Security Games over Lexicographic Orders

Stefan Rass[1(✉)], Angelika Wiegele[2], and Sandra König[3]

[1] Universitaet Klagenfurt, Institute of Applied Informatics, Klagenfurt, Austria
stefan.rass@aau.at
[2] Department of Mathematics, Universitaet Klagenfurt, Klagenfurt, Austria
angelika.wiegele@aau.at
[3] Austrian Institute of Technology, Center for Digital Safety & Security, Vienna, Austria
sandra.koenig@ait.ac.at

Abstract. Security is rarely single-dimensional and is in most practical instances a tradeoff between dependent, and occasionally conflicting goals. The simplest method of multi-criteria optimization and games with vector-valued payoffs, is transforming such games into ones with scalar payoffs, and looking for Pareto-optimal behavior. This usually requires an explicit weighting of security goals, whereas practice often only lets us rank security goals in terms of importance, but hardly admits a crisp numerical weight being assigned. Our work picks up the issue of optimizing security goals in descending order of importance, coming to the computation of an optimal solution w.r.t. lexicographic orders. This is interesting in two ways, as it (i) is theoretically nontrivial since lexicographic orders do not generally admit representations by continuous utility functions, hence render Nash's classical result inapplicable, and (ii) practically relevant since it avoids ambiguities by subjective (and perhaps unsupported) importance weight assignments. We corroborate our results by giving numerical examples, showing a method to design zero-sum games with a set of a-priori chosen Nash equilibria. This simple instance of mechanism design may be of independent interest.

Keywords: Lexicographic order · multi-criteria optimization · Mechanism design · Security economics

1 Introduction and Motivation

Security is in many practical instances a multi-dimensional matter. Basic security goals like confidentiality, integrity and availability (CIA) are known to be potentially conflicting. For example, encryption serves confidentiality but can be problematic for availability. Likewise, keeping systems or data redundant to increase availability makes confidentiality more complex and may add to the attack surface. Generally, the different dimensions of security can depend on parameter settings (for example, threshold secret sharing has different resilience

Q. Zhu et al. (Eds.): GameSec 2020, LNCS 12513, pp. 422–441, 2020.
https://doi.org/10.1007/978-3-030-64793-3_23

against passive or active adversaries, regarding recoverability of the secret but also its confidentiality [16,21,26,35]), or security properties themselves (such as is the case for sanitizable signatures [4]). The general problem reaches much beyond basic cryptographic matters, since also risk management is by default a multi-dimensional challenge [28] of keeping financial losses, damages to reputation, legal liabilities and many more under control.

Generally, the priority of security goals depends on the application domain, and we can broadly distinguish two example domains with opposite priorities regarding CIA (we spare the full spectrum of security requirements here, as it would take us much beyond the scope of this section and work):

- data-processing enterprises will have confidentiality as their highest good, followed by integrity, and lastly followed by availability of the personal data in their possession.
- production-oriented enterprises, on the contrary, will not necessarily rely on continuous processing of personal data, but rather will protect their production lines, i.e., availability is their top priority. Likewise, production control signals need integrity protection (as second priority goal), followed by confidentiality as the least important among the three goals.

We will later illustrate our methods by showing how to apply them in an enterprise whose main business is processing personal data, or producing goods. It will turn out that the results are somewhat different, yet the method of computing them is the same in both cases, without the need to become explicit on a numeric difference in terms of importance of the security goals.

1.1 Related Work

Our work addresses the problem of multi-criteria optimization and -game theory, which is traditionally approached by scalarizing the vector-valued revenues gained by each player. While the concept of a Nash equilibrium is straightforward to generalize to a Pareto-Nash equilibrium [19] or -security strategy, the hereby involved importance weights put practitioners to the challenge of finding a meaningful quantification of importance for the relevant goals, or more generally, prioritization of security requirements [14,24], whose importance is widely recognized throughout security practitioners [18,25,34]. Sophisticated methods and heuristics to do this include the Choquet integral [12], fuzzy logic [31], or resorting to Pareto-optimality [33].

The problem of optimization over lexicographic orders itself is in fact not new, yet has seen surprisingly little attention for security applications, despite the fact that security goals are often in a very strict order of importance. The theoretical toolbox is rich, and includes axiomatic studies and characterizations [8–10,15] and questions of equilibria in lexicographically ordered vector-payoffs [17] and min-max optimization [22,23]. The latter is most closely related to our work, yet ours is conceptually different and uses a sequential application of criteria. Essentially, we adapt the "lexicographic method" (also called preemtive

approach; see [5]) used in multi-criteria optimization, to the computation of equilibria in zero-sum games for security in a new form.

1.2 Our Contribution

The main contribution made in this work is the description of a simple method to compute a game-theoretic optimum if several goals are involved that obey a total order of priority without resorting to scalarization by using importance weights. As an implied consequence, we get a method to thereby refine ambiguous equilibria if they are intended as defense strategies, or as pointers to neuralgic spots in a system; the latter occurring if we interpret the optimal attack strategies in this way.

Independently, we corroborate our results by illustrating the computation using a method of mechanism design, allowing us to construct zero-sum games with a set of defined, i.e., designed, equilibria for both players. We remark that our focus on two-person zero-sum games is what makes the construction challenging, as it is conceptually not difficult to construct certain multi-player nonzero-sum games with desired optima: in the most trivial instance, we can just let the payoff for a player be independent of the other player's actions, and let it attain a maximum at the desired location(s). More generally, we may look for functions (e.g., polynomials) that, when having their domain restricted to points where the opponents have their optima, still have optima at desired positions. While this is, strictly speaking, no longer a strategic interaction with conflicts, and hence an uninteresting case for game theory or security, it nonetheless yields a theoretically valid instance of a general game. The most extreme instance is thus with exactly opposite goals of the players that depend on the player's mutual actions, i.e., a zero-sum game. This class of games is also useful in security modeling, since it delivers worst-case defenses without the need for accurate adversary models or -profiling [29,30,32,36].

2 Preliminaries

2.1 Notation

We let vectors appear as bold-printed lower-case letters, and matrices will use uppercase bold printed letters. Sets will appear as upper case letters in normal font. Families of sets are written in calligraphic font. Let $\leq_{lex}$ be the lexicographic order over $\mathbb{R}^n \times \mathbb{R}^n$, defined in the usual way by setting $\mathbf{a} = (a_1, \ldots, a_n) \leq_{lex} (b_1, \ldots, b_n)$ if and only if $[a_1 < b_1] \vee [(a_1 = b_1) \wedge (a_2, \ldots, a_n) \leq_{lex} (b_2, \ldots, b_n)]$. In the following, let U be a metric vector space, and let $\leq$ be an ordering relation on it. Typically, U will be a space spanned over $\mathbb{R}$.

For a finite set $X = \{x_1, \ldots, x_n\}$, we let the symbol $\Delta(X)$ be the simplex over X, i.e., the set $\Delta(X) := \{\sum_{i=1}^n \lambda_i \cdot x_i | \lambda_i \geq 0 \text{ for all } i, \text{ and } \sum_{i=1}^n \lambda_i = 1\}$. Games are triples $(N, \mathcal{S}, H)$, with N being a finite set of players, S being a

family of finite sets, one $S_i \in \mathcal{S}$ associated with each player giving its actions, and H is a collection of functions $u_i : S_i \times S_{-i} \to \mathbb{R}$ of utility functions. Herein, the notation S_{-i} is the cartesian product of all elements in $\mathcal{S}$, excluding S_i. It expresses the joint actions taken by player i's opponents. Throughout this work, we will have $N = \{1, 2\}$, $\mathcal{S} = \{S_1 = \Delta(AS_1), S_2 = \Delta(AS_2)\}$ for finite action spaces AS_1, AS_2 with n elements for player 1, and m elements for player 2. In this special case, we can represent the whole game by an $(n \times m)$-payoff matrix $\mathbf{A}$, and abbreviate our notation by referring to the matrix $\mathbf{A}$ to synonymously mean the game that it defines.

2.2 Representability of the Lexicographic Order

Given an ordering relation $\leq$ on a set $U \times U$, we say that $\leq$ is *representable* by a function $f : U \to \mathbb{R}$ if, $[a \leq b] \iff [f(a) \leq f(b)]$ for all $a, b \in U$ that are comparable under $\leq$.

It is well known that the lexicographic order, in general, does *not* lend itself to a representation by any continuous function. This folklore result is made precise as Proposition 1, whose proof we let follow in Appendix A for convenience of the reader.

Proposition 1. *On the totally ordered set* $([0,1]^2, \leq_{lex})$*, there is no continuous function* $f : [0,1]^2 \to \mathbb{R}$ *with the property that* $(x_1, x_2) \leq_{lex} (y_1, y_2) \iff f(x_1, x_2) \leq f(y_1, y_2)$.

Proposition 1 makes lexicographic orders generally difficult to handle in optimization algorithms, since it lacks the minimal requirement of continuity, assumed in many optimization methods (up to the stronger requirement of differentiability). On the bright side, this lack of continuity only holds in the most general setting, and special situations may still admit a continuous representation, or other means of efficient optimization, as we outline next.

3 Finding Lex-Order Optimal Strategies

Our algorithm to compute equilibria over lexicographic orders will decompose a vector-valued game matrix into a sequence of games, in which the i-th game has a payoff matrix composed from the respective i-th coordinates in each vector. That is, given the vector-valued utility function $u : S_1 \times S_2 \to \mathbb{R}^d$ for a player, on the discrete strategy spaces S_1, S_2, we define the k-th game for $k = 1, 2, \ldots, d$ via the matrix $\mathbf{A}_k = (u_k(r, c))_{(r,c) \in S_1 \times S_2}$. Note that all game matrices have the same $n \times m$-shape.

Towards finding an optimum, i.e., an equilibrium, over the lex-order, we induct over the coordinate $k = 1, 2, \ldots, d$, and prove the existence of equilibria along the way. The goal is finding strategies that are lex-order optimal for each player, in light of unilateral deviation.

Induction start $k = 1$: The game $\mathbf{A}_1$ is only about scalar payoffs, and the equilibrium w.r.t. the $\leq$-order over $\mathbb{R}$ is also lex-order optimal, since the two relations coincide on scalars.

Remark 1. We emphasize that the solution for $k = 1$ is the only stage at which the optima are the same as equilibria, since the lex-order and the real-valued $\leq$ coincide only in that case. As the idea will be seeking optima in the next stage among the optima found for (all of the) previous games, the term "equilibrium" will hereafter be understood *conditional* on the strategies constrained to be also equilibria for previous games (namely $\mathbf{A}_{k-1}, \mathbf{A}_{k-2}, \ldots$). Clearly, the solution may not be an equilibrium for any of the games when considered in isolation and independent of the others.

Remark 2. (Number of optima is finite). The characterization of equilibria by linear programs (appearing in Appendix B in the context of constructing games with desired equilibria) defines the feasible set of solutions via a finite number of inequalities. Therefore, the overall solution set, though generally infinite (as all convex combinations of optima are themselves also optimal; cf. Lemma 1), is representable by a finite (though in the worst case exponentially large) number of points, whose convex combination represents all feasible solutions, and hence also all optima therein. This finiteness property in fact holds in a measure-theoretic sense for almost all games [13].

Induction step $k-1 \to k$: For the induction until $k-1$, we can assume a finite set $E_{k-1} = \left\{(\mathbf{x}^*_{k-1,1}, \mathbf{y}^*_{k-1,1}), \ldots, (\mathbf{x}^*_{k-1,n_{k-1}}, \mathbf{y}^*_{k-1,n_{k-1}})\right\}$ of optima (cf. Remark 1). For the induction hypothesis, assume that all of them are also optima of the $(k-1)$-th game $\mathbf{A}_{k-1}$.

Our goal for the induction step is refining the equilibria into an optimum for the game $\mathbf{A}_k$ on the k-th coordinate. The basic idea is to play the game $\mathbf{A}_k$ restricted only to strategies that are already optimal for $\mathbf{A}_{k-1}$, so as to retain optimality in the previous games, when optimizing our behavior in the next game $\mathbf{A}_k$. We materialize this plan now.

From the set E_{k-1}, we define an auxiliary game $\mathbf{B}_k$: its strategy spaces are $S_{k,1} = \left\{\mathbf{x}^*_{k-1,i} | i = 1, \ldots, n_{k-1}\right\}$ and $S_{k,2} = \left\{\mathbf{y}^*_{k-1,i} | i = 1, \ldots, n_{k-1}\right\}$. The implied $(n_{k-1} \times n_{k-1})$-payoff structure is the matrix

$$\mathbf{B}_k := ((\mathbf{x}^*_{k-1,i})^T \cdot \mathbf{A}_k \cdot \mathbf{y}^*_{k-1,j})_{i,j=1}^{n_{k-1}}. \tag{1}$$

The so-constructed zero-sum game has its own equilibria, the entire set of which is enumerable by known algorithms [1]. Moreover, we have convenient topological properties, assuring that the set among which we look for optima is convex and closed, and the strategy spaces of $\mathbf{B}_k$ are compact sets. This is Lemma 1.

Lemma 1. *Let* $\mathbf{A}$ *be a matrix inducing a zero-sum game, on the strategy spaces* $S_1 \subset \Delta(\mathbb{R}^n), S_2 \subset \Delta(\mathbb{R}^m)$. *If* $\leq$ *is a continuous ordering*[1]*, and* $u : S_1 \times S_2 \to \mathbb{R}$

[1] An ordering $\leq$ is called *continuous*, if all bounded sequences (a_n) with $a_n \leq b$ for all $n \in \mathbb{R}$, have a limit that also satisfies the same bound $\lim_{n\to\infty} a_n \leq b$, if that limit exists. The lexicographic order is discontinuous w.r.t. this definition, since the sequence $(1/n, 0) \geq_{lex} (0, 1)$ for all $n \in \mathbb{N}$, but $\lim_{n\to\infty}(1/n, 0) = (0, 0) \leq_{lex} (0, 1)$.

is continuous w.r.t. a norm on $\mathbb{R}^n$ *and the order topology induced by* $\leq$*, then the set of mixed Nash equilibria for the zero sum game* $\mathbf{A}$ *is nonempty, convex and compact.*

Combining Lemma 1 with Glicksberg's theorem [11] yields an equilibrium $(\boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*) \in \mathbb{R}^{n_{k-1}} \times \mathbb{R}^{n_{k-1}}$ in the game $\mathbf{B}_k$, which by definition satisfies the saddle point condition $\boldsymbol{\alpha}^T \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^* \leq (\boldsymbol{\alpha}^*)^T \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^* \leq (\boldsymbol{\alpha}^*)^T \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}$, for all $\boldsymbol{\alpha}, \boldsymbol{\beta}$.

By the same token as in Remark 2, we can assume a finite number n_k of equilibria in $\mathbf{B}_k$, and let us put these as rows into two matrices $\mathbf{X}_{k-1} \in \mathbb{R}^{n_k \times n_{k-1}}$ and $\mathbf{Y}_{k-1}^{n_k \times n_{k-1}}$. These two relate to $\mathbf{B}_k$ via $\mathbf{B}_k = \mathbf{X}_{k-1} \cdot \mathbf{A}_k \cdot \mathbf{Y}_{k-1}^T$.

Since the pure strategies in $\mathbf{B}_k$ are all equilibria in $\mathbf{A}_{k-1}$, any mixed strategy pair $(\boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*) \in \Delta(S_{k,1}) \times \Delta(S_{k,2})$ played in $\mathbf{B}_k$ induces an equilibrium

$$\mathbf{x}'_k := (\boldsymbol{\alpha}_k^*)^T \cdot \mathbf{X}_{k-1}, \quad \text{and} \quad \mathbf{y}'_k := (\boldsymbol{\beta}_k^*)^T \cdot \mathbf{Y}_{k-1} \tag{2}$$

for the game $\mathbf{A}_{k-1}$ by Lemma 1 (note that the payoff is continuous, since the auxiliary game $\mathbf{B}_k$ is a standard matrix game and as such with continuous payoffs over mixed strategies). Thus, the pair $(\mathbf{x}'_k, \mathbf{y}'_k)$ satisfies the saddle point condition in $\mathbf{A}_{k-1}$, being $\mathbf{x}^T \cdot \mathbf{A}_{k-1} \cdot \mathbf{y}'_k \leq (\mathbf{x}'_k)^T \cdot \mathbf{A}_{k-1} \cdot \mathbf{y}'_k \leq (\mathbf{x}'_k)^T \cdot \mathbf{A}_{k-1} \cdot \mathbf{y}$ for all $\mathbf{x}, \mathbf{y}$.

Our goal is showing that the pair $(\mathbf{x}'_k, \mathbf{y}'_k)$ is also optimal in the game $\mathbf{A}_k$. To this end, we will adopt player 1's perspective, playing some arbitrary but fixed $\mathbf{x} \neq \mathbf{x}'_k$, while player 2 sticks to $\mathbf{y}'_k$.

Towards a contradiction, suppose that player 1 could improve in $\mathbf{A}_k$ by playing $\mathbf{x}$,

$$\mathbf{x}^T \cdot \mathbf{A}_k \cdot \mathbf{y}'_k > \mathbf{x}'_k \cdot \mathbf{A}_k \cdot \mathbf{y}'_k. \tag{3}$$

Substituting the definition of $\mathbf{x}'_k$ and $\mathbf{y}'_k$ by means of $\mathbf{X}_{k-1}, \mathbf{Y}_{k-1}$ gives on the left side of (3)

$$(\boldsymbol{\alpha}_k^*)^T \cdot \underbrace{\mathbf{X}_{k-1} \cdot \mathbf{A}_k \cdot \mathbf{Y}_{k-1}^T}_{=\mathbf{B}_k} \cdot \boldsymbol{\beta}_k^* = (\boldsymbol{\alpha}_k^*)^T \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^*.$$

With the same substitution on the right side of (3), we get $\mathbf{x}^T \cdot \mathbf{A}_k \cdot \mathbf{Y}_{k-1}^T \cdot \boldsymbol{\beta}_k^* > \boldsymbol{\alpha}_k^* \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^*$. Now, if there were some $\tilde{\mathbf{x}}$ such that $\mathbf{x}^T = \tilde{\mathbf{x}}^T \cdot \mathbf{X}_{k-1}$, we could rewrite the last inequality into $\tilde{\mathbf{x}}^T \cdot \mathbf{X}_{k-1} \cdot \mathbf{A}_k \cdot \mathbf{Y}_{k-1}^T \cdot \boldsymbol{\beta}_k^* = \tilde{\mathbf{x}}^T \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^* > \boldsymbol{\alpha}_k^* \cdot \mathbf{B}_k \cdot \boldsymbol{\beta}_k^*$, to contradict the fact that $(\boldsymbol{\alpha}_k^*, \boldsymbol{\beta}_k^*)$ is an equilibrium in $\mathbf{B}_k$, and thereby finally refute (3). But such an $\tilde{\mathbf{x}}$ is in fact easy to find, since the possible actions are restricted to equilibrium strategies in $\mathbf{A}_{k-1}$. The vector $\mathbf{x}$ must thus be a mix of rows from the matrix $\mathbf{X}_{k-1}$, putting $\mathbf{x}$ into the row-space of $\mathbf{X}_{k-1}$. In that case, the equation system $\mathbf{x}^T = \tilde{\mathbf{x}}^T \cdot \mathbf{X}_{k-1}$, even if over-determined, has a solution being the sought $\tilde{\mathbf{x}}$. Hence, (3) cannot hold, and $\mathbf{x}'_k$ is also optimal in the game $\mathbf{A}_k$, when the opponent plays $\mathbf{y}'_k$. By symmetry, the argument for the second player works analogously, and the induction step is complete, delivering the sought simultaneous optimum $(\mathbf{x}'_k, \mathbf{y}'_k)$ as given by (2). Figure 1 summarizes the construction as an algorithm.

Remark 3. From another angle, the above procedure is viewable as starting in the game on the first coordinate, with a possibly large set E_1 of equilibria, and then narrowing down, resp. *refining*, this set to those elements $E_2 \subseteq E_1$ that are also optimal in the game $\mathbf{A}_2$. Repeating this procedure, we then further restrict the set to $E_3 \subseteq E_2$, in which only those strategies are retained that are also optimal in the game $\mathbf{A}_3$, etc. It is obvious that the equilibria in $\mathbf{A}_1$ can be disjoint from those in $\mathbf{A}_2$, which means that a strategy carried over from E_k to E_{k+1} may no longer be an equilibrium in $\mathbf{A}_{k+1}$. This brings us back to the initial remark made at the induction step, calling the solutions "conditional optimal" rather than equilibria. However, since we are after optimality w.r.t. a lexicographic order, all we count is the chances of worsening the outcome upon an unilateral deviation from the final solution. Since the final set $E_d \subseteq E_{d-1} \subseteq \cdots \subseteq E_1 = \{$ all equilibria in $\mathbf{A}_1\}$ contains only equilibria for the first game, deviating from it would worsen our situation on the first coordinate, and hence irrespectively of the other coordinates, would make the result lex-order worse. Upon equality, the second coordinate kicks in, and so on.

Now, let us wrap up by putting the result in the context of security: like a conventional, scalar-valued, zero-sum game, our lex-ordered optima here have the same property of being a worst-case defense against any adversarial behavior, as long as the defender adheres to the final optimum.

Input: A set of payoff matrices $\mathbf{A}_1, \ldots, \mathbf{A}_d \in \mathbb{R}^{n \times m}$,
Output: A lex-order optimal strategy $\mathbf{x}^*, \mathbf{y}^*$ with the properties told by Prop. 2
Procedure:

1. Compute all equilibria in the game $\mathbf{A}_1$ (e.g., using the algorithm from [1]), and collect the optima for player 1 in the set $S_{1,1}$ and the optima for player 2 in the set $S_{1,2}$.
2. Put $k \leftarrow 1$
3. **if** $k = d$, **then return** any $(\mathbf{x}^*, \mathbf{y}^*) \in S_{k,1} \times S_{k,2}$ as the final result and **terminate**.
4. Otherwise (if $k < d$), update $k \leftarrow k + 1$
5. Arrange the elements of $S_{k-1,1}$ as rows of a matrix $\mathbf{X}$ and arrange the elements of $S_{k-1,2}$ as rows of a matrix $\mathbf{Y}$, and compute the matrix $\mathbf{B}_k = \mathbf{X} \cdot \mathbf{A}_k \cdot \mathbf{Y}^T$.
6. Compute all equilibria in the matrix game $\mathbf{B}_k$ and collect them in a set $E_k := \{(\boldsymbol{\alpha}_1^*, \boldsymbol{\beta}_1^*), \ldots, (\boldsymbol{\alpha}_{n_k}^*, \boldsymbol{\beta}_{n_k}^*)\}$.
7. Iterate over $i = 1, 2, \ldots, n_k$ pairs $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) \in E_k$ and for each pair, compute $\mathbf{x}_i' = (\boldsymbol{\alpha}_i)^T \cdot \mathbf{X}, \mathbf{y}_i' = (\boldsymbol{\beta}_i)^T \cdot \mathbf{Y}$. Put all the resulting $\mathbf{x}_i'$ into the set $S_{k,1}$ and the resulting $\mathbf{y}_i'$ into the set $S_{k,2}$.
8. go back to step 3.

Fig. 1. Computation of lexicographically optimal multi-goal security strategies

More formally, we have:

Proposition 2. *Let* $\mathbf{A}_1, \ldots, \mathbf{A}_d$ *be a series of game-matrices, all of* $(n \times m)$*-shape, describing a security game between a defender as player 1, and an attacker as player 2, whose unknown payoffs may be* $\mathbf{U}_1, \ldots, \mathbf{U}_d$*, in the same game.*

Let the defending player 1 substitute $\mathbf{U}_k := -\mathbf{A}_k$ *for all* $k = 1, 2, \ldots, d$*, in lack of better knowledge, and let* $\mathbf{x}^* = \mathbf{x}'_d, \mathbf{y}^* = \mathbf{y}'_d$ *be the output computed by the lexicographic method as described in Fig. 1.*

Then, conditional on the attacker acting within its own action space (i.e., not playing any strategy whose payoff is not captured by the columns in the $\mathbf{A}_k$*'s), we have the actual payoff in the* k*-th game for* $k = 1, 2, \ldots, d$ *for the defender satisfy*

$$\mathbf{x}^* \cdot \mathbf{A}_k \cdot \mathbf{y}^* \leq \mathbf{x}^* \cdot \mathbf{A}_k \cdot \mathbf{y},$$

for any true behavior $\mathbf{y}$ *of the attacker.*

The proof is a simple consequence of the equilibrium property that we have shown to hold for each $\mathbf{A}_k$: it means that each $\mathbf{x}'_d, \mathbf{y}'_d$ is a saddle point

$$\mathbf{x} \cdot \mathbf{A}_k \cdot \mathbf{y}'_d \leq \mathbf{x}'_d \cdot \mathbf{A}_k \cdot \mathbf{y}'_d \leq \mathbf{x}'_d \cdot \mathbf{A}_k \cdot \mathbf{y},$$

for all $k = 1, 2, \ldots, d$, since each $\mathbf{x}'_k, \mathbf{y}'_k$ is a convex combination of saddle points. If player 2 has a different incentive than engaging in a zero-sum competition, then the saddle point property will ensure that player 1's revenue can only increase by the unilateral deviation of player 2. The worst case is attained for exactly opposite intentions, i.e., a zero-sum regime.

4 Applications and Examples

4.1 Refining Ambiguous Attack or Defense Strategies

One appeal of using game theory to analyze attacker/defender scenarios is its simultaneous account for the natural opposition of interests. Thereby, it delivers optimal defenses and optimal attacks, but their practical value depends on matters of plausibility (e.g., for attacks), or feasibility (e.g., for defenses).

Implausible equilibria arise in models that neglect certain interests of the attacker, or under oversimplifying assumptions on the attack models. Likewise, infeasible defenses can result from models missing on certain cost or efforts imposed on the defender if it were to implement the game-theoretically optimal choice. The existence of ambiguities in Nash equilibria is well documented, and we can state the following explicit result for security games as zero-sum competitions:

Lemma 2. *Pick any set of vectors* $E_1 = \{\mathbf{x}^*_1, \ldots, \mathbf{x}^*_{k_1}\} \subset \mathbb{R}^n$ *and* $E_2 = \{\mathbf{y}^*_1, \ldots, \mathbf{y}^*_{k_2}\} \subset \mathbb{R}^m$*, where* $k_1 < n$ *and* $k_2 < m$*. Then, there is a finite two-person zero-sum game whose equilibria are exactly the set* $\Delta(E_1) \times \Delta(E_2)$*.*

Lemma 2 is proven in Appendix B. It essentially tells that any desired equilibrium for both, the defender and the attacker can be "enforced" by a proper mechanism designed, which is not per se surprising, but this mechanism can take the form of a simple security game. Conversely, this means that we may expect either a unique or an infinitude of equilibria in even the simplest instances of security games, making a method of refining them necessary. The usual way of resorting to more advanced concepts of equilibria is not necessarily also a practical way to go, since it still can leave practically relevant aspects untouched, see some examples following below.

The lexicographic method of computing equilibria does not require a priori knowledge of all relevant dimensions, and can refine ambiguous or implausible results "as needed" by bringing in further utility values. For example, several ways of defending a system by randomization of actions can be judged upon the following additional criteria:

Cost of Changing from Between Strategies: Randomization itself causes friction losses, and this is in conflict with the common practice of "never touch a running system". Thus, changing configurations requires efforts (e.g., people can be reluctant to change their password) and proper modeling [6] that can lead to further utility values for the lexicographic optimization.

Predictability of a Defense for the Attacker: Since a randomized action is essentially describing a random variable, we can – as an additional "utility" – ask for the uncertainty that our defense has against a guessing adversary. To measure this, we could define the randomized choice rule's min-entropy as another utility of interest (not Shannon-entropy, since this is in general not a good measure of unpredictability against guessing).

Cost or Times to Exploit: For the attacker, even if there is a vulnerability to exploit, it is typically a complex choice to pick the "easiest" one. Methods like the Common Vulnerability Scoring (CVSS) associate several scores with a vulnerability, such as required background knowledge, necessary form and duration of access, etc. All these lend themselves to defining their own utility values for the attacker, and can be brought into a lexicographic optimization for the defender to narrow down the set of optimal defenses in such multiple dimensions.

4.2 Example 1: The Pure Algorithm (Numerical Illustration)

Let us now give a numerical example of the computational method from Sect. 3 on a game with two payoffs per player. We start with a constructed matrix $\mathbf{A}_1$ (obtained by application of the techniques to prove Lemma 2; see Appendix B),

$$\mathbf{A}_1 = \begin{pmatrix} 0.955986 & -0.272557 & 0.316327 & -0.405844 & 0.102397 & -0.662056 \\ 0.0454297 & -0.0580642 & -0.178636 & -0.187195 & 0.130912 & 0.204854 \\ -0.298982 & 0.05127 & -0.17908 & 0.0170827 & 0.0593234 & 0.292065 \\ -0.436331 & 0.223209 & -0.113101 & 0.453724 & -0.301461 & 0.340507 \\ -0.558309 & 0.0324137 & 0.0676309 & -0.0335246 & 0.251095 & -0.076376 \end{pmatrix}, \quad (4)$$

known to have three equilibria $\mathbf{x}_1^*, \ldots, \mathbf{x}_3^*$ for player 1, and 2 optimal strategies $\mathbf{y}_1^*, \mathbf{y}_2^*$ for player 2 (given explicitly in Appendix B as (5) and (6)), and another matrix

$$\mathbf{A}_2 = \begin{pmatrix} 5\,4\,5\,3\,3\,2 \\ 5\,5\,3\,2\,5\,1 \\ 3\,3\,1\,1\,2\,4 \\ 2\,2\,5\,5\,2\,3 \\ 2\,3\,2\,4\,3\,2 \end{pmatrix}$$

chosen purely at random, but with the same shape as $\mathbf{A}_1$. A systematic construction of $\mathbf{A}_2$ would be possible (e.g., using a selection of the vectors used above to construct $\mathbf{A}_1$), but the optimization does not hinge on such constructed input(s).

Now, the auxiliary game matrix $\mathbf{B}_2$ arises from computing the equilibria of the scalar-valued matrix game $\mathbf{A}_1$, which we know to be given by any combination of $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^*\} \times \{\mathbf{y}_1^*, \mathbf{y}_2^*\} = \{(\mathbf{x}_{1,i}^*, \mathbf{y}_{1,i}^*)\}_{i=1}^{n_1=6}$. The matrix has as many rows as player 1 has equilibria, and as many columns as player 2 has equilibria, being

$$\mathbf{B}_2 = \begin{pmatrix} 3.27905 & 3.35008 & 3.31098 \\ 3.0755 & 3.1093 & 3.00523 \end{pmatrix}.$$

In this game, an equilibrium is computable by linear programming, explicitly stated as primal (P) and dual problem (D) in Appendix B, using the GNU linear programming kit [20]. An equilibrium is found as

$$\boldsymbol{\alpha}_2^* = (1, 0), \quad \text{and} \quad \boldsymbol{\beta}_2^* = (1, 0, 0),$$

so that the final optimum for both players is obtained by evaluating (2), giving

$$\mathbf{x}'_{2,1} = (0.236624, 0.259513, 0.011683, 0.330247, 0.161933) \quad (= \mathbf{x}_1^*)$$

and

$$\mathbf{y}'_{2,1} = (0.11090, 0.13516, 0.22033, 0.12635, 0.23811, 0.16914) \quad (= \mathbf{y}_1^*).$$

From here onwards, the process would continue likewise by enumerating all equilibria that exist in the game $\mathbf{B}_2$, to make up a list of strategies $\mathbf{x}'_{2,1}, \ldots, \mathbf{x}'_{2,n_2}$ and $\mathbf{y}'_{2,1}, \ldots, \mathbf{y}'_{2,n_2}$ to define the game $\mathbf{B}_3$ and so on. Since the process is repetitive, we let our example stop here, with a unique solution obtained for the second coordinate already. Once we are left with a single solution, the iteration may safely stop, since considering further payoff matrices for higher coordinates cannot further narrow down the set of equilibria; it would remain the same optimum over all coordinates $> k$, once the solution is unique at stage k.

4.3 Example 2: Data Download

Let us recap the two distinct settings of a company processing personal data or running a production line. In both cases, we assume that data is being downloaded from redundant servers, some of which may be compromised by an adversary. The settings are different for the two companies in the following way:

– for the *production-oriented* enterprise, Alice will download software or control scripts, neither of which has a particular demand for confidentiality, but must be available and integrity protected, in this order of importance for the two. The overall goals are thus

"Availability" > "Integrity" > "Confidentiality".

– for the *personal data processing* enterprise, it may not matter too much if the data of a particular person is not instantly accessible, but it must remain confidential in all cases, and needs protection from manipulation. The priorities are thus "Confidentiality" > "Integrity" > "Availability".

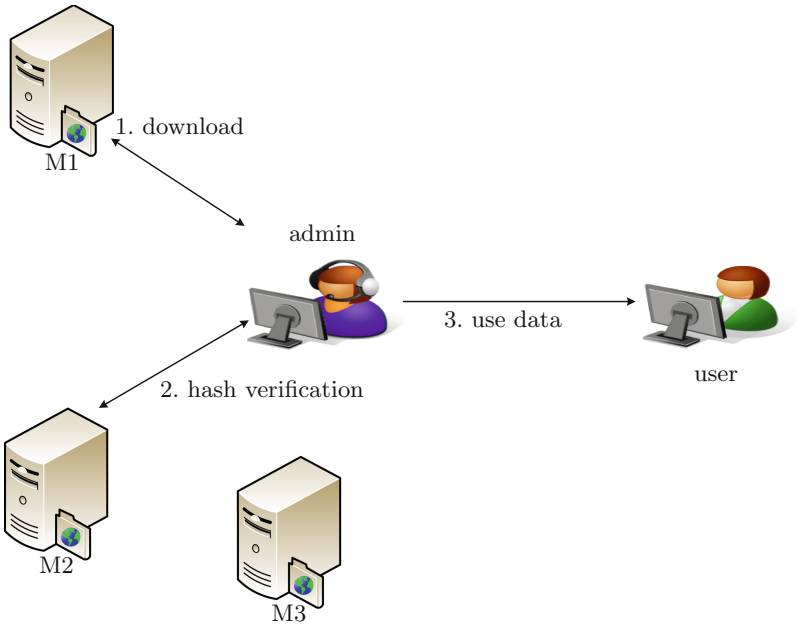

Fig. 2. Example scenario for data download

Towards a specific example, suppose that an administrator has three servers to potentially retrieve data from (where "data" can be a software or personal data), and that the policy prescribes to do a majority vote verification. That is, data retrieved from one server M_i needs to be checked against another redundant server M_j[2]; if the results are consistent, we have a 2-out-of-3 majority pointing to the downloaded data as correct, since the data is apparently the same as if it were when downloaded from the verification server M_j and checked against the other server M_i. If the verification fails, the data could be checked for consistency with a third server.

Let the situation be as depicted in Fig. 2, and consider the following likelihoods for a man-made hacker attack (probabilities p_1, p_2, p_3) or a natural outage of a server. Natural failures are hereby considered individually different, e.g., due to varying traffic

[2] This is indeed the standard idea behind putting cryptographic hash fingerprints on download sites for open-source software, addressing the possibility of a forged installation bundle. The package's fingerprint as put on the website next to the download is for verification against independent other mirrors that offer the same download.

loads, distinct hard- and software configurations, different administrative procedures applying to a server and its mirror, or similar, leading to different probabilities q_1, q_2 and q_3 assumed here:

mirror	M_1	M_2	M_3
probability of being hacked	$p_1 = 0.1$	$p_2 = 0.05$	$p_3 = 0.01$
probability of failure (reliability)	$q_1 = 0.1$	$q_2 = 0.2$	$q_3 = 0.15$

With this, we can set up a payoff structure with the strategies $\{M_1, M_2\}$, $\{M_2, M_3\}$ and $\{M_1, M_3\}$, meaning that both servers are used by the players; for player 1, using $\{M_i, M_j\}$ means download from M_i and verify against M_j. The same strategy for the attacker means that exactly M_i and M_j are being attacked, with success chances as given in the table above.

Assuming stochastic independence, we get the following generic payoff structure, where ℓ_i is a likelihood later to be substituted by either p_i or q_i,

	$\{M_1, M_2\}$	$\{M_1, M_3\}$	$\{M_2, M_3\}$
$\{M_1, M_2\}$	$(1-\ell_1)(1-\ell_2)$	$1-\ell_1$	$1-\ell_2$
$\{M_1, M_3\}$	$1-\ell_1$	$(1-\ell_1)(1-\ell_3)$	$1-\ell_3$
$\{M_2, M_3\}$	$1-\ell_2$	$1-\ell_3$	$(1-\ell_2)(1-\ell_3)$

which, upon substituting the values $\ell_i = p_i$ or $\ell_i = q_i$ values, gives the confidentiality game $\mathbf{A}_{conf}$ and availability game $\mathbf{A}_{avail}$

$$\mathbf{A}_{conf} := \begin{pmatrix} 0.855 & 0.9 & 0.95 \\ 0.9 & 0.891 & 0.99 \\ 0.95 & 0.99 & 0.9405 \end{pmatrix}, \quad \text{and} \quad \mathbf{A}_{avail} = \begin{pmatrix} 0.72 & 0.9 & 0.8 \\ 0.9 & 0.765 & 0.85 \\ 0.8 & 0.85 & 0.68 \end{pmatrix}.$$

Now, the optimization is either w.r.t. the goal priorities "Confidentiality" > "Availability", coming to the payoff vector $(\mathbf{x}^T \cdot \mathbf{A}_{conf} \cdot \mathbf{y}, \mathbf{x}^T \cdot \mathbf{A}_{avail} \cdot \mathbf{y})$, or with the reversed goal priorities "Availability" > "Confidentiality", giving the (reversed) payoff vector $(\mathbf{x}^T \cdot \mathbf{A}_{avail} \cdot \mathbf{y}, \mathbf{x}^T \cdot \mathbf{A}_{conf} \cdot \mathbf{y})$.

Let us compute the results in both cases separately:

1. For confidentiality as the highest goal, we find only a single equilibrium being

$$\mathbf{x}_1^* = (0, 0.09548, 0.90452), \quad \text{and} \quad \mathbf{y}_1^* = (0.49749, 0, 0.50251)$$

with the saddle point value $v_1 = 0.94523$, i.e., an $\approx 94\%$ chance of the data being not compromised (w.r.t. confidentiality).
Since this equilibrium is unique, it carries through to the second coordinate without any further change, giving the 1×1-payoff structure $B_2 = (0.7526)$. This is, at the same time, the best achievable payoff regarding availability, so we have the lex-order optimum being (Pr(Confidentiality), Pr(Availability)) = $(0.94523, 0.7526)$.
2. If availability is the highest-priority goal, we first look for a saddle point on $\mathbf{A}_{avail}$, giving

$$\mathbf{x}_2^* = \mathbf{y}_2^* = (0.40564, 0.55531, 0.03905),$$

with the availability likelihood being the saddle point value $v_2 = 0.82308$, i.e., an $\approx 82\%$ chance for the data to be downloadable.
As before, this equilibrium is unique, and hence no change upon proceeding to further coordinates will happen. The game $\mathbf{B}_2$ is again 1×1 and rewards the (constant) value 0.89537. The lex-order optimum is thus (Pr(Availability), Pr(Confidentiality)) $= (0.82308, 0.89537)$.

Generally, the procedure will make the goal with highest priority matter most, with the multi-criteria optimization subsequently refining the set of equilibria existing for the first goal. This is in contrast to other instances of multi-goal treatment, where the goals may play equally important roles. From the complexity perspective, the enumeration of equilibria per goal can take worst-case exponential time (in the number of strategies), but practically, there may not be a need to compute all equilibria in all cases; the method will never shrink the set towards emptiness, since once the set is singleton at stage k, it will remain unchanged for $k+1, k+2, \ldots, d$. Thus, as long as equilibria remain plausible in the context at hand, the computational burden may be kept in feasible bounds. Nonetheless, this may still call for other refinements like perfect equilibria or others. Imposing such additional constraints on the equilibria per stage is a straightforward matter.

5 Conclusion

We described a simple method to do multi-criteria optimization with goal priorities as optimization over lexicographic orders. As a natural side-effect, our algorithm narrows down a potentially large set of ambiguous Nash equilibria to fewer ones, and therefore is a natural refinement of the general Nash equilibrium in case of multiple criteria. It is, however, fair to admit that this process is in general not guaranteed to establish ultimate uniqueness of the equilibrium. In general, the so-found optimum depends on the specific goal prioritization, reflecting the fact that security strategies are expectedly different depending on the application domain. The method is algorithmically simple, and implemented as open source and freely available (see [2] for a software to enumerate equilibria, and [27] for the source code behind the examples given in this work).

While our method to construct games with desired ambiguous equilibria (see Appendix B) is here used only for the sake of illustrating, it may be of independent interest, e.g., for teaching general game theory to construct examples.

Acknowledgement. The authors would like to thank the anonymous reviewers for valuable and constructive feedback on this work.

A Proof of Proposition 1

Towards a contradiction, suppose there were such a function f then, it obviously cannot be constant, for otherwise, it were meaningless. Thus, there must be some value x for which $f(x,0) \neq f(x,1)$, and the interval $I(x) := [f(x,0), f(x,1)]$ has nonzero width.

Furthermore, any two such intervals $I(x), I(y)$ are disjoint: if there were x, y such that the intervals overlap, then we would have $f(x,0) < f(y,0) < f(x,1) < f(y,1)$, which, since f represents the ordering, entails $(x,0) <_{lex} (y,0) <_{lex} (x,1) <_{lex} (y,1)$,

in which the first $<_{lex}$ implies $x < y$ and the second implies $y < x$, which is not possible at the same time.

Let us pick some particular (arbitrary) x for which $f(x,0) \neq f(x,1)$ in the following. Since f is continuous, so is the function $f(h) := f(x+h,0) - f(x+h,1)$. Our choice of x makes $f(0) > 0$, and this relation holds in an entire compact neighborhood f of 0. The compactness of f implies that f attains a minimum $\varepsilon > 0$ on f.

Each $h \in f$ gives rise to a set $I(x+h)$, whose length is by construction $\geq \varepsilon$. Furthermore, all these uncountably many sets are pairwise disjoint, so that adding up their lengths would add up to infinity.

This is, however, impossible given the fact that this all happens within the unit interval $[0,1]$, whose length is 1. This final contradiction refutes the initial assumption on the existence of a continuous function f to represent the lexicographic order.

B Proof of Lemma 2

Suppose that we have picked a set of vectors $0 \leq \mathbf{x}_1^*, \ldots, \mathbf{x}_{k_1}^* \in \mathbb{R}^n$ for $k_1 < n$, to be equilibrium strategies for player 1, and likewise, let $0 \leq \mathbf{y}_1^*, \ldots, \mathbf{y}_{k_2}^* \in \mathbb{R}^m$ with $k_2 < m$ be a set of chosen equilibria for player 2 in our zero-sum game to be constructed.

Let the matrix $\mathbf{X}$ be such that all $\mathbf{x}_i^* \in N(\mathbf{X})$, when $N(\mathbf{X})$ denotes the null-space of the matrix $\mathbf{X}$. This matrix is directly constructible by taking the singular value decomposition of the matrix whose rows are exactly the desired $\mathbf{x}_i^*$. In defining $\mathbf{X}$ in this way, each (mixed) strategy $\mathbf{x}_i^*$ makes the other player indifferent in its response, since $\mathbf{X} \cdot \mathbf{x}_i^* = 0$.

Analogously, we can construct a matrix $\mathbf{Y}$ whose null-space is spanned by $\{\mathbf{y}_1^*, \ldots, \mathbf{y}_{k_2}^*\}$, thus achieving $(\mathbf{y}_i^*)^T \cdot \mathbf{Y}^T = 0$ for all $i = 1, 2, \ldots, k_2$.

Finally, pick any random matrix $\mathbf{Z}$ with a conformable shape to have the matrix product $\mathbf{A} = \mathbf{X}^T \cdot \mathbf{Z} \cdot \mathbf{Y} \in \mathbb{R}^{n_A \times m_A}$ well-defined[3]. By associativity, $\mathbf{A}$ retains the properties of $\mathbf{X}$ and $\mathbf{Y}$, so that we still have $(\mathbf{x}_i^*)^T \cdot \mathbf{A} = 0$ and $\mathbf{A} \cdot \mathbf{y}_j^* = 0$ for all i, j. Now, take $\mathbf{A}$ as the $(n_A \times m_A)$-payoff matrix in the game. It is well known that we can obtain an equilibrium for a maximizing player by solving the linear program

$$(P) \quad \min \overbrace{\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}^T}^{=:\mathbf{c}^T} \cdot \overbrace{\begin{pmatrix} v \\ \boldsymbol{\mu} \end{pmatrix}}^{=:\mathbf{x}} \quad \text{s.t.} \quad \overbrace{\left(\begin{array}{c|c} -\mathbf{A}^T & \mathbf{1} \\ \hline \mathbf{1} & 0 \end{array}\right)}^{=:\mathbf{B}} \cdot \begin{pmatrix} \boldsymbol{\mu} \\ v \end{pmatrix} \begin{matrix} \geq \\ = \end{matrix} \overbrace{\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}}^{=:\mathbf{b}}$$
$$\text{and } \mu_i \geq 0 \text{ for all } i = 1, \ldots, n_A$$

in which the conditions given here in matrix notation evaluate to the minimization of the saddle-point value v, upper-bounding the payoff obtained from the matrix $\mathbf{A}$ by playing the i-th row with probability μ_i, i.e., $\boldsymbol{\mu}^T \cdot \mathbf{A} \cdot \mathbf{e}_i \leq v$ for all i when $\mathbf{e}_i$ is the i-th unit vector. The lower block in the product $\mathbf{B} \cdot \boldsymbol{\mu} = 1$ is then just the condition that the sum of all μ_i should equal 1.

Now, look at the dual program for the other player being

[3] Here, n_A and m_A are new variables to describe the shape; their values depend on how many equilibria we want to enforce, and whether these are linearly dependent. This determines the dimension of the nullspaces, which sets the values for n_A, m_A.

$$(D) \quad \max \mathbf{b}^T \cdot \overbrace{\begin{pmatrix} \boldsymbol{\nu} \\ v \end{pmatrix}}^{=:\mathbf{y}}, \text{ s.t. } \quad \mathbf{y}^T \cdot \left(\begin{array}{c|c} -\mathbf{A} & \mathbf{1} \\ \hline \mathbf{1} & 0 \end{array}\right) \leq \overbrace{\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}^T}^{=\mathbf{c}^T}$$
$$\text{and } \nu_i \geq 0 \text{ for all } i = 1, \ldots, m_A.$$

The point of our construction is that in the matrix products $\mathbf{B} \cdot \mathbf{x}$ and $\mathbf{y}^T \cdot \mathbf{B}$, the following happens:

- in (P), we get the expression $-\mathbf{A} \cdot \boldsymbol{\mu} = 0$ for every $\boldsymbol{\mu} \in \left\{\mathbf{x}_1^*, \ldots, \mathbf{x}_{k_1}^*\right\}$ or linear combinations thereof. Thus, the constraint ≥ 0 on this row is satisfied with equality if $v = 0$.
- Likewise, evaluating the constraints in (D) creates the inner term $-\boldsymbol{\nu}^T\mathbf{A} = 0$ for all $\boldsymbol{\nu} \in \left\{\mathbf{y}_1^*, \ldots, \mathbf{y}_{k_2}^*\right\}$ (and any linear combinations thereof). Thus, the dual constraint ≤ 0 is also satisfied with equality.

Now, an equilibrium $(\boldsymbol{\mu}, \boldsymbol{\nu})$ in the zero-sum game $\mathbf{A}$ is characterized by $\boldsymbol{\mu}$ being an optimum in (P) and $\boldsymbol{\nu}$ being an optimum in (D), and by strong duality, this happens if both are feasible for the respective constraints, and the respective optima are equal. Putting these conditions together, we find $(\boldsymbol{\mu}, \boldsymbol{\nu})$ to be an equilibrium if and only if the following conditions are all satisfied:

1. $\mathbf{B} \cdot \mathbf{x} \geq \mathbf{b}$, i.e., feasibility for (P): this holds by construction, even with equality in all rows.
2. $\mathbf{y}^T \cdot \mathbf{B} \leq \mathbf{c}^T$, i.e., feasibility for (D): this also holds by construction with equality.
3. $\mathbf{c}^T \cdot \mathbf{x} \leq \mathbf{y}^T \cdot \mathbf{b}$, which can only hold if the two values are equal. But we constructed all equilibria to create the value $v = \boldsymbol{\mu}^T \cdot \mathbf{A} = \mathbf{A} \cdot \boldsymbol{\nu} = 0$, so equality holds here too.

Thus, all pairs $(\mathbf{x}_i^*, \mathbf{y}_j^*)$ are equilibria of our matrix game $\mathbf{A}$.

Remark 4. Switching the players's directions between minimization and maximization, as well as changing the saddle-point value from $v = 0$ into some chosen $v' \neq 0$ is easy by a proper affine transformation $\mathbf{A}' \mapsto \pm\mathbf{A} + v'$.

It is easy to see that the so-constructed game has the designed equilibria, but also many others, since not only the convex-combination, but any linear combination of the chosen vectors will be in the nullspace. Let us take a short break here to numerically illustrate the intermediate construction.

B.1 Example

We implemented the algorithm in GNU Octave [7], with sources are available from [27]: for the example, let us fix the strategy spaces for player 1 and 2 to have five, resp. six, actions. Furthermore, let us pick two equilibria for player 1, and three equilibria (all mixed for both players) at random, sampling uniformly random values from $[0, 1]$, and normalizing the vector to unit sum. For a random instance, these equilibria were

$$\begin{array}{lccccc} \text{strategy} & 1 & 2 & 3 & 4 & 5 \\ \mathbf{x}_1^* = & (0.236624, & 0.259513, & 0.0116831, & 0.330247, & 0.161933) \\ \mathbf{x}_2^* = & (0.26241, & 0.117688, & 0.21289, & 0.284324, & 0.122688) \end{array} \tag{5}$$

and

$$\begin{array}{rrrrrrr} \text{strategy} & 1 & 2 & 3 & 4 & 5 & 6 \\ \mathbf{y}_1^* = & (0.110901, & 0.13516, & 0.220331, & 0.126352, & 0.238114, & 0.169142) \\ \mathbf{y}_2^* = & (0.1328, & 0.45488, & 0.0802542, & 0.040265, & 0.236992, & 0.0548095) \\ \mathbf{y}_3^* = & (0.148226, & 0.0651162, & 0.0286501, & 0.31977, & 0.375297, & 0.0629404) \end{array} \tag{6}$$

With these values, the matrices $\mathbf{X}_1$ and $\mathbf{Y}_1$ from the previous section are easily found using the `null` function in Octave (that internally computes a singular value decomposition), to find

$$\mathbf{X} = \begin{pmatrix} -0.490976 & 0.689481 & 0.497453 & -0.183545 & -0.0490909 \\ -0.617055 & -0.273987 & 0.0243345 & 0.724245 & -0.138025 \\ -0.263877 & -0.192006 & 0.0534469 & -0.120881 & 0.935967 \end{pmatrix}$$

and

$$\mathbf{Y} = \begin{pmatrix} -0.554165 & 0.271458 & 0.237081 & 0.640731 & -0.372757 & -0.116278 \\ -0.72375 & -0.0592711 & 0.0703592 & -0.314963 & 0.585821 & -0.159171 \\ -0.278293 & 0.0898957 & -0.51704 & 0.0385987 & -0.0337394 & 0.802816 \end{pmatrix}$$

and with a randomly chosen matrix $\mathbf{Z}$, we find the payoff structure

$$\mathbf{A} = \begin{pmatrix} 0.955986 & -0.272557 & 0.316327 & -0.405844 & 0.102397 & -0.662056 \\ 0.0454297 & -0.0580642 & -0.178636 & -0.187195 & 0.130912 & 0.204854 \\ -0.298982 & 0.05127 & -0.17908 & 0.0170827 & 0.0593234 & 0.292065 \\ -0.436331 & 0.223209 & -0.113101 & 0.453724 & -0.301461 & 0.340507 \\ -0.558309 & 0.0324137 & 0.0676309 & -0.0335246 & 0.251095 & -0.076376 \end{pmatrix}$$

which is exactly the matrix (4) used in Sect. 4.2.

Solving the linear programs (P) and (D), we find the following mixed equilibrium for the game $\mathbf{A}$:

$$\begin{array}{r} \mathbf{x}^* = (0.28381,\ 0,\ 0.37985,\ 0.24622,\ 0.09012), \\ \text{and } \mathbf{y}^* = (0.06237,\ 0,\ 0.48687,\ 0,\ \ 0.11092,\ \ 0.33984) \end{array}$$

which is not among the equilibria listed in (5) or (6). However, it is a simple matter to put the vectors $\mathbf{x}^*, \mathbf{y}^*$ into span $\{\mathbf{x}_1^*, \mathbf{x}_2^*\}$ and span $\{\mathbf{y}_1^*, \mathbf{y}_2^*, \mathbf{y}_3^*\}$ via

$$\begin{aligned} \mathbf{x}^* &= -0.8298 \cdot \mathbf{x}_1^* + 1.8298 \cdot \mathbf{x}_2^* \quad \text{and} \\ \mathbf{y}^* &= 2.55931 \cdot \mathbf{y}_1^* - 0.62699 \cdot \mathbf{y}_2^* - 0.93232 \cdot \mathbf{y}_3^*. \end{aligned}$$

B.2 Restricting the Equilibria to the Desired Set

Now, to complete the proof of Lemma 2, it remains to modify the game so that no solution outside the convex hull of our chosen equilibrium points is possible.

The simplest method of to exclude equilibria outside the desired set is adding a penalty term to the goal function that vanishes on the desired set of optima. An obvious choice is letting δ be a distance measure, such as

$$\delta(M, \mathbf{y}) := \inf \{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x} \in M\},$$

for a set $M \subset \mathbb{R}^n$ and a point $\mathbf{x} \in \mathbb{R}^n$, using any norm $\|\cdot\|$ on $\mathbb{R}^n$. Put E_1 as the set of desired equilibria of player 1, and let E_2 be the set of desired equilibria for player 2.

Then, any action outside E_1 shall decrease the revenue for player 1, while any deviation to the exterior of E_2 shall increase the payoff for player 1, so that there is an incentive for player 1 to stay within the desired set of equilibria, and another incentive for player 2 to do the same (zero-sum game). Thus, we change the expected payoff function from $u(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{y}$ into

$$u(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{y} + \delta(\mathbf{y}, \Delta(E_2)) - \delta(\mathbf{x}, \Delta(E_1)). \tag{7}$$

This function is no longer linear, and hence the optimization problems (P) and (D) no longer apply as such. But strong duality still holds, since Slater's condition [3] is satisfied: note that the change in the payoff functional manifests itself in the primal problem (P) as the inequality $u(\mathbf{x}, \mathbf{e}_i) \leq v$ for e_i being the i-th unit vector running over all strategies of the second player (the likewise converse inequality would arise in the dual problem (D)). This is due to the fact that we still do a min-max optimization $\max_{\mathbf{x}} \min_{\mathbf{y}} u(\mathbf{x}, \mathbf{y})$, where the inner optimization is easy because we have only a finite number of choices (or any convex combination of them), making $\min_{\mathbf{y}} u(\mathbf{x}, \mathbf{y}) = \min_{i=1,\ldots,m} u(\mathbf{x}, \mathbf{e}_i)$.

More formally, let $B^o := \left\{(\mathbf{x}, \mathbf{y}) : \|\mathbf{x}\|_1 < 1, \|\mathbf{y}\|_1 < 1\right\}$ be the interior of the unit balls defining the feasible set of probability distributions, i.e., mixed strategies for both players. Moreover, let E be the convex hull of all equilibria that are admissible by design. For Slater's condition, we look for an inner point that satisfies the constraints with strict inequality. Note that the affine hull $\mathrm{aff}(E)$ is unbounded, and therefore extends over the bounded convex set E. Moreover, by construction of the penalized utility (7), we have nonzero contributions of the distance terms outside E. Now, distinguish two cases:

Case 1: If $B^o \setminus E = \emptyset$, then all probability distributions are admissible equilibria by design, and there is nothing to restrict (the penalty terms never become active, and always add zero to the overall utility).
Case 2: Otherwise, the affine hull $\mathrm{aff}(E)$ must contain a point $(\mathbf{x}_0, \mathbf{y}_0) \in (B^o \cap \mathrm{aff(E)}) \setminus E$ outside the admissible set E but in the interior of the unit ball. Look at the terms that sum up to the penalized utility:

$$\begin{aligned} \mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{y} = 0, &\quad \text{because } (\mathbf{x}_0, \mathbf{y}_0) \text{ are still in the nullspace of } \mathbf{A}; \\ \delta(\mathbf{x}_0, \Delta(E_1)) > 0, &\quad \text{because we are outside } \Delta(E_1) \subset E; \\ \delta(\mathbf{y}_0, \Delta(E_2)) > 0, &\quad \text{because we are outside } \Delta(E_2) \subset E. \end{aligned}$$

So, whenever $\delta(\mathbf{x}, \Delta(E_1)) \neq \delta(\mathbf{y}, \Delta(E_2))$, we are done since we have a nonzero utility for the respective player and hence a Slater point (for one of the players, i.e., either the primal or the dual problem). Otherwise, if $\delta(\mathbf{x}, \Delta(E_1)) = \delta(\mathbf{y}, \Delta(E_2))$, we can slightly move $\mathbf{x}$ farer away from E, since B^o is an open set. This move from $\mathbf{x}_0$ to $\mathbf{x}_0'$ with $\delta(\mathbf{x}_0, \Delta(E_1)) \neq \delta(\mathbf{x}_0', \Delta(E_1))$ again makes the penalty term overall negative, and we have $(\mathbf{x}_0', \mathbf{y}_0)$ as the sought Slater point. The existence of a Slater point certifies strong duality to hold for the optimization problems. The design of the respective utilities (having opposite signs since we are playing a zero-sum regime) then assures that all feasible solutions must be inside the set $\Delta(E_1) \times \Delta(E_2)$. By strong duality, no solution outside this region is possible, and Lemma 2 is proven.

References

1. Avis, D., Rosenberg, G., Savani, R., Stengel, B.: Enumeration of nash equilibria for two-player games. Econ. Theory **42**, 9–37 (2010)

2. Avis, D.: lrs home page (2020). http://cgm.cs.mcgill.ca/~avis/C/lrs.html
3. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Brzuska, C., et al.: Security of sanitizable signatures revisited. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 317–336. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00468-1_18
5. Cococcioni, M., Pappalardo, M., Sergeyev, Y.D.: Lexicographic multi-objective linear programming using grossone methodology: theory and algorithm. Appl. Math. Comput. **318**, 298–311 (2018). https://doi.org/10.1016/j.amc.2017.05.058, https://linkinghub.elsevier.com/retrieve/pii/S0096300317303703
6. Davidson, C.C., Andel, T.R.: Feasibility of applying Moving Target Defensive Techniques in a SCADA System. ACPI, Boston University, Boston (2016). https://doi.org/10.13140/RG.2.1.5189.5441, http://rgdoi.net/10.13140/RG.2.1.5189.5441
7. Eaton, J.W., Bateman, D., Hauberg, S., Wehbring, R.: GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations (2020). https://www.gnu.org/software/octave/doc/v5.2.0/
8. Ehrgott, M.: Discrete decision problems, multiple criteria optimization classes and lexicographic max-ordering. In: Fandel, G., Trockel, W., Stewart, T.J., van den Honert, R.C. (eds.) Trends in Multicriteria Decision Making. Lecture Notes in Economics and Mathematical Systems, vol. 465, pp. 31–44. Springer, Heidelberg (1998). https://doi.org/10.1007/978-3-642-45772-2_3
9. Ehrgott, M.: A Characterization of Lexicographic Max-Ordering Solutions (1999). https://kluedo.ub.uni-kl.de/frontdoor/index/index/docId/484
10. Fishburn, P.C.: Exceptional paper-lexicographic orders, utilities and decision rules: a survey. Manag. Sci. **20**(11), 1442–1471 (1974). https://doi.org/10.1287/mnsc.20.11.1442, http://pubsonline.informs.org/doi/abs/10.1287/mnsc.20.11.1442
11. Glicksberg, I.L.: A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points, vol. 3, pp. 170–174 (1952). http://dx.doi.org/10.2307/2032478
12. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. Eur. J. Oper. Res. **89**(3), 445–456 (1996). https://doi.org/10.1016/0377-2217(95)00176-X, https://linkinghub.elsevier.com/retrieve/pii/037722179500176X
13. Harsanyi, J.C.: Oddness of the number of equilibrium points: a new proof. Int. J. Game Theory **2**(1), 235–250 (1973). https://doi.org/10.1007/BF01737572, http://link.springer.com/10.1007/BF01737572
14. Herrmann, A., Morali, A., Etalle, S., Wieringa, R.: Risk and business goal based security requirement and countermeasure prioritization. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) BIR 2011. LNBIP, vol. 106, pp. 64–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29231-6_6
15. Isermann, H.: Linear lexicographic optimization. Oper. Res. Spektrum **4**(4), 223–228 (1982). https://doi.org/10.1007/BF01782758, http://link.springer.com/10.1007/BF01782758
16. Karnin, E.D., Greene, J.W., Hellman, M.E.: On secret sharing systems. IEEE Trans. Inf. Theory IT **29**(1), 35–41 (1983). http://ieeexplore.ieee.org/document/1056621/
17. Konnov, I.: On lexicographic vector equilibrium problems. J. Optim. Theory Appl. **118**(3), 681–688 (2003). https://doi.org/10.1023/B:JOTA.0000004877.39408.80
18. Kotler, R.: How to prioritize IT security projects (2020). https://www.helpnetsecurity.com/2020/01/30/prioritize-it-security-projects/, library Catalog: www.helpnetsecurity.com

19. Lozovanu, D., Solomon, D., Zelikovsky, A.: Multiobjective games and determining pareto-nash equilibria. Buletinul Academiei de Stiinte a Republicii Moldova Matematica **3**(49), 115–122 (2005)
20. Makhorin, A.: GLPK - GNU Project - Free Software Foundation (FSF) (2012) https://www.gnu.org/software/glpk/
21. McElice, R.J., Sarwate, D.V.: On sharing secrets and reed-solomon codes. Commun. ACM **24**(9), 583–584 (1981)
22. Ogryczak, W.: Lexicographic max-min optimization for efficient and fair bandwidth allocation. In: International network optimization conference (INOC) (2007)
23. Ogryczak, W., Śliwiński, T.: On direct methods for lexicographic min-max optimization. In: Gavrilova, M., et al. (eds.) ICCSA 2006. LNCS, vol. 3982, pp. 802–811. Springer, Heidelberg (2006). https://doi.org/10.1007/11751595_85
24. Park, K.-Y., Yoo, S.-G., Kim, J.: Security requirements prioritization based on threat modeling and valuation graph. In: Lee, G., Howard, D., Ślęzak, D. (eds.) ICHIT 2011. CCIS, vol. 206, pp. 142–152. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24106-2_19
25. Perkowski, M.: Why You Need To Prioritize Cyber Risks (2018). https://www.securityroundtable.org/everything-cant-be-urgent-why-you-need-to-prioritize-cyber-risks/
26. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: STOC 1989, pp. 73–85. ACM (1989). http://dx.doi.org/10.1145/73007.73014
27. Rass, S., Wiegele, A., König, S.: Source code to run the examples in Appendix B.1 available from https://www.syssec.at/de/publikationen/description/games-over-lex-orders. online (2020)
28. Rass, S., Schauer, S.: Game Theory for Security and Risk Management: From Theory to Practice. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-75268-6
29. Rios Insua, D., Couce-Vieira, A., Rubio, J.A., Pieters, W., Labunets, K., Rasines, D.G.: An Adversarial Risk Analysis Framework for Cybersecurity. Risk Analysis (2019). https://doi.org/10.1111/risa.13331, https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.13331
30. Rios Insua, D., Rios, J., Banks, D.: Adversarial Risk Analysis. J. Am. Stat. Assoc. **104**(486), 841–854 (2009). http://pubs.amstat.org/doi/abs/10.1198/jasa.2009.0155
31. Ross, T., Booker, J.M., Parkinson, W.J.: Fuzzy logic and probability applications: bridging the gap. In: ASA SIAM (2002)
32. Rothschild, C., McLay, L., Guikema, S.: Adversarial risk analysis with incomplete information: a level-k approach. Risk Anal. **32**(7), 1219–1231 (2012). https://doi.org/10.1111/j.1539-6924.2011.01701.x, http://doi.wiley.com/10.1111/j.1539-6924.2011.01701.x
33. Stanimirovic, I.P.: Compendious lexicographic method for multi-objective optimization. Facta Universitatis **27**(1), 55–66 (2012), https://pdfs.semanticscholar.org/25c6/8b5d4d9adfef3684dddbf0096a38fcbd1923.pdf?_ga=2.42418213.1664777629.1591959638-2095801372.1591959638
34. The Recorded Future Team: You Can't Do Everything: The Importance of Prioritization in Security (2018). https://www.recordedfuture.com/vulnerability-threat-prioritization/, library Catalog: www.recordedfuture.com Section: Cyber Threat Intelligence

35. Tompa, M., Woll, H.: How to share a secret with cheaters. J. Cryptol. **1**(3), 133–138 (1989). https://doi.org/10.1007/BF02252871, http://link.springer.com/10.1007/BF02252871
36. Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: attacks and defenses for deep learning. arXiv:1712.07107 [cs, stat] (2018), http://arxiv.org/abs/1712.07107, arXiv: 1712.07107