Information Leakage Games

Mário S. Alvim[1], Konstantinos Chatzikokolakis[2], Yusuke Kawamoto[3], and Catuscia Palamidessi[4]

[1] Universidade Federal de Minas Gerais, Brazil
[2] CNRS and École Polytechnique, France
[3] AIST, Japan
[4] INRIA and École Polytechnique, France

Abstract. We consider a game-theoretic setting to model the interplay between attacker and defender in the context of information flow, and to reason about their optimal strategies. In contrast with standard game theory, in our games the utility of a mixed strategy is a convex function of the distribution on the defender's pure actions, rather than the expected value of their utilities. Nevertheless, the important properties of game theory, notably the existence of a Nash equilibrium, still hold for our (zero-sum) leakage games, and we provide algorithms to compute the corresponding optimal strategies. As typical in (simultaneous) game theory, the optimal strategy is usually mixed, i.e., probabilistic, for both the attacker and the defender. From the point of view of information flow, this was to be expected in the case of the defender, since it is well known that randomization at the level of the system design may help to reduce information leaks. Regarding the attacker, however, this seems the first work (w.r.t. the literature in information flow) proving formally that in certain cases the optimal attack strategy is necessarily probabilistic.

1 Introduction

A fundamental problem in computer security is the leakage of sensitive information due to correlation of *secret information* with *observable information* publicly available, or in some way accessible, to the attacker. Correlation in fact allows for the use of Bayesian inference to guessing the value of the secret. Typical examples are *side channels attacks*, in which (observable) physical aspects of the system, such as the execution time of a decryption algorithm, may be exploited by the attacker to restrict the range of the possible (secret) encryption keys. The branch of security that studies the amount of information leaked by a system is called *Quantitative Information Flow* (QIF), and it has seen growing interest over the past decade. See for instance [10,15,27,3,4], just to mention a few.

In general, it has been recognized that randomization can be very useful to obfuscate the link between secrets and observables. Examples include various anonymity protocols (for instance, the dining cryptographers [9] and Crowds [23]), and the renown framework of differential privacy [11]. The *defender* (the system designer, or the user) is, therefore, typically probabilistic. As for the attacker, most works in the literature consider only *passive attacks*, limited to observing the system's behavior. Notable exceptions are the works of Boreale and Pampaloni [4], and of Mardziel et al. [18], which consider *adaptive attackers* who interact with and influence the system. We note that, however, [4] does not consider probabilistic strategies for the attacker. As for [18], although their model allows them, none of their extensive case-studies needs probabilistic attack strategies to maximize leakage. This may seem surprising, since, as mentioned before, randomization is known to be useful (and, in general, crucial) for the defender to undermine the attack and protect the secret. Thus there seems to be an asymmetry between attacker and defender w.r.t. probabilistic strategies in QIF. Our thesis is that there is indeed an asymmetry, but this does not mean that the attacker has nothing to gain from randomization: when the defender can change his own strategy according to the attacker's actions, it becomes advantageous for the attacker to try to be *unpredictable* and, consequently, adopt a probabilistic strategy. For the defender, while randomization is useful for the same reason, it is also useful because *it reduces the information leakage*, and since information leakage constitutes the gain of the attacker, this reduction influences his strategy. This latter aspect introduces the asymmetry mentioned above.

In the present work, we consider scenarios in which both attacker and defender can make choices that influence the system during the attack. We aim, in particular, at analyzing the attacker's strategies that can maximize information leakage, and the defender's most appropriate strategies to counterattack and keep the system as secure as possible. As argued before, randomization can help both attacker and defender make their moves unpredictable. The most suitable framework for analyzing this kind of interplay is, naturally, game theory, where the use of randomization can be modeled by the notion of *mixed strategies*, and where the interplay between attacker and defender, and their struggle to achieve the best result for themselves, can be modeled in terms of *optimal strategies* and *Nash equilibrium*. It is important to note, however, that one of the two advantages that randomization has for the defender, namely the reduction of information leakage, has no counterpart in standard game theory. Indeed, we demonstrate that this property makes the utility of a mixed strategy be a convex function of the distribution of the defender. In contrast, in standard game theory the utility of a mixed strategy is the expectation of the utility of the pure strategies of each player, and therefore it is an affine function on each of the players' distributions. As a consequence, we need to consider a new kind of games, which we call *information leakage games*, where the utility of a mixed strategy is a function affine on the attacker's strategy, and convex on the defender's. Nevertheless, the fundamental results of game theory, notably the minimax theorem and the existence of Nash equilibria, still hold for our zero-sum leakage games. We also propose algorithms to compute the optimal strategy, namely, the strategies for the attacker and the defender that lead to a Nash equilibrium, where no player has anything to gain by unilaterally changing his own strategy.

For reasoning about information leakage, we employ the well-established information-theoretic framework, which is by far the most used in QIF. A central notion in this model is that of *vulnerability*, which intuitively measures how easily the secret can be discovered (and exploited) by the attacker. For the sake of generality, we adopt the notion of vulnerability as any convex and continuous function [4,2], which has been shown to subsume most previous measures of the QIF literature [2], including *Bayes vulnerability* (a.k.a. min-vulnerability [27,8]), *Shannon entropy* [25], *guessing entropy* [19], and *g-vulnerability* [3].

We note that vulnerability is an expectation measure over the secrets. In this paper we assume the utility to be such average measure, but, in some cases, it could be advantageous for the defender to adopt different strategies depending on the value of the secret. We leave this refinement for future work.

The main contributions of this paper are the following:

- We define a general framework of *information leakage games* to reason about the interplay between attacker and defender in QIF scenarios.
- We prove that, in our framework, the utility is a convex function of the mixed strategy of the defender. To the best of our knowledge, this is a novelty w.r.t. traditional game theory, where the utility of a mixed strategy is defined as expectation of the utilities of the pure strategies.
- We provide methods for finding the solution and the equilibria of leakage games by solving a convex optimization problem.
- We show examples in which Nash equilibria require a mixed strategy. This is, to the best of our knowledge, the first proof in QIF that in some cases the optimal strategy of the attacker must be probabilistic.
- As a case study, we consider the Crowds protocol in a MANET (Mobile Adhoc NETwork). We study the case in which the attacker can add a corrupted node as an attack, the defender can add an honest node as a countermeasure, and we compute the defender component of the Nash equilibrium.

Plan of the paper In Section 2 we review the basic notions of game theory and QIF. In Section 3 we introduce some motivating examples. In section 4 we discuss the difference of our leakage games from those of standard game theory. In Section 5 we prove the convexity of the utility of the defender. In Section 6 we present algorithms for computing the Nash equilibria and optimal strategies for leakage games. In Section 7 we apply our framework to a version of the Crowds protocol. In Section 8 we discuss related work. Section 9 concludes.

2 Preliminaries

In this section we review some basic notions from game theory and QIF.

We use the following notation. Given a set $\mathcal{I}$, we denote by $\mathbb{D}\mathcal{I}$ the *set of all probability distributions* over $\mathcal{I}$. Given $\mu \in \mathbb{D}\mathcal{I}$, its *support* $\mathsf{supp}(\mu)$ is the set of its elements with positive probabilities, i.e., $\mathsf{supp}(\mu) = \{i \in \mathcal{I} : \mu(i) > 0\}$. We write $i \leftarrow \mu$ to indicate that a value $i \in \mathcal{I}$ is sampled from a distribution μ on $\mathcal{I}$.

2.1 Two-player, simultaneous games

We review basic definitions from *two-player games*, a model for reasoning about the behavior of strategic players. We refer to [22] for more details.

In a game, each player has at its disposal a set of *actions* that he can perform, and obtains some payoff (gain or loss) depending on the outcome of the actions chosen by both players. The payoff's value to each player is evaluated using a *utility function*. Each player is assumed to be *rational*, i.e., his choice is driven by the attempt to maximize his own utility. We also assume that the set of possible actions and the utility functions of both players are *common knowledge*.

In this paper we only consider *finite games*, namely the cases in which the set of actions available to each player is finite. Furthermore, we only consider simultaneous games, meaning that each player chooses actions without knowing the actions chosen by the other. Formally, such a game is defined as a tuple[5] $(\mathcal{D}, \mathcal{A}, u_\mathsf{d}, u_\mathsf{a})$, where $\mathcal{D}$ is a nonempty set of *defender's actions*, $\mathcal{A}$ is a nonempty set of *attacker's actions*, $u_\mathsf{d} : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$ is the *defender's utility function*, and $u_\mathsf{a} : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$ is the *attacker's utility function*.

Each player may choose an action deterministically or probabilistically. A *pure strategy* of the defender (resp. attacker) is a deterministic choice of an action, i.e., an element $d \in \mathcal{D}$ (resp. $a \in \mathcal{A}$). A pair (d, a) is a *pure strategy profile*, and $u_\mathsf{d}(d, a)$, $u_\mathsf{a}(d, a)$ represent the defender's and the attacker's utilities.

A *mixed strategy* of the defender (resp. attacker) is a probabilistic choice of an action, defined as a probability distribution $\delta \in \mathbb{D}\mathcal{D}$ (resp. $\alpha \in \mathbb{D}\mathcal{A}$). A pair (δ, α) is called a *mixed strategy profile*. The defender's and the attacker's *expected utility functions* for mixed strategies are defined, respectively, as:

$$U_\mathsf{d}(\delta, \alpha) \overset{\text{def}}{=} \underset{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}}{\mathbb{E}} u_\mathsf{d}(d, a) = \sum_{\substack{d \in \mathcal{D} \\ a \in \mathcal{A}}} \delta(d)\alpha(a)u_\mathsf{d}(d, a)$$

$$U_\mathsf{a}(\delta, \alpha) \overset{\text{def}}{=} \underset{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}}{\mathbb{E}} u_\mathsf{a}(d, a) = \sum_{\substack{d \in \mathcal{D} \\ a \in \mathcal{A}}} \delta(d)\alpha(a)u_\mathsf{a}(d, a)$$

A defender's mixed strategy $\delta \in \mathbb{D}\mathcal{D}$ is a *best response* to an attacker's mixed strategy $\alpha \in \mathbb{D}\mathcal{A}$ if $U_\mathsf{d}(\delta, \alpha) = \max_{\delta' \in \mathbb{D}\mathcal{D}} U_\mathsf{d}(\delta', \alpha)$. Symmetrically, $\alpha \in \mathbb{D}\mathcal{A}$ is a *best response* to $\delta \in \mathbb{D}\mathcal{D}$ if $U_\mathsf{a}(\delta, \alpha) = \max_{\alpha' \in \mathbb{D}\mathcal{A}} U_\mathsf{d}(\delta, \alpha')$. A *mixed-strategy Nash equilibrium* is a profile (δ^*, α^*) such that δ^* is a best response to α^* and vice versa. Namely, no unilateral deviation by any single player provides better utility to that player. If δ^* and α^* are point distributions concentrated on some $d^* \in \mathcal{D}$ and $a^* \in \mathcal{A}$, respectively, then (δ^*, α^*) is a *pure-strategy Nash equilibrium*, and will be denoted by (d^*, a^*). While not all games have a pure strategy Nash equilibrium, every finite game has a mixed strategy Nash equilibrium.

2.2 Zero-sum games and Minimax Theorem

A game $(\mathcal{D}, \mathcal{A}, u_\mathsf{d}, u_\mathsf{a})$ is *zero-sum* if for any $d \in \mathcal{D}$ and any $a \in \mathcal{A}$, $u_\mathsf{d}(d, a) = -u_\mathsf{a}(d, a)$, i.e., the defender's loss is equivalent to the attacker's gain. For brevity, in zero-sum games we denote by u the attacker's utility function u_a, and by U

[5] Following the convention of *security games*, we set the first player to be the defender.

the attacker's expected utility U_{a}.[6] Consequently, the goal of the defender is to minimize U, and the goal of the attacker is to maximize it.

In simultaneous zero-sum games the Nash equilibrium corresponds to the solution of the *minimax* problem (or equivalently, the *maximin* problem), namely, the profile (δ^*, α^*) such that $U(\delta^*, \alpha^*) = \min_\delta \max_\alpha U(\delta, \alpha)$. The von Neumann's minimax theorem ensures that such solution (which always exists) is stable:

Theorem 1 (von Neumann's minimax theorem). *Let $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{Y} \subset \mathbb{R}^n$ be compact convex sets, and $U : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a continuous function such that $U(x, y)$ is convex in $x \in \mathcal{X}$ and concave in $y \in \mathcal{Y}$. Then it is the case that $\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} U(x, y) = \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} U(x, y)$.*

A related property is that, under the conditions of Theorem 1, there exists a *saddle point* (x^*, y^*) s.t., for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, $U(x^*, y) \leq U(x^*, y^*) \leq U(x, y^*)$.

2.3 Quantitative information flow

Finally, we briefly review the standard framework of quantitative information flow, which is used to measure the amount of information leakage in a system.

Secrets and vulnerability A *secret* is some piece of sensitive information the defender wants to protect, such as a user's password, social security number, or current location. The attacker usually only has some partial knowledge about the value of a secret, represented as a probability distribution on secrets called a *prior*. We denote by $\mathcal{X}$ the set of possible secrets, and we typically use π to denote a prior belonging to the set $\mathbb{D}\mathcal{X}$ of probability distributions over $\mathcal{X}$.

The *vulnerability* of a secret is a measure of the utility of the attacker's knowledge about the secret. In this paper we consider a very general notion of vulnerability, following [2], and define a vulnerability $\mathbb{V}$ to be any continuous and convex function of type $\mathbb{D}\mathcal{X} \to \mathbb{R}$. It has been shown in [2] that these functions coincide with the set of g-vulnerabilities, and are, in a precise sense, the most general information measures w.r.t. a set of basic axioms. [7]

Channels, posterior vulnerability, and leakage Systems can be modeled as information theoretic channels. A *channel* $C : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a function in which $\mathcal{X}$ is a set of *input values*, $\mathcal{Y}$ is a set of *output values*, and $C(x, y)$ represents the conditional probability of the channel producing output $y \in \mathcal{Y}$ when input $x \in \mathcal{X}$ is provided. Every channel C satisfies $0 \leq C(x, y) \leq 1$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and $\sum_{y\in\mathcal{Y}} C(x, y) = 1$ for all $x \in \mathcal{X}$.

[6] Conventionally in game theory the utility u is set to be that of the first player, but we prefer to look at the utility from the point of view of the attacker to be in line with the definition of utility as *vulnerability*, as we will introduce in Section 2.3.

[7] More precisely, if posterior vulnerability is defined as the expectation of the vulnerability of posterior distributions, the measure respects the data-processing inequality and yields non-negative leakage iff vulnerability is convex.

A distribution $\pi \in \mathbb{D}\mathcal{X}$ and a channel C with inputs $\mathcal{X}$ and outputs $\mathcal{Y}$ induce a joint distribution $p(x,y) = \pi(x)C(x,y)$ on $\mathcal{X} \times \mathcal{Y}$, with marginal probabilities $p(x) = \sum_y p(x,y)$ and $p(y) = \sum_x p(x,y)$, and conditional probabilities $p(x|y) = {}^{p(x,y)}/_{p(y)}$ if $p(y) \neq 0$. For a given y (s.t. $p(y) \neq 0$), the conditional probabilities $p(x|y)$ for each $x \in \mathcal{X}$ form the *posterior distribution* $p_{X|y}$.

A channel C in which $\mathcal{X}$ is a set of secret values and $\mathcal{Y}$ is a set of observable values produced by a system can be used to model computations on secrets. Assuming the attacker has prior knowledge π about the secret value, knows how a channel C works, and can observe the channel's outputs, the effect of the channel is to update the attacker's knowledge from a prior π to a collection of posteriors $p_{X|y}$, each occurring with probability $p(y)$.

Given a vulnerability $\mathbb{V}$, a prior π, and a channel C, the *posterior vulnerability* $\mathbb{V}[\pi, C]$ is the vulnerability of the secret after the attacker has observed the output of C. Formally: $\mathbb{V}[\pi, C] \overset{\text{def}}{=} \sum_{y \in \mathcal{Y}} p(y)\mathbb{V}\left[p_{X|y}\right]$.

The *information leakage* of a channel C under a prior π is a comparison between the vulnerability of the secret before the system was run—called the *prior* vulnerability—and the posterior vulnerability of the secret. The leakage reflects by how much the observation of the system's outputs increases the utility of the attacker's knowledge about the secret. It can be defined either *additively* ($\mathbb{V}[\pi, C] - \mathbb{V}[\pi]$), or *multiplicatively* (${}^{\mathbb{V}[\pi,C]}/_{\mathbb{V}[\pi]}$).

3 A motivating example

We present some simple examples to motivate our information leakage games.

3.1 The two-millionaires problem

The "two-millionaires problem" was introduced by Yao in [33]. In the original formulation, there are two "millionaires", Alice and Don, who want to discover who is the richest among them, but neither wants to reveal to the other the amount of money that he or she has.

We consider a (conceptually) asymmetric variant of this problem, where Alice is the attacker and Don is the defender. Don wants to learn whether or not he is richer than Alice, but does not want Alice to learn anything about the amount x of money he has. To this purpose, Don sends x to a trusted server Jeeves, who in turn asks Alice, privately, what is her amount a of money. Jeeves then checks which among x and a is greater, and sends the result y back to Don.[8] However, Don is worried that Alice may intercept Jeeves' message containing the result of the comparison, and exploit it to learn more accurate information about x by tuning her answer a appropriately (since, given y, Alice can deduce whether a is an upper or lower bound on x). We assume that Alice may get to know Jeeves' reply, but not the messages from Don to Jeeves.

[8] The reason to involve Jeeves is that Alice may not want to reveal a to Don, either.

We will use the following information-flow terminology: the information that should remain secret (to the attacker) is called *high*, and what is visible to (and possibly controllable by) the attacker is called *low*. Hence, in the program run by Jeeves a is a *low input* and x is a *high input*. The result y of the comparison (since it may be intercepted by the attacker) is a *low output*. The problem is to avoid the *flow of information* from x to y (given a).

One way to mitigate this problem is to use randomization. Assume that Jeeves provides two different programs to ensure the service. Then, when Don sends his request to Jeeves, he can make a random choice d among the two programs 0 and 1, sending d to Jeeves along with the value x. Now if Alice intercepts the result y, it will be less useful to her since she does not know which of the two programs has been run. As Don of course knows which program was run, the result y will still be just as useful to him. [9]

In order to determine the best probabilistic strategy that Don should apply to select the program, we analyze the problem from a game-theoretic perspective. For simplicity, we assume that x and a both range in $\{0,1\}$. The two alternative programs that Jeeves can run are shown in Table 1.

`Program 0`	`Program 1`
`High Input:` $x \in \{0,1\}$	`High Input:` $x \in \{0,1\}$
`Low Input:` $a \in \{0,1\}$	`Low Input:` $a \in \{0,1\}$
`Output:` $y \in \{T,F\}$	`Output:` $y \in \{T,F\}$
return $x \leq a$	return $x \geq a$

Table 1. The two programs run by Jeeves.

The combined choices of Alice and Don determine how the system behaves. Let $\mathcal{D} = \{0,1\}$ represent Don's possible choices, i.e., the program to run, and $\mathcal{A} = \{0,1\}$ represent Alice's possible choices, i.e., the value of the low input a. We shall refer to the elements of $\mathcal{D}$ and $\mathcal{A}$ as *actions*. For each possible combination of actions d and a, we can construct a channel C_{da} with inputs $\mathcal{X} = \{0,1\}$ (the set of possible high input values) and outputs $\mathcal{Y} = \{T,F\}$ (the set of possible low output values), modeling the behavior of the system *from the point of view of the attacker*. Intuitively, each channel entry $C_{da}(x,y)$ is the probability that the program run by Jeeves (which is determined by d) produces output $y \in \mathcal{Y}$ given that the high input is $x \in \mathcal{X}$ and that the low input is a. The resulting four channel matrices are represented in Table 2. Note that channels C_{01} and C_{10} do not leak any information about the input x (output y is constant), whereas channels C_{00} and C_{11} completely reveal x (output y is in a bijection with x).

We want to investigate how the defender's and the attacker's strategies influence the leakage of the system. For that we can consider the (simpler) notion of posterior vulnerability, since, for a given prior, the value of leakage is in a

[9] Note that d should not be revealed to the attacker: although d is not sensitive information in itself, knowing it would help the attacker figure out the value of x.

$a = 0$ $\quad$ $a = 1$

$d = 0 \quad (x \leq a?)$

C_{00}	$y = T$	$y = F$
$x = 0$	1	0
$x = 1$	0	1

C_{01}	$y = T$	$y = F$
$x = 0$	1	0
$x = 1$	1	0

$d = 1 \quad (x \geq a?)$

C_{10}	$y = T$	$y = F$
$x = 0$	1	0
$x = 1$	1	0

C_{11}	$y = T$	$y = F$
$x = 0$	0	1
$x = 1$	1	0

Table 2. The two-millionaires system, from the point of view of the attacker.

one-to-one (monotonic) correspondence with the value of posterior vulnerability. For this example, we consider posterior Bayes vulnerability [8,27], defined as $\mathbb{V}[\pi, C] = \sum_y \max_x C(x, y)\pi(x)$. Intuitively, Bayes vulnerability measures the probability of the adversary guessing the secret correctly in one try, and it can be shown that $\mathbb{V}[\pi, C]$ coincides with the converse of the Bayes error.

For simplicity, we assume a uniform prior distribution π_u. It has been shown that, in this case, the posterior Bayes vulnerability of a channel C can be computed as the sum of the greatest elements of each column of C, divided by the high input-domain size [7]. Namely, $\mathbb{V}[\pi_u, C] = \sum_y {}^{\max_x C(x,y)}/_{|\mathcal{X}|}$. It is easy to see that we have $\mathbb{V}[\pi_u, C_{00}] = \mathbb{V}[\pi_u, C_{11}] = 1$ and $\mathbb{V}[\pi_u, C_{01}] = \mathbb{V}[\pi_u, C_{10}] = {}^1/_2$. Thus we obtain the utility table shown in Table 3, which is similar to that of the well-known "matching-pennies" game.

As in standard game theory, there may not exist an optimal pure strategy profile. The defender as well as the attacker can then try to minimize/maximize the system's vulnerability by adopting a mixed strategy δ and α, respectively. A crucial task is *evaluating the vulnerability* of the system under such mixed strategies. This evaluation is naturally performed from the point of view of the attacker, who knows his own choice a, but *not the defender's choice* d. As a consequence, the attacker sees the system as the convex combination $C_{\delta a} = \sum_d \delta(d)\, C_{ad}$, i.e., a probabilistic choice between the channels representing the defender's actions. Hence, the overall vulnerability of the system will be given by the vulnerability of $C_{\delta a}$, averaged over all attacker's actions.

$\mathbb{V}$	$a = 0$	$a = 1$
$d = 0$	1	${}^1/_2$
$d = 1$	${}^1/_2$	1

Table 3. Utility table for the two-millionaires game.

We now define formally the ideas illustrated above.

Definition 1. *An* information-leakage game *is a tuple* $(\mathcal{D}, \mathcal{A}, C)$ *where* $\mathcal{D}, \mathcal{A}$ *are the sets of actions of the attacker and the defender, respectively, and* $C = \{C_{da}\}_{da}$ *is a family of channel matrices indexed on pairs of actions* $d \in \mathcal{D}, a \in \mathcal{A}$. *For a given vulnerability* $\mathbb{V}$ *and prior* π, *the utility of a pure strategy* (d, a) *is given by* $\mathbb{V}[\pi, C_{da}]$. *The utility* $\mathbb{V}(\delta, \alpha)$ *of a mixed strategy* (δ, α) *is defined as:*

$$\mathbb{V}(\delta, \alpha) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{a \leftarrow \alpha} \mathbb{V}[\pi, C_{\delta a}] = \textstyle\sum_a \alpha(a)\, \mathbb{V}[\pi, C_{\delta a}] \quad \textit{where} \quad C_{\delta a} \stackrel{\text{def}}{=} \textstyle\sum_d \delta(d)\, C_{ad}$$

In our example, δ is represented by a single number p: the probability that the defender chooses $d = 0$ (i.e., Program 0). From the point of view of the attacker, Utility table for $a = 0$

C_{p0}	$y = T$	$y = F$
$x = 0$	1	0
$x = 1$	$1 - p$	p

Utility table for $a = 1$

C_{p1}	$y = T$	$y = F$
$x = 0$	p	$1 - p$
$x = 1$	1	0

Table 4. The two-millionaires mixed strategy of the defender, from the point of view of the attacker, where p is the probability the defender picks action $d = 0$.

once he has chosen a, the system will look like a channel $C_{pa} = p\, C_{0a} + (1-p)\, C_{1a}$. For instance, in the case $a = 0$, if x is 0 Jeeves will send T with probability 1, but, if x is 1, Jeeves will send F with probability p and T with probability $1 - p$. Similarly for $a = 1$. Table 4 summarizes the various channels modelling the attacker's point of view. It is easy to see that $\mathbb{V}[\pi_u, C_{p0}] = (1+p)/2$ and $\mathbb{V}[\pi_u, C_{p1}] = (2-p)/2$. In this case $\mathbb{V}[\pi_u, C_{pa}]$ coincides with the expected utility with respect to p, i.e., $\mathbb{V}[\pi_u, C_{pa}] = p\, \mathbb{V}[\pi_u, C_{0a}] + (1-p)\, \mathbb{V}[\pi_u, C_{1a}]$.

Assume now that the attacker choses $a = 0$ with probability q and $a = 1$ with probability $1 - q$. The utility is obtained as expectation with respect to the strategy of the attacker, hence the total utility is: $\mathbb{V}(p, q) = q\,(1+p)/2 + (1-p)\,(2-p)/2$, which is affine in both p and q. By applying standard game-theoretic techniques, we derive that the optimal strategy is $(p^*, q^*) = (1/2, 1/2)$.

In the above example, things work just like in standard game theory. However, in the next section we will show an example that fully exposes the difference of our games with respect to those of standard game theory.

3.2 Binary sum

The previous example is an instance of a general scenario in which a user, Don, delegates to a server, Jeeves, a certain computation that requires also some input from other users. Here we will consider another instance, in which the function to be computed is the binary sum $\oplus$. We assume Jeeves provides the programs in Table 5. The resulting channel matrices are represented in Table 6.

```
Program 0
High Input: $x \in \{0,1\}$
Low Input: $a \in \{0,1\}$
Output: $y \in \{0,1\}$
return $x \oplus a$
```

```
Program 1
High Input: $x \in \{0,1\}$
Low Input: $a \in \{0,1\}$
Output: $y \in \{0,1\}$
return $x \oplus a \oplus 1$
```

Table 5. The two programs for $\oplus$ and its complement.

We consider again Bayes posterior vulnerability as utility. It is easy to see that we have $\mathbb{V}[\pi_u, C_{00}] = \mathbb{V}[\pi_u, C_{11}] = \mathbb{V}[\pi_u, C_{01}] = \mathbb{V}[\pi_u, C_{10}] = 1$. Thus for the pure strategies we obtain the utility table shown in Table 7. This means that all pure strategies have the same utility 1 and therefore they are all equivalent. In standard game theory this would mean that also the mixed strategies have the same utility 1, since they are defined as expectation. In our case, however, the utility of a mixed strategy of the defender is convex on the

$d = 0 \quad (x \oplus a)$

$a = 0$

C_{00}	$y=0$	$y=1$
$x=0$	1	0
$x=1$	0	1

$a = 1$

C_{01}	$y=0$	$y=1$
$x=0$	0	1
$x=1$	1	0

$d = 1 \quad (x \oplus a \oplus 1)$

$a = 0$

C_{10}	$y=0$	$y=1$
$x=0$	0	1
$x=1$	1	0

$a = 1$

C_{11}	$y=0$	$y=1$
$x=0$	1	0
$x=1$	0	1

Table 6. The binary-sum system, from the point of view of the attacker.

distribution, so it may be convenient for the defender to adopt a mixed strategy. Let $p, 1-p$ be the probabilities of the defender choosing Program 0 and Program 1, respectively. From the point of view of the attacher, for each of his choices of a, the system will appear as the probabilistic channel C_{pa} represented in Table 8.

$\mathbb{V}$	$a=0$	$a=1$
$d=0$	1	1
$d=1$	1	1

Table 7. Utility table for the binary-sum game.

$a = 0$

C_{p0}	$y=T$	$y=F$
$x=0$	p	$1-p$
$x=1$	$1-p$	p

$a = 1$

C_{p1}	$y=T$	$y=F$
$x=0$	$1-p$	p
$x=1$	p	$1-p$

Table 8. The binary-sum mixed strategy of the defender, from the point of view of the attacker, where p is the probability the defender picks action $d = 0$.

It is easy to see that $\mathbb{V}[\pi_u, C_{p0}] = \mathbb{V}[\pi_u, C_{p1}] = 1 - p$ if $p \leq 1/2$, and $\mathbb{V}[\pi_u, C_{p0}] = \mathbb{V}[\pi_u, C_{p1}] = p$ if $p \geq 1/2$. On the other hand, with respect to a mixed strategy of the attacker the utility is still defined as expectation. Since in this case the utility is the same for $a = 0$ and $a = 1$, it remains the same for any strategy of the attacker. Formally, $\mathbb{V}(p, q) = q\, \mathbb{V}[\pi_u, C_{p0}] + (1 - q)\, \mathbb{V}[\pi_u, C_{p1}] = \mathbb{V}[\pi_u, C_{p0}]$, which does not depend on q and it is minimum for $p = 1/2$. We conclude that the point of equilibrium is $(p^*, q^*) = (1/2, q^*)$ for any value of q^*.

4 Leakage games vs. standard game theory models

In this section we explain the differences between our information leakage games and standard approaches to game theory. We discuss: (1) why the use of vulnerability as a utility function makes our games non-standard w.r.t. von Neumann-Morgenstern's treatment of utility, (2) why the use of concave utility functions to model risk-averse players does not capture the behavior of the attacker in our games, and (3) how our games differ from traditional convex-concave games.

4.1 The von Neumann-Morgenstern's treatment of utility

In their treatment of utility, von Neumann and Morgenstern [29] demonstrated that the utility of a mixed strategy equals the expected utility of the corresponding pure strategies when a set of axioms is satisfied for player's preferences over probability distributions (a.k.a. *lotteries*) on payoffs. Since in our leakage games the utility of a mixed strategy is *not* the expected utility of the corresponding pure strategies, it is relevant to identify how exactly our framework fails to meet von Neumann and Morgenstern (vNM) axioms.

Let us first introduce some notation. Given two mixed strategies σ, σ' for a player, we write $\sigma \preceq \sigma'$ (or $\sigma' \succeq \sigma$) when the player prefers σ' over σ, and $\sigma \sim \sigma'$ when the player is indifferent between σ and σ'. Then, the vNM axioms can be formulated as follows [24]. For every mixed strategies σ, σ' and σ'':

A1 *Completeness*: it is either the case that $\sigma \preceq \sigma'$, $\sigma \succeq \sigma'$, or $\sigma \sim \sigma'$.
A2 *Transitivity*: if $\sigma \preceq \sigma'$ and $\sigma' \preceq \sigma''$, then $\sigma \preceq \sigma''$.
A3 *Continuity*: if $\sigma \preceq \sigma' \preceq \sigma''$, then there exist $p \in [0,1]$ s.t. $p\,\sigma + (1-p)\,\sigma'' \sim \sigma'$.
A4 *Independence*: if $\sigma \preceq \sigma'$ then for any σ'' and $p \in [0,1]$ we have $p\,\sigma + (1-p)\,\sigma'' \preceq p\,\sigma' + (1-p)\,\sigma''$.

For any fixed prior π on secrets, the utility function $u(C) = \mathbb{V}[\pi, C]$ is a total function on $\mathcal{C}$ ranging over the reals, and therefore it satisfies axioms A1, A2 and A3 above. However, $u(C)$ does not satisfy A4, as the next example illustrates.

Example 1. Consider the following three channel matrices from input set $\mathcal{X} = \{0, 1\}$ to output set $\mathcal{Y} = \{0, 1\}$, where ϵ is a small positive constant:

C_1	$y=0$	$y=1$
$x=0$	$1-\epsilon$	ϵ
$x=1$	ϵ	$1-\epsilon$

C_2	$y=0$	$y=1$
$x=0$	1	0
$x=1$	0	1

C_3	$y=0$	$y=1$
$x=0$	0	1
$x=1$	1	0

If we focus on Bayes vulnerability, it is clear that an attacker would prefer C_2 over C_1, i.e., $C_1 \preceq C_2$. However, for the probability $p = 1/2$ we would have:

$p\,C_1 + (1-p)\,C_3$	$y=0$	$y=1$
$x=0$	$(1-\epsilon)/2$	$(1+\epsilon)/2$
$x=1$	$(1+\epsilon)/2$	$(1-\epsilon)/2$

and

$p\,C_2 + (1-p)\,C_3$	$y=0$	$y=1$
$x=0$	$1/2$	$1/2$
$x=1$	$1/2$	$1/2$

Since channel $p\,C_1 + (1-p)\,C_3$ clearly reveals no less information about the secret than channel $p\,C_2 + (1-p)\,C_3$, we have that $p\,C_1 + (1-p)\,C_3 \succeq p\,C_2 + (1-p)\,C_3$, and the axiom of independence is not satisfied.

It is actually quite natural that vulnerability does not satisfy independence: a convex combination of two "leaky" channels (i.e., high-utility outcomes) can produce a "non-leaky" channel (i.e., a low-utility outcome). As a consequence, the traditional game-theoretic approach to the utility of mixed strategies does not apply to our information leakage games. However the existence of Nash equilibria is still granted, as we will see in Section 5, Corollary 1.

4.2 Risk functions

At a first glance, it may seem that our information leakage games could be expressed with some clever use of the concept of *risk-averse players* (in our case, the attacker), which is also based on convex utility functions (cf. [22]). There is, however, a crucial difference: in the models of risk-averse players, the utility function is convex *on the payoff of an outcome of the game*, but the utility of a mixed strategy is still *the expectation of the utilities of the pure strategies*, i.e., it is linear on the distributions. On the other hand, the utility of mixed strategies in our information leakage games is *convex on the distribution of the defender*. This difference arises precisely because in our games utility is defined as the vulnerability of the channel perceived by the attacker, and, as we discussed, this creates an extra layer of uncertainty for the attacker.

4.3 Convex-concave games

Another well-known model from standard game-theory is that of convex-concave games, in which each of two players can choose among a continuous set of actions yielding convex utility for one player, and concave for the other. In this kind of game the Nash equilibria are given by *pure strategies* for each player.

A natural question would be why not represent our systems as convex-concave games in which the pure actions of players are the mixed strategies of our leakage games. Namely, the real values p and q that uniquely determine the defender's and the attacker's mixed strategies, respectively, in the two-millionaires game of Section 3, could be taken to be the choices of pure strategies in a convex-concave game in which the set of actions for each player is the real interval $[0,1]$.

This mapping from our games to convex-concave games, however, would not be natural. One reason is that utility is still defined as expectation in the standard convex-concave games, in contrast to our games. Consider two strategies p_1 and p_2 with utilities u_1 and u_2, respectively. If we mix them using the coefficient $q \in [0,1]$, the resulting strategy $q\,p_1 + (1-q)\,p_2$ will have utility $u = q\,u_1 + (1-q)\,u_2$ in the standard convex-concave game, while in our case the utility would in general be strictly smaller than u. The second reason is that a pure action corresponding to a mixed strategy may not always be realizable. To illustrate this point, consider again the two-millionaires game, and the defender's mixed strategy consisting in choosing Program 0 with probability p and Program 1 with probability $1-p$. The requirement that the defender has a pure action corresponding to p implies the existence of a program (on Jeeves' side) that makes internally a probabilistic choice with bias p and, depending on the outcome, executes Program 0 or Program 1. However, it is not granted that Jeeves disposes of such a program. Furthermore, Don would not know what choice has actually been made, and thus the program would not achieve the same functionality, i.e., let Don know who is the richest. (Note that Jeeves should not communicate to Don the result of the choice, because of the risk that Alice intercepts it.) This latter consideration underlines a key practical aspect of leakage games, namely, the defender's advantage over the attacker due to his knowledge of the result of his own random choice (in a mixed strategy). This advantage would be lost in a convex-concave representation of the game since the random choice would be "frozen" in its representation as a pure action.

5 Convexity of vulnerability w.r.t. channel composition

In this section we show that posterior vulnerability is a convex function of the strategy of the defender. In other words, given a set of channels, and a probability distribution over them, the vulnerability of the composition of these channels according to the distribution is smaller than or equal to the composition of their vulnerabilities. As a consequence, we derive the existence of the Nash equilibria.

In order to state this result formally, we introduce the following notation: given a channel matrix C and a scalar a, $a\,C$ is the matrix obtained by multiplying every element of C by a. Given two *compatible* channel matrices C_1 and C_2, namely matrices with the same indices of rows and columns[10], $C_1 + C_2$ is obtained by adding the cells of C_1 and C_2 with same indices. Note that if μ is a probability distribution on $\mathcal{I}$, then $\sum_{i\in\mathcal{I}} \mu(i)\, C_i$ is a channel matrix.

Theorem 2 (Convexity of vulnerability w.r.t. channel composition). *Let $\{C_i\}_{i\in\mathcal{I}}$ be a family of compatible channels, and μ be a distribution on $\mathcal{I}$. Then, for every prior distribution π, and every vulnerability $\mathbb{V}$, the corresponding posterior vulnerability is convex w.r.t. to channel composition. Namely, for any probability distribution μ on $\mathcal{I}$, we have $\mathbb{V}[\pi, \sum_i \mu(i)\, C_i] \leq \sum_i \mu(i)\, \mathbb{V}[\pi, C_i]$.*

Proof. Define $p(y) = \sum_x \pi(x) \sum_i \mu(i)\, C_i(x,y)$. Then:

$$\begin{aligned}
\mathbb{V}[\pi, \textstyle\sum_i \mu(i)\, C_i] &= \textstyle\sum_y p(y)\, \mathbb{V}\Big[\frac{\pi(\cdot)\, \sum_i \mu(i)\, C_i(\cdot,y)}{p(y)}\Big] && \text{(by def. of posterior } \mathbb{V}) \\
&= \textstyle\sum_y p(y)\, \mathbb{V}\Big[\sum_i \mu(i)\, \frac{\pi(\cdot)\, C_i(\cdot,y)}{p(y)}\Big] \\
&\leq \textstyle\sum_y p(y)\, \sum_i \mu(i)\, \mathbb{V}\Big[\frac{\pi(\cdot)\, C_i(\cdot,y)}{p(y)}\Big] && (*) \\
&= \textstyle\sum_i \mu(i)\, \sum_y p(y)\, \mathbb{V}\Big[\frac{\pi(\cdot)\, C_i(\cdot,y)}{p(y)}\Big] \\
&= \textstyle\sum_i \mu(i)\, \mathbb{V}[\pi, C_i] && \text{(by def. of posterior } \mathbb{V})
\end{aligned}$$

where (*) follows from the convexity of $\mathbb{V}$ w.r.t. the prior (cf. Section 2.3). □

The existence of Nash equilibria immediately follows from the above theorem:

Corollary 1. *For any (zero-sum) information-leakage game there exist a Nash equilibrium, which in general is given by a mixed strategy.*

Proof. Given a mixed strategy (δ, α), the utility $\mathbb{V}(\delta, \alpha)$ given in Definition 1 is affine (hence concave) on α. Furthermore, by Theorem 2, $\mathbb{V}(\delta, \alpha)$ is convex on δ. Hence we can apply the von Neumann's minimax theorem (Section 2.2), which ensures the existence of a saddle point, i.e., a Nash equilibrium. □

[10] Note that two channel matrices with different column indices can always be made compatible by adding appropriate columns with 0-valued cells in each of them.

6 Computing equilibria of information leakage games

Our goal is to solve information leakage games, in which the success of an attack a and a defence d is measured by a vulnerability measure $\mathbb{V}$. The attack/defence combination is a pure strategy profile (d, a) in this game, and is associated with a channel C_{da} modeling the behavior of the system. The attacker clearly knows his own choice a, whereas the defender's choice is assumed to be hidden. Hence the utilty of a mixed strategy profile (δ, α) will be given by Definition 1, that is:

$$\mathbb{V}(\delta, \alpha) = \textstyle\sum_a \alpha(a)\, \mathbb{V}[\pi, \sum_d \delta(d)\, C_{da}]$$

Note that $\mathbb{V}(\delta, \alpha)$ is convex on δ and affine on α, hence Theorem 1 guarantees the existence of an equilibrium (i.e. a saddle-point) (δ^*, α^*) which is a solution of both the minimax and the maximin problems. The goal in this section is to compute a) a δ^* that is part of an equilibrium, which is important in order to optimize the defence, and b) the utility $\mathbb{V}(\delta^*, \alpha^*)$, which is important to provide an upper bound on the effectiveness of an attack when δ^* is applied.

This is a convex-concave optimization problem for which various methods have been proposed in the literature. If $\mathbb{V}$ is twice differentiable (and satisfies a few extra conditions) then the Newton method can be applied [6]; however, many such measures, most notably Bayes-vulnerability, our main vulnerability measure of interest, are not differentiable. For non-differentiable functions, [21] proposes a subgradient method that iterates on both δ, α at each step. We have applied this method and it does indeed converge to $\mathbb{V}(\delta^*, \alpha^*)$, with one important caveat: the solution δ that it produces is not necessarily an equilibrium (note that $\mathbb{V}(\delta, \alpha) = \mathbb{V}(\delta^*, \alpha^*)$ does not guarantee that (δ, α) is a saddle point). Producing an optimal δ^* is of vital importance in our case.

The method we propose is based on the idea of solving the minimax problem $\hat{\delta} = \operatorname{argmin}_\delta \max_\alpha \mathbb{V}(\delta, \alpha)$, since its solution is guaranteed to be part of an equilibrium.[11] To solve this problem, we exploit the fact that $\mathbb{V}(\delta, \alpha)$ is affine on α (not just concave). For a fixed δ, maximizing $\sum_a \alpha(a)\, \mathbb{V}[\pi, \sum_d \delta(d)\, C_{da}]$ simply involves picking the a with the highest $\mathbb{V}[\pi, \sum_d \delta(d)\, C_{da}]$ and assigning probability 1 to it. Hence, our minimax problem is equivalent to $\hat{\delta} = \operatorname{argmin}_\delta f(\delta)$ where $f(\delta) = \max_a \mathbb{V}[\pi, \sum_d \delta(d)\, C_{da}]$; that is, we have to minimize the max of finitely many convex functions, with δ being the only variables.

For this problem we can employ the *projected subgradient* method, given by:

$$\delta^{(k+1)} = P(\delta^{(k)} - \alpha_k g^{(k)})$$

where $g^{(k)}$ is any subgradient of f on $\delta^{(k)}$ [5]. Note that the subgradient of a finite max is simply a subgradient of any branch that gives the max at that point. $P(x)$ is the projection of x on the domain of f; in our case the domain is the probability simplex, for which there exist efficient algorithms for computing

[11] Note that this is true only for δ, the α-solution of the minimax problem is not necessarily part of an equilibrium; we need to solve the maximin problem for this.

the projection [30]. Finally α_k is a step-size, for which various choices guarantee convergence [5]. In our experiments we found $\alpha_k = 0.1/\sqrt{k}$ to perform well.

As the starting point $\delta^{(1)}$ we take the uniform distribution; moreover the solution can be approximated to within an arbitrary $\epsilon > 0$ by using the stopping criterion of [5, Section 3.4]. Note that the obtained $\hat{\delta}$ approximates the equilibrium strategy δ^*, while $f(\hat{\delta})$ approximates $\mathbb{V}(\delta^*, \alpha^*)$. Hence we achieve both desired goals, as formally stated in the following result.

Proposition 1. *If $\mathbb{V}$ is Lipschitz then the subgradient method discussed in this section converges to a δ^* that is part of an equilibrium of the game. Moreover, let $\hat{\delta}$ be the solution computed within a given $\epsilon > 0$, and let (δ^*, α^*) be an equilibrium. Then it holds that:*

$$\mathbb{V}(\hat{\delta}, \alpha) - \epsilon \leq \mathbb{V}(\hat{\delta}, \alpha^*) \leq \mathbb{V}(\delta, \alpha^*) + \epsilon \qquad \forall \delta, \alpha$$

which also implies that $f(\hat{\delta}) - \mathbb{V}(\delta^*, \alpha^*) \leq \epsilon$.

Proof. (*Sketch*) $\operatorname{argmin}_\delta f(\delta)$ is equivalent to the minimax problem whose δ-solution is guaranteed to be part of an equilibrium. Convergence is ensured by the subgradient method under the Lipschitz condition, and given that $||\delta^{(1)} - \delta^*||$ is bounded by the distance between the uniform and a point distribution. □

Finally, of particular interest is the Bayes-vulnerability measure [8,27], given by $\mathbb{V}[\pi, C] = \sum_y \max_x \pi(x)\, C(x, y)$, since it is widely used to provide an upper bound to all other measures of information leakage [3]. For this measure, $\mathbb{V}$ is Lipschitz and the subgradient vector $g^{(k)}$ is given by $g_d^{(k)} = \sum_y \pi(x_y^*)\, C_{da^*}(x_y^*, y)$ where a^*, x_y^* are the ones giving the max in the branches of $f(\delta^{(k)})$. Note also that, since f is piecewise linear, the convex optimization problem can be transformed into a linear one using a standard technique, and then solved by linear programming. However, due to the large number of max branches of $\mathbb{V}$, this conversion can be a problem with a huge number of constraints. In our experiments we found that the subgradient method described above is significantly more efficient than linear programming.

Note also that, although the subgradient method is general, it might be impractical in applications where the number of attacker or defender actions is very large. Application-specific methods could offer better scalability in such cases, we leave the development of such methods as future work.

7 Case study

In this section, we apply our game-theoretic analysis to the case of anonymous communication on a mobile ad-hoc network (MANET). In such a network, nodes can move in space and communicate with other nearby nodes. We assume that nodes can also access some global (wide area) network, but such connections are neither anonymous nor trusted. Consider, for instance, smartphone users who can access the cellular network, but do not trust the network provider. The goal is to send a message on the global network without revealing the sender's identity to the provider. For that, users can form a MANET using some short-range communication method (e.g., bluetooth), and take advantage of the local network to achieve anonymity on the global one.

Crowds [23] is a protocol for anonymous communication that can be employed on a MANET for this purpose. Note that, although more advanced systems for anonymous communication exist (e.g. Onion Routing), the simplicity of Crowds makes it particularly appeling for MANETs. The protocol works as follows: the *initiator* (i.e., the node who wants to send the message) selects some other node connected to him (with uniform probability) and forwards the request to him. A *forwarder*, upon receiving the message, performs a probabilistic choice: with probability p_f he keeps forwarding the message (again, by selecting uniformly a user among the ones connected to him), while with probability $1 - p_f$ he delivers the message on the global network. Replies, if any, can be routed back to the initiator following the same path in reverse order.

Anonymity comes from the fact that the *detected* node (the last in the path) is most likely not the initiator. Even if the attacker knows the network topology, he can infer that the initiator is most likely a node close to the detected one, but if there are enough nodes we can achieve some reasonable anonymity guarantees. However, the attacker can gain an important advantage by deploying a node himself and participating to the MANET. When a node forwards a message to this *corrupted* node, this action is observed by the attacker and increases the probability of that node being the initiator. Nevertheless, the node can still claim that he was only forwarding the request for someone else, hence we still provide some level of anonymity. By modeling the system as a channel, and computing its posterior Bayes vulnerability [27], we get the probability that the attacker guesses correctly the identity of the initiator, after performing his observation.

In this section we study a scenario of 30 nodes deployed in an area of 1km×1km, in the locations illustrated in Fig. 1. Each node can communicate with others up to a distance of 250 meters, forming the network topology shown in the graph. To compromise the anonymity of the system, the attacker plans to deploy a corrupted node in the network; the question is which is the *optimal location* for such a node. The answer is far from trivial: on the one side being connected to many nodes is beneficial, but at the same time these nodes need to be "vulnerable", being close to a highly connected clique might not be optimal. At the same time, the administrator of the network is suspecting that the attacker is about to deploy a corrupted node. Since this action cannot be avoided (the network is ad-hoc), a countermeasure is to deploy a *deliverer* node at a location that is most vulnerable. Such a node

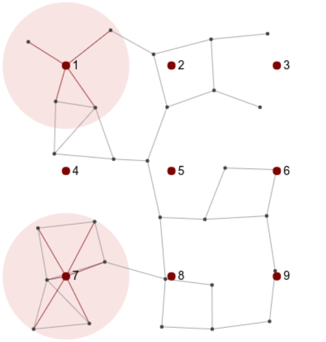

Fig. 1. A MANET with 30 users in a 1km×1km area.

directly delivers all messages forwarded to it on the global network; since it never generates messages its own anonymity is not an issue, it only improves the anonymity of the other nodes. Moreover, since it never communicates in the local network its operation is invisible to the attacker. But again, the optimal location for the new deliverer node is not obvious, and most importantly, the choice depends on the choice of the attacker.

To answer these questions, we model the system as a game where the actions of attacker and defender are the locations of newly deployed corrupted and honest nodes, respectively. We assume that the possible locations for new nodes are the nine ones shown in Fig. 1. For each pure strategy profile (d, a), we construct the corresponding network and use the PRISM model checker to construct the corresponding channel C_{da}, using a model similar to the one of [26]. Note that the computation considers the specific network topology of Fig. 1, which reflects the positions of each node at the time when the attack takes place; the corresponding channels need to be recomputed if the network changes in the future. As leakage measure we use the posterior Bayes vulnerability (with uniform prior π), which is the attacker's probability of correctly guessing the initiator given his observation in the protocol. According to Definition 1, for a mixed strategy profile (δ, α) the utility is $\mathbb{V}(\delta, \alpha) = \mathbb{E}_{a \leftarrow \alpha} \mathbb{V}[\pi, C_{\delta a}]$.

Defender's action	Attacker's action								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	7.38	6.88	6.45	6.23	7.92	6.45	9.32	7.11	6.45
2	9.47	6.12	6.39	6.29	7.93	6.45	9.32	7.11	6.45
3	9.50	6.84	5.46	6.29	7.94	6.45	9.32	7.11	6.45
4	9.44	6.92	6.45	5.60	7.73	6.45	9.03	7.11	6.45
5	9.48	6.91	6.45	6.09	6.90	6.13	9.32	6.92	6.44
6	9.50	6.92	6.45	6.29	7.61	5.67	9.32	7.11	6.24
7	9.50	6.92	6.45	5.97	7.94	6.45	7.84	7.10	6.45
8	9.50	6.92	6.45	6.29	7.75	6.45	9.32	6.24	6.45
9	9.50	6.92	6.45	6.29	7.92	6.24	9.32	7.11	5.68

Table 9. Utility for each pure strategy profile.

The utilities (posterior Bayes vulnerability %) for each pure profile are displayed in Table 9. Note that the attacker and defender actions substantialy affect the effectiveness of the attack, with the probability of a correct guess ranging between 5.46% and 9.5%. Based on the results of Section 6, we can then compute the best strategy for the defender, which turns out to be (probabilities expressed as %):

$$\delta^* = (34.59, 3.48, 3.00, 10.52, 3.32, 2.99, 35.93, 3.19, 2.99)$$

This strategy is part of an equilibrium and guarantees that for any choice of the attacker the vulnerability is at most 8.76%, and is substantially better that the best pure strategy (location 1) which leads to a worst vulnerability of 9.32%. As expected, δ^* selects the most vulnerable locations (1 and 7) with the highest probability. Still, the other locations are selected with non-negligible probability, which is important for maximizing the attacker's uncertainty about the defense.

8 Related work

There is an extensive literature on game theory models for security and privacy in computer systems, including network security, vehicular networks, cryptography, anonymity, location privacy, and intrusion detection. See [17] for a survey.

In many studies, security games have been used to model and analyze utilities between interacting agents, especially an attacker and a defender. In particular, Korzhyk et al. [16] present a theoretical analysis of security games and investigate the relation between Stackelberg and simultaneous games under various forms of uncertainty. In application to network security, Venkitasubramaniam [28] investigates anonymous wireless networking, which they formalize as a zero-sum game between the network designer and the attacker. The task of the attacker is to choose a subset of nodes to monitor so that anonymity of routes is minimum whereas the task of the designer is to maximize anonymity by choosing nodes to evade flow detection by generating independent transmission schedules.

Khouzani et al. [14] present a framework for analyzing a trade-off between usability and security. They analyze guessing attacks and derive the optimal policies for secret picking as Nash/Stackelberg equilibria. Khouzani and Malacaria [13] investigate properties of leakage when perfect secrecy is not achievable due to the limit on the allowable size of the conflating sets, and show the existence of universally optimal strategies for a wide class of entropy measures, and for g-entropies (the dual of g-vulnerabilities). In particular, they show that designing a channel with minimum leakage is equivalent to Nash equilibria in a corresponding two-player zero-sum games of incomplete information for a range of entropy measures.

Concerning costs of security, Yang et al. [32] propose a framework to analyze user behavior in anonymity networks. Utility is modeled as a combination of weighted cost and anonymity utility. They also consider incentives and their impact on users' cooperation.

Some security games have considered leakage of information about the defender's choices. For example, Alon et al. [1] present two-player zero-sum games where a defender chooses probabilities of secrets while an attacker chooses and learns some of the defender's secrets. Then they show how the leakage on the defender's secrets influences the defender's optimal strategy. Xu et al. [31] present zero-sum security games where the attacker acquires partial knowledge on the security resources the defender is protecting, and show the defender's optimal strategy under such attacker's knowledge. More recently, Farhang et al. [12] present two-player games where utilities are defined taking account of information leakage, although the defender's goal is different from our setting. They consider a model where the attacker incrementally and stealthily obtains partial information on a secret, while the defender periodically changes the secret after some time to prevent a complete compromise of the system. In particular, the defender is not attempting to minimize the leak of a certain secret, but only to make it useless (for the attacker). Hence their model of defender and utility is totally different from ours. To the best of our knowledge there have been no works exploring games with utilities defined as information-leakage measures.

Finally, in game theory Matsui [20] uses the term "information leakage game" with a meaning different than ours, namely, as a game in which (part of) the strategy of one player may be leaked in advance to the other player, and the latter may revise his strategy based on this knowledge.

9 Conclusion and future work

In this paper we introduced the notion of information leakage games, in which a defender and an attacker have opposing goals in optimizing the amount of information leakage in a system. In contrast to standard game theory models, in our games the utility of a mixed strategy is a convex function of the distribution of the defender's actions, rather than the expected value of the utilities of the pure strategies in the support. Nevertheless, the important properties of game theory, notably the existence of a Nash equilibrium, still hold for our zero-sum leakage games, and we provided algorithms to compute the corresponding optimal strategies for the attacker and the defender.

As future research, we would like to extend leakage games to scenarios with repeated observations, i.e., when the attacker can repeatedly observe the outcomes of the system in successive runs, under the assumption that both the attacker and the defender may change the channel at each run. Furthermore, we would like to consider the possibility to adapt the defender's strategy to the secret value, as we believe that in some cases this would provide significant advantage to the defender. We would also like to consider the cost of attack and of defense, which would lead to non-zero-sum games.

Acknowledgments The authors are thankful to Arman Khouzani and Pedro O. S. Vaz de Melo for valuable discussions. This work was supported by JSPS and Inria under the project LOGIS of the Japan-France AYAME Program, and by the project Epistemic Interactive Concurrency (EPIC) from the STIC AmSud Program. Mário S. Alvim was supported by CNPq, CAPES, and FAPEMIG. Yusuke Kawamoto was supported by JSPS KAKENHI Grant Number JP17K12667.

References

1. N. Alon, Y. Emek, M. Feldman, and M. Tennenholtz. Adversarial leakage in games. *SIAM J. Discrete Math.*, 27(1):363–385, 2013.
2. M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith. Axioms for information leakage. In *Proc. of CSF*, pages 77–92, 2016.
3. M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith. Measuring information leakage using generalized gain functions. In *CSF*, pages 265–279, 2012.
4. M. Boreale and F. Pampaloni. Quantitative information flow under generic leakage functions and adaptive adversaries. *Log. Meth. Comp. Sci.*, 11(4), 2015.
5. S. Boyd and A. Mutapcic. Subgradient methods. *Lecture notes of EE364b, Stanford University, Winter Quarter*, 2007, 2006.
6. S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, 2004.
7. C. Braun, K. Chatzikokolakis, and C. Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proc. of MFPS*, volume 249 of *ENTCS*, pages 75–91. Elsevier, 2009.
8. K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. On the Bayes risk in information-hiding protocols. *J. of Comp. Security*, 16(5):531–571, 2008.

9. D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.
10. D. Clark, S. Hunt, and P. Malacaria. A static analysis for quantifying information flow in a simple imperative language. *J. of Comp. Security*, 2007.
11. C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of TCC*, pages 265–284, 2006.
12. S. Farhang and J. Grossklags. Flipleakage: A game-theoretic approach to protect against stealthy attackers in the presence of information leakage. In *Proc. of GameSec*, pages 195–214, 2016.
13. M. Khouzani and P. Malacaria. Relative perfect secrecy: Universally optimal strategies and channel design. In *Proc. of CSF*, pages 61–76. IEEE, 2016.
14. M. H. R. Khouzani, P. Mardziel, C. Cid, and M. Srivatsa. Picking vs. guessing secrets: A game-theoretic analysis. In *Proc. of CSF*, pages 243–257, 2015.
15. B. Köpf and D. A. Basin. An information-theoretic model for adaptive side-channel attacks. In *Proc. of CCS*, pages 286–296. ACM, 2007.
16. D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Intell. Res.*, 41:297–327, 2011.
17. M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux. Game theory meets network security and privacy. *ACM Comput. Surv.*, 45(3):25:1–25:39, 2013.
18. P. Mardziel, M. S. Alvim, M. W. Hicks, and M. R. Clarkson. Quantifying information flow for dynamic secrets. In *Proc. of S&P*, pages 540–555, 2014.
19. Massey. Guessing and entropy. In *Proc. of ISIT*, page 204. IEEE, 1994.
20. A. Matsui. Information leakage forces cooperation. *Games and Economic Behavior*, 1(1):94 – 115, 1989.
21. A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.
22. M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.
23. M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Trans. on Information and System Security*, 1(1):66–92, 1998.
24. A. Rubinstein. *Lecture Notes in Microeconomic Theory.* Princeton University Press, second edition, 2012.
25. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 625–56, 1948.
26. V. Shmatikov. Probabilistic analysis of anonymity. In *CSFW*, pages 119–128, 2002.
27. G. Smith. On the foundations of quantitative information flow. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 288–302. Springer, 2009.
28. P. Venkitasubramaniam and L. Tong. A game-theoretic approach to anonymous networking. *IEEE/ACM Trans. Netw.*, 20(3):892–905, 2012.
29. J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior.* Princeton university press, 2007.
30. W. Wang and M. A. Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.
31. H. Xu, A. X. Jiang, A. Sinha, Z. Rabinovich, S. Dughmi, and M. Tambe. Security games with information leakage: Modeling and computation. In *Proc. of IJCAI*, pages 674–680, 2015.
32. M. Yang, V. Sassone, and S. Hamadou. A game-theoretic analysis of cooperation in anonymity networks. In *Proc. of POST*, pages 269–289, 2012.
33. A. C. Yao. Protocols for secure computations. *IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 160–164, 1982.