Data Integrity and Availability Verification Game in Untrusted Cloud Storage

Brahim Djebaili[1], Christophe Kiennert[1], Jean Leneutre[1], and Lin Chen[2]

[1] Télécom ParisTech, 46 rue Barrault, 75013 Paris, France
[2] Université Paris Sud, 15 rue Georges Clémenceau, 91400 Orsay, France

Abstract. The recent trends towards outsourcing data to the Cloud as well as various concerns regarding data integrity and availability created an increasing interest in enabling secure Cloud data-centers. Many schemes addressing data integrity issues and complying with various requirements came to place: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data. Yet, a critical question remains: how to use these schemes efficiently, i.e. how often should data be verified. Constantly checking is a clear waste of resources but only checking at times increases risks. This paper attempts to resolve this thorny issue by formulating the data integrity check problem as a non-cooperative game and by performing an in-depth analysis on the Nash Equilibrium and the engineering implications behind. Based on our game theoretical analysis, the course of action was to anticipate the Cloud provider's behavior; we then derive the minimum verification resource requirement, and the optimal strategy of the verifier. Finally, our game theoretical model is validated by showing correctness of the analytical results via simulation on a case study.

Keywords: Cloud computing, Game theory, Data integrity, Data availability, Nash equilibrium.

1 Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [11].

However, all the benefits brought by the cloud, such as lower costs and ease of use, come with a tradeoff. Users will have to entrust their data to a potentially untrustworthy cloud provider (CP). As a result, cloud security has become an important issue for both industry and academia [2].

One important security problem with cloud data storage is data integrity and availability, since the client lacks control over his data, entailing difficulties in ensuring that data stored in the Cloud are indeed left intact. Moreover, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide data errors from the clients for his own benefit. On top of that,

R. Poovendran and W. Saad (Eds.): GameSec 2014, LNCS 8840, pp. 287–306, 2014.

for both money and storage space saving purposes, the service provider might deliberately delete rarely accessed data files that belong to an ordinary client.

In order to solve these problems, many verification schemes are provided in the literature [10]. In all these works, it has taken major efforts to design solutions that meet various requirements: low time complexity, stateless verification, unbounded use of queries and retrievability of data, etc. In spite of these numerous features, knowing how to use these schemes efficiently remains a major issue. Indeed, it would be a waste of time and resources if the verifier checks the data all the time while the CP is being honest. On the other hand, it would be risky if the verifier checks the data just a few times while the CP is being dishonest. The best approach for the verifier is to find the right frequency of verification for the minimum cost, while maintaining accuracy and consistency of data. The natural way to achieve this last condition is to use *game theory*, by modeling the process of data verification as a game that contains two players, the defender (verifier) and the attacker (CP).

Considering the role of the verifier, all the proposed schemes fall into two categories: private verification, in which the client performs the auditing operation himself, and public verification, that consists in using a third party auditor (TPA). In this paper, we focus on the latter, because in many cases, clients do not know how to check data integrity, nor do they know which protocol they should use. Moreover, a client who owns a considerable amount of outsourced data (like a company) will have no incentive to check his data, as this process requires considerable resources and time.

In such an environment, the major questions are: What is the expected behavior of a rational attacker (CP)? What is the optimal strategy of the defender (TPA)?

In this paper, we answer these questions by developing a non-cooperative game model of Cloud storage verification problem, analyzing the resulting equilibria, investigating the engineering implications behind the analytical results, and then deriving the optimal strategy for the defender. It is worth noting that the different cases taken into account in this work represent realistic situations, in which a client expects a specific service level from the TPA as stated in his contract with the TPA, which can be seen as an *Audit Level Agreement*.

Our main contributions can be summarized as follows:

1) We provide a game theoretical framework of cloud storage verification, by analyzing as a first model the case of deterministic verification. Then, as extensions, we study the case of the Leader/Follower game (Stackelberg game) in the second model, and probabilistic verification in the third one.

2) For each model, we derive the expected behavior of a rational attacker, the minimum verification resource requirements of the defender, as well as his optimal strategy in terms of resource allocation.

The remainder of the paper is organized as follows: In Section 2, we describe the technical background on which our work is based on. In Section 3, we study the Nash equilibrium (NE) of the Cloud storage game for deterministic verification. In Section 4, we explore several variants and extensions of the game, by

analyzing the case of the Stackelberg game, and the case of probabilistic verification. Section 5 provides numerical results of the game theoretical framework. Finally, our concluding remarks are given in Section 6.

2 Technical Background

2.1 Integrity Verification Schemes

In recent years, a considerable amount of data integrity schemes were proposed by different researchers, and have been gradually adapted to specific use cases such as outsourced databases and Cloud Computing. Among these schemes, Provable Data Possession (PDP) for ensuring possession of data, and Proof of Retrievability (POR) for data possession and retrievability are the two main directions explored by researchers.

The main idea of PDP is that a data owner generates some metadata information for a data file to be used later for verification purposes. Many extensions of this scheme managed to decrease the communication cost and complexity, as well as to allow dynamic operations on data such as insertion, modification, or deletion. Moreover, [18] and [16] proposed PDP schemes fitting requirements specific to Cloud Computing.

The POR scheme is considered as a complementary approach to PDP. [9] was among the first papers to consider formal models for POR schemes. In this scheme, disguised blocks (called sentinels) are embedded into the data before outsourcing. The verifier checks randomly picked sentinels which would be influenced with a certain probability if data are corrupted. [10] gives a detailed survey of the contributions of numerous extensions of the PDP and POR schemes.

The aforementioned schemes primarily focus on a single data file copy. Yet, other schemes, such as [6], allow the verifier to check multiple copies of a data file over multiple Cloud servers.

2.2 Approaches Related to Game Theory

Several works handle cloud-related problems using game theory. Most focus on solutions such as resource allocation and management [8] or Cloud service negotiation [17], while few papers addressed the problem of Cloud security [12,13]. [12] addressed Cloud integrity issues by proposing a model where a client checks the correctness of calculations made on the data by the CP. They considered the case where for two CPs, the client sends a query to one of the two servers chosen randomly, and with a fixed probability, he sends the query to the other server as well.

Nix and Kantarcioglu also proposed in [13] to study the case of querying one single cloud provider, since checking data at multiple CPs is prohibitively expensive. [12,13] focused on checking that the queries sent to the CP are being computed correctly, under the condition that the stored data is intact. On a side note, they did not mention which type of verification protocol (deterministic

or probabilistic) they used. Besides the Cloud, game theory has already been applied to study network security [7] [1], intrusion detection [5], Botnet defense [4], etc. The work presented in this paper was actually strongly inspired by [5].

3 Untrusted Cloud Storage Game for Deterministic Verification

As a first step, we considered a basic model in which the data integrity verification protocol is deterministic and always returns correct information. The main problem of deterministic verification schemes is the fact that they are computationally expensive, since the TPA performs the verification process on the entire data. After solving this game and finding its Nash Equilibrium (NE), which describes the optimal strategies of both players from which neither of them has incentive to deviate unilaterally, we will progressively refine this model by taking more realistic hypotheses into account.

3.1 Game Features

- Players: The game features two players, the auditor (TPA: third party auditor) and the outsourced server (CP: Cloud provider).

- Information: The CP stores the client's data $D = \{D_1, D_2, ..., D_N\}$, with different importances and sizes. We consider that the TPA checks the data by using a deterministic scheme guaranteeing a probability of detecting data modification or deletion equal to 1.

- Actions: We consider *mixed strategies* where a probability is assigned to each strategy of each player. Thus, for each data D_i, the auditor may choose to check its integrity and availability with probability t_i that stems from a probability distribution $t = \{t_1, t_2, ..., t_N\}$. On the other side, the CP can modify or delete data D_i with probability p_i steming from a probability distribution $p = \{p_1, p_2, ..., p_N\}$. Both TPA and CP have resource constraints respectively designated by $T \leq 1$ and $P \leq 1$.

- Payoffs: The two TPA possible actions are *Check* and *Not Check*. Meanwhile, the CP may *Modify/Delete* a data or not, hence possibly leading to *Corrupted/Unavailable data*.

If the corrupted or unavailable data D_i is not checked, then the CP gains S_i, which represents the size of the data, with $S_1 \geq S_2 \geq ... \geq S_N$, while the TPA loses data value and importance designated by F_i. If the TPA decides not to verify, and the CP has the correct data, then both players will neither lose nor gain anything.

Table 1. Cloud Storage Game with Deterministic Verification

CP \ TPA	Check	Not check
Correct/Available data	0 , $-C^tS_i - C^sS_i$	0 , 0
Corrupted/Unavailable data	$-C^sS_i - S_i$, $-C^tS_i + F_i$	S_i , $-F_i$

Let C^tS_i be the cost of the verification process by the TPA, and C^sS_i be the cost of executing the verification query by the Cloud Provider. Both costs are proportional to the size of data D_i.

If the TPA verifies the data whereas the CP has the correct data, we then consider that the TPA should pay the cost of CP verification process C^sS_i, since the data are intact. However, when the CP chose to modify or delete the data, the TPA will gain F_i, which is the the importance of data D_i, minus the verification cost C^tS_i, while the CP will lose S_i, minus the cost of verification C^sS_i. Table 1 illustrates the matrix payoff of both players (CP/TPA) in the strategic form.

The overall payoffs of the TPA (U_t) and the CP (U_p) are defined as follows:

$$U_t(t,p) = \sum_{i=1}^{N} t_i[p_i(2F_i + C^sS_i) - (C^tS_i + C^sS_i)] - \sum_{i=1}^{N} p_iF_i$$

$$U_p(t,p) = \sum_{i=1}^{N} p_iS_i[1 - t_i(2 + C^s)]$$

We finally define the Cloud storage verification game G.

Definition 1: the two players Cloud storage verification game G is defined as:

Players: Attacker (CP), Verifier (TPA).

Strategy type: Mixed strategy.

Strategy set: Attacker:

$$W_P = \left\{ p : p \in [0,P]^N, \sum_{i=1}^{N} p_i \leq P \right\}$$

Verifier:

$$W_T = \left\{ t : t \in [0,T]^N, \sum_{i=1}^{N} t_i \leq T \right\}$$

Payoff: U_p for attacker, U_t for verifier.

Game rule: The attacker/verifier selects his strategy $p/t \in W_P/W_T$ to maximize U_p/U_t.

3.2 Solving the Game

For non-cooperative games like ours, the most essential solution concept is the Nash Equilibrium (NE), which can be considered as the optimal agreement between the players, i.e. an equilibrium in which no player has any incentive to unilaterally deviate from his current strategy in order to maximize his payoff.

1) Data Distribution

Since the attacker has limited attack resources, a relevant approach consists in determining if a rational attacker will target any data, or if he will tend to focus on specific data only. This question will be studied before starting the NE analysis.

First, we introduce two sets that will be of use to clarify data distribution: the attractive set D_A, and the unattractive set D_U. In order to do so, we will introduce the notations $\mathcal{N} = \{1, ..., N\}$, $\mathcal{N}_A = \{i \in \mathcal{N} / D_i \in D_A\}$, and $\mathcal{N}_U = \{i \in \mathcal{N} / D_i \in D_U\}$.

Definition 2: The two sets D_A and D_U are defined as follows:

We set $C = \dfrac{|\mathcal{N}_A| \left(\frac{1}{2+C^s}\right) - T}{\sum_{j \in \mathcal{N}_A} \left(\frac{1}{2S_j + C^s S_j}\right)}$

$$\begin{cases} S_i > C, & \forall i \in \mathcal{N}_A \\ S_i < C, & \forall i \in \mathcal{N}_U \end{cases}$$

where $|\mathcal{N}_A|$ is the number of data contained in $\mathcal{N}_A$. The case where $S_i = C$ does not need to be taken into account, since it happens with very low probability and since these values rely on estimations. Therefore, should this case happen, replacing S_i with a slightly different estimation $S_i + \epsilon$ or $S_i - \epsilon$ would be enough to solve the situation.

Lemma 1: Given a Cloud provider that stores N Data, $\mathcal{N}_A$ is uniquely determined and consists of N_S data with the biggest sizes, such that:

1) if $S_N > \dfrac{N\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{N} \left(\frac{1}{2S_j + C^s S_j}\right)}$, then $N_S = N$.

2) if $S_N < \dfrac{N\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{N} \left(\frac{1}{2S_j + C^s S_j}\right)}$, N_S is determined as follows:

$$\begin{cases} S_{N_S} > \dfrac{N_S\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{N_S} \left(\frac{1}{2S_j + C^s S_j}\right)} \\ S_{N_S+1} < \dfrac{N_S\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{N_S} \left(\frac{1}{2S_j + C^s S_j}\right)} \end{cases}$$

Proof: See Appendix I.

Now we will study the implication of data distribution on the players' decisions.

Theorem 1: A rational attacker has no incentive to attack any data $D_i \in D_U$.

Proof: See Appendix II.

The theorem shows that the attacker only needs to attack data that belong to D_A in order to maximize his payoff. From this point, the defender has no incentive to verify data that will not be attacked. The meaning of the theorem is to assert the existence of data that are too small to be worth attacking to free significant space. As a consequence, it would be a waste of resource for the TPA

to verify the integrity of such data.

Guideline 1: A rational defender has only to verify the integrity and the availability of data in D_A.

2) ***NE Analysis***

Definition 3: A strategy profile (p^*, q^*) is a Nash Equilibrium of the Cloud storage verification game G, when both players (CP and TPA) cannot improve their payoff by unilaterally deviating from their current strategy.

As G is a two-player game with mixed strategies, it admits at least one NE, according to Theorem 1 in [14]. Let (t^*, p^*) denote the NE, it holds that:

$$0 \leq p_i^*(2F_i + C^s S_i) - (C^t S_i + C^s S_i) = p_j^*(2F_j + C^s S_i) - (C^t S_i + C^s S_i) \geq p_k^*(2F_k + C^s S_i) - (C^t S_i + C^s S_i) \; \forall i,j,k \in \mathcal{N}, t_i^*, t_j^* > 0, t_k^* = 0 \quad (1)$$

Equation (1) can be shown by noticing the TPA payoff function. Indeed, if the TPA gain when verifying D_k is lower than when verifying D_i, then in order to maximize his payoff, the TPA will not have incentive to verify D_k and will set $t_k = 0$. The same thing remains valid for the CP, and by noticing his payoff function, it holds that:

$$0 \leq S_i(1 - 2t_i^*) - t_i^* C^s S_i = S_j(1 - 2t_j^*) - t_j^* C^s S_i \geq S_k(1 - 2t_k) - t_k C^s S_i \; \forall i,j,k \in \mathcal{N}, p_i^*, p_j^* > 0, p_k^* = 0 \quad (2)$$

These two equations allow us to find the NE, which we study in two different cases according to the players resource constraints. The NE is hence defined in the following cases:

Case 1: $\sum_{i \in \mathcal{N}} t_i^* = T$ and $\sum_{i \in \mathcal{N}} p_i^* = P$:

In this case, both TPA and CP use all their resources in order to verify/attack data. The game can be seen as a resource allocation problem, in which each player seeks to choose the most profitable strategy.

By combining (1) and (2), we get the NE displayed hereby:

$$t_i^* = \begin{cases} \dfrac{T - \dfrac{N_S}{2 + C^s} + S_i \sum_{j=1}^{N_S} \left(\dfrac{1}{2S_j + C^s S_j} \right)}{(2S_i + C^s S_i) \sum_{j=1}^{N_S} \left(\dfrac{1}{2S_j + C^s S_j} \right)}, & i \in \mathcal{N}_A \\ 0, & i \in \mathcal{N}_U \end{cases}$$

$$p_i^* = \begin{cases} \dfrac{P - \sum_{j=1}^{N_S} \left(\dfrac{(C^t + C^s)(S_j - S_i)}{2F_j + C^s S_j} \right)}{(2F_i + C^s S_i) \sum_{j=1}^{N_S} \left(\dfrac{1}{2F_j + C^s S_j} \right)}, & i \in \mathcal{N}_A \\ 0, & i \in \mathcal{N}_U \end{cases}$$

The necessary condition for the obtained result to be a NE is:

$$\begin{cases} p_i^*(2F_i + C^sS_i) - (C^tS_i + C^sS_i) \geq 0, \\ S_i[1 - t_i^*(2 + C^s)] \geq 0 \end{cases} \qquad i \in \mathcal{N}_A$$

$$\Longrightarrow \begin{cases} \dfrac{P}{C^t + C^s} \geq \sum_{i=1}^{N_S} \left(\dfrac{1}{\dfrac{2F_i}{S_i} + C^S} \right) \\ N_S \geq T(2 + C^s) \end{cases}$$

It is worth noting that $U_t(t^*, p^*)/U_p(t^*, p^*)$ is monotonously increasing in T/P, which means that the more resources are available to both players, the more payoff they will get.

This case is actually the most realistic situation to be considered, for both the TPA and the CP. The number of data that are usually outsourced in the Cloud is high enough to prevent both the attacker and the verifier from targeting every data. Actions, both in attack and verification, are therefore limited to the attractive data set D_A.

Case 2: $\sum_{i \in \mathcal{N}} t_i^* < T$ and $\sum_{i \in \mathcal{N}} p_i^* < P$:

In this case, both the CP and the TPA have sufficient resources, so they do not use up all their resources to respectively attack and verify data. Noticing U_t and U_p, we have: $\begin{cases} S_i(1 - 2t_i^* - t_i^* C^s) = 0 \\ p_i^*(2F_i + C^sS_i) - (C^tS_i + C^sS_i) = 0, \end{cases} \quad i \in \mathcal{N}$

$$\Longrightarrow NE = \begin{cases} t_i^* = \dfrac{1}{2 + C^s}, & i \in \mathcal{N} \\ p_i^* = \dfrac{C^tS_i + C^sS_i}{2F_i + C^sS_i}, & i \in \mathcal{N} \end{cases}$$

At the NE, we have:

$$\begin{cases} U_t(p^*, t^*) = -\sum_{i=1}^{N} \left(\dfrac{F_i\,(C^tS_i + C^sS_i)}{2F_i + C^sS_i} \right) \\ U_p(p^*, t^*) = 0 \end{cases}$$

In this case, the necessary condition for this result to be a NE is $N < T(2+C^s)$. Lemma 1 then states that $N_S = N$, which means that $D_U = \emptyset$. This is an expected result since both players have enough resources to target any data.

Moreover, from the above utility, it appears that having sufficient resources drags the utility of the attacker to zero, and leads the defender to be able to face greater risks by verifying more valuable data. The fact that the NE does not depend on the available resources is therefore consistent. Finally, the NE values show that the TPA will spend the necessary amount in order to prevent the CP

from gaining anything. In other words, the CP cannot expect to gain anything when the TPA has enough resources to verify all the outsourced data.

However, for medium and large companies, it is very unlikely that this case could actually occur given the amount and the wide diversity of data that are usually outsourced.

In the previous analysis, we identified the specific amount of resources that both the TPA and the CP should allocate for respectively verifying and attacking the attractive data set, in two different cases. A numerical analysis of this model is provided in section V.

However, this model obviously lacks some more realistic hypotheses, such as taking into account the fact that both players are more likely to act one after the other rather than at the same time, or taking into account a probabilistic integrity checking protocol instead of a deterministic one. The next section therefore considers such extensions of our primary model.

4 Extensions

4.1 Cloud Storage for Stackelberg Game

In the previous model, we considered that the two players take their decisions locally and simultaneously. However, a player can follow a certain strategy taking into account his opponent's decision (meaning that the follower makes his choice only after knowing the other's strategy). In this extended model, we address this case by modeling the interaction between TPA and CP as a Stackelberg game. The leader begins by choosing his best strategy, then the follower, after being informed about the leader's choice, chooses his own strategy which will maximize his payoff. We define the Stackelberg game for the Cloud storage verification like this: In this definition, the TPA is assumed as a leader, and the CP as a follower.

Players:	Leader : verifier side;
	Follower : attacker side;
Strategy type:	Mixed strategy.
Strategy:	$t \in W_T$ and $p \in W_P$
Payoff:	U_T for leader and U_P for follower
Game rule:	the leader decides t first, the follower decides p after knowing t.

Follower's problem:

According to the leader's chosen strategy, the follower chooses the strategy that maximizes his payoff (best response). Formally, for any chosen strategy t, the follower solves the following optimization problem:

$$p(t) = \arg\max_{p \in W_P} U_p(p, t)$$

Leader's problem:

The leader chooses his strategy which will maximize his payoff, given the follower will subsequently choose his best strategy. In other words, the leader

Table 2. Payoff matrix of the lead-or-follow game in extensive form

TPA/CP	Lead (p^L)	Follow (p^F)
Lead (t^L)	$U_p = -\delta \sum_{i\in\mathcal{N}} \left(\frac{S_i(C^t + C^s)(2 + C^s)}{\frac{2F_i}{S_i} + C^s} \right)$ $U_t = -\epsilon \sum_{i\in\mathcal{N}} \left(\frac{(2F_i + C^s S_i)}{2 + C^s} \right)$ $- \sum_{i\in\mathcal{N}} F_i \left(\frac{C^t S_i + C^s S_i}{2F_i + C^s S_i} - \epsilon \right)$	$U_p = 0$ $U_t = - \sum_{i\in\mathcal{N}} \left(\frac{C^t S_i + C^s S_i}{2 + C^s} \right)$ $-\delta \sum_{i\in\mathcal{N}} (C^t S_i + C^s S_i)$
Follow (t^F)	$U_p = \sum_{i\in\mathcal{N}} S_i \left(\frac{C^t S_i + C^s S_i}{2F_i + C^s S_i} - \epsilon \right)$ $U_t = - \sum_{i\in\mathcal{N}} F_i \left(\frac{C^t S_i + C^s S_i}{2F_i + C^s S_i} - \epsilon \right)$	$U_p = 0$ $U_t = 0$

chooses his strategy that gives the maximum gain in the worst case scenario. Formally, the leader solves the following optimization problem:

$$t(p) = \arg\max_{t\in W_T} U_t(p(t), t)$$

In most cases, Stackelberg games are solved by the backward induction technique. The solution consists of taking the follower's best response strategy as a function of the leader's strategy. Then, giving follower's best chosen response, the leader chooses his best strategy. The obtained equilibrium is referred to as a Stackelberg equilibrium (SE) or Stackelberg– Nash equilibrium (SNE).

Next, we address all possible cases, starting by considering the attacker as a leader and the verifier as a follower, then the verifier as a leader and the attacker as a follower, then we lastly examinate with the case when a player decides to be a leader or a follower without knowing the adversary's choice. In our study, we focus on the scenario where the attacker and the verifier have sufficient resources.

1) *Leader:* Attacker side; *Follower:* Verifier side

As the attacker will choose his strategy before the verifier, we have to find his best strategy subject to the constraint that the verifier makes a decision according to his best response function. We first start solving the verifier's best response by performing backward induction as follows:

$$t_i(t) = \begin{cases} = 0, & p_i < H_i, \quad i \in \mathcal{N} \\ \in [0, 1], & p_i = H_i, \quad i \in \mathcal{N} \\ = 1, & p_i > H_i, \quad i \in \mathcal{N} \end{cases}$$

Where $H_i = \dfrac{C^t S_i + C^s S_i}{2F_i + C^s S_i}$.

By noticing the leader's utility function $\sum_{i\in\mathcal{N}} p_i S_i [1 - t_i(2 + C^s)]$, we obtain the following SNE :

$$\begin{cases} t_i^S = 0, & i \in \mathcal{N} \\ p_i^S = H_i, & i \in \mathcal{N} \end{cases}$$

The corresponding payoff of both TPA and CP is as follows:

$$\begin{cases} U_t(t^S, p^S) = -\sum_{i\in D} F_i \ H_i, & i \in \mathcal{N} \\ U_p(t^S, p^S) = \sum_{i\in D} S_i \ H_i, & i \in \mathcal{N} \end{cases}$$

The fact that $U_t(t^S, p^S) = U_t(t', p^S), \forall t' \in W_T$ makes the above solution a weak Stackelberg equilibrium. Hence, the leader risks getting a negative payoff $(U_p(t^S, p^S) = -\sum_{i\in\mathcal{N}} H_i\big(S_i(1 + C^s)\big))$, since the follower can set $t_i = 1$ for all targets instead of t^S. This is clearly not acceptable for the attacker while his payoff is 0 when doing nothing.

As a solution, the attacker has to decrease his strategy a little bit by setting $p_i = p_i^S - \epsilon = H_i - \epsilon$, where ϵ is a small positive number, in order to guarantee that TPA will operate on t^S. As a result, the payoff will be $\sum_{i\in\mathcal{N}} S_i H_i - \epsilon \sum_{i\in\mathcal{N}} S_i$, which is slightly less than his desired payoff, since ϵ is sufficiently small.

2) *Leader:* Verifier side; *Follower:* Attacker side:

In this case, as the verifier plays the role of the leader, we will try to find the maximum value of his minimum payoff. Following the same analysis of the first case, The SNE is:

$$\begin{cases} t_i^S = \dfrac{1}{2 + C^s}, & i \in \mathcal{N} \\ p_i^S = 0, & i \in \mathcal{N} \end{cases}$$

In order to make sure that the attacker will operate on p^s, the verifier needs to increase his strategy a little bit by setting $t_i = t_i^S + \delta = \big(1/(2 + C^s)\big) + \delta$, where δ is a small positive number. In such a situation, the TPA payoff will be $-\sum_{i\in\mathcal{N}}\big(C^t S_i + C^s S_i/(2 + C^s)\big) - \delta \sum_{i\in\mathcal{N}}(C^t S_i + C^s S_i)$, which is a slightly less than his desired payoff at the SNE.

3) *Lead or Follow*:

Here we look at an interesting scenario where each player decides to choose the leader or the follower strategy, without knowing his adversary's choice. In this case, we aim to address the following questions: Is being a leader a better strategy than being a follower? Does the leader always control the behavior of the follower?

We formulate the (lead or follow) Cloud storage verification game as follows: the players are the verifier and the attacker; each player seeks to maximize his payoff by operating either on the leader strategy that we denote by t^L and p^L, respectively, or the follower strategy denoted by t^F and p^F, respectively. $\forall i \in \mathcal{N}$, we have:

$$t_i^L = \frac{1}{2 + C^s} + \delta, \quad t_i^F = 0, \quad p_i^L = H_i - \epsilon, \quad p_i^F = 0$$

Table 2 shows the payoff of both the attacker and the verifier. We ignored the terms that contain $\epsilon\delta$ due to their small value.

For the verifier, we can notice from Table 2 that the first row is strictly dominated by the second row, which means that it is better off for the verifier to be the follower. Hence, (p^L, t^F) is the NE of the game; the case when the attacker plays the role of the leader and the verifier follows.

From the above result, we can notice that the NE of the game is more favorable to the CP than the TPA, since the leader can control the behavior of the follower and pushes him to keep silent. Nevertheless, the TPA (follower) can influence the attacker's strategy, since both the strategy and the payoff of the attacker at the NE depends on the verification cost of the verifier. That being said, if $C^t \ll F_i$; both p_i and U_p are very small at the NE.

For the TPA, we would like to mention that his strategy at the NE $t_i^F = 0$ does not mean that no defender is needed, since before reaching the equilibrium, both players may try different strategies before choosing the one that maximizes their payoff.

Guideline 2: The TPA should choose the follower strategy in order to maximize his payoff, while leader is the best strategy for the CP.

4.2 Cloud Storage Game for Probabilistic Verification

Unlike the previous models, in which we consider that the TPA uses a deterministic verification protocol that guarantees a probability of detecting data modification or deletion equal to 1, in this extended model, we analyze the case of a probabilistic verification protocol that guarantees a detection probability inferior to one ($a < 1$) such as [3,9,15], since the TPA only performs verification on some parts of the data, in order to alleviate the verification cost. This means that there is a possibility that the TPA could not detect the incorrectness of the data with probability ($1 - a > 0$). On top of that, we now consider that the CP loses some storage cost when he does not attack the data while the TPA does not verify it.

Table 3 shows the matrix payoff of both players (CP/TPA) in the following extensive form: when the CP does not attack the data while the TPA does not verify it, the CP loses a payoff proportional to the size of the data, denoted by BS_i, where $B \in [0, 1]$. If the TPA verifies the data when it happens to be corrupted, then the TPA will gain ($-C^sS_i+F_i$) while CP gets ($-C^sS_i-S_i$), with probability a. With probability $(1 - a)$, the TPA has to pay the cost of the verification that is executed in both parts and also loses the data size, which means ($-C^sS_i-C^tS_i-F_i$) while CP gains ($-C^sS_i+C^sS_i+S_i$) $= S_i$.

The utility functions of CP and TPA are defined as follows:

$$U_t(t,p) = \sum_{i\in\mathcal{N}} t_i \Big[p_i a\big(2F_i + C^s S_i\big) - \big(C^t S_i + C^s S_i\big)\Big] - \sum_{i\in\mathcal{N}} p_i F_i$$

Table 3. Cloud storage game for probabilistic verification

CP \ TPA	Check	Not check
Available/ Correct data	$U_p = 0$ $U_t = -C^tS_i - C^sS_i$	$U_p = -BS_i$ $U_t = 0$
Unavailable/ Corrupted data	$U_p = (1-2a)S_i - aC^sS_i$ $U_t = -(1-2a)F_i - (1-a)C^sS_i - C^tS_i$	$U_p = S_i$ $U_t = -F_i$

$$U_p(t,p) = \sum_{i\in\mathcal{N}} p_i \Big[t_i\big(-(B+2a)S_i - aC^s S_i\big) + (1+B)S_i \Big] - \sum_{i\in\mathcal{N}} (1-t_i)BS_i$$

For data distribution, we keep the same characteristics as in the first model, in which data are distributed in two sets: the attractive set D_A, and the unattractive set D_U. The sets $\mathcal{N}_A$ and $\mathcal{N}_U$ are defined as in section III as well.

Now, we will investigate the NE of the game, according to players resource constraints. In this model, D_A and D_U are defined as follows.

Let $W = \dfrac{(1+B)\,|\mathcal{N}_A| - T(B + a(2+C^s))}{(1+B)\sum_{j\in\mathcal{N}_A} \frac{1}{S_j}}$.

Then :

$$\begin{cases} S_i > W, & \forall i \in \mathcal{N}_A \\ S_i < W, & \forall i \in \mathcal{N}_U \end{cases}$$

It is interesting to note that the detection rate a has a real influence on the constitution of the data sets D_A and D_U, since it follows from the preceding definition that D_A grows as a increases. This remark can be interpreted as follows: when the detection rate is low, the CP can target the most interesting data to corrupt without being detected, whereas with a high detection rate, the CP will have to take more targets into consideration in order to mitigate the risk of being detected.

As in section 3, the NE can be analyzed following two different cases, depending on the players resource constraints.

Case 1: $\sum_{i\in\mathcal{N}} t_i^* = T$ and $\sum_{i\in\mathcal{N}} p_i^* = P$:

This case represents the most frequent situation, encountered when both players do not have enough resources to attack or defend every target.

The NE, obtained by a reasoning similar to section 3, is as follows:

$$t_i^* = \begin{cases} \dfrac{T + \dfrac{1+B}{B+2a+aC^s} \displaystyle\sum_{j=1}^{N_S} \left(\dfrac{S_i - S_j}{S_j}\right)}{S_i \sum_{j=1}^{N_S} \left(\dfrac{1}{S_j}\right)}, & i \in \mathcal{N}_A \\ 0, & i \in \mathcal{N}_U \end{cases}$$

$$p_i^*=\begin{cases} \dfrac{P+\sum_{j=1}^{N_S}\left(\dfrac{(C^t+C^s)(S_i-S_j)}{a(2F_j+C^sS_j)}\right)}{a(2F_i+C^sS_i)\sum_{j=1}^{N_S}\left(\dfrac{1}{2F_j+C^sS_j}\right)}, & i\in\mathcal{N}_A \\ 0, & i\in\mathcal{N}_U \end{cases}$$

The necessary condition for the solution to be a NE is:

$$\begin{cases} \dfrac{P}{C^t+C^s}\geq\sum_{i=1}^{N_S}\left(\dfrac{1}{a\left(\frac{2F_i}{S_i}+C^S\right)}\right) \\ N_S(1+B)\geq T(B+a(2+C^s)) \end{cases}$$

In this case, as in the deterministic verification model, both players try to use the maximum of their resources in order to maximize their payoff. Moreover, calculating $U_t(t^*,p^*)$ shows, as expected, that improving the detection rate of the protocol used by the TPA (i.e., increasing a) can increase his utility and alleviate the attack intensity.

Case 2: $\sum_{i\in\mathcal{N}} t_i^* < T$ and $\sum_{i\in\mathcal{N}} p_i^* < P$:

Both players have enough resources to attack and verify every data. The NE is then:

$$\begin{cases} t_i^*=\dfrac{1+B}{B+2a+aC^s}, & i\in\mathcal{N} \\ p_i^*=\dfrac{C^tS_i+C^sS_i}{a(2F_i+C^sS_i)}, & i\in\mathcal{N} \end{cases}$$

Where the necessary condition is $N(1+B)<T(B+a(2+C^s))$.

As shown in the payoff values at the NE given below, having sufficient resources for both players is not suitable for the CP, who gets a negative payoff due to the fact that he loses some storage cost even when he does not attack. Since the TPA can target every data for verification, the CP has overall no chance to gain anything when attacking a data, and also suffers some loss, at least in this model, when doing nothing.

At the NE, the corresponding payoffs are indeed:

$$\begin{cases} U_t(t^*,p^*)=-\sum_{j=1}^{N}\left(\dfrac{F_i(C^tS_i+C^sS_i)}{a(2F_i+C^s)}\right) \\ U_p(t^*,p^*)=-\sum_{j=1}^{N}\left[BS_i\left(1-\dfrac{(1+B)}{B+a(2+C^s)}\right)\right] \end{cases}$$

It is also interesting to note that when the detection rate a increases, the TPA payoff increases, and the CP payoff decreases, which is a consistent result since a higher detection rate means that the TPA will have less failed verification attempts, while it will be harder for the CP to behave fraudulently without being detected.

From this analysis, we conclude that this theoretical model is realistic and consistent, and we were able to deduce the optimal strategies for both players in the two preceding cases, while putting into relief the importance of the detection rate a in the data distribution as well as in the players payoffs. A numerical study will now allow us to confirm these theoretical results.

5 Numerical Study

In this section, we validate the analytical results of the previous models by performing a numerical study.

In order to simplify the analysis, we consider that a client stores 20 data in the Cloud provider's data center with different sizes and sensibilities. We therefore consider that each data D_i has a size S_i and an importance F_i equal to $(21-i)*0.05$, $(i=1,2,...,20)$. As we mentioned earlier, the client delegates the check process to a special third party auditor TPA, that is equipped with high-performance verification modules and powerful processing capabilities. Thus, we set $C_t = C_s = 0.1$ for the case of deterministic verification schemes, and $C_t = C_s = 0.01$ for probabilistic schemes, since these schemes are much lighter, in terms of complexity, than deterministic ones.

For the deterministic verification model, according to Definition 2, our data are distributed into two sets: the first nine data belong to the attractive set D_A, whereas the remaining data are unattractive.

In the third model, where the verification process is probabilistic, we set $B = 0.001$. As expected, the data distribution is influenced by the probability of detecting data tampering a. In the case where $a = 0.9$ the attractive data are almost identical to the first model, since a is not so far from 1, while for $a = 0.5$, the number of attractive data decreases to 5, until reaches 3 for $a = 0.1$. This observation confirms our remark made in the previous section about the effect of a on the size of the data sets D_A and D_U. To further evaluate our analytical results, we investigate the case where TPA deviates from the NE. We thus simulate 10000 random strategies for the TPA under the condition that the CP chooses always his best response for each random strategy, in order to maximize his payoff.

For the deterministic model, Table 5 shows the strategies and the utility functions for both players at the NE, while Table 6 shows the payoffs of the TPA when he deviates from the NE. $U_t(t^r, p')_B$ is the best and the maximum payoff that the TPA can gain, where t^r is the random strategy for TPA, and p' is the CP's best response. $U_t(t^r, p')_W$ is the worst and minimal gain for TPA, while $U_t(t^r, p')_A$ is the average of all 10000 random strategies.

Table 5 and 6 clearly show that the best strategy for the TPA that maximizes his payoff is the NE, since $U_t(t^r, p')_B < U_t(t^*, p^*)$.

Fig.1 shows the utility functions of the TPA and the CP in the probabilistic verification model, under different values of the detection rate a. The valuable information that can be drawn here is that the TPA loss increases every time a decreases, while the CP gains more payoff every time a decreases, due to the fact

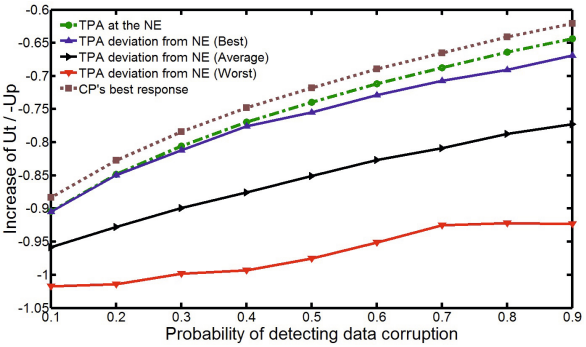

Fig. 1. Influence of the detection rate a on the TPA payoff in the probabilistic model

Table 4. Deterministic Verification Nash Equilibrium

The Defender (TPA)	The Attacker (CP)
$t_1^*= 0.19189$	$p_1^*= 0.10759$
$t_2^*= 0.17692$	$p_2^*= 0.10824$
$t_3^*= 0.16030$	$p_3^*= 0.10897$
$t_4^*= 0.14172$	$p_4^*= 0.10978$
$t_5^*= 0.12081$	$p_5^*= 0.11068$
$t_6^*= 0.09712$	$p_6^*= 0.11171$
$t_7^*= 0.07004$	$p_7^*= 0.11289$
$t_8^*= 0.03880$	$p_8^*= 0.11425$
$t_9^*= 0.00235$	$p_9^*= 0.11583$
$t_{10}^* - t_{20}^*= 0$	$p_{10}^* - p_{20}^*= 0$
$U_t(t^*, p^*) = -0.77100$	$U_p(t^*, p^*) = 0.59702$

that the more resources the CP uses to attack the first data in the attractive set, the more space he gains. Moreover, it appears that the TPA gets less payoff when he deviates from the NE.

These numerical results therefore corroborate our analysis of these theoretical models, and prove the consistency of the NE concept as the optimal strategy from which no player has any incentive to deviate in order to maximize his payoff.

6 Conclusion

In this paper, we focused on the problem of verifying data integrity in the case of data outsourced to an untrusted Cloud provider. We formulated the interaction between the verifier and the Cloud provider as a noncooperative game with mixed strategies, before performing an in-depth analysis on a deterministic model and on two extensions, namely the Stackelberg game for deterministic verification model, and a probabilistic verification model. Based on our analytical

Table 5. TPA Deviation From NE in Deterministic Verification Model

$U_t(t^r, p')_B$	$U_t(t^r, p')_A$	$U_t(t^r, p')_W$
-0.79884	-0.89058	-1.01050

results, we presented the expected behavior of a rational attacker, then derived the minimum verification resource requirement and the optimal strategy of the defender. We were also able to validate our analytical results by performing simulations.

However, the usual hypothesis of perfectly rational players limit the results of this work to very experienced attackers and verifiers who had a thoughtful approach of their actions. While not being unrealistic, given the fact that the CP and TPA entities are both very rational players by nature, this hypothesis remains a potential limitation to the superposition of this model with the objective behaviour of such entities in the reality.

Moreover, this work does not take into account several variants of the situation, such as the introduction of a penalty symbolizing the reputation loss in case of fraud from the CP, possibility to outsource numerous versions of a data to a CP, or the possibility for a CP to store multiple copies of each data with replication. Also, both the TPA and the CP can target more than one data at a time, which can be represented by a multiple-shot game. These variants will be the subject of future works that will aim at deepening this study in order to refine the model and integrate the hypotheses that are closer to reality.

References

1. Alpcan, T., Basar, T.: Network Security: A Decision and Game-Theoretic Approach. Cambridge University Press (2010)
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
3. Ateniese, G., Di Pietro, R., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, p. 9. ACM (2008)
4. Bensoussan, A., Kantarcioglu, M., Hoe, S(C.): A game-theoretical approach for finding optimal strategies in a botnet defense model. In: Alpcan, T., Buttyán, L., Baras, J.S. (eds.) GameSec 2010. LNCS, vol. 6442, pp. 135–148. Springer, Heidelberg (2010)
5. 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)
6. Curtmola, R., Khan, O., Burns, R., Ateniese, G.: Mr-pdp: Multiple-replica provable data possession. In: The 28th International Conference on Distributed Computing Systems, ICDCS 2008, pp. 411–420. IEEE (2008)
7. Gueye, A., Marbukh, V.: A game-theoretic framework for network security vulnerability assessment and mitigation. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 186–200. Springer, Heidelberg (2012)

8. Hassan, M.M., Song, B., Huh, E.-N.: Distributed resource allocation games in horizontal dynamic cloud federation platform. In: 2011 IEEE 13th International Conference on High Performance Computing and Communications (HPCC), pp. 822–827. IEEE (2011)
9. Juels, A., Kaliski Jr., B.S.: Pors: Proofs of retrievability for large files. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 584–597. ACM (2007)
10. Kochumol, A., Win, M.J.: Proving possession and retrievability within a cloud environment: A comparative survey. International Journal of Computer Science and Information Technologies 5(1), 478–485 (2014)
11. Mell, P., Grance, T.: The NIST definition of cloud computing (draft). NIST Special Publication 800(145), 7 (2011)
12. Nix, R., Kantarcioglu, M.: Contractual agreement design for enforcing honesty in cloud outsourcing. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 296–308. Springer, Heidelberg (2012)
13. Nix, R., Kantarcioglu, M.: Efficient query verification on outsourced data: A game-theoretic approach. arXiv preprint arXiv:1202.1567 (2012)
14. Ben Rosen, J.: Existence and uniqueness of equilibrium points for concave n-person games. Econometrica: Journal of the Econometric Society, 520–534 (1965)
15. Sebé, F., Domingo-Ferrer, J., Martinez-Balleste, A., Deswarte, Y., Quisquater, J.: Efficient remote data possession checking in critical information infrastructures. IEEE Transactions on Knowledge and Data Engineering 20(8), 1034–1038 (2008)
16. Yang, J., Wang, H., Wang, J., Tan, C., Yu, D.: Provable data possession of resource-constrained mobile devices in cloud computing. JNW 6(7), 1033–1040 (2011)
17. Zheng, X., Martin, P., Powley, W., Brohman, K.: Applying bargaining game theory to web services negotiation. In: 2010 IEEE International Conference on Services Computing (SCC), pp. 218–225. IEEE (2010)
18. Zhu, Y., Wang, H., Hu, Z., Ahn, G.-J., Hu, H., Yau, S.S.: Efficient provable data possession for hybrid clouds. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, pp. 756–758. ACM (2010)

Appendix I : Proof of Lemma 1

Here, we will prove that $\mathcal{N}_A$ contains d data with the biggest sizes, and $d = N_S$ by showing that neither $d < N_S$ nor $d > N_S$ is achieved.

In this proof, We need to only focus on the second case of the lemma, since the first case is straightforwardly evident. Before delving into the proof that $\mathcal{N}_A$ is unique, we should mention that it clearly appears that the N_S data with the biggest sizes that satisfy the second case of the lemma constitute the attractive data set $\mathcal{N}_A$, since the very definition of $\mathcal{N}_A$ given in Definition 2 is satisfied.

We first show that if $i \in \mathcal{N}_A$, then $\forall j < i$ $(S_j \geq S_i)$, it holds that $j \in \mathcal{N}_A$. Suppose this is not the case. Then, there exist $j_0 < i$ $(S_{j_0} \geq S_i)$ such that $j_0 \in \mathcal{N} - \mathcal{N}_A$. It follows that $S_{j_0} \leq C$. On the other hand, from Definition 2, we have $S_i > C$. It follows that $S_i > S_{j_0}$, which contradicts with $S_{j_0} \geq S_i$. Hence, $\mathcal{N}_A$ consist of the d data with the biggest sizes.

Now, we have to prove that $d = N_S$. Suppose first that $d < N_S$. From case 2 of the Lemma, we have:

$$S_{N_S} > \frac{N_S\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right)} \Longrightarrow S_{N_S}\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right) > N_S\left(\frac{1}{2+C^s}\right) - T$$

$$\Longrightarrow S_{N_S}\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right) - \frac{1}{2+C^s}(N_S-d) > d\frac{1}{2+C^s} - T.$$

Noticing that $S_{N_S} \leq S_i, \forall i \leq N_S$ and $d < N_S$ (i.e. $S_{d+1} \geq S_{N_S}$), we have:

$$\begin{aligned} S_{d+1}\sum_{j=1}^{d}\left(\frac{1}{2S_j+C^sS_j}\right) &\geq S_{N_S}\sum_{j=1}^{d}\left(\frac{1}{2S_j+C^sS_j}\right) \\ &\geq S_{N_S}\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right) - S_{N_S}\sum_{j=d+1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right) \\ &> S_{N_S}\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right) - \frac{1}{2+C^s}(N_S-d) > d\frac{1}{2+C^s} - T \end{aligned}$$

Hence, $S_{d+1} > \dfrac{d\left(\frac{1}{2+C^s}\right) - T}{\sum_{j=1}^{d}\left(\frac{1}{2S_j+C^sS_j}\right)}$. On the other hand, from Definition 2, we have

$S_{d+1} \leq \left(d\left((1/(2+C^s)\right) - T\right) / \left(\sum_{j=1}^{d}\left((1/2S_j + C^sS_j)\right)\right)$. This contradiction shows that it is impossible that $d < N_S$. Similarly, we can show that it is impossible that $d > N_S$. Hence, $d = N_S$ is uniquely determined, and so is $\mathcal{N}_A$. It follows obviously that $\mathcal{N}_U$ is also uniquely determined.

Appendix II : Proof of Theorem 1

The proof consists of showing that regardless of the verifier's strategy t, for any $p \in W_P$ such that $\exists i \in \mathcal{N}_U$, $p_i > 0$, we can construct another strategy p' such that $p'_i = 0$, $\forall i \in \mathcal{N}_U$ and $U_p(t,p) < U_p(t,p')$.
If $S_N \geq C$, then $\mathcal{N}_U = \emptyset$; the theorem holds evidently. We focus in our proof in the case where $S_N < C$, in other words, $\mathcal{N}_U \neq \emptyset$.
We consider a vector $t^0 = (t_1^0, t_2^0, ..., t_N^0)$ where:

$$t_i^0 = \begin{cases} \dfrac{T - \dfrac{N_S}{2+C^s} + S_i\sum_{j=1}^{N_S}\left(\dfrac{1}{2S_j+C^sS_j}\right)}{(2S_i+C^sS_i)\sum_{j=1}^{N_S}\left(\dfrac{1}{2S_j+C^sS_j}\right)}, & i \in \mathcal{N}_A \\ 0, & i \in \mathcal{N} - \mathcal{N}_A \end{cases}$$

It holds that $t_i^0 \geq 0$ and $\sum_{i=1}^{N_S} t_i^0 = T$. Let $t = (t_1, t_2, ..., t_N)$ denote the verification probability distribution of the verifier, with $\sum_{i=1}^{N_S} t_i \leq T$. By the Pigeon Hole Principle, it holds that $\exists m \in \mathcal{N}_A$ such that $t_m \leq t_m^0$.
We now consider any attacker strategy $p = (p_1, p_2, ..., p_N) \in W_P$ satisfying $\sum_{i\in\mathcal{N}_U} p_i > 0$, i.e; the attacker attacks at least one target outside the attractive data set with nonzero probability. We construct another attacker strategy profile p' based on p such that:

$$p'_i = \begin{cases} p_i, & i \in \mathcal{N}_A \text{ and } i \neq m \\ p_m + \sum_{j \in \mathcal{N}_U} p_j, & i = m \\ 0, & i \in \mathcal{N}_U \end{cases}$$

By comparing the attacker payoff at p and p', noticing that $\forall i \in \mathcal{N}_U$, $S_i < \frac{N_S\left((1/(2+C^s)\right)-T\Big)}{\left(\sum_{j=1}^{N_S}\left((1/2S_j+C^sS_j)\right)\right)}$, we obtain:

$$\begin{aligned} U_P(p) - U_P(p') &= \sum_{i\in\mathcal{N}} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) - \sum_{i\in\mathcal{N}} p'_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) \\ &= \sum_{i\in\mathcal{N}} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) \\ &\quad - \left(\sum_{i\in\mathcal{N}_A, i\neq m} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) + \left(p_m + \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i\right) S_m\Big(1 - t_m\big(2 + C^s\big)\Big)\right) \\ &= \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) - \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_m\Big(1 - t_m\big(2 + C^s\big)\Big) \\ &\leq \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) - \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_m\Big(1 - t_m^0\big(2 + C^s\big)\Big) \\ &= \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_i\Big(1 - t_i\big(2 + C^s\big)\Big) - \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i \left(\frac{N_S \frac{1}{2+C^s} - T}{\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right)}\right) \\ &\leq \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i S_i \; - \sum_{i\in\mathcal{N}-\mathcal{N}_A} p_i \left(\frac{N_S \frac{1}{2+C^s} - T}{\sum_{j=1}^{N_S}\left(\frac{1}{2S_j+C^sS_j}\right)}\right) < 0 \end{aligned}$$

Hence, the strategy p' gives more payoff to the CP than the strategy p. A rational CP therefore has no incentive to attack any data $D_i \in D_U$.