f^s(k_1, h_0) \geq \max\left(\frac{\delta_0}{2}, \delta_0 - \frac{\delta_1}{2}\right).$$

In reality, the structure of f strongly depends on how control of the resource can be hardened. For example, if the resource contains a large number of identical and independent subsystems, so that the control becomes more secure as more subsystems are hardened, then one may model f as

$$f(k_1, h_0) = k_1 + \lambda h_0, \tag{4}$$

with $\lambda \geq 0$ signifies how effective the hardening process is. Another method is to follow an idea similar to that from Gordon and Loeb [7], in which the new cost of attack increases as more is spent on hardening the control. However, such increase should not be linear as in (4), but at a decreasing rate. Also, [4] and [3] suggest a weakest-link model in which attack cost increases linearly step by

step. Based on these results, we devise another reasonable construction for f as follows:

$$f(k_1, h_0) = k_1 + \frac{\mu h_0}{h_0 + \lambda}, \tag{5}$$

where $\mu \geq 0$ is the least upperbound on the increase of attack cost, and $\lambda > 0$ represents the effectiveness of the hardening process, so that it is more effective when λ is small. It is not difficult to see that since $f'(h_0) > 0$ and $f''(h_0) < 0$, the attack cost increases with the hardening cost, but at a decreasing rate, thus agreeing with Gordon and Loeb's model. Also, with the same hardening cost, the attack cost is raised by the same amount after each attack. Define $H = \{(\alpha, h) | \alpha > 0, h \geq 0\}$ to be a set of *periodic state checking with hardening* strategies as described above, we study in the following theorems recommendations for the defender in response to periodic attacks.

Theorem 3. *Consider the game FlipIt(H, $S \cup P$) with $f(k_1, h_0) = k_1 + \lambda h_0$. The defender's best response is $h_0 = B/\lambda$, where B is the attacker's expected utility for the first attack. The game ends after one successful attack.*

Theorem 4. *Consider the game FlipIt(H, $S \cup P$) with $f(k_1, h_0) = k_1 + \frac{\mu h_0}{h_0 + \lambda}$. Let $B > 0$ be the attacker's expected utility for the first attack, and let $n_a = B/\mu$. Let $L > 0$ be the defender's expected loss[3] per attack excluding h_0. Then*

- *for any hardening cost h_0, at least $\lceil n_a \rceil$ attacks occur before the game ends.*
- *the optimal hardening cost is*

$$h_0 = \frac{\lambda n_a}{s - n_a} \text{ where } s = \left\lceil n_a + \frac{1}{2}\left(\frac{\sqrt{L + 4n_a^2\lambda}}{\sqrt{L}} - 1\right)\right\rceil \tag{6}$$

is the corresponding number of attacks.

The above theorems stress a need for appropriate decision over the investment for hardening the resource control. In terms of information security, hardening may mean, for example, system patching, penetration testing, adding security layers, etc. However, an improvement in security does not necessarily imply a better return on security investment, as one can infer from Theorem 4. This happens when security does not just improve with the hardening cost, but depends on other factors, such as information. For example, a system may become more secure not via deployment of new measures, but rather because it gets fixed after suffering more and more attacks. While this idea is captured in (5), Theorem 4 suggests that the defender should spend enough to, for example, sufficiently patch the vulnerability, so that the attack cost would be raised by an amount close to μ. Any additional expense becomes less effective as the increase is bounded by μ. In contrast, situations modelled by (4) represent security that can be strengthened with little information. A common example is when an attack occurs against a device in a homogeneous network. In this case, it is always better to patch all devices, whether they have been compromised or not.

[3] This loss includes the attacker's occupation of the resource and the cost spent on protecting the resource.

5 Dealing with Complex Systems

In this section, we study a different extension to the model in Section 3 to capture situations in which the control of a resource might be difficult to measure, and that state checking might be inaccurate. This disproves an inherent but hidden assumption made in previous models, that with a cost u_i, player i can always determine who is in control of the resource. Again, it addresses another important issue with organisational information security by exacerbating the question "are we compromised?" by "how certain are we that we are compromised?". An answer to such question reflects not just how often security should be assessed, but also how the assessment should be done.

We extend the previous state-checking model with a probability p that the state check succeeds in determining a loss of control, applied to the defender only. The reason for such bias is obvious: while the defender must examine every component of its system as a mean of state checking, the attacker only needs to consider what it has previously compromised, which normally happens with certainty. To simplify our modelling, we explicitly make two assumptions as follows.

A1. There exists no false positive in state checking, i.e., no false alarm on attack exists.

A2. Once a false negative occurs, it will persist until the attacker's next interaction with the resource, i.e., either via a state check, or a move/flip.

Based on these assumptions, we may reformulate the defender's utility functions from what is given in (1), with the help of Lemma 1. It is important to notice that while the average state checking cost remains the same, the average flipping/moving cost lessened by a factor of p, since only a p-fraction of losses in control are followed by a flip/move. These yield the following utility function

$$\beta_0(\alpha_0, \alpha_1) = \begin{cases} p\left(1 - \frac{\alpha_1}{2\alpha_0}\right) - u_0\alpha_0 - pk_0\alpha_1 & \text{if } \alpha_0 \geq \alpha_1 \\ p\left(\frac{\alpha_0}{2\alpha_1}\right) - u_0\alpha_0 - pk_0\alpha_0 & \text{if } \alpha_0 < \alpha_1 \end{cases} \tag{7}$$

Lemma 1. *Consider the FlipIt games in which the defender plays a periodic state checking strategy. Then the defender's average control rate is $p\gamma$, where p is the success probability to detect an attack, and γ is the defender's average control rate when every state check occurs with certainty, i.e., when $p = 1$.*

Similar to the its predecessor, with this model we are also interested in the conditions under which state checking is preferred to mere flipping, and vice versa. This concern is reflected in Theorem 5, which generalises the result given in Theorem 1, and thus emphasises a preference for strategies involving inexpensive state checking, i.e., equal to at most a $p/4$-fraction of the flipping cost. The subtle threshold for the attack rate α_1 in (8) explains the fact that if the attacker infrequently interacts with the resource, then by assumption **A2** it is difficult to detect an attack, and thus periodic flipping is more desirable.

Theorem 5. *Consider the game FlipIt($P \cup S$, $P \cup S$) in which there is a probability p that the defender can detect a take-over attack with the state-checking action. The defender does not prefer periodic moving if*

$$u_0 \leq \frac{k_0 p}{4} \quad \text{and} \quad \alpha_1 \geq \frac{1}{2k_0} \min\left(1, \left[\frac{2(1-p)}{p}\right]^2\right). \tag{8}$$

Corollary 2. *Consider the game FlipIt($P \cup S$, $P \cup S$) in which there is a probability p that the defender can detect an attack. Let $\overline{\alpha_1}$ (resp. $\overline{\alpha_1}^*$) be the minimum value for the attacker's move rate α_1 to drop a periodic-moving (resp. state-checking) defender from the game. Then, $\overline{\alpha_1}^* \geq \overline{\alpha_1}$ if and only if $u_0 \leq k_0 p/4$.*

The need for $u_0 \leq k_0/4$ is further strengthened by Corollary 2 which addresses the situation when the attacker plays too fast, e.g., $\alpha_1 > 1/(2k_0)$, and periodic moving cannot afford for positive payoff, leading to the system being indefensible [14]. This issue becomes more realistic when the attacker is given chances to perform state checking, since in the information security realm, the attacker's state checking can be inexpensive, e.g., reconnecting to backdoors, re-logging in with stolen passwords, etc. In this case, periodic state checking is more robust as they survive higher attack rates.

Another intrinsic part of Theorem 5 is its implication over what is the right cost for state checking. Indeed, flipping in security often involves procedures with high certainty (system reset, backup restores, failovers, etc.), hence their costs are normally determined rather than decided. In contrast, an organisation may choose to invest arbitrarily in administering its security, for example through guard patrolling, antivirus software, firewalls, etc., subject to how much it desires the situation to be in control. While the goal is to satisfy the condition $u_0 \leq pk_0/4$, it is hindered by an inherent constraint that p typically decreases/increases with u_0, that is, less efforts for state checking yields less certainty on its effectiveness.

We study this issue by modelling the connection between u_0 and p, along with an environment parameter $v > 0$ specifying how effectively the amount u_0 might be spent. For example, this parameter may deteriorate as the resource becomes increasingly more sophisticated. On the other hand, it may increase with the skills of the team performing state checking. We can model p as the function of u_0, parameterised by v in the following way

$$p_v(u_0) = 1 - \frac{1}{vu_0 + 1}. \tag{9}$$

It is not difficult to see that, by modelling the probability of successful state checking as in (9), the value $1/v$ represents the cost required for detection of attacks to succeed with a fair coin-flipping chance, i.e., 50%. Note that this does not mean state checking with cost $u_0 \leq 1/v$ can be replaced by "coin-flipping detection" of attacks, as it may violate assumption **A1** to create many false positives, and hence waste moves would become a credible threat to the net utility. We now analyse the threshold under which the cost for state checking suggests it to overpower merely periodic flipping strategies.

Corollary 3. *Consider the game FlipIt($P \cup S$, $P \cup S$) in which there is a probability p that the defender can detect an attack, with p satisfying (9). Then, if $u_0 \leq k_0/4 - 1/v$ and $\alpha_1 \geq \frac{1}{2k_0} \min\left[1, \frac{4}{(u_0 v)^2}\right]$, it is better for the defender to perform periodic state checking.*

From the threshold for state checking cost given in Corollary 3, we may also evaluate whether state checking is at all justifiable given specific characteristics of the environments. Indeed, if the productivity of information security is too low, i.e., $v \leq 4/k_0$, the use of state checking in most cases would not improve the overall utility, as too much cost is required to produce little benefit. This refers to situations when there is a mismatch between the scope of the resource being administered, and of the team performing administration, which means either the resource is too complex, or the administration is immature. In turn, such situations may apply to fast-growing organisations with slower catching-up with technology as well as security evaluation. Another example is with small to medium-sized firms whose businesses strongly rely on information systems, as many of them would spend little research in foreseeing the nontrivial impact of low security administration to the net income.

In overall, Corollary 3 recommends firms not just about hiring an administration team with highest quality-price ratio, but also to spend their concerns on easing the administration of their resource. In reality, the latter can be accomplished in a variety of ways, such as removing redudant components, restructuring the system toward simplification, avoiding complicated dependencies using separation of duties, etc. Otherwise, even the most desirable administration team might still be insufficient for a positive return on investment.

6 Conclusion

In this paper we investigate the concern on the choices of long-term strategic security plans for protecting organisational assets. These choices are represented by questions such as "are we vulnerable?" and "are we compromised?" This concern has become increasingly more important for large businesses as well as governmental units in the era where attackers are advanced, and have the resources to be persistent.

To do so, we extend the FlipIt game between an attacker and a defender periodically taking over a resource from each other, with the tradeoff between the cost of taking over, and the duration of the control. In our model, in addition from taking over, we allow players to check who is controlling the resource. We compare between blind take-over strategies and those that involve "check first, then take over", and show a threshold for the checking cost, under which the latter tactic is preferred.

In further extensions, we study strategic plans on how organisations would rationally invest in security improvement to discourage attackers. Our analysis on specific models proposed suggests that there are cases in which a system must suffer from many attacks to become sufficiently secure to deter attackers. In reality, this is because security breaches serve as valuable information for improving

system security. In another aspect, we relax our hidden assumption so that state checking might be incorrect, and study not just the frequency of security assessment, but also how quality-price-ratio may even discourage assessment of security. Since our models mostly deal with the defender's utility, the lessons learned may apply to not just advanced persistent threats (APTs), but also a pool of non-persistent threats that occurs with known frequency, e.g., from a community of underground hackers.

References

1. Bejtlich, R.: Testimony before the USCC Hearing on "Developments in China's Cyber and Nuclear Capabilities" (March 26, 2012), http://www.uscc.gov/hearings/2012hearings/written_testimonies/hr12_03_26.php
2. Billo, C.G.: Cyber warfare: An analysis of the means and motivations of selected nation states. Technical report, Institute for Security Technology Studies at Darthmouth College (2004)
3. Böhme, R., Félegyházi, M.: Optimal Information Security Investment with Penetration Testing. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 21–37. Springer, Heidelberg (2010)
4. Böhme, R., Moore, T.: The iterated weakest link: A model of adaptive security investment. In: Workshop on the Economics of Information Security, WEIS (2009)
5. Chabrow, E.: Identifying undetected breaches identifying undetected breaches: How data scientists analyze big data to spot vulnerabilities (April 20, 2012), http://www.bankinfosecurity.co.uk/interviews/identifying-undetected-breaches-i-1542
6. Coviello, A.: Open letter to RSA customers (March 17, 2011), http://www.rsa.com/node.aspx?id=3872
7. Gordon, L.A., Loeb, M.P.: The economics of information security investment. ACM Transactions in Information System Security 5(4), 438–457 (2002)
8. Kunreuther, H., Heal, G.: Interdependent Security. Journal of Risk and Uncertainty 26(2), 231–249 (2007)
9. Langner, R.: Stuxnet: Dissecting a cyber warfare weapon. IEEE Security & Privacy 9(3), 49–51 (2011)
10. Manshaei, M.H., Zhu, Q., Alpcan, T., Basar, T., Hubaux, J.: Game Theory Meets Network Security and Privacy. Technical report, EPFL (2010)
11. National Institute of Standards and Technology. Technical Guide to Information Security Testing and Assessment. Special Publication 800–115 (2008)
12. National Institute of Standards and Technology. Recommended security controls for federal information systems and organizations. Special Publication 800–53 (2009)
13. Raya, M., Shokri, R., Hubaux, J.: On the tradeoff between trust and privacy in wireless ad hoc networks. In: Proceedings of the Third ACM Conference on Wireless Network Security, WiSec 2010, pp. 75–80. ACM, New York (2010)
14. van Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: Flipit: The game of "stealthy takeover". Cryptology ePrint Archive, Report 2012/103 (2012)
15. Vijayan, J.: Breach, undetected since 2005, exposes data on Kingston customers (July 17, 2007), http://www.computerworld.com/s/article/9027220/Breach_undetected_since_05_exposes_data_on_Kingston_customers

Defending against the Unknown Enemy: Applying FLIPIT to System Security

Kevin D. Bowers[1], Marten van Dijk[1], Robert Griffin[2], Ari Juels[1], Alina Oprea[1], Ronald L. Rivest[3], and Nikos Triandopoulos[1]

[1] RSA Laboratories, Cambridge, MA, USA
[2] RSA, The Security Division of EMC, Zurich, Switzerland
[3] MIT, Cambridge, MA, USA

Abstract. Most cryptographic systems carry the basic assumption that entities are able to preserve the secrecy of their keys. With attacks today showing ever increasing sophistication, however, this tenet is eroding. "Advanced Persistent Threats" (APTs), for instance, leverage zero-day exploits and extensive system knowledge to achieve *full* compromise of cryptographic keys and other secrets. Such compromise is often silent, with defenders failing to detect the loss of private keys critical to protection of their systems. The growing virulence of today's threats clearly calls for new models of defenders' goals and abilities.

In this paper, we explore applications of FLIPIT, a novel game-theoretic model of system defense introduced in [14]. In FLIPIT, an *attacker* periodically gains *complete* control of a system, with the unique feature that system compromises are *stealthy*, i.e., not immediately detected by the system owner, called the *defender*. We distill out several lessons from our study of FLIPIT and demonstrate their application to several real-world problems, including password reset policies, key rotation, VM refresh and cloud auditing.

1 Introduction

Targeted attacks against computing systems have recently become significantly more sophisticated. One major consequence is erosion of the main principle on which most cryptographic systems rely for security: That "secret" keys remain strictly secret. Attacks known as Advanced Persistent Threats (APTs), for instance, exploit deep, system-specific knowledge and zero-day vulnerabilities to compromise a system completely, revealing sensitive information that can include *full* cryptographic keys. Moreover, this compromise is *stealthy*, meaning that it's not immediately detected by the system owner or defender. We have previously introduced a game-theoretic model for this volatile new security world called FLIPIT [14], The Game of "Stealthy Takeover."

FLIPIT is a game between two players, known as the *attacker* and *defender*. Players compete to control a shared sensitive resource (e.g., a secret key, a password, or an entire infrastructure, depending on the setting being modeled). A player may take control of the resource at any time by executing a *move*; the player pays a certain (fixed) cost to do so. The fact that moves are *stealthy* in FLIPIT distinguishes it from other games in the literature. A player in FLIPIT doesn't immediately know when her opponent has made a move, but discovers it only when she subsequently moves herself. Each

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 248–263, 2012.

player's objective is to maximize the fraction of time she controls the resource, while minimizing her cumulative move cost.

The goal of this paper is twofold: (1) To present some general principles of effective play in FLIPIT and (2) To demonstrate application of these principles to defensive strategy design in real-world cyberdefense settings. Thus our contributions are:

Principles of Effective FLIPIT Play: We introduce general principles for effective FLIPIT play by a defender facing a more powerful attacker. These principles fit into three categories: (A) Principles governing defender strategy selection based on knowledge about the class of strategies employed by the attacker; (B) Principles regarding game setup, specifically, effective cost-structure design choices for the defender; and (C) Principles regarding gameplay feedback, namely how the defender can best maximize feedback via system design for effective gameplay. All principles have solid theoretical underpinnings in our analysis in [14].

Application of FLIPIT to Real-World Security Problems: We explore the application of FLIPIT to the problem of managing *credentials* used for user authentication. In particular, we're interested in enabling system owners (defenders) to schedule the expiration or refresh of their credentials most effectively. We focus primarily on *passwords* (namely, password-reset policies) and *cryptographic keys* (key refresh, also known as *key rotation*). Specifically, we show the benefits of *randomizing* password reset intervals (in sharp distinction to the widespread 90-day password reset policy). We also quantify the importance of frequent rotation of keys protecting critical defender assets.

We briefly touch on other applications of the FLIPIT framework, including virtual machine refresh and cloud auditing for service-level-agreement (SLA) enforcement. Our FLIPIT design principles bring to light defensive strategies that improve on current practices in these settings. Study of these applications also introduces new and interesting variants of the basic FLIPIT game whose analysis provides interesting open questions for the community.

Organization. In Section 2, we present the FLIPIT framework and detail on the lessons learned from [14] in the form of set of principles. In Section 3, we apply these principles to password reset and key management, and in Section 4 to virtualization and cloud auditing. We review related work in Section 5 and conclude in Section 6.

2 Framework and Principles

We start this section by introducing the FLIPIT framework. We then introduce several principles for designing defensive strategies in various computer security scenarios derived from our theoretical analysis presented in [14]. In the following sections, we present several applications of the framework and show how the principles introduced here result in effective defensive strategies.

2.1 FLIPIT Framework

We present FLIPIT by the example of "host takeover" where the target resource is a computing device. The goal of the attacker is to compromise the device by exploiting

a software vulnerability or credential compromise. The goal of the defender is to keep the device clean through software reinstallation, patching, or other defensive steps.

An action/move by either side carries a cost. For the attacker, the cost of host compromise may be that of, e.g., mounting a social-engineering attack that causes a user to open an infected attachment. For the defender, cleaning a host may carry labor and lost-productivity costs. The resource can be controlled (or "owned") by either of two *players* (attacker / defender). When a player moves he takes control of the resource; ownership will change back and forth as the players make moves. A distinctive feature of FLIPIT is its *stealthy* aspect, that is the players *don't know* when the other player has taken over. Nor do they know the current ownership of the resource unless they perform a move. For instance, the defender does not find out about the machine compromise immediately, but potentially only after he moves himself; or, the attacker might find out about the host cleanup at a later time, not immediately when the defender moves.

The goal of each player is to maximize the time that he or she controls the resource, while minimizing their move costs; players thus have a disincentive against moving too frequently. A move results in a "takeover" when ownership of the resource changes hands. If the player who moves already had ownership of the resource, then the move was wasted (since it did not result in a takeover). The only way a player can determine the state of the game is to move. Thus a move by either player has two consequences: it acquires control of the resource (if not already controlled by the mover), but at the same time, it reveals information about the state of the resource prior to the player taking control. This knowledge may be used to determine information about the opponent's moves and assist in scheduling future moves.

FLIPIT provides guidance to both players on how to implement a cost-effective move schedule. For instance, it helps the defender answer the question: "How regularly should I clean my system?" and the attacker: "When should I launch my next attack?".

We show a graphical representation of the game in Figure 1. The control of the resource is graphically depicted through shaded rectangles, a blue rectangle (dark gray in grayscale) representing a period of defender control, a red rectangle (light gray in grayscale) one of attacker control. Players' moves are graphically depicted with shaded circles. A vertical arrow denotes a takeover, when a player (re)takes control of the resource upon moving. In this example, a move costs the equivalent of one second of ownership. Thus, at any time t, each player's net score is the number of seconds he has had ownership of the resource, minus his number of moves up to time t.

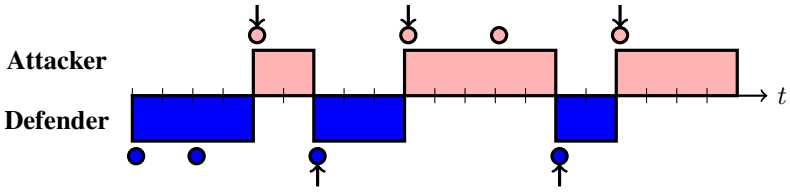


Fig. 1. The FLIPIT game. Blue and red circles represent defender and attacker moves, respectively. Takeovers are represented by arrows. Shaded rectangles show the control of the resource—blue (dark gray in grayscale) for the defender and red (light gray in grayscale) for the attacker. We assume that upon initialization at time 0, the defender has control.

The most interesting aspect of this game is that the players do *not* automatically find out when the other player has moved in the past; moves are *stealthy*. A player must move himself to find out (and reassert control). We distinguish various types of feedback that a player may obtain upon moving:

- **Nonadaptive [NA].** No feedback is given to the player upon moving.
- **Last move [LM].** The player moving at time $t > 0$ finds out the exact time when the opponent played last before time t.
- **Full history [FH].** The mover finds out the complete history of moves made by both players so far.

Since feedback is the means by which a player acquires more knowledge in FLIPIT, we are now ready to define the view or knowledge of a player. The *view* of a player after playing his nth move is the history of the game from this player's viewpoint from the beginning of the game up to and including his nth move. It lists every time that player moved, and the feedback received for that move, up to and including his nth move.

We can now define a player's strategy in the game. Informally, a strategy for a player defines how moves in the game are chosen as a function of time, the knowledge about the opponent acquired before the game starts and the amount of feedback received by a player during the game. More formally, a *strategy* for playing FLIPIT is a (possibly randomized) mapping S from views to positive real numbers. If S is a strategy and v a view up to and including the player's nth move, then $S(v)$ denotes the time the player waits before making his $(n+1)$st move.

Strategies can be grouped into several classes. For instance, the class of *non-adaptive strategies* includes all strategies for which players do not receive any feedback during the game. The class of *renewal strategies* is a subset of non-adaptive strategies in which the intervals between a player's consecutive moves are generated by a renewal process. As such, the inter-arrival times between moves in a renewal strategy are independent and identical distributed random variables chosen from the same probability density function. The class of *adaptive strategies* encompasses strategies in which players receive feedback during the game according to either LM or FH notions defined above.

A game instance in FLIPIT is given by two classes of strategies, one for the attacker and one for the defender, from which the players can select their strategies before the game starts. The strategy can be randomized and adapted according to the feedback received during the game. We denote by $\mathsf{FlipIt}(\mathcal{C}_0, \mathcal{C}_1)$ the FLIPIT game in which player i chooses a strategy from class $\mathcal{C}_i$, for $i \in \{0, 1\}$. Here, we identify the defender with 0 and the attacker with 1.

For a particular choice of strategies $S_0 \in \mathcal{C}_0$ and $S_1 \in \mathcal{C}_1$, the benefit $\beta_i(S_0, S_1)$ of player i is defined in the following way:

- We define k_0 and k_1 as the cost of the defender's and attacker's moves, respectively.
- By $\alpha_i(t)$ we denote the *average move rate* by player i up to time t. In other words, $\alpha_i(t)$ is equal to the total number of moves by player i up to time t divided by t.
- By $\gamma_i(t)$ we denote the *average gain rate* for player i defined as the fraction of time that player i has been in control of the game up to time t.
- Now we are ready to define player i's *average benefit rate* up to time t as $\beta_i(t) = \gamma_i(t) - k_i\alpha_i(t)$. This is equal to the fraction of time the resource has been owned by player i, minus the cost rate for moving.

– The *benefit* β_i of player i is defined as the liminf of player i's benefit rate up to time t as t tends to infinity; $\liminf_{t\to\infty} \beta_i(t)$.

The average move, gain and benefit rates all depend on the exact strategies S_0 and S_1 played by defender and attacker. (Here, benefits represent the notion of *utility*.)

In [14], we have presented a detailed definition of the FLIPIT game and a rigorous analysis of several aspects of the game, including Nash equilibria for certain FlipIt instances and an analysis of dominated strategies within certain classes of strategies.

2.2 Principles for Designing Defensive Strategies

FLIPIT was motivated by the observation that systems should nowadays be designed to be resilient to very powerful adversaries that can eventually fully compromise the system. Defenders protecting sensitive resources (including sensitive personal information, cryptographic keys, national secrets) face increasingly sophisticated attackers and traditional defensive techniques are no longer effective. The framework provided by FLIPIT provides a model of continuous interaction between a defender and attacker in controlling a resource, which can be used to study this new reality. Based on our theoretical analysis in [14] we outline in this section several principles for designing effective defensive strategies when dealing with various security situations.

There are three main categories of principles, detailed in the rest of the section:

(A) Principles about selecting a defensive strategy given some knowledge about the class of strategies employed by the attacker;
(B) Principles about the setup of the FLIPIT game resulting in various system design choices for the defender;
(C) Principles about the amount of feedback received by the defender and made available to the attacker during the game.

(A) Principles related to strategy selection. The first type of principles are as follows.

Residual Game [RG]. There is an assumption in game theory that a rational player does not choose to play a strategy that is strongly dominated by other strategies. Therefore, iterative elimination of strongly dominated strategies for both players is a standard technique used to reduce the space of strategies available to each player (see, for instance, the book by Myerson [8]). For a game instance $\mathsf{FlipIt}(\mathcal{C}_0, \mathcal{C}_1)$, we denote by $\mathsf{FlipIt}^*(\mathcal{C}_0, \mathcal{C}_1)$ the *residual* FLIPIT game consisting of surviving strategies after elimination of strongly dominated strategies from classes $\mathcal{C}_0$ and $\mathcal{C}_1$. A rational player will always choose a strategy from the residual game resulting in the following principle:

> *RG Principle: Given a particular game instance, a defensive strategy should be selected from the residual game.*

For instance, one set of results in [14] analyzes the game instance in which both players can select strategies from the class of renewal strategies $\mathcal{R}$ (i.e., to set the time between moves according to a fixed probability distribution) or that of periodic strategies $\mathcal{P}$ with random phase (i.e., to set the first move uniformly at random, while all next moves are

chosen according to a fixed period). For this game instance, the periodic strategy with random phase strongly dominates the renewal strategies of similar play rate, i.e., the residual game $\mathsf{Fliplt}^*(\mathcal{R} \cup \mathcal{P}, \mathcal{R} \cup \mathcal{P})$ turns out to be equal to $\mathsf{Fliplt}(\mathcal{P}, \mathcal{P})$.

A second example is a scenario in which an LM adaptive defender plays against an attacker employing an exponential strategy (i.e., the intervals between moves are selected according to an exponential distribution). Then the defender's strongly dominant strategy among all adaptive strategies is periodic play.

Randomized Strategy [RS]. In a FLIPIT game in which an NA defender plays against an adaptive LM attacker (i.e., the attacker acquires additional knowledge through last move feedback), the defender should either introduce randomness when selecting her moves or not play at all (assuming that the attacker plays with some positive rate).

A deterministic, predictable strategy for the defender (e.g., periodic play) results in total loss of control: an adaptive attacker finding out the exact last move time of the defender can predict the time of the defender's next move and move right after the defender. With this strategy, the attacker controls the resource virtually at all times. Therefore, adding randomization to the intervals between defender's moves has the advantage of increasing the attacker's uncertainty about the defender's strategy. This results in the following principle:

> *RS Principle: The defender should use randomization in her strategy (or not play at all) when confronted with an adaptive attacker moving with positive rate.*

While introducing randomization when selecting defensive moves against an adaptive attacker has a clear benefit in increasing the defender's benefit, the amount of variability in the defender's strategy has to be carefully calibrated to not deviate too much from the optimal strategy. We'd like to highlight here that finding the strongly dominant non-adaptive (randomized) defensive strategy against an adaptive attacker is an open problem (see [14]).

Drop Out Principle [DOP]. For some applications the resource is so valuable that the loss of control (even for small fraction of time) results in highly negative benefit for the defender. For such scenarios, the strongly dominant strategy for the defender is to play fast enough in order to force a rational attacker to drop out of the game.

In [14] we showed two results:

- If the defender plays periodic with rate $\alpha > 1/k_1$, then the attacker's strongly dominant adaptive strategy is to drop out of the game.
- If the defender plays periodic with rate $\alpha > 1/(2k_1)$, then the attacker's strongly dominant non-adaptive strategy is to drop out of the game.

These findings result in the following principle:

> *DOP Principle: For valuable resources, the defender should play fast enough to force the attacker to drop out of the game.*

To force the attacker out of the game, the move rate of the defender is dependent on the attacker's move cost. In order for the defender's benefit to be positive, her move

cost should be lower than the attacker's ($k_0 < k_1$). As the ratio between the attacker's and defender's move costs increases, the defender improves his benefit. Achieving such conditions is discussed below.

(B) Principles related to game setup. In the security situations that we model, typically the defender has the advantage that she is responsible for setting up the game. Typically, the resource is initially controlled by the defender, and she can make various design choices that can result in different game parameters. Below we highlight two principles related to controlling the attacker and defender move costs.

Move Cost Principles [MCP]. The defender's benefit increases if she arranges the game so that her moves cost much less than the attacker's moves. Lower move cost for the defender implies that the defender can play more frequently, and control the resource more. For some situations, a reduction in defender's move cost results in the ability of the defender to play with sufficiently high rate that it eventually forces the attacker to drop out of the game (as illustrated in the DOP principle). This observation leads to the following principle:

> *MCP Principle 1: The defender should arrange the game so that her moves cost much less than the attacker's.*

An interesting research challenge for system designers is how to design an infrastructure in which refresh/clean costs are very low. We believe that virtualization has huge potential in this respect. For instance, refreshing a virtual machine has much lower cost than refreshing a physical machine. For cleaning a physical machine, full system wiping and reinstallation of all software is needed, while a virtual machine imagine can be simply restored from a clean-state version in a couple of minutes.

Moreover, the defender should make design choices that increase the attacker's move costs. This will result in the attacker playing less frequently, which in turn also implies higher control for the defender. Thus, the following principle can be derived:

> *MCP Principle 2: The defender should arrange the game so that she increases the move cost of the attacker.*

Another interesting research problem for system designers is how to setup an infrastructure that increases the attacker's move costs in practice. For instance, sensitive data or cryptographic keys can be split (shared) over multiple storage servers such that only by accessing all servers can the sensitive data can be reconstructed while no information is obtained if at least one of the servers is not accessed/controlled. This reduces the attack surface: in order to compromise the sensitive data, the attacker needs to obtain control of all servers, effectively multiplying his move cost by the number of servers.

In our analysis from [14], we showed, for instance, that when playing with an exponential distribution against an LM-adaptive attacker, the defender can achieve benefits ranging from 0.1262 to 0.75 as the move cost ratio k_1/k_0 varies from 1 to 4. Similarly, when playing with a delayed exponential distribution the benefit achieved by the defender varies between 0.15 and 0.85 as the move cost ratio k_1/k_0 changes from 1 to 4. These examples clearly illustrate the MCP principles.

(C) Principles related to feedback received during the game. Our theoretical analysis in [14] demonstrates that any amount of feedback (even limited) received during

the game about the opponent benefits a player in FLIPIT. Both players can control to some extent the amount of feedback received by the opponent, but again the defender has some advantage in setting up and knowing all the details of the internal infrastructure of the resource that she protects.

Feedback Principles [FP]. The defender's benefit increases if the amount of feedback received during the game about the attacker's moves is increased. Defenders, therefore, should monitor their systems frequently to gain information about the attacker's strategy and detect potential attacks quickly after take over. Both monitoring and fast detection help a defender to more effectively schedule moves, which results in more control of the resource and less budget spent on moves, increasing the defender's benefit. As a consequence, the following principle follows naturally:

> *FP Principle 1: Defenders should monitor their resources to increase the amount of feedback received during the game.*

Moreover, limiting the amount of feedback available to the attacker upon moving can also contribute to an increased benefit for the defender. The defender can employ various techniques to hide information about the exact time when she performed a move. The defender may, e.g., decide not to log timing information about when a system was cleaned. Accordingly, the following principle can be derived:

> *FP Principle 2: Defenders should limit the amount of feedback available to the attacker during the game.*

3 Applications to Credential Expiration

In this section we highlight credential expiration as an application of particular practical interest. Credentials confirm the identity of a party. We focus on the two most common forms: *passwords* and *cryptographic keys*. The most common practice for managing credentials is to let credentials expire after a certain period. As we show the FLIPIT defending principles offer some simple, easy-to-deploy improvements to this practice.

We first discuss password reset and show the benefit of the randomized strategy principle. Next we discuss a storage service managed by a single enterprise that maintains directories with documents for its employees. Employees may update their documents, create new and remove old documents. We assume that access control to employee specific directories is managed by authentication keys. We discuss the well-established practice of key management by means of key-rotation and illustrate the DOP principle by a parameterization in which rational adversaries are forced to drop out. We extend our example by showing a reduction in defensive move costs when the storage service is outsourced to the cloud.

3.1 Password Reset

Knowledge of a password usually equates with control of a resource, such as an account. Thus we may view an adversary's attempt to compromise a password as a game of

control. On learning a password, the adversary seizes control of an account. By resetting the password, the account owner regains control.

FLIPIT for Password Reset. When resetting a password, a user typically obtains no feedback on whether it's been compromised. Conversely, though, on (re-)compromising a password, an attacker learns whether it has been reset, simply by observing whether the password has changed since the last compromise. Where fixed-period password-reset policies are in force, an attacker will also generally know the period, as it's a matter of organization-wide policy. Worse still, in many situations, the adversary may also know or learn over time the *phase* of a user's password reset schedule.

Password reset thus involves asymmetric knowledge. The defender receives no feedback, while the attacker learns whether a password is still valid. So a FLIPIT game for password reset is similar to the basic FLIPIT model with a non-adaptive defender and LM-adaptive attacker. The move cost for a defender is essentially the human overhead of creating and memorizing a new password. The cost of password compromise by the attacker depends on the environment: There are many vectors for password compromise, e.g., database breaches, password-stealing Trojans, etc.

Resetting passwords at fixed 90-day intervals, as commonly employed by most organizations today, is a poor defensive strategy. The Randomized Strategy (RS) principle offers a key insight into password refreshing:

> *To minimize adversarial control of a password-protected account, password resets should take place at* randomly *determined intervals.*

Case study. Consider the application of FLIPIT to the problem of password reset for corporate e-mail accounts. The cost to an attacker of compromising a password is perhaps most meaningfully reflected in the price of account passwords in underground markets. In a 2008 report on the underground economy, Symantec reported a price range of \$4-\$30 for a compromised e-mail password.[1] Quantifying the defender's cost in this setting is harder, as the overhead of password reset includes a substantial intangible burden on the user. Enterprise help-desk calls for password reset offer an indirect estimate of the human cost. A 2004 Gartner case study [15] documented an average cost of \$17.23 (here, rounded to \$17.00) per password reset call at a large beverage company.

Quantifying the benefit over time of account control is an even greater challenge, and depends largely on the attacker's control objective and its strategy for monetizing or otherwise exploiting a compromised account. We might notionally assume that the benefit of account control is equal for attacker and defender and also that the value of an account is much larger than the cost of password resets. With the cost of password reset at \$17 every 90 days, we assume that the value of the account is 10 times larger, resulting in approximately a value of \$2.00 per-day benefit. We'd like to highlight that these numbers are for illustration purposes only, the analysis can be easily adapted if some of the parameters change their values.

With these parameter settings, we can set $k_0 = 17/2 = 8.5$ and $k_1 \in [4/2, 30/2]$. For a defender playing with a 90-day period strategy against an adaptive attacker, the

[1] http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_underground_economy_report_11-2008-14525717.en-us.pdf

amount of control is 0 and her benefit is always negative at $0 - 8.5/90 = -0.09$. We quantify now the exact benefits for the attacker and defender in case the defender employs an exponential strategy and the attacker is LM adaptive. From our analysis in [14] (see Theorem 8), we distinguish two cases:

1. If $k_1 \geq k_0/0.854 = 10$, then the defender's optimal play is exponential with rate $\lambda = 1/k_1$ (and mean k_1), and the defender's benefit is $\beta_0 = 1 - k_0/k_1$. The attacker's best response is not playing at all and his benefit β_1 is zero.
2. If $k_1 < k_0/0.854 = 10$, the defender's maximum benefit is achieved by playing at rate $\lambda = (1 - (1 + z)e^{-z})/k_1$, where z is such that $(e^z - 1 - z)/z^3 = k_0/k_1$. The attacker's maximum benefit is achieved for playing periodically with period $\delta = z/\lambda$.

We present in Table 1 the defender's optimal average inter-move delay (in days), the attacker's period of play (in days), and the defender's and attacker's optimal benefits (expressed in dollars) for different values of k_1. As observed, the defender always achieves positive benefit when employing an exponential strategy. As expected, the defender's benefit increases with higher attacker cost, validating the Move Cost Principle (MCP). The Drop Out Principle (DOP) is also demonstrated as the attacker's optimal strategy is not playing at all once his move cost exceeds a certain threshold.

Table 1. Parameters and benefits for exponential defender strategy. The defender's average inter-move delay and attacker's period are given in days and the defender's and attacker's benefit in $.

k_1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Def. average	289	193	145	116	97	84	74	66	10	11	12	13	14	15
Att. period	35	36	37	38	39	40	41	42	∞	∞	∞	∞	∞	∞
Def. benefit	0.058	0.088	0.116	0.14	0.16	0.206	0.236	0.266	0.3	0.45	0.582	0.692	0.784	0.866
Att. benefit	1.768	1.64	1.548	1.46	1.34	1.24	1.144	1.05	0	0	0	0	0	0

This case study illustrates the principle that the defender should use randomization when facing an adaptive attacker. At the same time, it shows the limitations of employing a simple non-adaptive strategy. Clearly, the defender achieves much lower benefit than the attacker when the attacker's move cost are lower or equal to the defender's. According to our Feedback Principles (FP), the defender would improve his benefit if she is able to obtain more feedback during the game. For instance, certain defensive techniques such as monitoring the infrastructure to detect password compromises or requiring multi-factor authentication will enhance the defender's benefit.

Variant. We might consider an enhanced password-reset model with asymmetric play as well. The attacker then has a second action type available to it, a *check* that determines whether it still has control, and is distinct from a *move*. This move corresponds to an attacker attempt to use a password in order to check its validity. A check move, like an ordinary reset move, carries some cost: It increases the detection risk for the attacker. (In a system in which unsuccessful login events are logged, for instance, every check will potentially trigger an investigation by the defender.)

3.2 Key Management

We now explore the application of FLIPIT in the area of *key management*. We examine the use case of key management for authenticating employee directories, by considering two concrete, contrasting deployment models:

- **Deployment of key management within a single enterprise:** This deployment model is very commonly used in the industry today. Widely adopted key management products, e.g., from IBM, HP, EMC, Thales, Symantec/PGP and many other vendors, provide solutions for managing cryptographic keys at an enterprise level.
- **Deployment of key management within a cloud infrastructure**. This deployment model relies on the shared infrastructure of a cloud service provider on behalf of multiple tenant enterprises, and is emerging as a significant alternative to enterprise-based key-management infrastructures. Architectural and security considerations for this model have been discussed in the Cloud Security Alliance Security Guidance for Critical Areas of Focus in Cloud Computing 3.0.[2] We focus on the deployment of enterprise-specific key managers within a dedicated and isolated segment of a Cloud Service Provider infrastructure, a model already exemplified in commercial products, e.g., Microsoft Azure Trust Services.

The FLIPIT game offers an alternative way to look at the question of whether and when to use key rotation. As NIST SP 800-57, Part 2 [2] (pp. 45), suggests, evaluation of key rotation policy should take into account "the threat to the information (e.g., who the information is protected from, and what are their perceived technical capabilities and financial resources to mount an attack)." We achieve this with FLIPIT, by exploring whether there are ways to take advantage of key rotation that might invoke the Drop Out Principle, so that the best strategy for the attacker is to defect from the game.

FLIPIT for Key Rotation. In this game, the defender's moves implement key rotation in order to refresh keys. We will assume that the defender plays a non-adaptive periodic strategy because she does not see the attacker's moves until the point at which the compromise is exposed. We assume the following parameters for the defender:

- Refreshing a single authentication key costs about $1 (this estimate seems to be well supported due to the cost of interaction with the parties who need the authentication key for accessing their directory). Let u be the period (measured as a fraction of a year) at which the defender rotates each key. Then $1/u$ equals the defender's move cost in $ per key per year.
- If the attacker gets hold of an authentication key, then the loss to the defender due to the leakage of the protected documents (which are updated, created and removed continuously) is assumed to be about $200 /year (this estimate comes from Ponemon estimating in their 2012 report[3] that the costs for responding to a data breach incident typically equals $204 per stolen credit card record; here we assume a continued loss of $200 /year due to illegitimate access to protected

[2] https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

[3] Ponemon Institute, 2009 Annual Study.http://www.ponemon.org, "Cost of a Data Breach: Understanding Financial impact, Customer Turnover and Preventive Solutions."

documents). So, being in control of an authentication key means that the defender does not incur a loss at rate \$200 /year. We may model this as a gain of \$200 /year being in control. So, for γ_0 denoting the fraction of time the defender is in control of an authentication key, the defender's gain is equal to $\$200\gamma_0$ per key per year.

In FLIPIT notation, where benefit is normalized with respect to the value of being in control, the defender's benefit is equal to $\gamma_0 - k_0/u$, where $k_0 = 1/200$.

We assume a typical scenario of an enterprise with $10,000$ authentication keys (this estimate comes from [2]). For the attacker we assume the following parameters:

- To exfiltrate keys an attacker needs to launch an attack of $\$10,000$ (move) cost.
- Let n be the total number of exfiltrated keys per attack. As n increases, the defender detects the attack with higher probability. We model this by introducing a parameter r, the probability that the defender detects the compromise of a single key. Then, assuming that the detection of different key compromises are independent events, the probability of detecting the compromise of n keys is $1 - (1 - r)^n$. In the first example of deployment within the enterprise, we consider the worst case for the defender, in which the attacker is detected with probability 0. When deploying key management to the cloud, we assume a better detection mechanism ($r > 0$) as the cloud provider handles keys of multiple tenants.
- If the attacker gets an authentication key without being detected, then access to the protected documents leads to a gain of about \$4 /year (monetizing leaked data is orders of magnitude less than the loss to the defender caused by their leakage).

Summarizing, a single attack in which n keys are extracted costs $\$10,000$ and leads to an expected value of being in control of $[(1 - (1 - r)^n) \cdot 0 + (1 - r)^n \cdot 4] \cdot n$ \$/year. In FLIPIT notation: $k_1 = 10000/[4(1 - r)^n n]$.

An adaptive attacker drops out if he can get no benefit at all: this happens if $u < k_1$. This shows that if the defender chooses u slightly less than $k_1 = 2500/[(1 - r)^n n]$, then the attacker must drop out in order to avoid a negative benefit. In the enterprise deployment example, in the worst case for the defender, $r = 0$ (no detection) and $n = 10000$ (all keys within the enterprise are stolen). For these parameters the defender should choose $u < 0.25$, i.e., the defender's period is at most 3 months. The defender's benefit per key per year is equal to $\$(200 - 1/u) = \$196$ which is very close to \$200. Hence, for a small cost of \$4 per key per year, no documents will be stolen by a rational adversary playing a periodic strategy. Overall, the Drop Out Principle offers a key insight for effective key rotation:

> *To minimize the possibility of exfiltration of documents protected using authentication keys, key rotation should be applied sufficiently often; at least every 3 months in our concrete setting above.*

For a non-adaptive attacker the drop out condition is $u < 2k_1$, and key rotation must occur at least every 6 months in which case for a small cost of \$2 per key per year for the defender no documents will be stolen by a rational adaptive attacker. So, by reducing the feedback available to the attacker, the defender halves his cost per key per year. This demonstrates the Feedback Principle:

To minimize the possibility of exfiltration of documents protected using authentication keys, key rotation should be applied sufficiently often; for a non-adaptive attacker at least every 6 months in our concrete setting above.

As an extension to this case study we now assume that the enterprise outsources its document storage and key management to a cloud provider. Since the cloud provider manages the keys of several enterprises/tenants, we may assume that more detection mechanisms are available to detect stolen keys, e.g., let us assume that the probability of detecting the event of stealing one key is equal to $r = 1/10000$.

The optimal defender's period u against an adaptive attacker is now equal to $k_1 = 2500/[(1-r)^n n]$, which is in the worst case minimized for n equal to the minimum of $-1/\ln(1-r)$ and 10,000, the total number of keys of a single enterprise. This results in an optimal defender's period of $u = 0.68$ or 248 days. This shows that the increased risk to the attacker allows the defender/cloud to choose a period which is 2.72 times larger than the 3 months key rotation period for enterprise key management. This leads to a reduction from \$4 cost per key to \$1.47 cost per key. (A second benefit of having a cloud provider manage enterprise keys is a reduction in the initial start up costs which is now shared among all the tenants of the cloud provider.)

The main goal in both the enterprise and the cloud service provider game is to invoke the Drop Out Principle, creating such a significant advantage for the defender that the rational strategy for the attacker is to quit the game. A key factor in achieving this result is the risk to the attacker. We have formulated this risk in terms of the possibility that the value of the stolen information will be negated if and when the attack is exposed. Such a result can be demonstrated in a number of real-life situations in which the rapid discovery of an attack prevented the attacker from deriving value from their theft, such as in the case of the 2011 attack on Lockheed-Martin that attempted to use information stolen from RSA, as a vector in the attack.[4]

4 Other Applications

We next consider two more applications of FLIPIT, emphasizing its breadth of application, rather than detailed analysis. We first examine defensive virtual-machine refresh. While less mature a practice than password reset and key rotation, it's an emerging approach that fits well within the basic FLIPIT framework. Secondly, we consider FLIPIT as a model for automated (cryptographic) cloud service auditing.

4.1 Virtual-Machine Refresh

Virtualization is seeing heavy use today in the deployment of servers in data centers. As individual servers may experience periods of idleness, consolidating multiple servers as VMs on a single physical host often results in greater hardware utilization. Similarly, Virtual Desktop Infrastructure (VDI) is an emerging workplace technology that provisions users with VMs (desktops) maintained in centrally managed servers. In this

[4] See article in Infosecurity Magazine at http://www.infosecurity-magazine.com/view/18299

model, users are not bound to particular physical machines. They can access their virtual desktops from any endpoint device available to them, even smart phones.

While virtualization exhibits many usability challenges, one key advantage is a security feature: VMs can be periodically refreshed (or built from scratch) from "clean" images.

Takeover of a VM results in a game very similar to that for a physical host. Virtualization is of particular interest in the context of FLIPIT, though, because FLIPIT offers a means of measuring its security benefits. Refreshing a VM is much less cumbersome than rebuilding the software stack in a physical host. In other words, virtualization lowers the move cost for the defender illustrating the Move Cost Principle (MCP):

> *When designing system infrastructures, virtualization is a key technique useful in reducing the defender's move cost.*

4.2 Cloud Service Auditing

When a cloud service provider furnishes a resource to a client, it's desirable for the client, or an auditor acting on its behalf, to *audit* the provider. A provider generally furnishes resources to clients under a Service-Level Agreement (SLA), a contractual specification of configuration options and minimum service levels. Compliance or deviation from an SLA, however, isn't always readily apparent to clients—particularly for security or reliability objectives.

Although real-time cloud-service auditing, applied as remote spot-checks, isn't common today, it will inevitably become a regular practice, as recognized by the growth of supporting standards such as SCAP and CloudTrust. The growing literature on remote testing of cloud security properties [1,3] largely neglects the question of how challenges should be scheduled, or assumes a simplistic partitioning of time into epochs. Overall, FLIPIT offers a more refined temporal framework for these protocols.

Our Randomization Principle (RP) shows that the defender must adopt a randomized strategy to perform well, i.e., audit spot checks must be unpredictable to be effective in an adversarial environment.

> *An optimal cloud service auditing strategy is* adaptive*, i.e., conditions challenge times on the observed compliance or non-compliance of the provider.*

The defender has a disadvantage in the game defined so far as the cloud provider has complete feedback about the defender's moves. The Feedback Principle (FP) teaches us that the defender further benefits in the game if the exact audit times are not divulged to the provider. To implement such a defensive technique, the defender might, for instance, use an auditing technique having the property that audit requests are indistinguishable from normal requests. This allows the defender to spread and hide her audits at slow rate among the normal requests to the cloud. We believe that designing such an auditing techniques is an interesting topic of future work.

5 Related Work

FLIPIT was first presented at an invited talk by Ron Rivest at CRYPTO 2011[12]; [14] introduces FLIPIT and gives a formal treatment with theoretical analysis.

In the game theory literature, FLIPIT is related to "repeated games" (see, for example, the excellent text by Mailath and Samuelson [5]), but it differs from them through its stealthy aspect and continuous time. Nonetheless, at a higher level, FLIPIT does share some qualitative characteristics with repeated games. If both players of FLIPIT play adaptively, then FLIPIT acquires the rich complexity of repeated Prisoner's Dilemma, where players may choose to cooperate for their mutual benefit.

FLIPIT is also related to a game of timing [11] where (1) there is an infinite time interval and a finite amount of resources (moves) within each finite subinterval, and (2) the resources/moves of a player are either silent (i.e., the other player does not learn when the moves take place) or noisy with delay till the other player moves (i.e., the other player learns the full history of moves when he moves himself). As future work, we plan to investigate how the theory of games of timing applies to FLIPIT.

Conventional game theory has a long history of application to and enhancement by cryptography and network security; see [4,6] for two surveys. More pertinent to FLIPIT are games modeling system security. Roy et al. [13] offer a taxonomy and survey of game-theoretic models in network security in particular. They note a preponderance of games differing from FLIPIT in two ways: The games assume perfect information (i.e., players know the full game state) and synchronous moves by players.

Some recent information security modeling has made use of extensive forms, which permit complex modeling, strictly within a framework of synchronous moves. This approach gives rise to security games with imperfect information, as in a game devised by Moore et al. [7] to model zero-day disclosure by competing entities. Nguyen et al. [9] consider an abstract, repeated security game with imperfect information and also incomplete information, in the sense that players don't know one another's payoffs. Related work also includes a synchronous territorial game of incomplete, perfect information proposed by Pavlovic [10] which models two-player cybersecurity scenarios for information gathering via deception.

6 Conclusion

While its rules are simple, we have shown that FLIPIT is a conceptually rich security model that yields both important general defensive principles and specific guidance in a number of real-world security scenarios. The Randomized Strategy Principle, for instance, yields a beneficial randomization of password-reset policies. The Drop Out Principle highlights the importance of key rotation frequency. The Move Cost Principle underscores one of the benefits of virtualization, namely its reduction in defender's move costs. FLIPIT offers similarly useful insights across a broad range of real-world security applications, of which we've presented only a small set here. It also gives rise to a wealth of variants applicable to diverse and potentially complex security scenarios.

While this paper provides a glimpse into the applications of FLIPIT, the underlying model of *complete* and *silent* compromise has countless uses, especially in a world where no system is safe and the longstanding assumptions of cryptographers and security system designers can no longer be taken for granted.

References

1. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable data possession at untrusted stores. In: Proc. 14th ACM Conference on Computer and Communication Security, CCS (2007)
2. Barker, E., Barker, W., Polk, W., Smid, M.: Recommendation for key management II: Best practices for key management organization. NIST SP (2/3), 1–79 (2005)
3. Juels, A., Kaliski, B.: PORs: Proofs of retrievability for large files. In: Proc. 14th ACM Conference on Computer and Communication Security (CCS), pp. 584–597 (2007)
4. Katz, J.: Bridging Game Theory and Cryptography: Recent Results and Future Directions. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 251–272. Springer, Heidelberg (2008)
5. Mailath, G.J., Samuelson, L.: Repeated Games and Reputations: Long-run relationships, Oxford (2006)
6. Manshaei, M., Zhu, Q., Alpcan, T., Basar, T., Hubaux, J.P.: Game Theory Meets Network Security and Privacy. Technical report, EPFL (2010)
7. Moore, T., Friedman, A., Procaccia, A.: Would a "cyber warrior" protect us? Exploring tradeoffs between attack and defense of information systems. In: NSPW, pp. 85–94 (2010)
8. Myerson, R.B.: Game Theory—Analysis of Conflict. Harvard University Press (1997)
9. Nguyen, K.C., Alpcan, T., Basar, T.: Security games with incomplete information. In: Proc. IEEE International Conference on Communications, ICC (2009)
10. Pavlovic, D.: Gaming security by obscurity, CoRR abs/1109.5542 (2011)
11. Radzik, T.: Results and problems in games of timing. Statistics, Probability and Game Theory 30 (1996)
12. Rivest, R.L.: Illegitimi non carborundum. Invited keynote talk given at CRYPTO 2011 (August 15, 2011), http://people.csail.mit.edu/rivest/pubs.html#Riv11b
13. Roy, S., Ellis, C., Shiva, S., Dasgupta, D., Shandilya, V., Wu, Q.: A survey of game theory as applied to network security. In: Int. Conf. on System Sciences (HICSS), pp. 1–10 (2010)
14. van Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: FlipIt: The game of "stealthy takeover". To appear in Journal of Cryptology (2012)
15. Witty, R.J., Brittain, K., Allen, A.: Justify identity management investment with metrics. Gartner Group report (February 23, 2004)

Incentives and Security in Electricity Distribution Networks

Saurabh Amin[1], Galina A. Schwartz[2], and Hamidou Tembine[3]

[1] Department of CEE, Massachusetts Institute of Technology - Cambridge, MA, USA
amins@mit.edu
[2] Department of EECS, University of California at Berkeley - Berkeley, CA, USA
schwartz@eecs.berkeley.edu
[3] Département de Télécommunications, École Supérieure d'Électricité
hamidou.tembine@supelec.fr

Abstract. We study incentive problems in electricity distribution when customer energy usage is imperfectly observable by the utility. Thus, we assume that each customer has private information about the amount of his consumed energy. Imperfect observability of individual user demand results is non-technical energy losses. In developing countries, these losses amount to $20-30\%$ per year, and are largely attributed to theft by residential customers. Reducing these losses will allow a marked increase in efficiency of the electricity distribution. Usage of smart energy management devices enables new functionalities and brings the potential for such increased efficiency. However, employing smart energy management devices also entails a new set of problems. Typically, such devices are commercially produced, and employ off-the-shelf information technology (IT) solutions with inherent security vulnerabilities. Thus, due to technology limitations and cost constraints, smart devices are vulnerable to tampering and may enable systemic energy theft, threatening to reduce, or even erase the gains in efficiency. In this paper, we address incentives of utility company to combat theft (i.e., non-technical losses), when utility is subject to rate (tariff) regulation. From our analysis, such regulated utilities invest less than socially optimal in theft reduction. We suggest that regulators should include explicit targets for the allowable losses to remedy the problem of incentive misalignment.

1 Introduction

Energy theft in emerging economies has been a wide-spread practice. World Bank report [1] states that up to 50% of electricity consumed in certain parts of developing countries is acquired by means of theft. Here physical security considerations range from default on payment to stealing of energy through equipment manipulation. Second, cyber security threats to Advanced Metering Infrastructures (AMIs) are abundant. The AMI technology aims to cut cost of utilities and increase energy efficiency by providing new functionalities, including reducing unmetered and unbilled consumption. Yet, the AMI technology does not employ security-by-design principles [2],[3]. Unsurprisingly, a number of studies

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 264–280, 2012.

have demonstrated that smart meters can be manipulated via tampering of physical and communication components as well as message spoofing [4],[5]. Also, [6] demonstrates the increased risk of energy theft which further justifies the importance of security considerations for AMIs. Finally, Anderson and Fuloria [7] point out that energy auditing and billing systems for AMIs suffer losses due to technical (e.g., transmission loss) and non-technical (e.g., fraud) reasons.

From the perspective of utility operator, the electricity losses in the distribution system are the amounts of electricity injected into the distribution network, which are not being paid by the users. These losses can be sub-divided on technical or non-technical. The resulting level of losses depends on the choices of utility operator and his customers. These choices are interdependent, and they also depend on technological and institutional environments.

Technical losses of the distributor occur due to the energy dissipation (i.e., current flowing) though resistive conductors and equipment used for power transmission, transformation, sub-transmission, and distribution. Non-technical losses occur due to the actions of (i) utility operator (for example, administrative losses due to errors in accounting and record keeping), (ii) customer theft (fraud or willful pilferage by bona-fide customers), (iii) customer non-payment (i.e., default), and lastly the theft by (iii) the outsiders (non-customers). In some cases, administrative errors are strategic, i.e., made with a purpose of assisting customer theft. Once the theft is detected, and the culprit is found, the losses (ii) and (iv) become subject of recovery, as in case (iii).

In this paper, we distinguish two main effects of the deployment of smart energy management devices. First, these devices permit to reduce the costs of utility operator via new functionalities, for example, the improved precision of dispatch, computerized metering and billing infrastructures. Second, these devices give customers new means for energy theft, for example via exploring IT insecurities. Our model could be straightforwardly modified to allows parametric assessment of these effects. And, while this paper focuses on the distribution system losses only, losses of the transmission system could be addressed in a similar manner.

The paper is organized as follows. In Section 2, we present an overview of non-technical losses in electricity distribution networks, and in Section 3 we briefly describe the regulatory regimes that are currently employed in different countries. Section 4 presents the model of interaction between consumers and a monopolist distributor. The consumers face a non-linear tariff for billed electricity, and are subject to an exogenous fine schedule for stolen/unbilled electricity (if detected). The distributor faces imperfect information about the consumer preferences and can partially recover the unbilled electricity using detection and enforcement mechanisms facilitated by Advanced Metering Infrastructures (AMIs). Under realistic assumptions on the probability of detection and fine schedule, we characterize the equilibrium consumption levels of billed and unbilled electricity, as well as the optimal tariff schedule and investment level of the distributor. In Section 5, we analyze the profit-maximizing tariff schedule and investment level when the distributor is subject to price cap

regulation. In particular, for average revenue regulation, we find that the investment level in AMIs can be sub-optimal relative to a perfectly informed regulator. In Section 6, we summarize our findings and conclude.

2 Non-technical Losses of Distribution System

The non-technical losses in electric distribution networks are due to theft, fraud, or uncollected (defaulted) payments. The consumers who fail to pay for electricity by acquiring it via stealing or defaulting on their bills, obtain the electricity at near zero prices. Effectively, electricity consumption of these non-paying parties is subsidized, because their consumption is paid by other members of the society. Specifically, the non-technical losses could be covered via (i) higher electricity tariffs for other consumers; (ii) the entire society (via taxes) if government subsidizes the distributor for these losses. In some cases, these losses remain uncovered for prolonged periods of time. Clearly, this situation negatively affects the efficiency of distribution system.

In most developed countries, the combined losses of transmission and distribution (T&D) systems do not exceed 10% [8]. First, the technical losses have been small due to historically adequate levels of investments in T&D, development and deployment of efficient transformers and other electric equipment, and transmission at higher voltages. For e.g., T&D losses in the US decreased from more than 16% in 1920 to less than 7% today. Only about half of these losses occur in the distribution system. Second, the non-technical losses in developed countries are also small, and in many countries even negligible. Industrialized countries have nearly 100% electrification, and for residential customers, expenses on electricity constitute a relatively low share of household incomes. For e.g., in today's US, electricity theft is considered unimportant. In comparison, the data for Italy suggests unusually high losses from theft. In the UK, the T&D losses are also high due to the aging grid infrastructure.

2.1 Losses in Developing Countries

In contrast with developed countries, many developing countries still experience high distribution system losses [9]. For South Asia (for example, India and Pakistan), and most sub-Saharan Africa countries, various official and unofficial estimates of T&D losses range from $15 - 50\%$ [10]. Especially high levels of non-technical losses ultimately bear on the electricity rates (which are typically regulated), or higher taxes, or both.

We now briefly discuss the barriers in reducing these losses: Oftentimes, certain categories of consumers (e.g., agricultural, rural, or underprivileged consumers) are unmetered or pay a flat rate, i.e., the payment does not depend on quantity of consumed electricity. Such customers tend to increase their connected loads without obtaining the required sanctions for the increases of their loads. The under-payment by these consumers is often recovered from industrial or commercial customers who face higher tariffs. Such cross-subsidization,

combined with unreliability in supply (e.g., poor frequency control and irregular load-shedding during high-demand periods) encourages commercial enterprises to install their own local supply such as diesel generators. These locally generated electricity is expensive, and thus its usage introduces inefficiencies. Moreover, the distribution utility has incentives to fudge the consumption figures. Its reports to the regulator (e.g., public utility commissions) tend to provide higher estimates of unmetered consumption to under-report the actual losses.

Below we summarize the main channels of non-technical losses:

- Theft via availing unauthorized/unrecorded supply by tapping into conductors, feeders, and tampering service wires.
- Theft via willful pilferage by customers includes damaging or manipulating electric equipment and meters installed in their premises.
- Theft that is assisted by corrupt distribution utility's employers, who could make intentional billing errors in favor of certain consumers.
- Administrative losses including the errors in metering and billing of the actually consumed quantity, and errors in collecting billed amounts.

Combatting non-technical losses requires reducing the losses at each channel. This could be achieved via the implementation of measures at technological (e.g., detection tools) and organizational (e.g., enforcement capabilities) levels. We distinguish the following four categories of technological and regulatory measures that could be adopted to limit the non-technical losses:

(1) *Technological (hardware) measures*: Installation of IT-supported meters at distribution transformers and feeders; Providing data-logging, remote monitoring and communication capabilities; Automated Meter Reading (AMR) and Advanced Metering Infrastructures (AMIs) to eliminate unmeasured and unbilled consumption.
(2) *Technological (software) measures*: Management information systems equipped with data analytics tools to improve metering, billing and collection processes, and detection of fraud and unmetered connections.
(3) *Regulatory measures*: Strengthening enforcement mechanisms (e.g., prosecution of theft); Publicizing theft cases for sharper public scrutiny (e.g., using the name and shame effect); Making consumers aware that electricity theft is a cognizable offense; Disconnection of customers related to fraud/debts and reconnection after clearance.
(4) *Institutional measures*: Fixing the skewed tariff structures; Providing coordination and transparency in distribution operations; Investing in hardware and software upgrades.

2.2 Reforms in Distribution Sector

During past three decades, the power sector has experienced reforms. Overall, the reforms have resulted in unbundling of power sector operations, introduction of competitive wholesale electricity markets, and privatization of existing companies at the generation, transmission, and distribution levels. In this paper, we limit our attention to the distribution sector.

Before the reforms, the electricity distribution was largely provided by utilities, operating as state-owned enterprises (SOE). In general, these utilities tend to suffer from poor operational performance. The institutional environments with SOE typically also feature ill-defined and conflicting regulations, which distort managerial incentives, and could result in corrupt monitoring and enforcement practices. In addition, the state-owned utilities inherit other difficulties typical for non-market environments, such as sub-standard investment and overall poor managerial incentives. This translates in poor productivity, manifested by high losses and overall costs, and low service quality. In many cases, significant provision inefficiencies have resulted in widespread customer dissatisfaction.

Such non-performance of state-owned utilities necessitate the reforms of distribution system: drastic reorganizations of regulatory regime, and utility operator practices, including changes in ownership structures of utilities. Publicly available data about reforms is scarce, but there are clear indications that successful reforms of distribution system have resulted in substantial reduction of losses [11]. For e.g., independent regulatory commissions have been set up for licensing, regulating tariff structures, and promoting efficiency and competition. Indeed, distribution sector reforms have achieved increased efficiency levels by cutting technical and non-technical losses, and improving service quality. These reforms can be sustained over-time provided that properly designed institutional and regulatory framework to eliminates losses exists, and utilities have adequate incentives to improve their performance.

Even though post-reform losses of electricity distribution in developing countries are substantially lower than the respective pre-reform losses, *why these countries still face substantially higher losses in comparison to the developed countries?* Our stylized model suggests that imperfectly designed regulations, in particular, suboptimal levels of investment levels in monitoring and enforcement, could be responsible for that. In the next section, we present relevant insights from regulation of distribution sector.

3 Regulated Electricity Distribution

Electricity delivery to the end *consumers* is provided by utility companies (*distributors*), which operate as regulated monopolists. Each distributor can be viewed as an exclusive franchise subject to tariff and performance regulation. The entity responsible for overseeing the distributor is referred as the *regulator*. Thus, the actions of three types of entities are relevant for electric distribution: the regulator, the distribution utilities, and the consumers. The principles for regulating tariff structures are broadly similar across ownership structures of utilities (publicly owned and investor owned). Typically, the regulator's objective is to maximize consumer surplus, subject to a participation constraint for utility, and possibly other requirements, such as minimal level of service quality. The regulator's objectives can be summarized as [8]: operational efficiency to ensure reliably delivery at lowest reasonable cost, dynamic efficiency to meet

future demand, and consumption efficiency to ensure lowest prices subject to distributor's cost recovery and investment incentives. We also refer the reader to [12,13] for a modern treatment of regulatory principles in electricity distribution.

3.1 Asymmetric Information

The central issue in the design of regulatory policies in electricity distribution is the presence of *asymmetric information*; see Fig. 1. If regulator has perfect information about the distributor's costs and efficiency levels, and the consumer demand, designing regulatory requirements is straightforward [14]. If regulator's information is imperfect, and especially if hidden information is present, regulatory design becomes difficult, see [15]. The distributor has better knowledge of consumers' demand and its own technological capabilities (e.g., operational costs) in comparison to the regulator. There is a well-developed body of work on designing optimal regulatory policies of a monopoly distributor where he has privileged information about his technological capabilities and customers' demand and the regulator has well-defined inter-temporal commitment powers. However, such a normative analysis assumes that the regulator, although imperfectly informed about distributor's technological capabilities and customers' demand, perfectly knows the structure of regulated environment and has a formal model of information asymmetry between the regulator and the distributor.

Still, in practical situations, the precise nature of information asymmetry and the full set of relevant constraints on the regulator and the distributor are difficult to characterize a priori. Thus, design of regulatory policies should importantly take the robustness into account [16]. That is "well designed" regulatory policy must be robust, i.e., it must perform "reasonably well" under broad conditions, although such a policy may be sub-optimal in each particular setting. There are two main regulatory regimes that have been theoretically well-studied and adapted for a variety of practical settings: (i) rate of return (dominant regime in USA) and (ii) price cap (dominant regime in many parts of European Union and in some developing countries). Below we briefly outline each regime, but subsequently limit our analysis to price cap regulation.

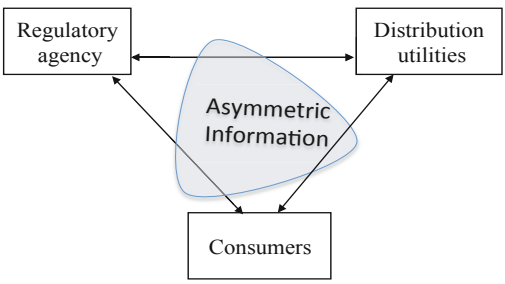


Fig. 1. Players in regulated electricity distribution

3.2 Rate-of-Return versus Price Cap Regulation

Under *rate-of-return regulation*, the distribution utility is allowed a rate-of-return, and the rate structures for the electricity delivery are adjusted as the cost changes to ensure that the distributor has the opportunity to earn an authorized return. Here the regulator bears the onus of setting the prices and ensures that the realized rate of return does not deviate significantly from the target rate. Since the prices are directly linked to the realized costs, the distributor is unlikely to engage in cost-reducing activities. A classical example is the Averch-Johnson effect, which shows that the rate-of-return regulation deviates from cost minimization. However, since distributor faces limited risks of expropriation of his sunk investments by the regulator, upgrades of distribution network can be sustained in this form of regulation. The investment behavior of regulated distributor is especially important, since the infrastructure upgrades (e.g., capacity expansion) and modernization (e.g., AMI installations) require substantial costs.

Under *price cap regulation*, the tariffs provided to customers could increase, on average, at a specified rate during a pre-specified time. The specified rate is typically linked to the overall rate of inflation, and may fail to reflect the distributor's short-term realized costs and/ or profit. Typically, under a price cap regulatory regime, only average prices are controlled by the regulator, and the utility is given the flexibility to control the pattern of relative prices subject to pre-defined constraints. Since the tariffs are fixed and / or change according to a pre-specified rate for relatively long periods of time, the distributor has incentives to minimize its operating costs, and thus to operate efficiently.

Notice, that price cap does not directly provide good incentives for long-term investments in production, such as distribution network upgrades and reduction of non-technical losses. Similarly, price cap does not incentivize the distributor to choose optimal allocation of service quality. In this paper, we demonstrate that price cap regulation fails to incentivize the distributor to invest in monitoring and enforcement efforts to reduce unbilled electricity (e.g., consumer theft) at socially optimal levels.

When the pricing flexibility of price cap regulation is combined with the rewards (resp. punishments) for performance improvement (resp. deterioration) relative to the regulator's benchmark, the resulting regime is termed *performance-based* (or incentive) regulation. Indeed, in the face of rapidly changing technological environment and evolving customer preferences, the regulated electricity distribution industry is moving toward incentive regulation. The goal of incentive regulation is to improve distributor's incentives by decoupling regulated price structure from the need to know the exact operating / maintenance costs.

4 Consumer-Distributor Model

4.1 Consumer Preferences

We consider a population of consumers in which the individual tastes vary according to a type parameter θ. Let θ be distributed across the population

according to the density $f^c(\theta)$ with cumulative distribution function $F^c(\theta)$ on an interval $[\underline{\theta}, \bar{\theta}]$ (where $0 \leq \underline{\theta} < \bar{\theta}$). The electric distribution utility (a monopolist) cannot distinguish the type of given consumer, but knows the distribution $F^c(\theta)$.

Let us denote the billed and unbilled quantities for type-θ consumer as $\mathrm{q_B}(\theta)$ and $\mathrm{q_U}(\theta)$, respectively. The total consumed quantity is $\mathrm{q} = \mathrm{q_B} + \mathrm{q_U}$. The distribution utility (or distributor) offers a tariff (i.e., pricing schedule) $\mathrm{T}(\mathrm{q_B})$, which specifies for each billed quantity $\mathrm{q_B}(\theta)$, the total sum that the type-θ consumer should pay to the distributor. Special cases includes a linear pricing schedule corresponding to a single price, i.e., $\mathrm{T}(\mathrm{q_B}) = \mathrm{pq_B}$, and affine pricing schedule corresponding to a two-part tariff, i.e., $\mathrm{T}(\mathrm{q_B}) = \mathrm{A} + \mathrm{pq_B}$. Here A is the fixed premium (or rental) and p is the charge varying with number of billed units. However, in general, the distributor can offer nonlinear tariff $\mathrm{T}(\mathrm{q_B})$.

The unbilled quantity $\mathrm{q_U}$ constitute non-technical losses to the distributor and result from theft, fraud, or payment default by the consumers. If the distribution utility deploys Advanced Metering Infrastructures (AMIs), it improves its monitoring and billing efficiency, and thus reduces $\mathrm{q_U}$. Let us denote the level of distributor's effort in deploying AMIs by $\mathrm{e} \in \mathbb{R}_+$.[1] The efficiency of recovering unbilled electricity increases with e, and can be modeled as $\rho : \mathbb{R}_+ \to (0, 1)$ which assigns for to each investment level e, a probability of detection. Thus, type-θ consumer's unbilled electricity is detected by the distributor with probability $\rho(\mathrm{e})$, and undetected with probability $(1 - \rho(\mathrm{e}))$.

Let $\mathrm{F^r}(\mathrm{q_U})$ denote the fine schedule that is exogenously fixed by the regulator (and hence the superscript r), and is known to consumers and the distributor. If the unbilled electricity $\mathrm{q_U}(\theta)$ were perfectly detected, a consumer of type θ would pay $\mathrm{F^r}(\mathrm{q_U}(\theta))$ to the distributor. However, under imperfect detection, the distributor only recovers payment for $\rho(\mathrm{e})\mathrm{q_U} < \mathrm{q_U}$ via fines, and the remaining quantity, $(1-\rho(\mathrm{e}))\mathrm{q_U}$, is considered stolen. In accordance with current detection technology and enforcement practices, we impose the following assumption:

Assumption 1 *$\rho(\cdot)$ is concave increasing in* e*, and $\mathrm{F^{r\prime}}(\cdot)$ is nondecreasing in* $\mathrm{q_U}$.

Suppose that each consumer has the following utility function:

$$\mathrm{U} = \begin{cases} \theta u(\mathrm{q_B} + \mathrm{q_U}) - \mathrm{T}(\mathrm{q_B}) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}) & \text{[AMIs deployed with effort e]} \\ \theta u(\mathrm{q_B} + \mathrm{q_U}) - \mathrm{T}(\mathrm{q_B}) & \text{[AMIs are not deployed]}, \end{cases} \tag{1}$$

where the function $u(\cdot)$ satisfies $u(0) = 0$, $u'(\mathrm{q}) > 0$, and $u''(\mathrm{q}) < 0$, i.e., there is a decreasing marginal utility of electricity consumption. Also, $u(\cdot)$ is assumed to be same for all consumers.

In our model, the unbilled electricity is undetectable when AMIs are not deployed. Then, $\mathrm{q_B} \equiv 0$ becomes a trivial solution, i.e., the consumers do not

[1] The theory of regulation [17] has considered the distributor's cost reducing effort e. In this paper, e is specific to deployment of AMIs, and specifies the monitoring and enforcement effort of the distributor for reducing the quantity of unbilled electricity.

prefer a quantity dependent tariff (although they may still pay a flat rate). Hence, we only consider the case when the distributor deploys AMIs at effort level e. A type-θ consumer facing the distributor's tariff schedule $\mathrm{T}(\mathrm{q_B})$ and a fine schedule $\mathrm{F^r}(\mathrm{q_U})$ obtains a net surplus $v(\theta)$ given by

$$v(\theta) \equiv \max_{\mathrm{q_B}\geq 0, \mathrm{q_U}\geq 0} \left[\theta u(\mathrm{q_B}+\mathrm{q_U}) - \mathrm{T}(\mathrm{q_B}) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U})\right]. \tag{2}$$

The first-order-conditions (FOCs) are:

$$\theta u'(\mathrm{q_B}+\mathrm{q_U}) - \mathrm{T}'(\mathrm{q_B}) = 0, \quad \text{and} \quad \theta u'(\mathrm{q_B}+\mathrm{q_U}) - \rho(\mathrm{e})\mathrm{F^{r\prime}}(\mathrm{q_U}) = 0, \tag{3}$$

which implies

$$\theta u'(\mathrm{q}(\theta)) = \mathrm{T}'(\mathrm{q_B}(\theta)) = \rho(\mathrm{e})\mathrm{F^{r\prime}}(\mathrm{q_U}(\theta)). \tag{4}$$

Hence, a small increase in the total quantity consumed by a type-θ consumer generates the marginal surplus $\theta u'(\mathrm{q}(\theta))$ equal to the marginal payment $\mathrm{T}'(\mathrm{q_B}(\theta))$ (resp. *expected* marginal fine $\rho(\mathrm{e})\mathrm{F^{r\prime}}(\mathrm{q_U}(\theta))$) for a small increase in the billed (resp. unbilled) quantity. Once the quantity functions $\mathrm{q_B}$ and $\mathrm{q_U}$ are known, the payment function can be obtained using (4). Also, since $\rho < 1$, we obtain

$$\mathrm{T}'(\mathrm{q_B}(\theta)) < \mathrm{F^{r\prime}}(\mathrm{q_U}(\theta)), \quad a.e.$$

Without loss of generality we assume that the tariff and fine schedules satisfy $\mathrm{T}(0) \leq 0$ and $\mathrm{F^r}(0) \leq 0$, respectively (because the consumers have the option of consuming nothing at zero cost). Under our assumptions, the following holds:

Lemma 1. *(i) $v(\cdot)$ is non-negative, increasing, convex, and differentiable almost everywhere (a.e.); (ii) For a type-θ consumer, the chosen (optimal) quantity of electricity, $\mathrm{q}(\theta) \equiv \mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)$, is unique, increasing in θ, and is given by $v'(\theta) = u(\mathrm{q}(\theta))$; (iii) the chosen billed $\mathrm{q_B}(\theta)$ and unbilled $\mathrm{q_U}(\theta)$ quantities are both unique, and satisfy*

$$\mathrm{T}'(\mathrm{q_B}(\theta)) = \frac{\theta v''(\theta)}{\left(d\mathrm{q}(\theta)/d\theta\right)}, \quad \mathrm{F^{r\prime}}(\mathrm{q_U}(\theta)) = \frac{\theta v''(\theta)}{\rho(\mathrm{e})\left(d\mathrm{q}(\theta)/d\theta\right)}, \quad a.e. \tag{5}$$

Remark 1. The distributor's collection efficiency can be defined as follows:

$$\eta \equiv 1 - \frac{\int_{\underline{\theta}}^{\bar{\theta}} (1-\rho(\mathrm{e}))\mathrm{q_U}(\theta) f^c(\theta) d\theta}{\int_{\underline{\theta}}^{\bar{\theta}} \left(\mathrm{q_B}(\theta)+\mathrm{q_U}(\theta)\right) f^c(\theta) d\theta}. \tag{6}$$

4.2 Distributor's Revenue

Let us introduce the revenue function of the distributor. For a quantity Q of total electricity provisioned by the distributor, we define the revenue function $\mathrm{R}(\mathrm{Q})$ as his maximum revenue, when he offers a tariff schedule $\mathrm{T}(\mathrm{q_B})$ for billed

quantity $\mathrm{q_B}(\theta)$, and implements a fine schedule $\mathrm{F^r}(\mathrm{q_U})$ to recover the unbilled quantity $\mathrm{q_U}(\theta)$ with probability $\rho(\mathrm{e})$:

$$\mathrm{R(Q)} = \max \int_{\underline{\theta}}^{\bar{\theta}} \left[\mathrm{T}(\mathrm{q_B}(\theta)) + \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\theta))\right] f^c(\theta) d\theta, \text{ subject to} \tag{7}$$

$$\forall \theta, \quad \left[\theta u(\mathrm{q_B} + \mathrm{q_U}) - \mathrm{T}(\mathrm{q_B}) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U})\right] \geq 0, \tag{7a}$$

$$\mathrm{q_B}(\theta), \mathrm{q_U}(\theta) \text{ maximize } \left[\theta u(\mathrm{q_B} + \mathrm{q_U}) - \mathrm{T}(\mathrm{q_B}) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U})\right], \tag{7b}$$

$$\mathrm{Q} \geq \int_{\underline{\theta}}^{\bar{\theta}} \left(\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)\right) f^c(\theta) d\theta, \tag{7c}$$

Here *individual-rationality* (IR) constraint (7a) ensures that all consumers are willing to purchase. Actually, it suffices to require that the lowest demand consumer (type-$\underline{\theta}$) is individually rational, i.e.,

$$\left[\underline{\theta} u(\mathrm{q_B}(\underline{\theta}) + \mathrm{q_U}(\underline{\theta})) - \mathrm{T}(\mathrm{q_B}(\underline{\theta})) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\underline{\theta}))\right] \geq 0.$$

The constraint (7b) ensures that the consumers do not exercise personal arbitrage. In other words, it requires that $\forall \theta$, $\tilde{\theta}$

$$\begin{aligned}\mathrm{U}(\theta) =& \theta u(\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)) - \mathrm{T}(\mathrm{q_B}(\theta)) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\theta)) \\ &\geq \theta u(\mathrm{q_B}(\tilde{\theta}) + \mathrm{q_U}(\tilde{\theta})) - \mathrm{T}(\mathrm{q_B}(\tilde{\theta})) - \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\tilde{\theta})),\end{aligned}$$

i.e., the type-θ consumer must not choose the same quantity bundles as chosen by the type-$\tilde{\theta}$ consumer (where $\tilde{\theta} \neq \theta$). These are known as *incentive compatibility* (IC) constraints. Finally, the constraint (7c) ensures that the billed plus unbilled quantity of electricity is no greater than Q.

We now seek an alternative representation of the revenue maximization problem (7). Let us write $\mathrm{T}(\mathrm{q_B}(\theta)) + \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\theta)) = \theta u(\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)) - v(\theta)$, and recall from Lemma 1 that $v'(\theta) = u(\mathrm{q}(\theta))$. We can express the net surplus of type-θ consumer as

$$v(\theta) = \int_{\underline{\theta}}^{\theta} u(\mathrm{q_B}(\zeta) + \mathrm{q_U}(\zeta)) d\zeta + v(\underline{\theta}) = \int_{\underline{\theta}}^{\theta} u(\mathrm{q_B}(\zeta) + \mathrm{q_U}(\zeta)) d\zeta,$$

where the second equality uses the IR constraint ($v(\underline{\theta}) = 0$). Then, the revenue maximization problem (7) can be re-written as:

$$\mathrm{R(Q)} = \max \int_{\underline{\theta}}^{\bar{\theta}} \left[\theta u(\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)) - \int_{\underline{\theta}}^{\theta} u(\mathrm{q_B}(\zeta) + \mathrm{q_U}(\zeta)) d\zeta\right] f^c(\theta) d\theta,$$

subject to the constraints (7b), and (7c).

It is straightforward to see that the constraint (7c) is binding, because R(Q) can be increased by allocating larger quantities to high consumer types. Integrating by parts, and noting that constraint (7b) is equivalent to imposing that $\mathrm{q}(\cdot)$ is nondecreasing in θ where $\mathrm{q_B}(\cdot)$ and $\mathrm{q_U}(\cdot)$ verify (5), we obtain

$$\mathrm{R(Q)} = \max \int_{\underline{\theta}}^{\bar{\theta}} \left[\theta f^c(\theta) - (1 - F^c(\theta))\right] u(\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta)) d\theta \text{ subject to} \tag{8}$$

$$\text{(i)} \quad \mathrm{q}(\theta) \equiv \mathrm{q_B}(\cdot) + \mathrm{q_U}(\cdot) \text{ nondecreasing, and (5) holds} \tag{8a}$$

$$\text{(ii)} \quad \mathrm{Q} = \int_{\underline{\theta}}^{\bar{\theta}} (\mathrm{q_B}(\theta) + \mathrm{q_U}(\theta))\, f^c(\theta) d\theta. \tag{8b}$$

In solving the above optimization problem, we initially ignore the constraint (i) but verify it *ex post*. Let $\lambda(\mathrm{Q})$ be the Lagrange multiplier associated with constraint (ii). From the Envelope Theorem, we obtain the following useful relation:

$$\mathrm{R}'(\mathrm{Q}) \equiv \lambda(\mathrm{Q}). \tag{9}$$

Since $\mathrm{R(Q)}$ is non-decreasing in Q, we conclude that $\lambda(\mathrm{Q})$ is non-negative. Moreover, the optimal response functions $\mathrm{q}^*(\theta) \equiv (\mathrm{q_B}^*(\theta) + \mathrm{q_U}^*(\theta))$ satisfy:

$$\mathrm{q}^*(\theta) = \arg\max_{\mathrm{q} \geq 0} \left[\theta - \frac{1 - F^c(\theta)}{f^c(\theta)}\right] u(\mathrm{q}) - \lambda(\mathrm{Q})\mathrm{q}, \tag{10}$$

The FOC for pointwise maximization of (10):

$$\theta u'(\mathrm{q}^*(\theta)) = \frac{\lambda(\mathrm{Q})}{\left[1 - \frac{1-F^c(\theta)}{\theta f^c(\theta)}\right]}. \tag{11}$$

We make the following standard assumption about the *hazard rate* of the type distribution, which holds for many common distributions including uniform, normal, logistic, exponential, etc.

Assumption 2 *The hazard rate of type distribution* $\frac{f^c(\theta)}{1-F^c(\theta)}$ *increases with* θ.

From Assumption (2), we observe that the expression $\left[\theta - \frac{1-F^c(\theta)}{f^c(\theta)}\right]$ increases with θ. Then, from (11) and the fact that u is concave, $\mathrm{q}^*(\theta)$ is increasing in θ. To complete checking the constraint (8a), see (12) below. The following lemma follows from [18]:

Lemma 2. *Under Assumption (2), the revenue function* $\mathrm{R(Q)}$ *is strictly concave.*

From (9) and Lemma 2, we observe that $\lambda(\mathrm{Q})$ decreases with Q.

Now let $\mathrm{p}^*(\mathrm{q_B}) \equiv (\mathrm{T}^*)'(\mathrm{q_B})$ denote the marginal price for the billed quantity corresponding to the optimal tariff schedule $\mathrm{T}^*(\mathrm{q_B})$. Similarly, let $\mathrm{p_f^r}(\mathrm{q_U}) \equiv (\mathrm{F^r})'(\mathrm{q_U})$ denote the fine for an extra unit of unbilled electricity (if detected), when the consumer has an unbilled amount $\mathrm{q_U}$. Equation (11), and the FOCs (3) for $\mathrm{q_B}^*(\theta) > 0$ and $\mathrm{q_U}^*(\theta) >$ to be optimal choices for type-θ consumer, imply the following result:

Theorem 3. *Let the assumption 2 hold. Then, for a quantity of total electricity* Q *and AMI investment level* e *by the distributor, the marginal price schedule and the marginal fine schedule satisfy*

$$\mathrm{p}^*(\mathrm{q_B}^*(\theta)) = \rho(\mathrm{e})\mathrm{p_f^r}(\mathrm{q_U}^*(\theta)) = \frac{\lambda(\mathrm{Q})}{\left[1 - \frac{1-F^c(\theta)}{\theta f^c(\theta)}\right]}, \tag{12}$$

where $\mathrm{q_B}^*(\theta) + \mathrm{q_U}^*(\theta) = \mathrm{q}^*(\theta)$*, with* $\mathrm{q}^*(\cdot)$ *given by* (10)*.*

Since we assume that $\mathrm{p_f^r}(\cdot)$ is nondecreasing in $\mathrm{q_U}$, (12) implies that the optimal consumer choice of billed (resp. unbilled) electricity is increasing (resp. non-increasing) in θ, i.e.,

Corollary 1. *Under assumptions 1 and 2,* $\mathrm{q_U}^*(\cdot)$ *(resp.* $\mathrm{q_B}^*(\cdot)$*) is non-increasing (resp. increasing) in* θ*.*

We now deduce the shape of optimal tariff schedule. Since $\mathrm{p}^*(\mathrm{q_B}^*(\theta))$ is decreasing in θ (from (12)), and $\mathrm{q_B}^*(\cdot)$ is increasing in θ (from Corollary 1), we conclude that $\mathrm{p}^*(\mathrm{q_B})$ is decreasing in $\mathrm{q_B}$. Thus, under the assumption on non-decreasing marginal fine schedule, we obtain that $\mathrm{T}^*(\cdot)$ is concave in $\mathrm{q_B}$. This is the classical quantity discount result for a revenue maximizing distributor [19]!

4.3 Unregulated Distributor

Consider an unregulated distributor with aggregate cost function $\mathrm{C}(\beta, \mathrm{e}, \mathrm{Q})$ of provisioning the total quantity of electricity Q and effort level $\mathrm{e} \geq 0$ for detecting unbilled electricity via AMIs. The parameter $\beta \in [\underline{\beta}, \bar{\beta}]$ signifies the distributor's technological efficiency. Thus, a distributor of type $\underline{\beta}$ (resp. $\bar{\beta}$) is most (resp. least) efficient in reducing nontechnical losses (and hence, unbilled electricity). We assume the following standard assumptions: $\partial_\beta \mathrm{C} > 0, \partial_\mathrm{e} \mathrm{C} < 0, \partial_\mathrm{Q} \mathrm{C} > 0$.

Let $\psi(\mathrm{e})$ denote the distributor's fixed cost of deploying AMIs at effort level e, where $\psi'(\mathrm{e}) > 0, \psi''(\mathrm{e}) > 0$. The problem of computing the profit maximizing quantity of electricity Q^* and optimal investment level e^* for an unregulated monopolist (who knows β) can be written as

$$\pi^m(\beta) = \max_{\mathrm{Q}\geq 0, \mathrm{e}\geq 0} \mathrm{R}(\mathrm{Q}) - \mathrm{C}(\beta, \mathrm{e}, \mathrm{Q}) - \psi(\mathrm{e}), \tag{13}$$

Using (9), the FOCs for (13) involve setting Q^* and e^* to satisfy

$$\partial_\mathrm{Q} \mathrm{C}(\beta, \mathrm{e}^*, \mathrm{Q}^*) = \lambda(\mathrm{Q}^*), \quad \text{and} \quad \partial_\mathrm{e} \mathrm{C}(\beta, \mathrm{e}^*, \mathrm{Q}^*) = -\psi'(\mathrm{e}^*). \tag{14}$$

Then, from Theorem 3, the distributor chooses a tariff schedule $\mathrm{T}^*(\mathrm{q})$ and investment level e^* such that its profit from supplying the total quantity Q^* is maximized.

For simplicity, let us assume the following cost function:

$$\mathrm{C}(\beta, \mathrm{e}, \mathrm{Q}) = (\beta - \mathrm{e})\mathrm{Q}, \tag{15}$$

where marginal cost of distribution $\beta - \mathrm{e} > 0$ over the relevant range of operation. For cost function (15), $\mathrm{e}^*(\beta)$ and $\mathrm{Q}^*(\beta)$ satisfy:

$$\mathrm{e}^*(\beta) = \beta - \lambda(\mathrm{Q}^*(\beta)), \quad \mathrm{Q}^*(\beta) = \psi'(\mathrm{e}^*(\beta)).$$

Notice from (12) that the highest demand consumer pays the marginal *aggregate* cost for the billed electricity, i.e.,

$$\mathrm{p}^*({\mathrm{q_B}}^*(\bar{\theta})) = \partial_{\mathrm{Q}}\mathrm{C}(\mathrm{e}^*, \mathrm{Q}^*) = \beta - \mathrm{e}^*(\beta),$$

where we have used the fact that $F^c(\bar{\theta}) = 1$.

Remark 2. Following (6), the distributor's collection efficiency becomes

$$\eta^*(\mathrm{e}^*, \mathrm{Q}^*) = 1 - \frac{(1-\rho(\mathrm{e}^*)) \int_{\underline{\theta}}^{\bar{\theta}} {\mathrm{q_U}}^*(\theta) f^c(\theta) d\theta}{\mathrm{Q}^*}, \quad \text{where } {\mathrm{q_U}}^*(\theta) \text{ satisfies (12).}$$

To summarize, the interaction between consumers and distributor can be viewed as a non-zero sum *Stackelberg game*, where the distributor acts as leader and the consumers act as followers.[2] The fine schedule $\mathrm{F^r}(\cdot)$ and detection probability function $\rho(\cdot)$ are common knowledge. Based on his prior belief of consumer types $f^c(\cdot)$, the monopolist distributor offers a tariff schedule $\mathrm{T}(\cdot)$, and also selects output level Q and AMI investment level e. A type-θ consumer, knowing the strategy of the distributor, chooses his consumption levels of billed ${\mathrm{q_B}}^*(\theta)$ and unbilled ${\mathrm{q_U}}^*(\theta)$ quantities to maximize his individual utility; see Section 4.1. The distributor, knowing the consumers' rationale, must choose optimal Q^*, e^*, and $\mathrm{T}^*(\cdot)$ to maximize his profit.

5 Price Cap Regulation

We now analyze a form of price cap regulation in which the distributor faces an average revenue constraint. The distributor can offer tariff T and enforce penalty $\mathrm{F^r}$ with AMI investment level e, only if the induced consumer demand functions $\mathrm{q_B}(\theta)$ and $\mathrm{q_U}(\theta)$ permit an average revenue that is no more than a regulator-specified price cap. Two possible average revenue constraints are:

$$\int_{\underline{\theta}}^{\bar{\theta}} \left[\mathrm{T}(\mathrm{q_B}(\theta)) + \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\theta))\right] f^c(\theta) d\theta \leq \bar{\mathrm{p}}\mathrm{Q} \tag{15a}$$

$$\int_{\underline{\theta}}^{\bar{\theta}} \left[\mathrm{T}(\mathrm{q_B}(\theta)) + \rho(\mathrm{e})\mathrm{F^r}(\mathrm{q_U}(\theta))\right] f^c(\theta) d\theta \leq \bar{\mathrm{p}}(\mathrm{Q} - \mathrm{Q_S}), \tag{15b}$$

where $\bar{\mathrm{p}}$ is the maximum permitted level of average revenue per unit of electricity determined by the regulator (i.e., $\bar{\mathrm{p}}$ is the *price cap*), Q is the distributor's total

[2] Stackelberg games have been used in the context of incentive design in both engineering [20,21,22] and economics [14,15,16].

output ($Q = \int_{\underline{\theta}}^{\bar{\theta}} (q_B(\theta) + q_U(\theta) f^c(\theta) d\theta)$, and Q_S is the net quantity of stolen electricity ($Q_S = (1-\rho(e)) \int_{\underline{\theta}}^{\bar{\theta}} q_U(\theta) f^c(\theta) d\theta$). Thus, (15b) is a stricter constraint in comparison to (15a) because it excludes the stolen electricity in computing the average revenue, and only accounts for the billed plus recovered quantities.

From the regulator's viewpoint, the constraint (15b) is more desirable because he considers the consumer surplus for given tariff and fine schedules to be $S(T, F^r) = \int_{\underline{\theta}}^{\bar{\theta}} v(q_B(\theta + \rho(e) q_U)) f^c(\theta) d\theta$. In determining the price cap $\bar{p}$, the regulator will not account for the consumers' surplus resulting from successfully stolen (undetected) electricity Q_S. From the distributor's viewpoint, the constraint (15a) is more desirable because it eases the regulatory constraint.

We first suppose that at the tariff schedule chosen by the distributor, the average revenue constraint (15a) is imposed by the regulator and is binding.[3] Then, the distributor's goal is to choose output level Q and AMI investment level e that solves the following maximization problem

$$\begin{aligned} \hat{\pi} = \max_{Q \geq 0, e \geq 0} & R - C(\beta, e, Q) - \psi(e) \text{ subject to} \\ & \text{(i)} \quad R = \bar{p} Q \\ & \text{(ii)} \quad R \leq R(Q), \end{aligned} \tag{16}$$

where the constraint (i) is the average revenue constraint, and (ii) specifies that R should indeed be attainable at total output Q. Now, (16) can be expressed as:

$$\max_{Q \geq 0, e \geq 0} \bar{p} Q - C(\beta, e, Q) - \psi(e), \quad \text{subject to } \bar{p} Q \leq R(Q).$$

From the concavity of R(Q) (see Lemma 2), we can conclude that there exists a unique $\hat{Q} > 0$ such that the following two conditions hold: first, $R(\hat{Q}) = \bar{p}\hat{Q}$, and second, $R(Q) \geq \bar{p} Q$ if and only if $Q \leq \hat{Q}$. Thus, (16) can be rewritten as

$$\max_{Q \geq 0, e \geq 0} \bar{p} Q - C(\beta, e, Q) - \psi(e), \quad \text{subject to } 0 \leq Q \leq \hat{Q}.$$

This observation leads to the following extension of the result by Armstrong, Cowan, and Vickers [18]:

Theorem 4. *Let $\hat{Q}$ be the unique level of output level satisfying $R(\hat{Q}) = \bar{p}\hat{Q}$, and let $\bar{p} \geq \partial_Q C(\beta, e, Q)$ for $Q \leq \hat{Q}$. Then, if the constraint (15a) binds, the distributor will choose output level $\hat{Q} > Q^*$ and $\hat{e} > e^*$, where Q^* and e^* respectively denote the profit-maximising output and AMI investment level of the unregulated monopolist distributor, and $\partial_e C(\beta, \hat{e}, \hat{Q}) = -\psi'(\hat{e})$.*

Furthermore, the distributor will offer a nonlinear tariff:

$$\hat{p}(\hat{q}_B(\theta)) \equiv (\hat{T})'(q_B) = \frac{\lambda(\hat{Q})}{\left[1 - \frac{1 - F^c(\theta)}{\theta f^c(\theta)}\right]}, \tag{17}$$

[3] Our analysis is similar to [23,18].

where $\hat{\mathrm{T}}(\cdot)$ is the optimal tariff schedule under binding regulatory constraint (15a), $\hat{\mathrm{p}}(\mathrm{q_B}) = \hat{\mathrm{T}}'(\mathrm{q_B})$ is the marginal price, and $\hat{\mathrm{q}}_\mathrm{B}(\theta)$ is the corresponding quantity of billed electricity chosen by type-θ consumer.

Thus, the marginal price schedule corresponding to the optimal tariff under binding regulatory constraint (15a) is of the same from as the corresponding marginal price schedule for unregulated monopoly. However, under binding regulatory constraint (15a), the total output and AMI investment level increases. Recall that $\lambda(\cdot)$ is decreasing in Q and $\partial_\mathrm{Q}\mathrm{C}(\beta, \mathrm{e}^*, \mathrm{Q}^*) = \lambda(\mathrm{Q}^*)$. If C is weakly convex in Q then we obtain

$$\lambda(\hat{\mathrm{Q}}) < \partial_\mathrm{Q}\mathrm{C}(\beta, \hat{\mathrm{e}}, \hat{\mathrm{Q}}) \tag{18}$$

From (17) and (18), we conclude that for type-$\bar{\theta}$ consumer, $\hat{\mathrm{p}}(\hat{\mathrm{q}_\mathrm{B}}(\theta)) < \partial_\mathrm{Q}\mathrm{C}(\beta, \hat{\mathrm{e}}, \hat{\mathrm{Q}})$. Thus, it is optimal for the distributor under average revenue regulation to set the marginal price schedule *below* the marginal cost for sufficiently high-demand consumers (higher θ). The pricing of billed electricity below marginal costs occurs because higher type θ consumers have higher demand.

Next, suppose that at the tariff schedule chosen by the distributor, the average revenue constraint (15b) is binding. The distributor's choices of Q and e solve:

$$\begin{aligned} \tilde{\pi} \max_{\mathrm{Q}\geq 0, \mathrm{e}\geq 0} \; & \mathrm{R} - \mathrm{C}(\beta, \mathrm{e}, \mathrm{Q}) - \psi(\mathrm{e}) \quad \text{subject to} \\ & \mathrm{R} = \bar{\mathrm{p}}[\mathrm{Q} - (1 - \rho(\mathrm{e})\mathrm{Q_U})] \\ & \mathrm{R} \leq \mathrm{R}(\mathrm{Q}). \end{aligned} \tag{19}$$

Rewriting, the problem (19) reduces to

$$\begin{aligned} \max_{\mathrm{Q}\geq 0, \mathrm{e}\geq 0} \; & \bar{\mathrm{p}}[\mathrm{Q} - (1 - \rho(\mathrm{e})\mathrm{Q_U})] - \mathrm{C}(\beta, \mathrm{e}, \mathrm{Q}) - \psi(\mathrm{e}) \quad \text{subject to} \\ & \mathrm{R}(\mathrm{Q}) \geq \bar{\mathrm{p}}[\mathrm{Q} - (1 - \rho(\mathrm{e})\mathrm{Q_U})], \end{aligned}$$

By using the definition of collection efficiency (6), (19) can be expressed as

$$\max_{\mathrm{Q}\geq 0, \mathrm{e}\geq 0} \bar{\mathrm{p}}\eta(\mathrm{Q}, \mathrm{e})\mathrm{Q} - \mathrm{C}(\beta, \mathrm{e}, \mathrm{Q}) - \psi(\mathrm{e}) \quad \text{subject to} \quad \mathrm{R}(\mathrm{Q}) \geq \bar{\mathrm{p}}\eta(\mathrm{Q}, \mathrm{e})\mathrm{Q}.$$

Again, from strict concavity of $\mathrm{R}(\cdot)$, there exists a unique $\tilde{\mathrm{Q}} > 0$ and $\tilde{\mathrm{e}} > 0$ satisfying $\mathrm{R}(\tilde{\mathrm{Q}}) = \bar{\mathrm{p}}\eta(\tilde{\mathrm{Q}}, \tilde{\mathrm{e}})\tilde{\mathrm{Q}}$, and $\mathrm{R}(\mathrm{Q}) \geq \bar{\mathrm{p}}\eta(\mathrm{Q}, \mathrm{e})$ if and only if $\mathrm{Q} \leq \tilde{\mathrm{Q}}$. The following result can be shown:

Claim. There exists a $\bar{\mathrm{p}}$, such that $\hat{\pi} > \tilde{\pi}$ and $\hat{\mathrm{e}} \leq \tilde{\mathrm{e}}$.

In this case, the distributor's preference is to induce the regulator in choosing (15a) as the binding regulatory constraint (since $\hat{\pi} > \tilde{\pi}$). However, this regime also leads to a sub-optimal AMI investment level $\hat{\mathrm{e}}$ relative to the level achieved under when (15b) is binding ($\hat{\mathrm{e}} \leq \tilde{\mathrm{e}}$), i.e., when the regulator is perfectly informed about $\mathrm{Q_U}$.

6 Concluding Remarks

In this paper, we propose to study incentives of a regulated utility to invest in theft reduction via monitoring and enforcement. We have shown that utility under a price cap regulation will underinvest in monitoring customer theft relative to social planner, i.e., a perfectly informed regulator. Thus, in equilibrium, profit maximizing utility operator incurs higher aggregate losses, and has higher equilibrium theft than would be socially optimal. This effect is driven by the regulatory threat of lower price cap, which will be optimal with a higher monitoring level, and thus lower aggregate equilibrium theft.

Our results are consistent with published empirical evidence on electricity distribution losses. Indeed, successful reforms of financially inept state-owned utilities tend to be accompanied by strengthening of monitoring, and dramatic reduction of losses due to non-technical reasons (theft plus billing errors). A combination of technological and institutional means is used in Chile, Brazil, and Argentina; see [1],[5]. Our analysis could be modified to address the theft in transmission system as well. In addition, we argue that deployment of the AMI technology in developed or advanced industrial countries may result in resurgence of non-technical losses. The problem could be especially acute under a bleak economic conditions, when the theft traditionally raises.

We suggest that regulators should include explicit targets for the allowable losses to remedy the problem of incentive misalignment. While institutional and regulatory aspect of reforms are important to improve distribution sector performance, continual adaptation of information technology tools is also essential to maintain operational performance. Without regulatory, institutional, and technological structures in place, the poor operational performance and fiscal discipline will continue to mar the electricity distribution sector.

Acknowledgement. We are grateful to the GameSec'12 TPC Chairs for their help, and to anonymous reviewers for providing useful suggestions. This work was supported by MIT faculty start-up grant and NSF TRUST Science & Technology Center.

References

1. Antmann, P.: Reducing technical and non-technical losses in the power sector. Technical report, World Bank Group Energy Sector (2009)
2. Anderson, R., Fuloria, S.: Who controls the off switch? In: 2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 96–101 (October 2010)
3. Jeff Smith, N.G.: Smart meters take bite out of electricity theft (September 2011), http://news.nationalgeographic.com/news/energy/2011/09/110913-smart-meters-for-electricity-theft/
4. Cleveland, F.M.: Cyber security issues for advanced metering infrastructure (ami). In: 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1–5. IEEE (2008)

5. Smith, T.B.: Electricity theft: a comparative analysis. Energy Policy 32(18), 2067–2076 (2004)
6. McLaughlin, S., Podkuiko, D., McDaniel, P.: Energy Theft in the Advanced Metering Infrastructure. In: Rome, E., Bloomfield, R. (eds.) CRITIS 2009. LNCS, vol. 6027, pp. 176–187. Springer, Heidelberg (2010)
7. Anderson, R., Fuloria, S.: On the security economics of electricity metering. In: The Ninth Workshop on the Economics of Information Security, Citeseer (2010)
8. Kassakian, J.G., Schmalensee, R.: The future of electric grid: An interdisciplinary MIT study. Technical report, Massachusetts Institute of Technology (2011)
9. Sam Dolnick, A.P.: Indian officials wage war on energy theft (July 2009), http://dailyreporter.com/2009/07/06/indian-officials-wage-war-on-energy-theft/
10. PPPIRC: Theft / non-technical losses (water and electricity) (2011), http://ppp.worldbank.org/public-private-partnership/legislation-regulation/laws/theft-nontechnical-loss
11. Victor, D.G., Heller, T.C. (eds.): The Political Economy of Power Sector Reform. Cambridge University Press (2009)
12. Joskow, P.L.: Incentive regulation in theory and practice: Electricity distribution and transmission networks. In: Economic Regulation and Its Reform: What Have We Learned? NBER Chapters. National Bureau of Economic Research, Inc. (Julio Dic 2011)
13. Vogelsang, I.: Electricity transmission pricing and performance-based regulation. CESifo Working Paper Series 1474, CESifo Group Munich (2005)
14. Laffont, J., Martimort, D.: The theory of incentives: the principal-agent model. Princeton Univ. Pr. (2002)
15. Bolton, P., Dewatripont, M.: Contract Theory, vol. 1. The MIT Press (2005)
16. Armstrong, M., Sappington, D.E.: Recent Developments in the Theory of Regulation. Handbook of Industrial Organization, vol. 3, pp. 1557–1700. Elsevier (2007)
17. Laffont, J.J., Tirole, J.: A Theory of Incentives in Procurement and Regulation, 1st edn., vol. 1. The MIT Press (1993)
18. Amrstong, M., Cowan, S., Vickers, J.: Nonlinear pricing and price cap regulation. Journal of Public Economics 58(1), 33–55 (1995)
19. Tirole, J.: A Theory of Industrial Organizaton. The MIT Press (1988)
20. Ho, Y.C., Luh, P., Muralidharan, R.: Information structure, stackelberg games, and incentive controllability. IEEE Transactions on Automatic Control 26(2), 454–460 (1981)
21. Ho, Y.C., Luh, P.B., Olsder, G.J.: A control-theoretic view on incentives. Automatica 18(2), 167–179 (1982)
22. Shen, H., Basar, T.: Optimal nonlinear pricing for a monopolistic network service provider with complete and incomplete information. IEEE Journal on Selected Areas in Communications 25(6), 1216–1223 (2007)
23. Armstrong, M., Vickers, J.: Welfare effects of price discrimination by a regulated monopolist. RAND Journal of Economics 22(4), 571–581 (1991)

Security Games and Risk Minimization for Automatic Generation Control in Smart Grid

Yee Wei Law, Tansu Alpcan, Marimuthu Palaniswami, and Subhrakanti Dey

Department of Electrical & Electronic Engineering, The University of Melbourne, Parkville, VIC 3010, Australia
{ywlaw,tansu.alpcan,palani,sdey}@unimelb.edu.au

Abstract. The power grid, on which most economic activities rely, is a critical infrastructure that must be protected against potential threats. Advanced monitoring technologies at the center of smart grid evolution increase its efficiency but also make it more susceptible to malicious attacks such as false data injection. This paper develops a game-theoretic approach to smart grid security by combining quantitative risk management with decision making on protective measures. Specifically, the consequences of data injection attacks are quantified using a risk assessment process based on simulations. Then, the quantified risks are used as an input to a stochastic game model, where the decisions on defensive measures are made taking into account resource constraints. Security games provide the framework for choosing the best response strategies against attackers in order to minimize potential risks. The theoretical results obtained are demonstrated using numerical examples.

Keywords: Smart grid, automatic generation control, security games.

1 Introduction

A power grid is a critical infrastucture that must be protected against potential threats. As it evolves to a "smart grid" with better efficiency, however, the security concerns increase due to emergence of new attack vectors exploiting increasing system complexity. While security is an important issue for grid operators, real world constraints such as resource limitations necessarily force adoption of a risk management approach to the problem. Protective measures are usually taken based on a cost-benefit analysis balancing available defensive resources with perceived security risks.

This paper investigates the important class of false data injection attacks to smart grids which directly affect the operation of automatic generation control systems and potentially lead to blackouts. The problem is formulated first as one of quantitative risk management which in turn is used as an input to a stochastic (Markov) security game. The resulting game analysis helps smart grid operators to make informed decisions on their security strategies while taking into account their resource constraints. Although the paper focuses on a certain type of attack and subsystem, the approach can be applied to similar security

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 281–295, 2012.

problems in smart grid, and hence, can be extended to develop the foundation of a systematic framework for smart grid security.

A simple but elegant definition of risk is "the probability and magnitude of a loss, disaster, or other undesirable event" [9]. **Security risk analysis** can be defined as "the process of identifying the risks to system security and determining the likelihood of occurrence, the resulting impact, and the additional safeguards that mitigate this impact" [19]. Most smart grid standards and guidelines, e.g. IEC 62351-1, NISTIR 7628, identify risk assessment as a critical part of a security framework. For instance, the Australian Government advocates the use of the Australian and New Zealand Standard for Risk Management (AS/NZS ISO 31000:2009) by owners and operators of critical infrastructure [3]. However, the standard ISO 31000:2009 is "not mathematically based", and has "little to say about probability, data, and models" [13].

Security games. provide an analytical framework for modeling the interaction between malicious attackers, who aim to compromise smart grid, and operators defending them. The "game" is played on smart grids, which are complex and interconnected systems. The rich mathematical basis provided by the field of game theory facilitates formalising the strategic struggle between attackers and defenders for the control of the smart grid [1]. Utilising the risk framework and some of the concepts of earlier studies [5,17], this work applies game theory to the modeling of attacks on and defenses for a critical power system component called **automatic generation control (AGC)**.

The **main contributions** of this work include

- Assessment and identification of risks faced by the automatic generation control system, which constitute an important part of smart grid, due to false data injection attacks.
- A discussion of the security threat model, potential attacks, and countermeasures.
- A stochastic (Markov) security game for analysis of best defensive actions building upon the risk analysis conducted and under resource limitations.
- A numerical study illustrating the framework developed.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 states the problem of assessing the cyber security risks of automatic generation control, an essential power system component. Section 4 presents our game and risk model. In Section 5, we specify an informal threat model; we also discuss attack and defense actions under this threat model. In Section 6, we apply the game and risk model to automatic generation control, and present our simulation results. Section 7 concludes this paper.

2 Related Work

Smart grid cyber security is an emerging area. Substantial research effort is still being dedicated to exploring cyber attacks and their effects on power grids.

Stamp et al. [22] develop a cyber-to-physical modeling approach called *Reliability Impacts from Cyber Attack*, for quantifying the degradation of system reliability for a given probability of cyber attack. Several metrics are investigated, including frequency of interruption, loss of load expectancy, load curtailed per interruption, etc. Kundur et al. [10] present two simulation studies on the effects of attacks against a single-generator system and a 13-bus system by injecting *three* levels of errors into a *single* sensor in the systems. Esfahani et al. [6,7] design elaborate schemes for controlling maliciously injected AGC *output signal* to maximally disrupt a grid. Our focus on AGC is in a way inspired by their work. However, we focus on one of the AGC *input signals* (i.e., frequency deviation), since from an attacker's perspective, compromising a meter potentially costs less than compromising an automatic generation controller.

Risk assessment has been garnering a lot of attention lately. We note that some authors erroneously refer to risk assessment as *vulnerability assessment*, which is a different concept [19]. *Attack trees* or attack graphs is a common starting point for most work in this area. An attack tree represents attacks against a system in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes. Ten et al. [24] propose a framework based on attack trees for evaluating system security. They focus on attacks originating from substations connecting to the control center through a *virtual private network*. They limit cyber intrusions to firewall penetration and password cracking, singling out password policies and port auditing as the two most important security measures – these assumptions are used in other work by the same research team [21,23]. Their framework define three vulnerability indices: the *system vulnerability index* is the maximum of *scenario vulnerability indices*, which are products of *leaf vulnerability indices*, which in turn depend on subjective definitions of port vulnerability and password strength. Liu et al. [14] take an attack tree as input, and assign a "difficulty level" to each action on the tree using Analytic Hierarchy Process. Their methodology produces a *vulnerability factor*, an artificial measure of the success probability of an attack. Analytic Hierarchy Process is a decision making methodology that is often applied to risk management, but for its reliance on subjective scoring and failure to satisfy several statistical axioms (e.g., transitivity), the risk management community is divided regarding its validity [9]. In comparison, only empirical evidence is used in this work.

The limitation of attack trees is not unrecognized. Sommestad et al. [20] propose *defense graphs* as an alternative to attack graphs, to take into account the countermeasures already in place within a system. They model defense graphs using *influence diagrams*, which are essentially Bayesian networks enhanced with indicators that express *beliefs* on *likelihood* values. The output of their assessment methodology is the expected loss associated with a successful attack. Hahn et al. [8] propose *privilege graphs* to model the privilege states in a system and the paths exploitable by an attacker. The essence of their proposal is an algorithm for computing an *exposure metric*, that takes into account (i) the number of attack paths through the security mechanisms protecting a target asset, and (ii) the path length representing the effort required to exploit a path.

Ten et al. [23] model attacks using *stochastic Petri Nets*, which encapsulate the probability and risk of attacks. They define the metric *system vulnerability* which is the maximum of all *scenario vulnerability* values, and the metric *impact factor* w.r.t to a substation disconnected by a successful attack. Sridhar et al. [21] use stochastic Petri Nets to model computers, firewalls and intrusion protection systems. To assess the *steady-state impact* of attacks on the power system itself, they present the impact study of six coordinated attack scenarios, where coordination is in the sense of targeting multiple power system components at the same time. They define risk as the product of the probability of a successful attack and the resultant shed load; we adopted this definition of risk. With the exception of [21], most risk assessment work discussed so far is ICT-centric, and does not consider the impact of cyber attacks on the power system itself. In comparison, our work involves the detailed modeling and simulation of attacks on the AGC system.

3 Automatic Generation Control in Power Grid

The most critical aspect of a power system is stability, and one of the most important parameters to stabilize is frequency. This is because the frequency of a power system rises/falls with decreased/increased loading. Failure to stabilize frequency may lead to damage to equipment (utility's or end users'), harm to human safety, reduction of or interruption to electricity supply. Violation of frequency stability criteria is one of the main reasons for numerous power blackouts [4]. Less tangible secondary impacts, including loss of data or information and damage to reputation, are equally undesirable.

The frequency control system operates at three levels. Primary frequency control takes the form of a turbine governor's *speed regulator*, a proportional controller of gain $1/R$, where R is the *droop characteristic* (drop in speed or frequency when combined machines of an area change from no load to full load). Secondary frequency control is for correcting the steady-state error residue left by the proportional controller, and may take the form of an integral controller; in which case, primary and secondary frequency control form a parallel proportional-integral controller, capable of driving frequency deviations to zero whenever a step-load perturbation is applied to the system. Tertiary frequency control is supervisory control based on offline optimizations for (i) ensuring adequate spinning reserve in the units participating in primary control, (ii) optimal dispatch of units participating in secondary control, (iii) restoration of bandwidth of secondary control in a given cycle. While primary and secondary control respond in seconds and tens of seconds respectively, tertiary control is usually manually activated minutes after secondary control. Our study concerns only the *dynamics* of frequency control, and hence does not consider tertiary control.

In an interconnected system with two or more *control areas*, in addition to frequency, the generation within each area must also be controlled to maintain scheduled power interchanges over *tie lines* (inter-area transmission lines). The control of both frequency and generation is called *load-frequency control*. Within

each area, each generation unit has primary control, while secondary control is centralized. Together, decentralized primary control and centralized secondary control achieve the purpose of load-frequency control. **Automatic generation control (AGC)** is load-frequency control with the additional objective of *economic dispatch* (distributing the required change in generation among units to minimize costs) [11,26]. However, AGC is sometimes referred to as automated (vs manual) load-frequency control [2], or even the entire frequency control system itself [16]. AGC is an indispensable part of the "central nervous system" of a power grid called the **energy management system (EMS)**, and possibly the only automatic closed loop between the IT and power system of a control area [6]; because of this, it is subject to attacks propagated through the IT system. A detailed threat model is given in Section 5.

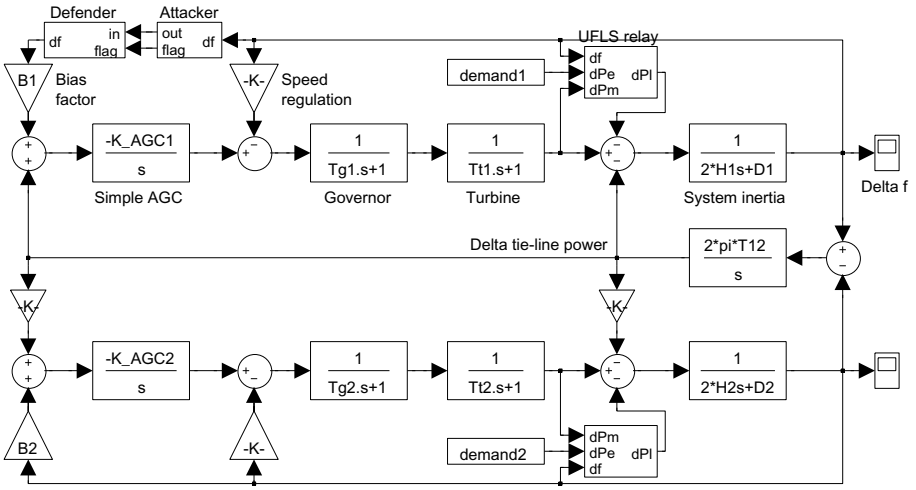
When system frequency deviates from the nominal frequency (60 Hz for Americas, 50 Hz for most other parts of the world) by a certain threshold, overfrequency and underfrequency protection relays execute tripping logic defined by a protection plan that varies from operator to operator. Assuming a nominal frequency of 60 Hz, overfrequency relays start tripping thermal plants when frequency rise exceeds 1.5 Hz [15,16], but these relays are usually set to tolerate deviations due to post-fault transients for short periods of time. Underfrequency relays perform **underfrequency load shedding (UFLS)**, which is the sole concern of our study because it results in directly measurable revenue loss. For our study, we adopt Mullen's UFLS scheme [18]. The gist of the scheme is, when the system frequency drops by more than 0.35 Hz below the nominal frequency, to shed this amount of load:

$$\Delta P_m - \Delta P_e - 0.3/R,$$

where ΔP_m is the change in generator's mechanical power, ΔP_e is the change in generator's electrical power, and R is the droop characteristic. Our aim is to model and quantify the risks posed by an attacker whose intention is to inflict revenue loss on the electricity provider by injecting false data to the automatic generation controller in the hope of triggering load shedding.

For this work, we use the two-area AGC system model and associated simulation parameters in Fig. 1. The automatic generation controller is an integral controller of gain K_{AGC}. We note that AGC design is an established area with designs dating back to the 1950s; a simple integral controller seems to be a logical starting point. The UFLS relay in each area decides on the necessity to shed load, and the amount of load to shed if necessary, using Mullen's algorithm [18]. Once the system frequency has stabilized for at least 30 s, the UFLS relays reconnect the shed loads in the reverse order they were shed.

In this sample configuration, the maximum sheddable loads are capped at 4 p.u. and 1 p.u. for the areas 1 and 2 respectively. "p.u." stands for "per unit" and is simply the ratio of an absolute value in some unit to a base/reference value in the same unit. The base load for both areas is taken to be 1000 MW.



Symbols	Definition	Symbols	Definition
H, D	Inertia constant, damping coefficient	T_g, T_t	Time constants for turbine governor
R	Droop characteristic	B	Frequency bias factor
K_{AGC}	Gain of secondary frequency control	T_{12}	Tie-line synchronizing coefficient

	$K_{\text{AGC}\,i}$ (s)	D_i (p.u./Hz)	$2H_i$ (p.u. s)	R_i (Hz/p.u.)	T_{gi} (s)	T_{ti} (s)	B_i (p.u./Hz)
$i = 1$	0.3	0.015	0.1667	3.00	0.08	0.40	0.3483
$i = 2$	0.2	0.016	0.2017	2.73	0.06	0.44	0.3827

Fig. 1. Simulink representation and simulation parameters for a two-area AGC system model based on Bevrani's [4, Fig. 2.10 and Table 2.2]. The top area is labeled area 1. The demand time series `demand1` and `demand2` are the demand profiles of Victoria on 4-5 June 2012 and of South Australia on 7-8 June 2012 respectively, provided by the Australian Energy Market Operator. Nominal frequency = 60 Hz.

4 Security Game Model

Our security game model is based on Alpcan and Başar's framework [1]. The concept of **risk states** is combined with this model. A system has a set of states, and a different level of risk is associated with each state. In this work, we define risk as *the product of the probability of a successful attack and the resultant shed load (in the unit of power)*. Clearly under this definition, risk ranges from 0 to the maximum sheddable load for all areas combined. As a starting point, we partition this risk space into only two states: s_0 where risk is zero (no load is shed), and s_1 where risk is nonzero (some load is shed). We model the state to evolve probabilistically according to a stochastic process with the Markov property. Accordingly, we model the interactions between an attacker and a defender using stochastic or Markov *security games*.

As a general basis for Markov security games, consider a 2-player (attacker vs. defender) zero-sum Markov game played on a finite state space, where each player has a finite number of actions to choose from. Let the attacker's action space be $\mathcal{A}^A \stackrel{\text{def}}{=} \{a_1, \ldots, a_{N_A}\}$, the defender's action space be $\mathcal{A}^D \stackrel{\text{def}}{=} \{d_1, \ldots, d_{N_D}\}$,

and the state space be $\mathcal{S} \stackrel{\text{def}}{=} \{s_1, \ldots, s_{N_S}\}$. It is assumed that the state evolves according to a discrete-time finite-state Markov chain which enables utilization of well-established analytical tools to study the problem. Then, the *state transitions* are determined by the mapping $\mathcal{M} : \mathcal{S} \times \mathcal{A}^A \times \mathcal{A}^D \to \mathcal{S}$. Let $\boldsymbol{p}^{\mathcal{S}}(t)$ be the probability distribution on the state space $\mathcal{S}$, i.e.,

$$\boldsymbol{p}^{\mathcal{S}}(t) \stackrel{\text{def}}{=} \left[\Pr[s(t) = s_1] \Pr[s(t) = s_2] \cdots \Pr[s(t) = s_{N_S}]\right]^T,$$

where $t \geq 1$ denotes the discrete time (stage) of the repeated Markov game. The mapping $\mathcal{M}$ can then be represented by the $N_S \times N_S$ state transition matrix $\boldsymbol{M}(a, d) = [M_{s_i,s_j}(a, d)]_{N_S \times N_S}$, which is parameterized by $a \in \mathcal{A}^A$ and $d \in \mathcal{A}^D$, such that

$$\boldsymbol{p}^S(t+1) = \boldsymbol{M}(a, d)\boldsymbol{p}^S(t). \tag{1}$$

The matrix entry $M_{s_i,s_j}(a, d)$ represents the probability of state s_i transitioning to state s_j under attacker action a and defender action d.

The mapping $\mathcal{M}$ can alternatively be parameterized by the state to obtain as many zero-sum game matrices $\boldsymbol{G}(s)$ as the number of states, each of dimension $N_A \times N_D$. In other words, given a state $s(t) \in \mathcal{S}$ at a stage t, the players play the zero-sum game $\boldsymbol{G}(s(t)) = [G_{a,d}(s(t))]_{N_A \times N_D}$. The matrix entry $G_{a,d}(s)$ represents the attacker's gain from risk state s by taking action a when the defender action is d. As a simplifying assumption, actions have no cost other than their "contribution" to load shedding, so $G_{a,d}(s)$ is the *expected total load shed* in state s under attacker action a and defender action d. In particular, $\boldsymbol{G}(s_0) = \boldsymbol{0}$. Due to the adopted zero-sum Markov game formulation, the attacker's gain (loss) equals the defender's loss (gain).

The attacker's strategy is defined as a probability distribution on $\mathcal{A}^A$ for a give state s, i.e., $p^A(s) \stackrel{\text{def}}{=} \left[\Pr[a(s) = a_1] \cdots \Pr[a(s) = a_{N_A}]\right]^T$. The defender's strategy is similarly defined. For the zero-sum Markov game formulation here, the defender aims to minimize its own expected total cost, $\bar{Q}$, in response to the attacker who tries to maximize it. The reverse is true for the attacker due to the zero-sum nature of the game. Hence, it is sufficient to describe the solution algorithm for only one player, which is the defender in our case.

The game is played in stages over an infinite time horizon. The defender's $\bar{Q}$ at the end of a game is the sum of all realized stage costs discounted by a scalar *discount factor* $\gamma \in [0, 1)$:

$$\bar{Q} \stackrel{\text{def}}{=} \sum_{t=1}^{\infty} \gamma^t G_{a(t),d(t)}(s(t)), \qquad a(t) \in \mathcal{A}^A, d(t) \in \mathcal{A}^D, s(t) \in \mathcal{S}, \tag{2}$$

where $G_{a(t),d(t)}(s(t))$ is the $(a(t), d(t))$-th element of the stage-t game matrix $\boldsymbol{G}(s(t))$. The discount factor γ is a logical construct for de-emphasizing the payoff of future stages (smaller $\gamma \implies$ smaller future payoffs). The defender can theoretically choose a different strategy $p^D(s(t))$ at each stage t of the game to minimize the final realized cost $\bar{Q}$ in (2). Fortunately, this complex problem can be simplified significantly. First, it can be shown that a stationary strategy

$p^D(s) = p^D(s(t)), \forall t$ is optimal, and hence there is no need to compute a separate optimal strategy for each stage. Second, the problem can be solved recursively using *dynamic programming* to obtain the stationary optimal strategy (solving a zero-sum matrix game at each stage). The optimal strategy can be mixed, i.e., stochastic for each state s. At a given stage t, the optimal cost $Q_t(a, d, s)$ (the dependency of s, a and d on t is omitted for notational brevity) can be computed iteratively using the dynamic programming recursion

$$Q_{t+1}(a, d, s) = G_{a,d}(s) + \gamma \sum_{s' \in \mathcal{S}} M_{s,s'}(a, d) \cdot \min_{p^D(s')} \max_{a} \sum_{d \in \mathcal{A}^D} Q_t(a, d, s') p_d^D(s'), \tag{3}$$

for $t = 0, 1, \ldots$ and a given initial condition Q_0. In (3), $p_d^D(s')$ is the element of $p^D(s')$ that corresponds to d. (3) converges to the optimal Q^* as $t \to \infty$.

There are multiple ways to implement (3). The algorithm called *value iteration* is prescribed here due to its scalability. To describe the algorithm, we first split (3) into two parts:

$$V(s) = \min_{p^D(s)} \max_{a} \sum_{d \in \mathcal{A}^D} Q_t(a, d, s) p_d^D(s), \tag{4}$$

$$Q_{t+1}(a, d, s) = G_{a,d}(s) + \gamma \sum_{s' \in \mathcal{S}} M_{s,s'}(a, d) V(s'), \quad t = 1, 2, \ldots \tag{5}$$

We can formulate (4) as a linear program:

$$\begin{aligned} &\min_{p^D(s)} V(s) \\ &\text{s.t.} V(s) \geq \sum_{d \in \mathcal{A}^D} Q_t(a, d, s) p_d^D(s), \forall a \in \mathcal{A}^A, \\ &p_d^D \geq 0, \sum_d p_d^D = 1, \forall d \in \mathcal{A}^D. \end{aligned} \tag{6}$$

The strategy $p^D(s), \forall s \in \mathcal{S}$ computed from (6) is the *minimax* strategy w.r.t. Q. The fixed points of equations (4) and (5), V^* and Q^*, lead to the optimal minimax solution for the defender. The value iteration algorithm, using (6) and (5) to find V^* and Q^*, is given in Algorithm 1.

Algorithm 1. The value iteration algorithm

Given arbitrary Q_0 and V
repeat
  for $a \in \mathcal{A}^A$ and $d \in \mathcal{A}^D$ **do**
    Update V and Q according to (6) and (5)
  end for
until $V(s) \to V^*$, i.e., $V(s)$ converges

5 Threat Analysis

Fig. 2 shows the communication architecture involving a control center and a substation based on the international standard IEC 61850 [23,26]. Access to the control system in either the control center or the substation is enabled through

a virtual private network (VPN). Some authors [24] equate the compromise of an entire control center or substation to the successful cracking of a VPN access password and the penetration of an Internet-facing firewall (see Fig. 2). This strong attacker model is not entirely unrealistic, however, our goal is to investigate the strategy of an attacker that has successfully penetrated the protected network but whose actions within the AGC system are bounded by several resource constraints. We assume the following resource constraints:

- The attacker cannot directly trip generators, or transmission lines (by opening circuit breakers).
- The attacker cannot tamper with turbine governors.
- The attacker cannot tamper with underfrequency load shedding (UFLS) relays. Some commercial relays (e.g., SEL-387E) have an integrated frequency meter, and are thereby not subject to false frequency data injection attacks.
- The attacker cannot tamper with the EMS.
- The attacker can reduce but not block the input/output of the EMS.

Without the above constraints, it is a trivial exercise for any attacker that has successfully penetrated the protected network to trigger cascading failures across the power grid. It is therefore conceivable that an energy provider would make protecting its generators, circuit breakers, turbine governors, UFLS relays, and EMS its foremost priority. Despite the above constraints, an attacker can forge and send false *frequency deviation* (Δf) data to the AGC software executing on one of the EMS servers, by compromising one of the meters in the substation (see Fig. 2). In the spirit of stealthy attacks as embodied by Stuxnet, Duqu and Flame, it is also conceivable that a persistent attacker would adopt this subtle and stealthy strategy. Then, it is up to the AGC software to detect such attacks.

Basic attacks: It is impossible to exhaust all injection patterns, but there are four basic patterns on which more sophisticated attacks are based:

- **Constant injection**: If an attacker injects a constant false Δf, then the it effectively disables the integral control loop, causing the system frequency to converge to a non-nominal frequency. If the false Δf is positive, then the

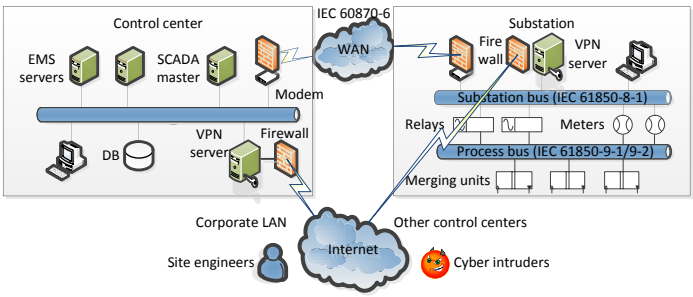


Fig. 2. Accessibility of a power system control center and substation from the Internet. AGC is executed on one of the EMS servers. In our threat model, an attacker can feed the AGC software with false frequency deviation data.

system will settle on a below-nominal frequency, causing loads to be shed; otherwise, the system will settle on an above-nominal frequency, causing generators to be tripped. Both cases lead to cascading failures.

- **Bias injection**: When the false Δf is a constant bias (displacement) from the true Δf, the effect is similar to that of constant injection because normally the true $\Delta f \approx 0$.
- **Overcompensation**: If the false Δf is k times the true Δf, where k is a large positive number, then the attack effectively causes overcompensation by the integral control loop, and consequently unstable oscillations. As the system frequency sweeps past the overfrequency and underfrequency thresholds, generators will be tripped and loads will be shed, followed by cascading failures. Fig. 3 shows the result of an attack using $k = 8$.
- **Negative compensation**: If the false Δf is $-k$ times the true Δf, where k is a positive number, then the attack effectively reverses the intended effect of the integral control loop, causing the system frequency to diverge from the nominal frequency (see Fig. 3). This attack directly triggers generator tripping, but not load shedding.

For our study, we concentrate only on the overcompensation attack, as it inflicts maximum damage in terms of triggering both load shedding and generator tripping.

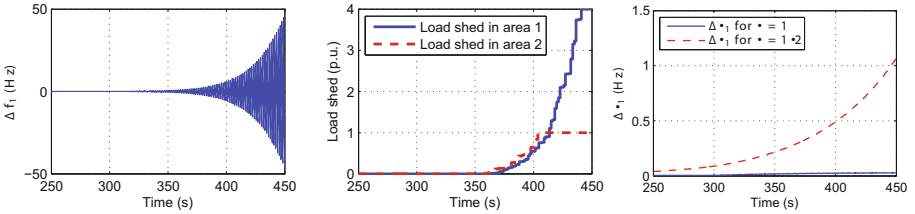


Fig. 3. (Left and Middle) An example of "overcompensation" attack, where the attacker substitutes Δf_1 with $8\Delta f_1$ as frequency input to the area-1 integral controller. As long as the attack persists, neither generator tripping nor load shedding helps stabilize the system. (Right) Negative compensation attack: for large enough k (e.g., 1.2), the system frequency $\to +\infty$.

Basic defenses which are applicable to the overcompensation attack include:

- **Saturation filter**: We can constrain the attack by limiting the Δf input to the integral controller to $[-4.5, 3.5]$ Hz (i.e., passing the input through a saturation filter), because at $\Delta f = -4.5$ Hz, not only should all sheddable loads have been shed, but also all generators would be tripped. At $\Delta f = 3.5$ Hz, all generators would be tripped as well [15].
- **Redundancy**: Measurement redundancy is routinely provisioned for critical grid parameters [12]. Multiple frequency meters of different grades can be installed, so that the likelihood of all meters being compromised is small and the AGC software has a non-zero chance of getting genuine frequency data.

– **Detection**: Saturation filtering and redundancy only limit the effect of an attack, stopping an attack requires the attack to be detected and the source be removed. A threshold-based algorithm can be designed to observe the quantity $\sum_t |\Delta f(t)|$; if the quantity is larger than a certain threshold, the system could be under attack. Alternatively, a clustering-based algorithm can be designed to count the number of clusters in the time series $\{\Delta f(t)\}$; if more than one cluster is observed, the system could be under attack.

There are unlimited ways to improve upon the overcompensation attack to counter the above defenses. Correspondingly, there are unlimited ways to detect these improved attacks with varying accuracy, and certainly there are more advanced controllers that are less susceptible to these attacks. Nevertheless, our interest is not on the design of attacks, defenses or the controller, but on the modeling of system risk dynamics under the actions of the attacker and defender for any given system.

6 Simulation Study

An AGC system under the interactions of an attacker and a defender is simulated in order to observe the state transition matrix $\boldsymbol{M}(a,d) = [M_{s_i,s_j}(a,d)]_{N_S \times N_S}$, and the game matrix $\boldsymbol{G}(s) = [G_{a,d}(s(t))]_{N_A \times N_D}$. $M_{s_i,s_j}(a,d)$ is readily obtained by fixing attacker action at a, defender action at d, and measuring the probability of a session starting in state s_i ends in state s_j. Based on our assumption that actions have no cost other than their "contribution" to load shedding, $\boldsymbol{G}(s_0) = \mathbf{0}$; $\boldsymbol{G}(s_1)$ is the expected total load shed in state s_1. To obtain $G_{a,d}(s_1)$, we fix attacker action at a, defender action at d, and measure the total load shed during the combined duration of s_1. Suppose the total energy shed is E_{s_1} and the combined duration of s_1 is T_{s_1}, then $G_{a,d}(s_1) = E_{s_1}/T_{s_1}$.

For numerical simplicity, we define only two attacker actions and two defender actions, although our approach can be applied to any finite number of attacker and defender actions. The chosen attacker actions are:

a_1 Send N samples, $N/2$ of which are false.
a_2 Send N samples, N of which are false.

a_1 and a_2 are two special cases of the general attacker action space $\mathcal{A}^A = \{$ Send N samples, i of which are false $(i = 1, \ldots, N)\}$. The attacker sets a false Δf to -4.5 Hz if the true Δf is negative, or 3.5 Hz if the true Δf is positive. This implements the overcompensation attack, and takes into account the saturation filter in Section 5.

The defender implements the saturation filter and redundancy measure described in Section 5. For redundancy, the defender reads N consecutive samples alternately from two frequency meters of different builds (one is more secure than the other). N consecutive samples from one meter constitute one *session/stage* (see Fig. 4(a)). Upon collecting N samples, the defender performs one of the following defender actions:

d_1 Run Detection Algorithm 1, a hypothetical algorithm with an attack detection probability of $1 - \alpha_1^{(x/N)^{\beta_1}}$, where x is the number of malicious samples among N samples, α_1 and β_1 are constants. Detection Algorithm 1 emulates a clustering-based anomaly detection algorithm.

d_2 Run Detection Algorithm 2, a hypothetical algorithm with an attack detection probability of $1/[1 + e^{-\alpha_2(x/N-\beta_2)}]$, where x is the number of malicious samples among N samples, α_2 and β_2 are constants. Detection Algorithm 2 emulates a threshold-based algorithm.

We assume that the defender can run only one detection algorithm at the end of each session due to time constraint. If the detection result is positive, the defender disinfects the meter (e.g., by refreshing its firmware, cryptographic keys and so on). Disinfection is assumed to complete within the time frame of one session (see Fig. 4(a)).

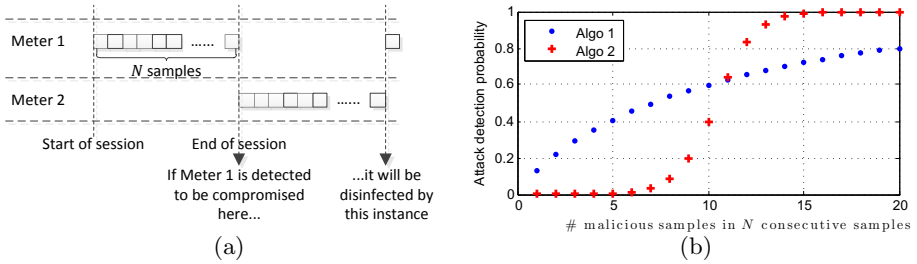


Fig. 4. (a) A session/stage in our security game. (b) Attack detection probabilities of Detection Algorithms 1 and 2 if $N = 20$, $\alpha_1 = 0.2$, $\beta_1 = 0.8127$, $\alpha_2 = 20$, $\beta_2 = 0.5203$.

For simulations, we use the two-area AGC system model and associated simulation parameters in Fig. 1. Since AGC signals are transmitted to the generating plant once every 2 to 4 seconds [11], we set the sampling rate of the "Defender" and "Attacker" blocks to 2 seconds. Attacks are simulated to start at time 100 s. We set $N = 20$, i.e., 20 samples are read from a meter in each session. The parameters of the detection algorithms are set according to the parameters in Fig. 4(b), such that Detection Algorithm 1 is good for low concentration of malicious samples, while Detection Algorithm 2 is good for high concentration of malicious samples. After a meter is detected to be compromised and disinfected, it will become compromised again after some time; Meter 1 and Meter 2 take 4 sessions and 20 sessions to compromise respectively. Using MATLAB/Simulink, each simulation is conducted for 30 virtual minutes. The obtained $\boldsymbol{M}$ and $\boldsymbol{G}$ are fed into Algorithm 2. Fig. 5 shows the simulation results and the following observations.

Effect of Sampling Rate: Since AGC signals are usually transmitted to the generating plant once every 2 to 4 seconds [11], we initially set the AGC sampling rate to 0.5 Hz. A lower sampling rate means a malicious sample will have longer effect on the controller. When we increase the AGC sampling rate to 1 Hz,

the amount of load shed drops conspicuously as evidenced by the lower-valued $\boldsymbol{G}(s_1)$ (less gain for the attacker). Thus, besides improving control precision, a sufficiently high sampling rate provides a good buffer against attacks. Fig. 5(f, g, h) indicates that except for low discount factors, increasing the sampling rate (diminishing the attacker's gain) tend to drive both attacker and defender to adopt a mixed strategy.

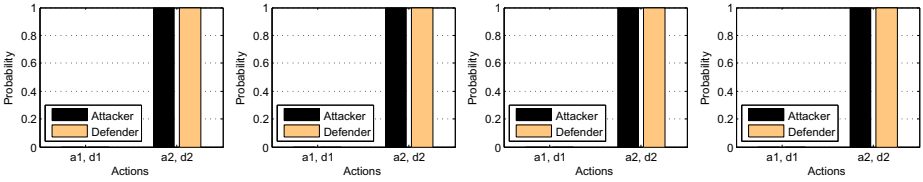
Effect of the Discount Factor: Fig. 5(f, g, h) shows that at a higher sampling rate, where the attacker's gain is lower, defender action d_1 increases in effectiveness as the discount factor increases (future payoffs get more emphasized). In the limit, a pure defense strategy using only d_2 should suffice.

AGC sampling rate: 0.5 Hz

$$\boldsymbol{M}(a_1,d_1)=\begin{bmatrix}7/11 & 4/11\\ 4/31 & 27/31\end{bmatrix} \qquad \boldsymbol{M}(a_1,d_2)=\begin{bmatrix}9/14 & 5/14\\ 4/28 & 24/28\end{bmatrix}$$

$$\boldsymbol{M}(a_2,d_1)=\begin{bmatrix}8/12 & 4/12\\ 3/30 & 27/30\end{bmatrix} \qquad \boldsymbol{M}(a_2,d_2)=\begin{bmatrix}7/10 & 3/10\\ 3/32 & 29/32\end{bmatrix}$$

$$\boldsymbol{G}(s_0)=\mathbf{0} \qquad \boldsymbol{G}(s_1)=\begin{bmatrix}0.5038 & 0.5884\\ 0.6643 & 0.6450\end{bmatrix}$$



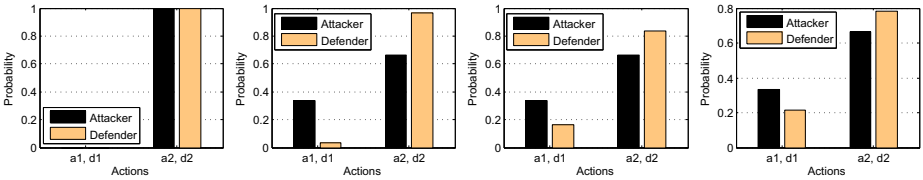
(a) 0.5 Hz, $\gamma = 0.1$ (b) 0.5 Hz, $\gamma = 0.3$ (c) 0.5 Hz, $\gamma = 0.7$ (d) 0.5 Hz, $\gamma = 0.9$

AGC sampling rate: 1 Hz

$$\boldsymbol{M}(a_1,d_1)=\begin{bmatrix}13/21 & 8/21\\ 7/64 & 57/64\end{bmatrix} \qquad \boldsymbol{M}(a_1,d_2)=\begin{bmatrix}3/8 & 5/8\\ 4/77 & 73/77\end{bmatrix}$$

$$\boldsymbol{M}(a_2,d_1)=\begin{bmatrix}3/9 & 6/9\\ 6/76 & 70/76\end{bmatrix} \qquad \boldsymbol{M}(a_2,d_2)=\begin{bmatrix}5/11 & 6/11\\ 8/74 & 66/74\end{bmatrix}$$

$$\boldsymbol{G}(s_0)=\mathbf{0} \qquad \boldsymbol{G}(s_1)=\begin{bmatrix}0.3046 & 0.3473\\ 0.3719 & 0.3505\end{bmatrix}$$



(e) 1 Hz, $\gamma = 0.1$ (f) 1 Hz, $\gamma = 0.3$ (g) 1 Hz, $\gamma = 0.7$ (h) 1 Hz, $\gamma = 0.9$

Fig. 5. Attack and defense strategies organized according to AGC sampling rate and discount factor γ

7 Conclusion

Risk assessment for power grids has been identified as a critical area by the public sector, industry and academia. However, existing risk management standards such as ISO 31000:2009 are more about general principles and guidelines than concrete mathematical techniques. In this work, we identify and assess the risks faced by a critical power system component called automatic generation control (AGC). Our discussion of potential attacks and countermeasures is based on an explicit security threat model. We propose a quantitative risk model capturing the probability and magnitude of security threats faced by the AGC system due to false data injection attacks. Building upon the risk analysis, we model attacker-defender interactions using stochastic (Markov) security games to analyze the best defensive actions under resource constraints. The developed framework is illustrated with a detailed AGC model and simulation results.

For our preliminary study, we have adopted a risk-neutral framework, such that the expected loss from a blackout tends to conceal the significance of rare events at the tail-end of a probability distribution. Financial risk measures such as value-at-risk and conditional value-at-risk have been proposed to account for these rare events [25], and are being explored in ongoing work. In addition to attacks on the frequency input to AGC, we will consider attacks on the tie-line power input, and the AGC output. We will also use more precise models for AGC, turbine governor, generator and underfrequency load shedding. In this work, generators are *as per convention* simulated as a lumped "System inertia" block, but fine-grained simulations of the electrical circuits in each control area, including the effects of generator tripping triggered by overfrequency protection and islanding, are desirable.

References

1. Alpcan, T., Başar, T.: Network Security: A Decision and Game Theoretic Approach. Cambridge University Press (2011)
2. Andersson, G.: Dynamics and control of electric power systems. Lecture notes 227-0528-00, ETH Zürich (February 2010)
3. Australian Government: Critical infrastructure resilience strategy (2010), http://www.tisn.gov.au/
4. Bevrani, H.: Robust Power System Frequency Control. Power Electronics and Power Systems. Springer Science+Business Media LLC (2009)
5. Bommannavar, P., Alpcan, T., Bambos, N.: Security risk management via dynamic games with learning. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–6 (June 2011)
6. Esfahani, P.M., Vrakopoulou, M., Margellos, K., Lygeros, J., Andersson, G.: A Robust Policy for Automatic Generation Control Cyber Attack in Two Area Power Network. In: IEEE Conference on Decision and Control (December 2010)
7. Esfahani, P.M., Vrakopoulou, M., Margellos, K., Lygeros, J., Andersson, G.: Cyber Attack in a Two-Area Power System: Impact Identification using Reachability. In: American Control Conference, Baltimore, MD, USA (June 2010)

8. Hahn, A., Govindarasu, M.: Cyber attack exposure evaluation framework for the smart grid. IEEE Transactions on Smart Grid 2(4), 835–843 (2011)
9. Hubbard, D.W.: The Failure of Risk Management: Why It's Broken and How to Fix It. Wiley (2009)
10. Kundur, D., Feng, X., Mashayekh, S., Liu, S., Zourntos, T., Butler-Purry, K.L.: Towards modelling the impact of cyber attacks on a smart grid. International Journal of Security and Networks 6(1/2011), 2–13 (2011)
11. Kundur, P.: Power System Stability and Control. McGraw-Hill Professional (1994)
12. Lefebvre, D., Bernard, S., Cutsem, T.V.: Undervoltage load shedding scheme for the Hydro-Québec system. In: IEEE Power Engineering Society General Meeting, vol. 2, pp. 1619–1624 (June 2004)
13. Leitch, M.: ISO 31000:2009—The New International Standard on Risk Management. Risk Analysis 30(6), 887–892 (2010)
14. Liu, N., Zhang, J., Zhang, H., Liu, W.: Security Assessment for Communication Networks of Power Control Systems Using Attack Graph and MCDM. IEEE Transactions on Power Delivery 25(3), 1492–1500 (2010)
15. Luo, C., Far, H., Banakar, H., Keung, P.K., Ooi, B.T.: Estimation of wind penetration as limited by frequency deviation. IEEE Transactions on Energy Conversion 22(3), 783–791 (2007)
16. Machowski, J., Bialek, J.W., Bumby, J.R.: Power System Dynamics: Stability and Control, 2nd edn. John Wiley and Sons, Ltd (2008)
17. Mounzer, J., Alpcan, T., Bambos, N.: Dynamic Control and Mitigation of Interdependent IT Security Risks. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–6 (May 2010)
18. Mullen, S.K.: Plug-In Hybrid Electric Vehicles as a Source of Distributed Frequency Regulation. Ph.D. thesis, University of Minnesota (2009)
19. NIST: Glossary of key information security terms. IR 7298 Revision 1 (February 2011)
20. Sommestad, T., Ekstedt, M., Nordstrom, L.: Modeling security of power communication systems using defense graphs and influence diagrams. IEEE Transactions on Power Delivery 24(4), 1801–1808 (2009)
21. Sridhar, S., Govindarasu, M., Liu, C.-C.: Risk analysis of coordinated cyber attacks on power grid. In: Control and Optimization Methods for Electric Smart Grids. Power Electronics and Power Systems, vol. 3, pp. 275–294. Springer, US (2012)
22. Stamp, J., McIntyre, A., Ricardson, B.: Reliability impacts from cyber attack on electric power systems. In: IEEE/PES Power Systems Conference and Exposition (PSCE 2009), pp. 1–8 (March 2009)
23. Ten, C.W., Liu, C.C., Manimaran, G.: Vulnerability Assessment of Cybersecurity for SCADA Systems. IEEE Trans. Power Syst. 23(4), 1836–1846 (2008)
24. Ten, C.W., Manimaran, G., Liu, C.C.: Cybersecurity for critical infrastructures: Attack and defense modeling. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 40(4), 853–865 (2010)
25. Varaiya, P., Wu, F., Bialek, J.: Smart operation of smart grid: Risk-limiting dispatch. Proceedings of the IEEE 99(1), 40–57 (2011)
26. Wu, F., Moslehi, K., Bose, A.: Power system control centers: Past, present, and future. Proceedings of the IEEE 93(11), 1890–1908 (2005)

Contractual Agreement Design for Enforcing Honesty in Cloud Outsourcing

Robert Nix and Murat Kantarcioglu

The University of Texas at Dallas,
500 W Campbell Rd,
Richardson, TX 75080
{rcn062000,muratk}@utdallas.edu

Abstract. To save time and money, businesses and individuals have begun outsourcing their data and computations to cloud computing services. These entities would, however, like to ensure that the queries they request from the cloud services are being computed correctly. In this paper, we use the principles of economics and competition to vastly reduce the complexity of query verification on outsourced data. Instead of building a specialized computation system for verifying the result of a single outsourced query, we rely on a second, non-colluding data outsourcing entity, whose services are required only a miniscule fraction of the time. Using a game theoretic model, we show that given the proper incentive structure, we can effectively deter dishonest behavior on the part of the data outsourcing services with a very small expected cost increase. We then prove that the incentive for an outsourcing service to cheat can be reduced to zero under this structure.

Keywords: game theory, data outsourcing, contracts, query verification.

1 Introduction

As the amount of data that we generate increases, so does the time and effort necessary to process and store the data. With an increase in time and effort comes an increase in monetary cost. To this end, many have turned to outsourcing their data processing to "the cloud." Cloud computing services are offered by many large companies, such as Amazon, IBM, Microsoft, and Google, as well as smaller companies such as Joyent and CSC. For example, Google [5] recently launched the Google BigQuery Service, which is designed for exactly this purpose: outsourced data processing. The distributed nature of these cloud services shortens data processing time significantly. In addition, these cloud services provide a massive amount of data storage.

In a perfect world, these cloud providers would impartially devote all the computation necessary to any task paid for by the subscribers. In such a world, the querying process would look like figure 1 (minus the verifier), where the subscriber outsources the data D to the cloud, sends queries (Q), and the cloud

J. Grossklags and J. Walrand (Eds.): GameSec 2012, LNCS 7638, pp. 296–308, 2012.

does the necessary calculations and returns the result ($Q(D)$). However, a cloud provider is a self-interested entity. Since it is very difficult for the users of the cloud to see the inner workings of the cloud service, a cloud provider could "cut corners," delivering a less accurate or incomplete computation result which would take fewer system resources to compute. This would, of course, save computational resources for the provider, provided the subscriber was unable to tell a false result from a true one. Because of this, query verification, or the assurance of query result correctness, has been identified as one of the major problems in data outsourcing [17].

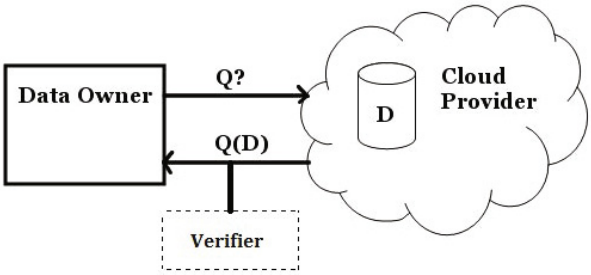


Fig. 1. Data Outsourcing with Verification

Many techniques have been developed and employed for query verification. In figure 1 above, the subscriber sends a query to the outsourcing service, and receives a response. Query verification would then be another process where the subscriber determines if the response is, in fact, the result of the query. The verification process may belong to the owner, or it may be another process entirely. In any case, the verifier aims to make sure that the outsourced server responded correctly. These verification techniques range from simple to extremely complex, and generally rely on the subscriber storing some sketch of the data (much smaller in size), or some cryptographic protocols. Such protocols do a good job verifying the data, but are often slow, or only work with specific types of queries. Many of them assume that the subscriber knows which queries he will execute in advance, so that a sketch can be created for each one. None of these, however, consider the heart of the problem: the self-interest of the parties.

The problem of data outsourcing, and the resultant query verification, is fundamentally a problem of *incentives*. A cloud subscriber wants to get the result of his queries accurately and efficiently, with as low a cost as possible. A cloud provider, however, is most concerned about the profitable use of its computing resources. These incentives can be at odds with each other. The natural way of analyzing competing incentives is to use *game theory*. An interaction between parties is cast as a game, where players use strategy to maximize their gains. The gains from an interaction can be offset artificially by contracts, which can be enforced by law. These adjustments can make actions which were once profitable, such as "cutting corners" in a calculation, less profitable through the use

of penalties. The contracts, therefore, aim not to detect whether a cloud provider is cheating, but to remove the incentive for the provider to cheat altogether.

We propose a game theory-based approach to query verification on outsourced data. We model the process of querying outsourced data as a game, with contracts used to enforce behavior. Data outsourcing does not take place in a vacuum. Service Level Agreements (SLAs) exist for all types of cloud services[12], and are enforceable contracts in court. Thus, we can augment the SLA with an incentive structure to encourage honest behavior. Using a very simple query verification technique, we show that even the threat of verification is enough to deter cheating by a cloud provider.

We consider the case where multiple, non-colluding cloud providers exist. Non-colluding means that the cloud providers do not share information. We believe this is realistic, since cloud providers are competing entities and do not wish to share data with their competitors. In this scenario, we show that without the use of special verification techniques, a data owner can guarantee correct results from rational cloud providers, while incurring an additional cost that is only a small fraction of the overall computation cost.

Our contributions can be summarized as follows:

- We develop a game theoretic model of query verification on outsourced data.
- We show that the model has an equilibrium where the cloud provider behaves honestly.
- Finally, we show that our incentives can improve the expected runtime of *any* query verification method, making it extremely flexible.

Our paper does not consider the privacy of the outsourced data (similar to [2]). However, any privacy-preserving technique for outsourcing data could still be used in our framework. The use of our game theoretic techniques will not affect the privacy-preserving properties of such schemes.

2 Related Work

Several works have outlined query verification methods. The vast majority of these works focus on specific types of queries. Some focus only on selection [1,3,8,18,20], while others focus on relational queries such as selection, projection, and joins [11,10]. Still others focus only on aggregation queries like sum, count, and average [6,19,21]. Some of these processes [16,21] require different verification schemes for each type of query, or even each individual query, requiring that the subscriber knows which queries will be asked in advance.

Many of the aforementioned schemes require complex cryptographic protocols. Some encrypt the data itself, relying on homomorphic schemes to allow the cloud provider to perform the computation [4,19]. A homomorphic operation will always be less efficient than the operation on the unencrypted data, rendering the overhead of these protocols greater by orders of magnitude. Others, such as [16], rely on relatively simpler cryptographic primitives, like secure hash functions. To maintain integrity, our scheme will also use hash functions. Our verification

framework is, however, simpler than these cryptography-based protocols, and can be used to improve the expected runtime of any of these verification schemes.

The work of Canetti, Riva, and Rothblum [2] also makes use of multiple outsourcing services for query verification. However, they make use of all the services all the time, and require a logarithmic number of rounds to ensure verifiability of computation. In addition, they assume that at least one of the cloud providers is in fact honest. We, in contrast, do not assume that any provider is honest, merely that they are *rational* (meaning that the provider wishes to maximize his profits), and we only use additional providers a fraction of the time. In addition, we only require one round of computation.

3 Cryptographic Background

In order to maintain the integrity of our outsourced data, we will need to employ some basic cryptographic primitives. We will need to employ a scheme that allows the owner to make sure that tuples he receives from the server are legitimate, and were not added or altered by the server. We can use a simple message authentication code protocol known as HMAC [13] (Hash-based Message Authentication Code) to do this. HMAC requires the use of *cryptographic hash functions*.

A *cryptographic hash function* or *one-way hash function* is a function mapping a large, potentially infinite, domain to a finite range. This function is simple to compute (taking polynomial time), but is difficult to invert. Equivalently, we can say that, for a cryptographic hash function f, it is difficult to find an x and y such that $x \neq y$ and $f(x) = f(y)$. Examples of cryptographic hash functions include MD5 [15], SHA-1, and SHA-256 [9].

The HMAC system creates a *keyed* hash function from an existing cryptographic hash function. Let m be the message for which we wish to create a code, and k be the key we wish to use. Let f be our cryptographic hash function, and let its required input size be n. If k has a length smaller than n, we pad k with zeroes until it has size n. If k is larger, we let k be $f(k)$ for the purposes of calculating the HMAC function. We define the HMAC function as follows:

$$HMAC(m, k) = f((k \oplus outpad)||f(k \oplus inpad)||m)$$

where outpad and inpad are two constants which are the length of f's block size (in practice, 0x5c...5c and 0x36...36, respectively).

Given a message m and its HMAC value h, if we have the key k, we can simply check to see if HMAC(m,k) matches h. If it does, then the probability that the message is not legitimate (i.e., fabricated or altered) is negligible. Someone who does not have the key k, however, will be unable to compute HMAC(m,k), and will therefore be unable to forge a correct message.

Some more sophisticated methods of verifying data exist, such as Merkle hash trees[7], which allow larger and smaller granularities of the message to be authenticated without authenticating the rest. These other methods of verification could be used to ensure data integrity if desired. In practice, any method of

ensuring data integrity once it is in the hands of the outsourced servers will suffice. We will use the simple HMAC protocol to do this. Data integrity will be a critical component of our second solution.

4 The First Solution

We consider the case where multiple non-colluding cloud providers exist. This means that the parties do not exchange strategies and do not exchange information. Since multiple providers exist, our strategy will be to choose two of them, checking the results of one against the other. We model the query verification process as a game. The game has the following characteristics:

Players (3): the Data Owner(O), and two outsourced servers (S_1 and S_2).

Actions: The data owner begins the game by selecting a probability α, and declares this probability to the servers. He then sends the query (Q) to one of the two servers, with equal probability. With probability α he also sends the query to the other server. If server S_i receives the query, they then respond to the query with either $Q(D)$, that is, the query result on the database D, or $Q'_i(D)$ which is some result other than $Q(D)$. We apply the subscript i to Q' to indicate that one player's method of cheating is different from the other players' method of cheating. We denote the honest action as h, and the cheating action as c. These actions are depicted in figure 4.

Information: Data Owner O has given his database D to S_1 and S_2, with an HMAC message authentication code appended to each tuple. Any message authentication scheme would work here, but its purpose and only effect is that it maintains the integrity of the data. This means that the servers cannot alter any tuples and cannot add any tuples without being detected. The players have entered into an agreement (a contract) before the game, and the contents of this contract are known to all players. The contract could contain the probability α. We assume that no updates are to be made to the database once they are outsourced (they are outsourced purely for the purposes of querying).

Payoffs: The owner recieves the information value of the results received, given by $I_v(Q)$, where Q is either $Q(D)$ or $Q'_i(D)$, minus the amount paid to the servers $P(Q)$. The servers recieve this payment, minus the cost of computing the query, $C(Q)$. For simplicity's sake, we assume that both outsourcing services have the same cost of computation and receive the same payment for the query. The logic below easily applies to the case where costs are different, but this assumption simplifies the equations involved. These payoffs are additionally adjusted by the aforementioned contract. We assume the reservation utility of all parties is zero, and if any party declines the contract, then none of the parties participate.

We assume that $I_v(Q(D)) \geq (1+\alpha)P(Q)$ and $P(Q) \geq C(Q)$. If this were not the case, then the game would not be individually rational without some outside subsidies (that is, some player's expected payout would be less than zero). In essence, we want to ensure that the data owner would want to pay $(1+\alpha)P(Q)$ to receive the result, and the cloud provider would accept $P(Q)$ for the computation. To do this, we make sure that the value that the data owner

places on the query is at least the expected payment, and the cost to the cloud providers is no more than the amount they would be paid. No one takes a loss on the transaction.

We now present two contracts, both of which provide simple solutions to the above game in which neither server has incentive to cheat. The first is very simple and requires no additional computation. The second is intuitively more fair, and thus more likely to be accepted in a real world scenario. Both contracts, however, would be accepted by rational players. It should be noted that both of these contracts are loosely based on the results from Auditing Game II and III in [14].

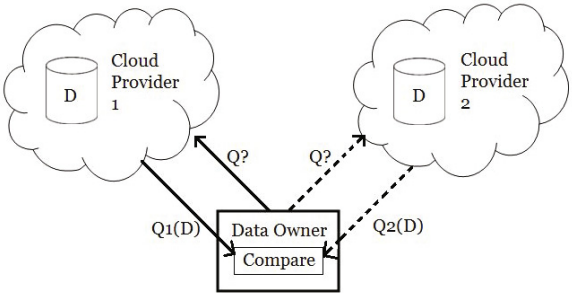


Fig. 2. The Two-Cloud Query Verification System

Contract 1. If the owner asks for query responses from both servers, and the results do not match, both servers pay a penalty of F to the owner, and return the money paid for the computation $P(Q)$ as well.

Theorem 4. The above game with contract 1 has an individually rational, incentive compatible equilibrium in which the servers behave honestly.

Proof: Let $C(Q'_i)$ be the cost of computing Q'_i for S_i. Note that, because S_1 and S_2 do not collude, S_1 does not know Q'_2, and S_2 does not know Q'_1. The only function both know for sure is Q. Without additional knowledge, we can assume that the probability that $Q'_1(D) = Q'_2(D)$ is negligible. For a player to even consider returning Q'_i instead of Q, we must have $C(Q'_i) \leq C(Q)$, since a player will not cheat if they do not gain anything from it. We also assume that $I_v(Q'_i(D)) < 0 < I_v(Q(D))$, since not only is the false result not what the owner asked for, but also appears to be the true result if not verified. If the wrong answer is believed to be correct, this would lead to wrong decisions, and ultimately, financial loss, on the part of the owner. Now, we can define the expected payoffs to each player, where $u_P(x, y)$ is the expected utility for player P when S_1 takes action x and S_2 takes action y. Note that, in these equations and throughout the rest of the paper, we omit the argument D from Q, since D is fixed. We begin with O. If both players are honest (equation 1), O recieves the value of the information gained from the query, minus the expected payment for the calculation, $1 + \alpha$ times $P(Q)$. If one player is dishonest (equations 2 and 3), then with probability α, O detects this and gets both the honest and

the dishonest result and the fine F from both players. With probability $1-\alpha$, he does not detect this, and gets either the correct value or the incorrect value with equal probability. In the event that both players cheat (equation 4), they are once again caught with probability α, but in this case, when they are not caught, O receives only bogus values. This results in the following equations:

$$u_O(h,h) = I_v(Q(D)) - (1+\alpha)P(Q) \tag{1}$$

$$u_O(h,c) = \alpha(2F + I_v(Q) + I_v(Q'_2)) + (1-\alpha)(\frac{1}{2}(I_v(Q) + I_v(Q'_2)) - P(Q)) \tag{2}$$

$$u_O(c,h) = \alpha(2F + I_v(Q) + I_v(Q'_1)) + (1-\alpha)(\frac{1}{2}(I_v(Q) + I_v(Q'_1)) - P(Q)) \tag{3}$$

$$u_O(c,c) = \alpha(2F + I_v(Q'_1) + I_v(Q'_2)) + (1-\alpha)(\frac{1}{2}(I_v(Q'_1) + I_v(Q'_2)) - P(Q)) \tag{4}$$

For the servers, if both servers are honest (equations 5 and 8), they receive the payment for the query, minus the cost of the query, provided they are selected to perform the calculation. This selection probability is why the equations below contain $\frac{1}{2}$. Otherwise, they gain nothing and lose nothing. If one player is dishonest, that player (equations 7 and 10), regardless of whether the other player is honest, with probability α is caught, and loses the fine F. With probability $1-\alpha$, the player is not caught, and gains the payment $P(Q)$, minus the cost of computing his cheat, $C(Q'_i)$, if he is chosen for the computation. If a player is honest while the other player is dishonest (equations 6 and 9), that player similarly is punished with probability α, but invests a cost of $C(Q)$ instead of $C(Q'_i)$ in the computation. This gives us the following equations:

$$u_{S_1}(h,h) = \frac{1}{2}(1+\alpha)(P(Q) - C(Q)) \tag{5}$$

$$u_{S_1}(h,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q)) - \alpha F \tag{6}$$

$$u_{S_1}(c,h) = u_{S_1}(c,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q'_1)) - \alpha F \tag{7}$$

$$u_{S_2}(h,h) = \frac{1}{2}(1+\alpha)(P(Q) - C(Q)) \tag{8}$$

$$u_{S_2}(c,h) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q)) - \alpha F \tag{9}$$

$$u_{S_2}(h,c) = u_{S_1}(c,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q'_2)) - \alpha F \tag{10}$$

We can now find the α for which the expected value for S_1 is less when he cheats than when he is honest, assuming S_2 is honest. By symmetry, this will be the same for S_2. Thus, we set:

$$\frac{1}{2}(1-\alpha)(P(Q)-C(Q_1'))-\alpha F \leq \frac{1}{2}(1+\alpha)(P(Q)-C(Q))$$

Let H represent the quantity $P(Q)-C(Q)$, and H' represent the quantity $P(Q)-C(Q_1')$. Distribute the $(1+\alpha)$ and $(1-\alpha)$ to get:

$$\frac{1}{2}(H')-\frac{\alpha}{2}(H')-\alpha F \leq \frac{1}{2}(H)+\frac{\alpha}{2}(H)$$

Rearranging and combining terms, we get:

$$\frac{1}{2}(C(Q)-C(Q_1')) \leq \alpha F+\alpha P(Q) + \frac{\alpha}{2}(C(Q)-C(Q_1'))$$

Let G represent the quantity $C(Q)-C(Q_1')$, that is, the amount the server would gain from cheating. Substituting this in and factoring out an α, we get:

$$\frac{1}{2}G \leq \alpha(F+P(Q)+\frac{1}{2}G)$$

Multiplying through by two, we get:

$$G \leq \alpha(2F+2P(Q)+G)$$

And, solving for α,

$$\frac{G}{2F+2P(Q)+G} \leq \alpha \tag{11}$$

Since we can define F to be whatever we want in the contract, we can make this minimum α value arbitrarily small. If α is at least this much, then S_1 (and by symmetry, S_2) has no incentive to cheat. If S_2 is not honest, then S_1 has no incentive to be honest, but the payout is less for both (much less, if F is large). Therefore, the best outcome is for both players to behave honestly.

Now, we need to show that choosing α is incentive compatible for O. Given that both players are honest, O's utility is given as:

$$u_O(h,h) = I_v(Q(D)) - (1+\alpha)P(Q)$$

which, by our assumption, is greater than or equal to zero. Thus, it is individually rational for O. If α is increased, it merely decreases this value, so O has no incentive to increase α. If we decrease α, then S_1 and S_2 will see cheating as the more profitable choice, and will begin cheating. This leads to:

$$u_O(c,c) = \alpha(2F+I_v(Q_1')+I_v(Q_2')) + (1-\alpha)(\frac{1}{2}(I_v(Q_1')+I_v(Q_2'))-P(Q))$$

Now, since our α is less than our prescribed value in equation (11), F is bounded above by $\frac{G}{\alpha} - 2P(Q) - G$. Because of this, as α approaches zero, the first term of the above equation decreases. The second term is negative (as $I_v(Q'_1)$ and $I_v(Q'_2)$ are less than zero), and gets worse as α approaches zero. Thus, if α is less than our prescribed value, O expects to lose value from cheating. So, O has no incentive to deviate from $\alpha = \frac{G}{2F+2P(Q)+G}$.

Now, in practice, O does not know G. Thus, he must choose the smallest α that he knows he can use. Since $P(Q) \geq C(Q) \geq G$, O can choose $\alpha = \frac{P(Q)}{2F+2P(Q)-P(Q)} = \frac{P(Q)}{2F-P(Q)}$.

Now, based on the above analysis, it is clear that a cheater will gain less than an honest player when the value of α is chosen as above, regardless of whether the other player is honest. Thus, S_1 and S_2 have no incentive to cheat, and this is incentive compatible for these players as well.

Now, quickly, a note on individual rationality: O has expected payout of $Iv(Q) - (1+\alpha)(P(Q)$. If this is greater than the reservation utility (zero), then the contract is individually rational for O. By our initial assumption about the value of the query, this is true. S_1 and S_2, in equilibrium, have an expected payout of $\frac{1}{2}(1+\alpha)(P(Q) - C(Q))$. Again, by the above assumption, this is true.

As this is both incentive compatible and individually rational for all players, this contract creates the best possible equilibrium where S_1 and S_2 do not cheat, and O pays only $(1+\alpha)$ times the price of a single computation (where α is small). □

5 A More Intuitively Fair Solution

Now, it might seem unfair to punish both players when only one player cheats. The rational player would see the above contract as completely fair, but humans are not always completely rational. Thus, we also examine a contract which identifies the cheater and punishes only the cheater.

Contract 2. If the owner asks for query responses from both servers, and the results do not match, the owner performs a potentially costly audit of the computation. Each server whose result does not match the result given by the owner's process pays a fine F to the owner.

Theorem 5. The above game under contract 2 also has an equilibrium where both servers remain honest.

Proof: Let all variables be defined as above, and let $c(A(Q, Q'_1, Q'_2))$ represent the cost of auditing the computation. The payout functions associated with this contract are as follows:

We begin with O. If both players are honest (equation 12), O recieves the value of the information gained from the query, minus the expected payment for the calculation, $1 + \alpha$ times $P(Q)$. If one player is dishonest (equations 13 and 14), then with probability α, O detects this and gets both the honest and the dishonest result and the fine F from the dishonest player. In this case, he also pays for a costly audit ($c(A(Q, Q'_1, Q'_2))$ to determine which player cheated. With

probability $1-\alpha$, he does not detect this, and gets either the correct value or the incorrect value with equal probability. In the event that both players cheat (equation 15), they are once again caught with probability α, and both pay the fine. However, O only recieves false values, and still pays for the audit. This results in the following equations:

$$u_O(h,h) = I_v(Q(D)) - (1+\alpha)P(Q) \tag{12}$$

$$u_O(h,c) = \alpha(F + I_v(Q) + I_v(Q'_2) - c(A(Q,Q'_1,Q'_2))) + (1-\alpha)(\frac{1}{2}(I_v(Q) + I_v(Q'_2)) - P(Q)) \tag{13}$$

$$u_O(c,h) = \alpha(F + I_v(Q) + I_v(Q'_1) - c(A(Q,Q'_1,Q'_2))) + (1-\alpha)(\frac{1}{2}(I_v(Q) + I_v(Q'_1)) - P(Q)) \tag{14}$$

$$u_O(c,c) = \alpha(2F + I_v(Q'_1) + I_v(Q'_2) - c(A(Q,Q'_1,Q'_2))) + (1-\alpha)(\frac{1}{2}(I_v(Q'_1) + I_v(Q'_2)) - P(Q)) \tag{15}$$

For the servers, if both servers are honest (equations 16 and 19), they receive the payment for the query, minus the cost of the query, provided they are selected to perform the calculation. This selection probability is why the equations below contain $\frac{1}{2}$. Otherwise, they gain nothing and lose nothing. If one player is dishonest, that player (equations 18 and 21), regardless of whether the other player is honest, with probability α is caught, and loses the fine F. With probability $1-\alpha$, the player is not caught, and gains the payment $P(Q)$, minus the cost of computing his cheat, $C(Q'_i)$, if he is chosen for the computation. In this case, if a player is honest while the other player is dishonest (equations 17 and 20), the player is not punished, and therefore receives exactly the same payment as if both players were honest. This gives us the following equations:

$$u_{S_1}(h,h) = \frac{1}{2}(1+\alpha)(P(Q) - C(Q)) \tag{16}$$

$$u_{S_1}(h,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q)) \tag{17}$$

$$u_{S_1}(c,h) = u_{S_1}(c,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q'_1)) - \alpha F \tag{18}$$

$$u_{S_2}(h,h) = \frac{1}{2}(1+\alpha)(P(Q) - C(Q)) \tag{19}$$

$$u_{S_2}(c,h) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q)) \tag{20}$$

$$u_{S_2}(h,c) = u_{S_1}(c,c) = \frac{1}{2}(1-\alpha)(P(Q) - C(Q'_2)) - \alpha F \tag{21}$$

We can now find the α for which the expected value for S_1 is less when he cheats than when he is honest, assuming S_2 is honest. By symmetry, this will be the same for S_2. Thus, we set:

$$\frac{1}{2}(1-\alpha)(P(Q)-C(Q'_1))-\alpha F \leq \frac{1}{2}(1+\alpha)(P(Q)-C(Q))$$

This inequality is exactly the same as in theorem 4. Thus, letting G represent the quantity $C(Q)-C(Q'_1)$, we get:

$$\frac{G}{2F+2P(Q)+G} \leq \alpha \tag{22}$$

Since we can define F to be whatever we want in the contract, we can make this minimum α value arbitrarily small. If α is at least this much, then S_1 (and by symmetry, S_2) has no incentive to cheat. If S_2 is not honest, then S_1 has no incentive to be honest, but the payout is less for both (much less, if F is large). Therefore, the best outcome is for both players to behave honestly.

Now, we need to show that choosing α is incentive compatible for O. Given that both players are honest, O's utility is given as:

$$u_O(h,h) = I_v(Q(D)) - (1+\alpha)P(Q)$$

which, by our assumption, is greater than or equal to zero. Thus, it is individually rational for O. If α is increased, it merely decreases this value, so O has no incentive to increase α. If we decrease α, then S_1 and S_2 will see cheating as the more profitable choice, and will begin cheating. This leads to:

$$\begin{aligned} u_O(c,c) = \alpha(2F + I_v(Q'_1) + I_v(Q'_2) - c(A(Q,Q'_1,Q'_2))) \\ + (1-\alpha)(\frac{1}{2}(I_v(Q'_1)+I_v(Q'_2)) - P(Q)) \end{aligned}$$

As in theorem 4, the first term of this equation decreases as α tends to zero (regardless of $c(A(Q,Q'_1,Q'_2))$, and the second term is negative. Thus, as α decreases, O's expected payout decreases. Therefore, O has no incentive to adjust α up or down, and this α is incentive compatible for O. The arguments in theorem 4 for the incentive compatibility of the servers and the individual rationality of both players continue to apply in this case. Thus, the contract is both incentive compatibile and individually rational for all parties involved. □

The audit process mentioned above could be done in several ways. The simplest, although most expensive, of these would be for the owner to retrieve all the data, then perform the query himself. Obviously, this defeats the purpose of data outsourcing. Based on the fact that the outsourced data uses some message authentication codes to keep the data from being modified, we can improve this. First, for selection queries, if one player fails any MAC checks, then they are obviously cheating. If one player returns fewer results than the other, then they are also obviously cheating. For aggregate queries, we can have each source return the tuples which were selected for the aggregation process. We can then check to see if the aggregate query result matches the values returned by the server. Finding a tuple set that matches a false query result might prove incredibly difficult if the false query was not generated from a sample. We can also apply the

same techniques used for selection queries, noting that the cloud that returns fewer tuples must be cheating (provided all tuples returned are authenticated). Essentially, for a given query, we end up asking the providers to "show their work," or face consequences.

Note the generality of this result. In contrast with many other results, it works for **any query** on **any database** (with the caveat that the query is deterministic), and it works in only one round of computation.

5.1 Conclusions

In summary, by thinking about the problem of query verification from a different perspective, namely, that of an economist, we can drastically reduce the computation required to ensure that the result asked for is the result received. Using the game-theoretic framework outlined here, we show that using a multiple servers, contracts can be designed that will ensure that results obtained from an outsourced computation service are genuine, while requiring only a fractional increase in cost. This is, of course, in contrast to most methods of query verification, which rely on complicated security technologies. The various query verification technologies that are out there are still quite useful, however. Specialized verification methods which take up very little space work well for common queries. They are, however, not generic and can rely on some expensive operations. The outside-the-box approach of using a redundant data service for verification vastly simplifies this process, and incurs a minimal cost.

5.2 Future Work

In the future, we will consider a similar auditing mechanism using only a single cloud service. This mechanism will use both a costly full audit and a less costly partial audit to achieve minimal cost. We also will consider the use of other verification methods in a framework such as ours, and how they can be improved through the use of incentives. Finally, we also wish to consider the effect of accidental errors. The steepness of the penalties involved in this project could lead more risk-averse players to balk at the contract. Nevertheless, a rational, risk-neutral player will have no problems with these contracts, and will be incentivized to check for errors before reporting results.

Acknowledgements. This work was partially supported by Air Force Office of Scientific Research MURI Grant FA9550-08-1-0265, National Institutes of Health Grant 1R01LM009989, National Science Foundation (NSF) Grant Career-0845803, and NSF Grant 0964350.

References

1. Atallah, M., Cho, Y., Kundu, A.: Efficient data authentication in an environment of untrusted third-party distributors. In: IEEE 24th International Conference on Data Engineering, pp. 696–704. IEEE (2008)

2. Canetti, R., Riva, B., Rothblum, G.: Practical delegation of computation using multiple servers. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 445–454. ACM (2011)
3. Chen, H., Ma, X., Hsu, W., Li, N., Wang, Q.: Access Control Friendly Query Verification for Outsourced Data Publishing. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 177–191. Springer, Heidelberg (2008)
4. Gennaro, R., Gentry, C., Parno, B.: Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 465–482. Springer, Heidelberg (2010)
5. Google. Google bigquery service (2011)
6. Haber, S., Horne, W., Sander, T., Yao, D.: Privacy-preserving verification of aggregate queries on outsourced databases. Technical report, Citeseer (2006)
7. Merkle, R.: Secrecy, authentication and public key systems (1979)
8. Mykletun, E., Narasimha, M., Tsudik, G.: Authentication and integrity in outsourced databases. ACM Transactions on Storage (TOS) 2(2), 107–138 (2006)
9. National Institute of Standards and Technology. FIPS 180-2, secure hash standard, federal information processing standard (FIPS), publication 180-2. Technical report, Department of Commerce (August 2002)
10. Pang, H., Jain, A., Ramamritham, K., Tan, K.: Verifying completeness of relational query results in data publishing. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 407–418. ACM (2005)
11. Pang, H., Zhang, J., Mouratidis, K.: Scalable verification for outsourced dynamic databases. Proceedings of the VLDB Endowment 2(1), 802–813 (2009)
12. Patel, P., Ranabahu, A., Sheth, A.: Service level agreement in cloud computing. In: Cloud Workshops at OOPSLA (2009)
13. F. Pub. 198, the keyed-hash message authentication code (hmac). Federal Information Processing Standards Publication, 198 (2002)
14. Rasmusen, E.: Games and information: An introduction to game theory. Wileyblackwell (2007)
15. Rivest, R.: The md5 message-digest algorithm (1992)
16. Sion, R.: Query execution assurance for outsourced databases. In: Proceedings of the 31st International Conference on Very Large Databases, pp. 601–612. VLDB Endowment (2005)
17. Sion, R.: Secure data outsourcing. In: Proceedings of the 33rd International Conference on Very large Databases, pp. 1431–1432. VLDB Endowment (2007)
18. Xie, M., Wang, H., Yin, J., Meng, X.: Integrity auditing of outsourced data. In: Proceedings of the 33rd International Conference on Very Large Databases, pp. 782–793. VLDB Endowment (2007)
19. Xu, J., Chang, E.: Authenticating aggregate range queries over multidimensional dataset. Technical report, Cryptology ePrint Archive, Report 2010/050 (2010)
20. Yang, Y., Papadias, D., Papadopoulos, S., Kalnis, P.: Authenticated join processing in outsourced databases. In: Proceedings of the 35th SIGMOD International Conference on Management of Data, pp. 5–18. ACM (2009)
21. Yi, K., Li, F., Cormode, G., Hadjieleftheriou, M., Kollios, G., Srivastava, D.: Small synopses for group-by query verification on outsourced data streams. ACM Transactions on Database Systems (TODS) 34(3), 1–42 (2009)

Author Index