Towards a strategic approach to security based on game theory: Untrusted Cloud Storage Game

J. Leneutre
20/01/2017
Outline

- Introduction
- Untrusted Cloud Storage Game
- Resource Constrained Network Security Games
- Conclusion
Introduction

Context: Security Risk Assessment

- Security mechanisms are available & validated
- Decision making help when defining security policy or facing security incidents
- With a limited defense budget

Challenges

- How to find the best trade-off between security level and other constraints?
- How to compare the respective efficiency on the security level of two distinct defense strategies?
- How to capture the dynamic between attacker and defense system?

⇒ Need for a Quantitative Approach for security assessment
Introduction

- **Existing approaches**
  - Traditional qualitative security risk assessment methods (EBIOS, TVRA)
    - Subjective assessment results
  - Approaches based on an explicit modeling of attacks:
    - Examples: extension of attack trees, attack graphs, extension of BDMP to security, etc.
    - Quantitative extensions
    - Scaling problem for large systems
  - Approaches based on security metrics definition
    - Examples: vulnerability prediction models, attack surface model, etc.
    - Difficult to assess the relevance of metrics

- **Alternative approach based on the use of Game theory?**
Introduction

- **Game theory**: study of conflict and cooperation between intelligent rational decision makers

- **Application to security**
  - Analytical framework for a quantitative modeling of the interaction between malicious attackers and security administrators
    - The attacker is strategic and adapts to the defender’s action
    - Security risk assessment & response modeled as dynamic resource allocation problems
  - Decision support to security
    - Where to deploy in priority the security resources given a limited budget?
    - How to optimally configure protection or monitoring mechanisms
Introduction

Game Theory glossary

- **Game**: description of the strategic interactions between a set of rational players under certain rules
- **Player**: a strategic decision maker (can be a person, a machine, etc.)
- **Action**: a move that can be carried out by the player at any given time
- **Utility function**: assigns a payoff for every possible outcome of the game for a given player taking into account other players’ actions
- **Strategy**: a plan of actions taken in the game by one player (pure/mixed)
- **Strategy profile**: set of strategy for each player

- **Rationality assumption**: every player acts in a way to maximize his utility function

- **Equilibrium**: the point where players maximize their payoffs taking into account other players’ strategies
  - **Nash Equilibrium**: strategy from which no player has an incentive to deviate unilaterally
**Introduction**

- **n-player strategic game (normal form)**
  - A finite set \( N \) of \( n \) **players**
  - For each player \( i \in N \) a non empty set \( A_i \) of **actions**, and \( S_i \) the derived set of **strategies**
  - For each in player \( i \in N \) a **utility function** \( u_i \)

- **Nash equilibrium (NE)**
  - A strategy profile \((s_1^*, ..., s_i^*, ..., s_n^*)\) is a Nash Equilibrium if

\[
\forall i \in N, \forall s_i \in S_i, u_i(s_1^*, ..., s_i^*, ..., s_n^*) \geq u_i(s_1^*, ..., s_i^*, ..., s_n^*)
\]

  - Every agent’s strategy at NE is a best response to the other agents’ strategies at NE
Introduction

Example: Forwarder’s dilemma

- **Goal:** device $p_1$ (resp. $p_2$) wants to send a packet to his receiver $r_1$ (resp. $r_2$) using $p_2$ (resp. $p_1$) as a forwarder, in each time slot

- **Players:** $p_1$ & $p_2$
- **Actions:** Forward (F) or Drop (D) a packet

- **Utility function:**
  - $c$ ($0 < c < 1$): cost representing the energy and computation spent for the forwarding action
  - Reward when packet arrive at destination: 1

- **Nash equilibrium:** strategy profile (D,D)

\[
\begin{array}{c|cc}
   & F & D \\
\hline
F & (1-c,1-c) & (-c,1) \\
D & (1,-c) & (0,0) \\
\end{array}
\]
Introduction

Different type of games

- Zero-sum vs. Non Zero-sum
- Cooperative vs Non cooperative
- Static vs. Dynamic games
  - Static game (one-shot game): all players choose their strategies simultaneously
  - Dynamic game (Stackelberg game, leader & follower game): players choose their actions in more than one stage
- Complete information vs. Incomplete information game
  - Complete Information game: players know each others’ strategies and payoffs
  - Incomplete Information game (Bayesian game): information about the characteristics (strategies, payoffs) of other players are incomplete
- Deterministic vs. Stochastic games
  - Stochastic game: game involving probabilistic transitions between different states of the system
Introduction

Security Games [Alpcan & Basar 2010]

- Games that study the interaction between malicious attackers and defenders
- Mostly two Player non cooperative games: attacker and security administrator
- Pioneered in the context of intrusion detection [Alpcan & Basar 2003], and later received a large attention
- Typically in these works, the defender decides where to allocate limited available security resources
- Utility function quantifies the loss of the attacker in term of security risk impact and cost of security countermeasures
### Introduction

#### Multiple models of Security Games: intrusion detection

<table>
<thead>
<tr>
<th>Classification</th>
<th>Article</th>
<th>Decision-making model</th>
<th>IDS optimization problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Allocation Optimization</td>
<td>[Kodialam and Lakshman 2003]</td>
<td>Zero-sum static game</td>
<td>Optimize network link sampling rates</td>
</tr>
<tr>
<td></td>
<td>[Otrock et al. 2008b]</td>
<td>Incomplete information zero-sum static game</td>
<td>Optimize cluster defense strategy in MANET</td>
</tr>
<tr>
<td></td>
<td>[Otrock et al. 2008a]</td>
<td>Zero-sum static game</td>
<td>Optimize resource sharing between nodes</td>
</tr>
<tr>
<td></td>
<td>[Agah et al. 2004]</td>
<td>Nonzero-sum static game / MDP</td>
<td>Compute the expected behavior of attackers</td>
</tr>
<tr>
<td></td>
<td>[Zhu et al. 2009]</td>
<td>N-player nonzero-sum static game</td>
<td>Optimize defense resource allocation on nodes</td>
</tr>
<tr>
<td></td>
<td>[Sallhammar et al. 2005]</td>
<td>Zero-sum stochastic game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Chen and Lenetre 2009]</td>
<td>Nonzero-sum static game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Nguyen et al. 2009]</td>
<td>Zero-sum stochastic game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Alpcan and Basar 2003]</td>
<td>Nonzero-sum static game</td>
<td>Configure IDS detection sensitivity</td>
</tr>
<tr>
<td></td>
<td>[Alpcan and Basar 2006]</td>
<td>Zero-sum stochastic game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Zhu and Basar 2009]</td>
<td>CMDP</td>
<td>Select the optimal set of detection libraries</td>
</tr>
<tr>
<td></td>
<td>[Zhu et al. 2010]</td>
<td>N+M player nonzero-sum stochastic game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Ghorbani and Hashemi 2015]</td>
<td>N+M player nonzero-sum stochastic game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Liebald et al. 2008]</td>
<td>Zero-sum static game</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Kantzavelou and Katsikas 2010]</td>
<td>Static game</td>
<td>Optimize the IDS reaction to user interactions</td>
</tr>
<tr>
<td></td>
<td>[Shen et al. 2012]</td>
<td>Zero-sum stochastic game</td>
<td>Optimize the survivability of a WSN</td>
</tr>
<tr>
<td></td>
<td>[Moosavi and Bui 2014b]</td>
<td>Incomplete information nonzero-sum stochastic game</td>
<td>Optimize attack mitigation in a WSN</td>
</tr>
</tbody>
</table>

**Example:** Comparison of game theoretic models for intrusion detection
Introduction

*Example: Intrusion detection resource allocation with non-correlated security assets Game [Chen & Leneutre 2009] (1)*

**Attacker**
- Attack (resp. not attack) target Ni with probability $p_i$ (resp 1-$p_i$)
- Attack resource budget: $P$
- Cost of attacking: $C_{a1}$

**Target node Ni with security asset level $l_i$**
- Monitor (resp. not monitor) target Ni with probability $q_i$ (resp 1-$q_i$)
- Defense resource budget: $Q$
- Detection rate $a$, false alarm rate $b$
- Cost of monitoring: $C_{m1}$
- Cost of false positive: $C_{f1}$

**Defender (IDS)**

- Strategy of defender: $q_i$, $i=1,…,n$ such that $\sum_{i=1}^{n} q_i \leq Q$
- Utility of defender is a function of: $q_i$, $p_i$, $l_i$, $a$, $b$, $C_m$, $C_f$
Introduction

Example: Intrusion detection resource allocation with non-correlated security assets Game [Chen & Leneutre 2009] (2)

- Contributions: model the interaction between attack and defender side as non-cooperative game
  - Study the possible equilibrium: Nash equilibrium
  - Look for efficient NE which is favorable for defender side
  - Study the attack’s strategy at the NE
  - Study the optimal strategy of defender side to maximize its payoff at the NE

- Main results
  - Rational attacker only focus on a subset of targets, others are “self secured”
    - We provide an algorithm to compute the above subset: sensible target set
    - We derive the minimum number of defenders to maintain the efficient NE
    - We derive optimal strategy for defenders to operate at the efficient NE
Introduction

- Real World application of game theory to (global) security
  
  - ARMOR project* (Assistant for Randomized Monitoring Over Routes) is a real world application which calculates optimum patrol patterns
    
    ➡ Federal Air Marshal Service use it to determine the optimum schedule to guard the most vulnerable flights and the location of checkpoints and canine patrols at LAX
    
    ➡ The Coast Guard use it to randomize patrols
    
    ➡ Used in rapid transit systems (metro, …) for fare evasion deterrence

Outline

• Introduction

• **Untrusted Cloud Storage Game**

• Resource Constrained Network Security Games

• Conclusion

*Joint Work: with B. Djebaili, Z. Ismail, C. Kiennert (TPT, LTCI), L. Chen (Université Paris-Sud, LRI), D. Bateman (EDF R&D)

B. Djebaili, C. Kiennert, J. Leneutre, L. Chen, Data Integrity and Availability Verification Game in Untrusted Cloud Storage, Conference on Decision and Game Theory for Security (GameSec), Los Angeles, CA, USA, November 2014, LNCS.

Untrusted Cloud Storage Game

Context

Cloud features:
- On-demand services
- Resource pooling via multi-tenancy
- Elasticity via dynamic provisioning of resources
- Device and location independence

Source of security problems:
- Reduced control over software and data
- Potential Interference between security and cloud optimization mechanisms

Security of data storage:
- Privacy / Confidentiality
- Integrity/availability
  - External (hackers) threats for data integrity or availability
  - Cloud Provider (CP) might behave unfaithfully
  - Users need strong evidence that their data have not been tampered or partially deleted
Untrusted Cloud Storage Game
Problem Statement

- Case of an Untrusted CP
  - Economically-motivated CP that may be tempted to erase (copies of) data to use less storage space
    ➡  How to check compliance of SLAs with regard to data replication?

- Efficient schemes for remote data integrity checking exist
  - New cryptographic protocols: proof of data possession (PDP), proof of retrieval (POR) …
    ➡  However verification costs computing resources

- How to optimize their use?
  - Frequency of the verification process?
  - Which data to check in priority?
  - Are there data not worth checking at all?
    ➡  Optimal verification policies needed
  - Trade-off between security & cost of verification
  - Obtained by a Game Theoretical analysis modelling interactions between Verifier & CP
Untrusted Cloud Storage Game

Underlying assumptions

- **Data replication rate is specified in SLAs**
  - Usually not covered in a *cloud storage service provider's* SLA
    - Rather provide guarantees in terms of uptime, or allowed number of retries, or how long a read request can take to be serviced
    - Offer some sort of tiered credits the users if the guarantees are not satisfied
  - May be negotiated in the case of *storage backup* or *cloud archive services*
    - Possible definition of precise retention policies

- **User is allowed to access to different copies of same data**
  - May be necessary to check geographical location of data
Untrusted Cloud Storage Game
Background: Integrity verification of outsourced data

- **Usual techniques for integrity control**
  - Hash functions, error-correcting code, checksum, …
  - … not suited for intentional modification of data!
Untrusted Cloud Storage Game

Background: Integrity verification of outsourced data

- Need for a new cryptographic primitive
  - Integrity checking challenge response protocol

- Metadata may also be outsourced
- Verification may be delegated to a third party auditor (TPA)
Untrusted Cloud Storage Game

Background: Integrity verification of outsourced data

- A naive scheme

1. Compute $H_k(F)$
2. Send the file $F$ by releasing $K$
3. Challenge the cloud by releasing $K'$ and asking him to compute $H_{k'}(F)$
4. Return a response $r$
5. Verify if $r = H_k(F)$

- Requires large metadata size
- Consumes too much bandwidth and computation
- Verifications limited to the number of precomputed hash values
A simple protocol based on DLP [Deswarte & alii, 2004]

- Metadata: Tag computed using an homomorphic function

\[ T = g^d \mod n \]

\[ C = g^r \mod n \]

\[ R = C^d \mod n \]

\[ T^r = C^d? \]

“d”: data  
“T”: tag (metadata)  
“C”: challenge  
“R”: response  
“n”: RSA modulus  
“r”: random integer  
“DLP”: discrete logarithm problem

Untrusted Cloud Storage Game

Background: Integrity verification of outsourced data

- **Two main approaches for data verification schemes**
  - Deterministic protocols: checks entire data
  - Probabilistic protocols: randomly checks blocks of data
    ➔ reduce the computing time of verification

- **Main efficient verification schemes**
  - PDP (Provable Data Possession) [Ateniese & alii 2011]
    - Minimize bandwidth
  - POR (Proofs of Retriability) [Juels, Kaliski 2007]
    - Ability to recover corrupted files by using error correcting codes

- **Other features**
  - Public verification
  - Management of dynamic data
  - Verification of multiple copies of a data


Untrusted Cloud Storage Game
Contributions

Define a basic model
- Static game with deterministic verification protocol
- CP stores only one copy of the data

Study different extensions of the model
- Static game with probabilistic verification protocol
- Dynamic game with deterministic verification (Stackelberg game)
- Extension where CP stores multiple copies of data

For each model:
- Prove the existence of an attractive data set on which both attacker and verifier should focus exclusively
- Find the Nash Equilibrium
- Analyze the results in terms of expected behaviours & deduce guidelines for optimal TPA data checking
Untrusted Cloud Storage Game
Static deterministic verification basic game

- Non-cooperative game
- Two rational players
  - Attacker (CP)
  - Verifier (TPA)
- Two actions per player for each data:
  - Attacker: Not replicating / Do nothing
  - Verifier: Check data integrity / Do nothing
- Strategies: distribution of attack/verification resources
  - For each data $D_i$, the attacker decides to not replicate (delete) data with probability $p_i$, and the verifier checks data with probability $q_i$
  - Available resources for attacker (resp. verifier) : $P$ (resp. $Q$)
  - Resource constraints: $\sum_{i=1}^{n} p_i \leq P \leq 1$ and $\sum_{i=1}^{n} q_i \leq Q \leq 1$

Untrusted Cloud Storage Game
Static deterministic verification basic game

- Game parameters
  - Amount of data stored at the CP: \( N \)
  - Financial storage cost of data \( D_i \) (proportional to its size): \( S_i \geq 0 \)
  - Financial value (integrity level) of data \( D_i \): \( F_i \)
  - Overall TPA probability of detecting fraud when checking data: \( a \)
    - \( a = 1 \) for deterministic verification protocols
    - \( a < 1 \) for probabilistic verification protocols
  - Verification processing costs for CP: \( C^s S_i \) with \( 0 \leq C^s \leq 1 \)
  - Verification processing costs for TPA: \( C^T S_i \) with \( 0 \leq C^t \leq 1 \)
Assumptions

- Cost related to network communications both on the CP and TPA sides are ignored.
- Possible storage flaws of an honest CP are out of scope of this model.
- The probability of data corruption remaining undetected by the TPA after a check is neglected, even when using a probabilistic protocol.
  - Each player aims at maximizing his payoff.
- TPA verification processing costs are taken in charge by the TPA.
- CP verification processing costs are taken in charge by the TPA (resp. CP) when verification leads to a positive (resp. negative) result.
Untrusted Cloud Storage Game
Static deterministic verification basic game

Utility functions of static game for deterministic verification

<table>
<thead>
<tr>
<th>CP \ TPA</th>
<th>Check</th>
<th>Not check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replicate</td>
<td>0, (-C^t S_i - C^s S_i)</td>
<td>0, 0</td>
</tr>
<tr>
<td>Not Replicate</td>
<td>(-C^s S_i - S_i, -C^t S_i + F_i)</td>
<td>(S_i, -F_i)</td>
</tr>
</tbody>
</table>

CP payoff:

\[
U_{CP}(p, q) = \sum_{i=1}^{N} [p_i q_i (-C^s S_i - S_i) + p_i (1 - q_i) S_i] = \sum_{i=1}^{N} p_i S_i [1 - q_i (2 + C^s)]
\]

TPA payoff:

\[
U_{TPA}(p, q) = \sum_{i=1}^{N} [(1 - p_i) q_i (-C^t S_i - C^s S_i) + p_i q_i (-C^t S_i + F_i) - p_i (1 - q_i) F_i]
\]

Ressource constraints: \(\sum_{i=1}^{N} p_i \leq P \leq 1\) and \(\sum_{i=1}^{N} t_i \leq Q \leq 1\)
Data distribution

- Does a rational attacker (CP) attack all data?

Existence of an Attractive Dataset

Actually, a rational attacker will only attack data with large enough sizes \( S_i \)

**Guideline 1:** A rational defender has only to verify data in the attractive dataset
Untrusted Cloud Storage Game
Static deterministic verification basic game

Solving the game

• The existence of NE depends on the resource constraints
• Value of NE when all resources are used by both players, \( \sum_{i \in N} p_i^* = P \) and \( \sum_{i \in N} q_i^* = Q \)

\[
p_i^* = \begin{cases} 
  P - \sum_{j=1}^{N_S} \left( \frac{(C^t + C^s)(S_j - S_i)}{2F_j + C^sS_j} \right), & i \in N_A \quad \text{(Attractive dataset)} \\
  0, & i \in N_U
\end{cases}
\]

\[
q_i^* = \begin{cases} 
  T - \frac{N_S}{2 + C^s} + S_i \sum_{j=1}^{N_S} \left( \frac{1}{2S_j + C^sS_j} \right), & i \in N_A \\
  0, & i \in N_U
\end{cases}
\]

Guideline 2: Verification resources to data should be allocated accordingly to the values of \( q_i^* \)
### Table 1: Payoff at the Nash Equilibrium (NE)

<table>
<thead>
<tr>
<th>Defender (TPA)</th>
<th>Attacker (CP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1^* = 0.19620$</td>
<td>$p_1^* = 0.37038$</td>
</tr>
<tr>
<td>$t_2^* = 0.18034$</td>
<td>$p_2^* = 0.18970$</td>
</tr>
<tr>
<td>$t_3^* = 0.16272$</td>
<td>$p_3^* = 0.12538$</td>
</tr>
<tr>
<td>$t_4^* = 0.14303$</td>
<td>$p_4^* = 0.09239$</td>
</tr>
<tr>
<td>$t_5^* = 0.12087$</td>
<td>$p_5^* = 0.07232$</td>
</tr>
<tr>
<td>$t_6^* = 0.12087$</td>
<td>$p_6^* = 0.05883$</td>
</tr>
<tr>
<td>$t_7^* = 0.06707$</td>
<td>$p_7^* = 0.04914$</td>
</tr>
<tr>
<td>$t_8^* = 0.03396$</td>
<td>$p_8^* = 0.04183$</td>
</tr>
<tr>
<td>$t_9^* \ldots t_{20}^* = 0$</td>
<td>$p_9^* \ldots p_{20}^* = 0$</td>
</tr>
</tbody>
</table>

$U_t(t^*, p^*) = -0.12369$

$U_p(t^*, p^*) = 0.60562$

### Table 2: Payoff Degradation due to deviation from NE

<table>
<thead>
<tr>
<th>Number of data</th>
<th>$n=20$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPA random strategy</td>
<td>$t^r$</td>
</tr>
<tr>
<td>CP best response</td>
<td>$p'$</td>
</tr>
<tr>
<td>TPA best &amp; maximum gain</td>
<td>$U_t(t^r, p')_B$</td>
</tr>
<tr>
<td>TPA average gain</td>
<td>$U_t(t^r, p')_A$</td>
</tr>
<tr>
<td>TPA minimal gain</td>
<td>$U_t(t^r, p')_W$</td>
</tr>
</tbody>
</table>

$U_t(t^r, p')_B = -0.13976$

$U_t(t^r, p')_A = -0.34721$

$U_t(t^r, p')_W = -0.61456$
Untrusted Cloud Storage Game
Stackelberg deterministic verification basic game

- **Players have sequential interaction**
  - the move of one player is conditioned by the move of the other player

- **Stackelberg Game principle**
  - The leader L moves first
  - The follower F observes the leader’s choice, then chooses his strategy

- **Backward induction**
  - Follower’s problem: for every strategy $s_L$ of L, F computes
    \[ s_F(s_L) = \arg\max_{s_F \in S_F} U_F(s_L, s_F) \]
  - Leader’s problem: F computes $s_L(s_F) = \arg\max_{s_L \in S_L} U_L(s_L, s_F(s_L))$
  - Subgame perfect equilibria or Stackelberg-NE
Players have sequential interaction: the move of one player is conditioned by the move of the other player

Stackelberg Game principle:
- The leader moves first
- The follower observes the leader’s choice, then chooses his strategy

Three cases analyzed:
- Case 1: Leader: CP, Follower: TPA
- Case 2: Leader: TPA, Follower: CP
- Case 3: Lead of Follow
  - Which strategy will be better for both TPA & CP?
  - Actually, Case 1 corresponds to the best strategy for both

Guideline 3: TPA should choose the follower strategy in order to maximize his payoff, while leader is the best strategy for the CP
Untrusted Cloud Storage Game
Game with multiple data copies

- Multiple copies of the same data on the CP servers
  - Parameters: same than generic game plus
    - Number of copies of data $D_i$: $R_i$
    - Reward the CP gets if he acts honestly: $\varepsilon F_i (\varepsilon \geq 0)$

<table>
<thead>
<tr>
<th>CP \ TPA</th>
<th>Check</th>
<th>Not check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct/Available copy</td>
<td>$\varepsilon F_i$, $-C^t S_i - C^s S_i$</td>
<td>0, 0</td>
</tr>
<tr>
<td>Incorrect/unavailable copy</td>
<td>$-C^s S_i - S_i$, $-C^t S_i + F_i$</td>
<td>$S_i$, $-F_i$</td>
</tr>
</tbody>
</table>

- Strategies
  - Probability that the CP deletes $i$ copies of data $D_m$ ($0 \leq i \leq R_m$): $p_i^m$
  - Probability that the TPA checks $i$ copies of data $D_m$ ($0 \leq i \leq R_m$): $q_i^m$
Untrusted Cloud Storage Game
Game with multiple data copies

Utility functions of game with multiple copies

Notation: \( \mathbb{1} \) denotes the indicator function

CP payoff:

\[
U_A(p, q) = \sum_{m=1}^{N} \left\{ - \sum_{i=1}^{R_m} \sum_{j=1}^{R_m} p_i^m q_j^m (i S^m + j C^s S^m) \mathbb{1}_{i > R_m - j} \\
+ \sum_{i=0}^{R_m} \sum_{j=1}^{R_m} \epsilon p_i^m q_j^m (j F^m) \mathbb{1}_{i \leq R_m - j} \\
+ \sum_{i=1}^{R_m} \sum_{j=0}^{R_m} p_i^m q_j^m (i S^m) \mathbb{1}_{i \leq R_m - j} \right\}
\]

TPA payoff:

\[
U_D(p, q) = \sum_{m=1}^{N} \left\{ \sum_{i=1}^{R_m} \sum_{j=1}^{R_m} p_i^m q_j^m (i F^m) \mathbb{1}_{i > R_m - j} \\
- \sum_{i=0}^{R_m} \sum_{j=1}^{R_m} p_i^m q_j^m (j C^s S^m) \mathbb{1}_{i \leq R_m - j} - \sum_{j=1}^{R_m} q_j^m C^t S^m j \\
- \sum_{i=1}^{R_m} \sum_{j=0}^{R_m} p_i^m q_j^m (i F^m) \mathbb{1}_{i \leq R_m - j} \right\}
\]
Untrusted Cloud Storage Game
Game with multiple data copies

**Two game settings**

- Independent strategies game
  - Player’s strategy for each data does not depend on other data:
  - for each data $D_m$: $\sum_{i=0}^{R^m} p_i^m = 1$ and $\sum_{j=0}^{R^m} q_j^m = 1$

- Correlated strategies game
  - Player’s strategy for each data depends on strategies for other data:
  - for N data: $\sum_{m=1}^{N} \sum_{i=0}^{R^m} p_i^m = 1$ and $\sum_{m=1}^{N} \sum_{j=0}^{R^m} q_j^m = 1$

➤ There exists a unique NE for the game with independent strategies
➤ There exists a unique NE for the game with correlated strategies when each player can target several types of data at each instance of the game
➤ If $\varepsilon>0$, there exists a unique NE for the game with correlated strategies when each player can target only one type of data at each instance of the game
Untrusted Cloud Storage Game
Game with multiple data copies

- Game with multiple copies and independant strategies

\[ \mu = (2, 0.5, 0.1, 0.5, 0.1) \]

\[ \mu = (R, F, C^s, C^t, \epsilon) \]
Untrusted Cloud Storage Game
Game with multiple data copies

Estimation of the parameters

- Number of Backup copies $R_m$: specified in the SLA
- Storage cost $S_m$: can be deduced from the size of data based on [Chen & Sion, 2011]
- Verification costs parameter $C^t$ & $C^s$: benchmarks from verification protocol implementations and cost of CPU cycles from [Chen & Sion, 2011]
- Financial value of Data $F_m$: deduced from the application of a risk assessment method

Perspectives

- Extension to an infinite repeated game
- Take into account location requirements for data

→ Give some guidelines to define an optimal verification strategy for data replication compliance checking

→ May be used to define “ALAs (Audit Level Agreements)”

Resource Constrained Network Security Game

Resource Constrained Network Security games

- N targets in the network to defend/attack
- Strategy of each player corresponds to the amount of resources allocated to attack/defend each target with resource constraints P/Q
- Utilities functions are such that
  - \( r_i, s_i, u_i > 0, \) and \( t_i \geq 0 \)
  - \( r_i', s_i', u_i' < 0 \) and \( t_i' \leq 0 \)
  - \( u_i \leq t_i, s_i' \leq u_i' \)

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defend</th>
<th>Not defend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( r_i, r_i' )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_i, t_i' )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not attack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( s_i, s_i' )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_i, u_i' )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Theo : If \( t_i-u_i \geq r_i-s_i \) and \( r_i'-t_i' \geq s_i'-u_i' \), a necessary condition for \( (p^*,q^*) \) to be a NE of a RNCS game is \( \sum_{i \in T} p_i^* = P \)
Outline

• Introduction

• **Untrusted Cloud Storage Game**

• Resource Constrained Network Security Games

• Conclusion

*Joint Work: with B. Djebaili, Z. Ismail, C. Kiennert (TPT, LTCI), L. Chen (Université Paris-Sud, LRI), D. Bateman (EDF R&D)

B. Djebaili, C. Kiennert, J. Leneutre, L. Chen, Data Integrity and Availability Verification Game in Untrusted Cloud Storage, Conference on Decision and Game Theory for Security (GameSec), Los Angeles, CA, USA, November 2014, LNCS.

Outline

• Introduction
• Game Theory
• Contributions
• Game Models
• Conclusion
Conclusion

Some limitations when applying game theory to security

- Relevance of the game solution (rationality assumption)
  - Need for an experimental evaluation methodology
    ➡ Take inspiration from [Taylor & al. 2010]

- Uncertainties about the attacker (Complete information vs. Incomplete information)
  - Need to take into account several profiles of attackers
    ➡ Use of Bayesian Stackelberg games [Taylor & al. 2010]

- High level of abstraction: limited action sets (attack/not attack), simple utility functions, ...
  - Definition of appropriate utility functions in term of security