Leakage and Protocol Composition in a Game-Theoretic Perspective

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

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

Abstract. In the inference attacks studied in Quantitative Information
Flow (QIF), the adversary typically tries to interfere with the system in
the attempt to increase its leakage of secret information. The defender,
on the other hand, typically tries to decrease leakage by introducing some
controlled noise. This noise introduction can be modeled as a type of pro-
tocol composition, i.e., a probabilistic choice among different protocols,
and its effect on the amount of leakage depends heavily on whether or
not this choice is visible to the adversary. In this work we consider op-
erators for modeling visible and invisible choice in protocol composition,
and we study their algebraic properties. We then formalize the interplay
between defender and adversary in a game-theoretic framework adapted
to the specific issues of QIF, where the payoff is information leakage. We
consider various kinds of leakage games, depending on whether players
act simultaneously or sequentially, and on whether or not the choices of
the defender are visible to the adversary. Finally, we establish a hierar-
chy of these games in terms of their information leakage, and provide
methods for finding optimal strategies (at the points of equilibrium) for
both attacker and defender in the various cases.

1   Introduction

A fundamental problem in computer security is the leakage of sensitive informa-
tion due to *correlation* of secret values with *observables*—i.e., any information
accessible to the attacker, such as, for instance, the system's outputs or execu-
tion time. The typical defense consists in reducing this correlation, which can
be done in, essentially, two ways. The first, applicable when the correspondence
secret-observable is deterministic, consists in coarsening the equivalence classes
of secrets that give rise to the same observables. This can be achieved with
post-processing, i.e., sequentially composing the original system with a program
that removes information from observables. For example, a typical attack on en-
crypted web traffic consists on the analysis of the packets' length, and a typical
defense consists in padding extra bits so to diminish the length variety [28].

The second kind of defense, on which we focus in this work, consists in adding controlled noise to the observables produced by the system. This can be usually seen as a composition of different protocols via probabilistic choice.

Example 1 (Differential privacy). Consider a counting query f, namely a function that, applied to a dataset x, returns the number of individuals in x that satisfy a given property. A way to implement differential privacy [12] is to add geometrical noise to the result of f, so to obtain a probability distribution P on integers of the form $P(z) = c\, e^{|z-f(x)|}$, where c is a normalization factor. The resulting mechanism can be interpreted as a probabilistic choice on protocols of the form $f(x), f(x)+1, f(x)+2, \ldots, f(x)-1, f(x)-2, \ldots$, where the probability assigned to $f(x) + n$ and to $f(x) - n$ decreases exponentially with n.

Example 2 (Dining cryptographers). Consider two agents running the dining cryptographers protocol [11], which consists in tossing a fair binary coin and then declaring the exclusive or $\oplus$ of their secret value x and the result of the coin. The protocol can be thought as the fair probabilistic choice of two protocols, one consisting simply of declaring x, and the other declaring $x \oplus 1$.

Most of the work in the literature of quantitative information flow (QIF) considers passive attacks, in which the adversary only observes the system. Notable exceptions are the works [8,21,4], which consider attackers who interact with and influence the system, possibly in an adaptive way, with the purpose of maximizing the leakage of information.

Example 3 (CRIME attack). Compression Ratio Info-leak Made Easy (CRIME) [25] is a security exploit against secret web cookies over connections using the HTTPS and SPDY protocols and data compression. The idea is that the attacker can inject some content a in the communication of the secret x from the target site to the server. The server then compresses and encrypts the data, including both a and x, and sends back the result. By observing the length of the result, the attacker can then infer information about x. To mitigate the leakage, one possible defense would consist in transmitting, along with x, also an encryption method f selected randomly from a set F. Again, the resulting protocol can be seen as a composition, using probabilistic choice, of the protocols in the set F.

In all examples above the main use of the probabilistic choice is to obfuscate the relation between secrets and observables, thus reducing their correlation— and, hence, the information leakage. To achieve this goal, it is essential that the attacker never comes to know the result of the choice. In the CRIME example, however, if f and a are chosen independently, then (in general) it is still better to choose f probabilistically, even if the adversary will come to know, afterwards, the choice of f. In fact, this is true also for the attacker: his best strategies (in general) are to chose a according to some probability distribution. Indeed, suppose that $F = \{f_1, f_2\}$ are the defender's choices and $A = \{a_1, a_2\}$ are the attacker's, and that $f_1(\cdot, a_1)$ leaks more than $f_1(\cdot, a_2)$, while $f_2(\cdot, a_1)$ leaks less than $f_2(\cdot, a_2)$. This is a scenario like *the matching pennies* in game

theory: if one player selects an action deterministically, the other player may exploit this choice and get an advantage. For each player the optimal strategy is to play probabilistically, using a distribution that maximizes his own gain for all possible actions of the adversary. In zero-sum games, in which the gain of one player coincides with the loss of the other, the optimal pair of distributions always exists, and it is called *saddle point*. It also coincides with the *Nash equilibrium*, which is defined as the point in which neither of the two players gets any advantage in changing unilaterally his strategy.

Motivated by these examples, this paper investigates the two kinds of choice, visible and hidden (to the attacker), in a game-theoretic setting. Looking at them as language operators, we study their algebraic properties, which will help reason about their behavior in games. We consider zero-sum games, in which the gain (for the attacker) is represented by the leakage. While for visible choice it is appropriate to use the "classic" game-theoretic framework, for hidden choice we need to adopt the more general framework of the *information leakage games* proposed in [4]. This happens because, in contrast with standard game theory, in games with hidden choice the utility of a mixed strategy is a convex function of the distribution on the defender's pure actions, rather than simply the expected value of their utilities. We will consider both simultaneous games—in which each player chooses independently—and sequential games— in which one player chooses his action first. We aim at comparing all these situations, and at identifying the precise advantage of the hidden choice over the visible one.

To measure leakage we use the well-known information-theoretic model. A central notion in this model is that of *entropy*, but here we use its converse, *vulnerability*, which represents the magnitude of the threat. In order to derive results as general as possible, we adopt the very comprehensive notion of vulnerability as any convex and continuous function, as used in [8] and [5]. This notion has been shown [5] to subsume most information measures, including *Bayes vulnerability* (aka min-vulnerability, aka (the converse of) Bayes risk) [27,10], *Shannon entropy* [26], *guessing entropy* [22], and *g-vulnerability* [6].

The main contributions of this paper are:

– We present a general framework for reasoning about information leakage in a game-theoretic setting, extending the notion of information leakage games proposed in [4] to both simultaneous and sequential games, with either hidden or visible choice.
– We present a rigorous compositional way, using visible and hidden choice operators, for representing adversary and defender's actions in information leakage games. In particular, we study the algebraic properties of visible and hidden choice on channels, and compare the two kinds of choice with respect to the capability of reducing leakage, in presence of an adaptive attacker.
– We provide a taxonomy of the various scenarios (simultaneous and sequential) showing when randomization is necessary, for either attacker or defender, to achieve optimality. Although it is well-known in information flow that the defender's best strategy is usually randomized, only recently it has

3

been shown that when defender and adversary act simultaneously, the adversary's optimal strategy also requires randomization [4].

– We use our framework in a detailed case study of a password-checking protocol. The naive program, which checks the password bit by bit and stops when it finds a mismatch, is clearly very insecure, because it reveals at each attempt the maximum correct prefix. On the other hand, if we continue checking until the end of the string (time padding), the program becomes very inefficient. We show that, by using probabilistic choice instead, we can obtain a good trade-off between security and efficiency.

Plan of the paper. The remaining of the paper is organized as follows. In Section 2 we review some basic notions of game theory and quantitative information flow. In Section 3 we introduce our running example. In Section 4 we define the visible and hidden choice operators and demonstrate their algebraic properties. In Section 5, the core of the paper, we examine various scenarios for leakage games. In Section 6 we show an application of our framework to a password checker. In Section 7 we discuss related work and, finally, in Section 8 we conclude.

2 Preliminaries

In this section we review some basic notions from game theory and quantitative information flow. 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* $\mathsf{supp}(\mu) \stackrel{\text{def}}{=} \{i \in \mathcal{I} : \mu(i) > 0\}$ is the set of its elements with positive probability. We use $i \leftarrow \mu$ to indicate that a value $i \in \mathcal{I}$ is sampled from a distribution μ on $\mathcal{I}$.

2.1 Basic concepts from game theory

Two-player games *Two-player games* are a model for reasoning about the behavior of two players. In a game, each player has at its disposal a set of *actions* that he can perform, and he obtains some gain or loss depending on the actions chosen by both players. Gains and losses are defined using a real-valued *payoff function*. Each player is assumed to be *rational*, i.e., his choice is driven by the attempt to maximize his own expected payoff. We also assume that the set of possible actions and the payoff functions of both players are *common knowledge*.

In this paper we only consider *finite games*, in which the set of actions available to the players are finite. Next we introduce an important distinction between *simultaneous* and *sequential* games. In the following, we will call the two players *defender* and *attacker*.

Simultaneous games In a simultaneous game, each player chooses his action without knowing the action chosen by the other. The term "simultaneous" here does not mean that the players' actions are chosen at the same time, but only

that they are chosen independently. 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 payoff function*, and $u_{\mathsf{a}} : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$ is the *attacker's payoff 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 called *pure strategy profile*, and $u_{\mathsf{d}}(d, a)$, $u_{\mathsf{a}}(d, a)$ represent the defender's and the attacker's payoffs, respectively. 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 *mixed strategy profile*. The defender's and the attacker's *expected payoff functions* for mixed strategies are defined, respectively, as: $U_{\mathsf{d}}(\delta, \alpha) \overset{\mathrm{def}}{=} \mathbb{E}_{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}} u_{\mathsf{d}}(d, a) = \sum_{\substack{d \in \mathcal{D} \\ a \in \mathcal{A}}} \delta(d)\alpha(a) u_{\mathsf{d}}(d, a)$ and $U_{\mathsf{a}}(\delta, \alpha) \overset{\mathrm{def}}{=} \mathbb{E}_{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}} u_{\mathsf{a}}(d, a) = \sum_{\substack{d \in \mathcal{D} \\ a \in \mathcal{A}}} \delta(d)\alpha(a) u_{\mathsf{a}}(d, a)$.

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 the best response to α^* and vice versa. This means that in a Nash equilibrium, no unilateral deviation by any single player provides better payoff 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.

Sequential games In a sequential game players may take turns in choosing their actions. In this paper, we only consider the case in which each player moves only once, in such a way that one of the players (*the leader*) chooses his action first, and commits to it, before the other player (*the follower*) makes his choice. The follower may have total knowledge of the choice made by the leader, or only partial. We refer to the two scenarios by the terms *perfect* and *imperfect information*, respectively.

We now give the precise definitions assuming that the leader is the defender. The case in which the leader is the attacker is similar.

A *defender-first sequential game with perfect information* is a tuple $(\mathcal{D}, \mathcal{D} \to \mathcal{A}, u_{\mathsf{d}}, u_{\mathsf{a}})$ where $\mathcal{D}$, $\mathcal{A}$, u_{d} and u_{a} are defined as in simultaneous games. Also the strategies of the defender (the leader) are defined as in simultaneous games: an action $d \in \mathcal{D}$ for the pure case, and a distribution $\delta \in \mathbb{D}\mathcal{D}$ for the mixed one. On the other hand, a pure strategy for the attacker is a function $s_{\mathsf{a}} : \mathcal{D} \to \mathcal{A}$, which represents the fact that his choice of an action s_{a} in $\mathcal{A}$ depends on the defender's choice d. An attacker's mixed strategy is a probabil-

---

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

ity distribution $\sigma_{\mathsf{a}} \in \mathbb{D}(\mathcal{D} \to \mathcal{A})$ over his pure strategies.[6] The defender's and the attacker's *expected payoff functions* for mixed strategies are defined, respectively, as $U_{\mathsf{d}}(\delta, \sigma_{\mathsf{a}}) \stackrel{\text{def}}{=} \mathbb{E}_{\substack{d \leftarrow \delta \\ s_{\mathsf{a}} \leftarrow \sigma_{\mathsf{a}}}} u_{\mathsf{d}}(d, s_{\mathsf{a}}(d)) = \sum_{\substack{d \in \mathcal{D} \\ s_{\mathsf{a}}:\mathcal{D} \to \mathcal{A}}} \delta(d)\sigma_{\mathsf{a}}(s_{\mathsf{a}})u_{\mathsf{d}}(d, s_{\mathsf{a}}(d))$ and $U_{\mathsf{a}}(\delta, \sigma_{\mathsf{a}}) \stackrel{\text{def}}{=} \mathbb{E}_{\substack{d \leftarrow \delta \\ s_{\mathsf{a}} \leftarrow \sigma_{\mathsf{a}}}} u_{\mathsf{a}}(d, s_{\mathsf{a}}(d)) = \sum_{\substack{d \in \mathcal{D} \\ s_{\mathsf{a}}:\mathcal{D} \to \mathcal{A}}} \delta(d)\sigma_{\mathsf{a}}(s_{\mathsf{a}})u_{\mathsf{a}}(d, s_{\mathsf{a}}(d))$.

The case of imperfect information is typically formalized by assuming an *indistinguishability (equivalence) relation* over the actions chosen by the leader, representing a scenario in which the follower cannot distinguish between the actions belonging to the same equivalence class. The pure strategies of the followers, therefore, are functions from the set of the equivalence classes on the actions of the leader to his own actions. Formally, a *defender-first sequential game with imperfect information* is a tuple $(\mathcal{D}, K_{\mathsf{a}} \to \mathcal{A}, u_{\mathsf{d}}, u_{\mathsf{a}})$ where $\mathcal{D}$, $\mathcal{A}$, u_{d} and u_{a} are defined as in simultaneous games, and K_{a} is a partition of $\mathcal{D}$. The *expected payoff functions* are defined as before, except that now the argument of s_{a} is the equivalence class of d. Note that in the case in which all defender's actions are indistinguishable from each other at the eyes of the attacker (*totally imperfect information*), we have $K_{\mathsf{a}} = \{\mathcal{D}\}$ and the expected payoff functions coincide with those of the simultaneous games.

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 $a \in \mathcal{A}$, the defender's loss is equivalent to the attacker's gain, i.e., $u_{\mathsf{d}}(d, a) = -u_{\mathsf{a}}(d, a)$. For brevity, in zero-sum games we denote by u the attacker's payoff function u_{a}, and by U the attacker's expected payoff U_{a}.[7] 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 strategy profile (δ^*, α^*) such that $U(\delta^*, \alpha^*) = \min_{\delta} \max_{\alpha} U(\delta, \alpha)$. The von Neumann's minimax theorem, in fact, 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 a convex function in $x \in \mathcal{X}$ and a concave function in $y \in \mathcal{Y}$. Then $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} U(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} U(x, y)$.*

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 $y \in \mathcal{Y}$: $U(x^*, y) \le U(x^*, y^*) \le U(x, y^*)$.

---

[6] The definition of the mixed strategies as $\mathbb{D}(\mathcal{D} \to \mathcal{A})$ means that the attacker draws a function $s_{\mathsf{a}} : \mathcal{D} \to \mathcal{A}$ *before* he knows the choice of the defender. In contrast, the so-called *behavioral strategies* are defined as functions $\mathcal{D} \to \mathbb{D}\mathcal{A}$, and formalize the idea that the draw is made after the attacker knows such choice. In our setting, these two definitions are equivalent, in the sense that they yield the same payoff.

[7] Conventionally in game theory the payoff u is set to be that of the first player, but we prefer to look at the payoff from the point of view of the attacker to be in line with the definition of payoff as vulnerability.

The solution of the minimax problem can be obtained by using convex optimization techniques. In case $U(x, y)$ is affine in x and in y, we can also use linear optimization.

In case $\mathcal{D}$ and $\mathcal{A}$ contain two elements each, there is a closed form for the solution. Let $\mathcal{D} = \{d_0, d_1\}$ and $\mathcal{A} = \{a_0, a_1\}$ respectively. Let u_{ij} be the utility of the defender on d_i, a_j. Then the Nash equilibrium (δ^*, α^*) is given by: $\delta^*(d_0) = {}^{(u_{11} - u_{10})}/_{(u_{00} - u_{01} - u_{10} + u_{11})}$ and $\alpha^*(a_0) = {}^{(u_{11} - u_{01})}/_{(u_{00} - u_{01} - u_{10} + u_{11})}$ if these values are in $[0, 1]$. Note that, since there are only two elements, the strategy δ^* is completely specified by its value in d_0, and analogously for α^*.

2.2 Quantitative information flow

Finally, we briefly review the standard framework of quantitative information flow, which is concerned with measuring the amount of information leakage in a (computational) 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 that it represents for the attacker. In this paper we consider a very general notion of vulnerability, following [5], and we 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 [5] 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. [8]

Channels, posterior vulnerability, and leakage Computational 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 $x \in \mathcal{X}$.

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 $\mathcal{X} \times \mathcal{Y}$, producing joint random variables X, 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* $p_{X|y}$.

---

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

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 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 the channel C. Formally: $\mathbb{V}[\pi, C] \stackrel{\text{def}}{=} \sum_{y \in \mathcal{Y}} p(y) \mathbb{V}[p_{X|y}]$.

It is known from the literature [5] that the posterior vulnerability is a convex function of π. Namely, for any channel C, any family of distributions $\{\pi_i\}$, and any set of convex coefficients $\{c_i\}$, we have: $\mathbb{V}[\sum_i c_i \pi_i, C] \leq \sum_i c_i \mathbb{V}[\pi_i, C]$.

The *(information) leakage* of channel C under prior π is a comparison between the vulnerability of the secret before the system was run—called *prior vulnerability*—and the posterior vulnerability of the secret. Leakage reflects by how much the observation of the system's outputs increases the attacker's information 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 An illustrative example

We introduce an example which will serve as running example through the paper. Although admittedly contrived, this example is simple and yet produces different leakage measures for all different combinations of visible/invisible choice and simultaneous/sequential games, thus providing a way to compare all different scenarios we are interested in.

Consider that a binary secret must be processed by a program. As usual, a defender wants to protect the secret value, whereas an attacker wants to infer it by observing the system's output. Assume the defender can choose which among two alternative versions of the program to run. Both programs take the secret value x as high input, and a binary low input a whose value is chosen by the attacker. They both return the output in a low variable y. [9] Program 0 returns the binary product of x and a, whereas Program 1 flips a coin with bias $a/3$ (i.e., a coin which returns heads with probability $a/3$) and returns x if the result is heads, and the complement $\bar{x}$ of x otherwise. The two programs are represented in Figure 1.

```
Program 0

High Input: x ∈ {0, 1}
Low Input: a ∈ {0, 1}
Output: y ∈ {0, 1}
y = x · a
return y
```

```
Program 1

High Input: x ∈ {0, 1}
Low Input: a ∈ {0, 1}
Output: y ∈ {0, 1}
c ← flip coin with bias a/3
if c = heads {y = x}
else {y = x̄}
return y
```

Fig. 1. Running example.

---

[9] We adopt the usual convention in QIF of referring to secret variables, inputs and outputs in programs as *high*, and to their observable counterparts as *low*.

The combined choices of the defender's and of the attacker's determine how the system behaves. Let $\mathcal{D}=\{0,1\}$ represent the set of the defender's choices—i.e., the index of the program to use—, and $\mathcal{A} = \{0,1\}$ represent the set of the attacker's 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 $d \in \mathcal{D}$ and $a \in \mathcal{A}$, we can construct a channel C_{da} modeling how the resulting system behaves. Each channel C_{da} is a function of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{X} = \{0,1\}$ is the set of possible high input values for the system, and $\mathcal{Y} = \{0,1\}$ is the set of possible output values from the system. Intuitively, each channel provides the probability that the system (which was fixed by the defender) produces output $y \in \mathcal{Y}$ given that the high input is $x \in \mathcal{X}$ (and that the low input was fixed by the attacker). The four possible channels are depicted as matrices below.

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

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

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

C_{11}	$y = 0$	$y = 1$
$x = 0$	$^1/_3$	$^2/_3$
$x = 1$	$^2/_3$	$^1/_3$

Note that channel C_{00} does not leak any information about the input x (i.e., it is *non-interferent*), whereas channels C_{01} and C_{10} completely reveal x. Channel C_{11} is an intermediate case: it leaks some information about x, but not all.

We want to investigate how the defender's and the attacker's choices influence the leakage of the system. For that we can just consider the (simpler) notion of posterior vulnerability, since in order to make the comparison fair we need to assume that the prior is always the same in the various scenarios, and this implies that the leakage is in a one-to-one correspondence with the posterior vulnerability (this happens for both additive and multiplicative leakage).

For this example, assume we are interested in Bayes vulnerability [10,27], defined as $\mathbb{V}(\pi) = \max_x \pi(x)$ for every $\pi \in \mathbb{D}\mathcal{X}$. Assume for simplicity that the prior is the uniform prior π_u. In this case we know from [9] that the posterior Bayes vulnerability of a channel is the sum of the greatest elements

$\mathbb{V}$	$a = 0$	$a = 1$
$d = 0$	$^1/_2$	1
$d = 1$	1	$^2/_3$

Table 1. Vulnerability of each channel C_{da} in the running example.

of each column, divided by the total number of inputs. Table 1 provides the Bayes vulnerability $\mathbb{V}_{da} \stackrel{\text{def}}{=} \mathbb{V}[\pi_u, C_{da}]$ of each channel considered above.

Naturally, the attacker aims at maximizing the vulnerability of the system, while the defender tries to minimize it. The resulting vulnerability will depend on various factors, in particular on whether the two players make their choice *simultaneously* (i.e. without knowing the choice of the opponent) or *sequentially*. Clearly, if the choice of a player who moves first is known by an opponent who moves second, the opponent will be in advantage. In the above example, for instance, if the defender knows the choice a of the attacker, the most convenient choice for him is to set $d = a$, and the vulnerability will be at most $^2/_3$. Vice versa, if the attacker knows the choice d of the defender, the most convenient choice for him is to set $a \neq d$. The vulnerability in this case will be 1.

Things become more complicated when players make choices simultaneously. None of the pure choices of d and a are the best for the corresponding player,

because the vulnerability of the system depends also on the (unknown) choice of the other player. Yet there is a strategy leading to the best possible situation for both players (the *Nash equilibrium*), but it is mixed (i.e., probabilistic), in that the players randomize their choices according to some precise distribution.

Another factor that affects vulnerability is whether or not the defender's choice is known to the attacker at the moment in which he observes the output of the channel. Obviously, this corresponds to whether or not the attacker knows what channel he is observing. Both cases are plausible: naturally the defender has all the interest in keeping his choice (and, hence, the channel used) secret, since then the attack will be less effective (i.e., leakage will be smaller). On the other hand, the attacker may be able to identify the channel used anyway, for instance because the two programs have different running times. We will call these two cases *hidden* and *visible* choice, respectively.

It is possible to model players' strategies, as well as hidden and visible choices, as operations on channels. This means that we can look at the whole system as if it were a single channel, which will turn out to be useful for some proofs of our technical results. Next section is dedicated to the definition of these operators. We will calculate the exact values for our example in Section 5.

4 Visible and hidden choice operators on channels

In this section we define matrices and some basic operations on them. Since channels are a particular kind of matrix, we use these matrix operations to define the operations of visible and hidden choice among channels, and to prove important properties of these channel operations.

4.1 Matrices, and their basic operators

Given two sets $\mathcal{X}$ and $\mathcal{Y}$, a *matrix* is a total function of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Two matrices $M_1 : \mathcal{X}_1 \times \mathcal{Y}_1 \to \mathbb{R}$ and $M_2 : \mathcal{X}_2 \times \mathcal{Y}_2 \to \mathbb{R}$ are said to be *compatible* if $\mathcal{X}_1 = \mathcal{X}_2$. If it is also the case that $\mathcal{Y}_1 = \mathcal{Y}_2$, we say that the matrices *have the same type*. The *scalar multiplication* $r \cdot M$ between a scalar r and a matrix M is defined as usual, and so is the *summation* $\left(\sum_{i \in \mathcal{I}} M_i\right)(x, y) = M_{i_1}(x, y) + \ldots + M_{i_n}(x, y)$ of a family $\{M_i\}_{i \in \mathcal{I}}$ of matrices all of a same type.

Given a family $\{M_i\}_{i \in \mathcal{I}}$ of compatible matrices s.t. each M_i has type $\mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$, their *concatenation* $\Diamond_{i \in \mathcal{I}}$ is the matrix having all columns of every matrix in the family, in such a way that every column is tagged with the matrix it came from. Formally, $\left(\Diamond_{i \in \mathcal{I}} M_i\right)(x, (y, j)) = M_j(x, y)$, if $y \in \mathcal{Y}_j$, and the resulting matrix has type $\mathcal{X} \times \left(\bigsqcup_{i \in \mathcal{I}} \mathcal{Y}_i\right) \to \mathbb{R}$. [10] When the family $\{M_i\}$ has only two elements we may use the *binary* version $\diamond$ of the concatenation operator. The

---

[10] $\bigsqcup_{i \in \mathcal{I}} \mathcal{Y}_i = \mathcal{Y}_{i_1} \sqcup \mathcal{Y}_{i_2} \sqcup \ldots \sqcup \mathcal{Y}_{i_n}$ denotes the *disjoint union* $\{(y, i) \mid y \in \mathcal{Y}_i, i \in \mathcal{I}\}$ of the sets $\mathcal{Y}_{i_1}, \mathcal{Y}_{i_2}, \ldots, \mathcal{Y}_{i_n}$.

following depicts the concatenation of two matrices M_1 and M_2 in tabular form.

M_1	y_1	y_2
x_1	1	2
x_2	3	4

$\diamond$

M_2	y_1	y_2	y_3
x_1	5	6	7
x_2	8	9	10

$=$

$M_1 \diamond M_2$	$(y_1, 1)$	$(y_2, 1)$	$(y_1, 2)$	$(y_2, 2)$	$(y_3, 2)$
x_1	1	2	5	6	7
x_2	3	4	8	9	10

4.2 Channels, and their hidden and visible choice operators

A channel is a *stochastic* matrix, i.e., all elements are non-negative, and all rows sum up to 1. Here we will define two operators specific for channels. In the following, for any real value $0 \le p \le 1$, we denote by $\bar{p}$ the value $1 - p$.

Hidden choice The first operator models a hidden probabilistic choice among channels. Consider a family $\{C_i\}_{i \in \mathcal{I}}$ of channels of a same type. Let $\mu \in \mathbb{D}\mathcal{I}$ be a probability distribution on the elements of the index set $\mathcal{I}$. Consider an input x is fed to one of the channels in $\{C_i\}_{i \in \mathcal{I}}$, where the channel is randomly picked according to μ. More precisely, an index $i \in \mathcal{I}$ is sampled with probability $\mu(i)$, then the input x is fed to channel C_i, and the output y produced by the channel is then made visible, but not the index i of the channel that was used. Note that we consider hidden choice only among channels of a same type: if the sets of outputs were not identical, the produced output might implicitly reveal which channel was used.

Formally, given a family $\{C_i\}_{i \in \mathcal{I}}$ of channels s.t. each C_i has same type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the *hidden choice operator* $\displaystyle\sideset{}{}\sum_{i \leftarrow \mu}$ is defined as $\displaystyle\sideset{}{}\sum_{i \leftarrow \mu} C_i = \sum_{i \in \mathcal{I}} \mu(i)\, C_i$.

Proposition 2 *Given a family $\{C_i\}_{i \in \mathcal{I}}$ of channels of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and a distribution μ on $\mathcal{I}$, the hidden choice $\displaystyle\sideset{}{}\sum_{i \leftarrow \mu} C_i$ is a channel of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.*

In the particular case in which the family $\{C_i\}$ has only two elements C_{i_1} and C_{i_2}, the distribution μ on indexes is completely determined by a real value $0 \le p \le 1$ s.t. $\mu(i_1) = p$ and $\mu(i_2) = \bar{p}$. In this case we may use the *binary* version $_p\oplus$ of the hidden choice operator: $C_{i_1} \,_p\oplus\, C_{i_2} = p\,C_{i_1} + \bar{p}\,C_{i_2}$. The example below depicts the hidden choice between channels C_1 and C_2, with probability $p = 1/3$.

C_1	y_1	y_2
x_1	$1/2$	$1/2$
x_2	$1/3$	$2/3$

$_{1/3}\oplus$

C_2	y_1	y_2
x_1	$1/3$	$2/3$
x_2	$1/2$	$1/2$

$=$

$C_1 \,_{1/3}\oplus\, C_2$	y_1	y_2
x_1	$7/18$	$11/18$
x_2	$4/9$	$5/9$

Visible choice The second operator models a visible probabilistic choice among channels. Consider a family $\{C_i\}_{i \in \mathcal{I}}$ of compatible channels. Let $\mu \in \mathbb{D}\mathcal{I}$ be a probability distribution on the elements of the index set $\mathcal{I}$. Consider an input x is fed to one of the channels in $\{C_i\}_{i \in \mathcal{I}}$, where the channel is randomly picked according to μ. More precisely, an index $i \in \mathcal{I}$ is sampled with probability $\mu(i)$, then the input x is fed to channel C_i, and the output y produced by the channel is then made visible, along with the index i of the channel that was used. Note

that visible choice makes sense only between compatible channels, but it is not required that the output set of each channel be the same.

Formally, given $\{C_i\}_{i \in \mathcal{I}}$ of compatible channels s.t. each C_i has type $\mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$, and a distribution μ on $\mathcal{I}$, the *visible choice operator* $\lfloor \cdot \rceil_{i \leftarrow \mu}$ is defined as $\lfloor \cdot \rceil_{i \leftarrow \mu} C_i = \diamondsuit_{i \in \mathcal{I}} \, \mu(i) \, C_i$.

Proposition 3 *Given a family $\{C_i\}_{i \in \mathcal{I}}$ of compatible channels s.t. each C_i has type $\mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$, and a distribution μ on $\mathcal{I}$, the result of the visible choice $\lfloor \cdot \rceil_{i \leftarrow \mu} C_i$ is a channel of type $\mathcal{X} \times \left(\bigsqcup_{i \in \mathcal{I}} \mathcal{Y}_i \right) \to \mathbb{R}$.*

In the particular case the family $\{C_i\}$ has only two elements C_{i_1} and C_{i_2}, the distribution μ on indexes is completely determined by a real value $0 \le p \le 1$ s.t. $\mu(i_1) = p$ and $\mu(i_2) = \bar{p}$. In this case we may use the *binary* version ${}_p \sqcup$ of the visible choice operator: $C_{i_1} \, {}_p \sqcup \, C_{i_2} = p \, C_{i_1} \diamond \bar{p} \, C_{i_2}$. The following depicts the visible choice betwee channels C_1 and C_3, with probability $p = 1/3$.

C_1	y_1	y_2
x_1	$1/2$	$1/2$
x_2	$1/3$	$2/3$

${}_{1/3} \sqcup$

C_3	y_1	y_3
x_1	$1/3$	$2/3$
x_2	$1/2$	$1/2$

$=$

$C_1 \, {}_{1/3} \sqcup \, C_3$	$(y_1, 1)$	$(y_2, 1)$	$(y_1, 3)$	$(y_3, 3)$
x_1	$1/6$	$1/6$	$2/9$	$4/9$
x_2	$1/9$	$2/9$	$1/3$	$1/3$

4.3 Properties of hidden and visible choice operators

We now prove algebraic properties of channel operators. These properties will be useful when we model a (more complex) protocol as the composition of smaller channels via hidden or visible choice.

Whereas the properties of hidden choice hold generally with equality, those of visible choice are subtler. For instance, visible choice is not idempotent, since in general $C \, {}_p \sqcup \, C \ne C$. (In fact if C has type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $C \, {}_p \sqcup \, C$ has type $\mathcal{X} \times (\mathcal{Y} \sqcup \mathcal{Y}) \to \mathbb{R}$.) However, idempotency and other properties involving visible choice hold if we replace the notion of equality with the more relaxed notion of "equivalence" between channels. Intuitively, two channels are equivalent if they have the same input space and yield the same value of vulnerability for every prior and every vulnerability function.

Definition 4 (Equivalence of channels) *Two compatible channels C_1 and C_2 with domain $\mathcal{X}$ are equivalent, denoted by $C_1 \approx C_2$, if for every prior $\pi \in \mathbb{D}\mathcal{X}$ and every posterior vulnerability $\mathbb{V}$ we have $\mathbb{V}[\pi, C_1] = \mathbb{V}[\pi, C_2]$.*

Two equivalent channels are indistinguishable from the point of view of information leakage, and in most cases we can just identify them. Indeed, nowadays there is a tendency to use *abstract channels* [23,5], which capture exactly the important behavior with respect to any form of leakage. In this paper, however, we cannot use abstract channels because the hidden choice operator needs a concrete representation in order to be defined unambiguously.

The first properties we prove regard idempotency of operators, which can be used do simplify the representation of some protocols.

Proposition 5 (Idempotency) *Given a family $\{C_i\}_{i\in\mathcal{I}}$ of channels s.t. $C_i = C$ for all $i \in \mathcal{I}$, and a distribution μ on $\mathcal{I}$, then: (a) $\sum_{i\leftarrow\mu} C_i = C$; and (b) $\bigsqcup_{i\leftarrow\mu} C_i \approx C$.*

The following properties regard the reorganization of operators, and they will be essential in some technical results in which we invert the order in which hidden and visible choice are applied in a protocol.

Proposition 6 ("Reorganization of operators") *Given a family $\{C_{ij}\}_{i\in\mathcal{I},j\in\mathcal{J}}$ of channels indexed by sets $\mathcal{I}$ and $\mathcal{J}$, a distribution μ on $\mathcal{I}$, and a distribution η on $\mathcal{J}$:*

(a) $\sum_{i\leftarrow\mu} \sum_{j\leftarrow\eta} C_{ij} = \sum_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij}$, *if all C_i's have the same type;*

(b) $\bigsqcup_{i\leftarrow\mu} \bigsqcup_{j\leftarrow\eta} C_{ij} \approx \bigsqcup_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij}$, *if all C_i's are compatible; and*

(c) $\sum_{i\leftarrow\mu} \bigsqcup_{j\leftarrow\eta} C_{ij} \approx \bigsqcup_{j\leftarrow\eta} \sum_{i\leftarrow\mu} C_{ij}$, *if, for each i, all C_{ij}'s have same type $\mathcal{X}\times\mathcal{Y}_j\to\mathbb{R}$.*

4.4 Properties of vulnerability w.r.t. channel operators

We now derive some relevant properties of vulnerability w.r.t. our channel operators, which will be later used to obtain the Nash equilibria in information leakage games with different choice operations.

The first result states that posterior vulnerability is convex w.r.t. hidden choice (this result was already presented in [4]), and linear w.r.t. to visible choice.

Theorem 7 *Let $\{C_i\}_{i\in\mathcal{I}}$ be a family of channels, and μ be a distribution on $\mathcal{I}$. Then, for every distribution π on $\mathcal{X}$, and every vulnerability $\mathbb{V}$:*

(a) *posterior vulnerability is convex w.r.t. to hidden choice:* $\mathbb{V}\left[\pi, \sum_{i\leftarrow\mu} C_i\right] \leq \sum_{i\in\mathcal{I}} \mu(i)\,\mathbb{V}\left[\pi, C_i\right]$ *if all C_i's have the same type.*
(b) *posterior vulnerability is linear w.r.t. to visible choice:* $\mathbb{V}\left[\pi, \bigsqcup_{i\leftarrow\mu} C_i\right] = \sum_{i\in\mathcal{I}} \mu(i)\,\mathbb{V}\left[\pi, C_i\right]$ *if all C_i's are compatible.*

The next result is concerned with posterior vulnerability under the composition of channels using both operators.

Corollary 8 *Let $\{C_{ij}\}_{i\in\mathcal{I},j\in\mathcal{J}}$ be a family of channels, all with domain $\mathcal{X}$ and with the same type, and let $\pi \in \mathbb{D}\mathcal{X}$, and $\mathbb{V}$ be any vulnerability. Define $U : \mathbb{D}\mathcal{I} \times \mathbb{D}\mathcal{J} \to \mathbb{R}$ as follows: $U(\mu,\eta) \overset{\text{def}}{=} \mathbb{V}\left[\pi, \sum_{i\leftarrow\mu} \bigsqcup_{j\leftarrow\eta} C_{ij}\right]$. Then U is convex on μ and linear on η.*

5 Information leakage games

In this section we present our framework for reasoning about information leakage, extending the notion of *information leakage games* proposed in [4] from only simultaneous games with hidden choice to both simultaneous and sequential games, with either hidden or visible choice.

In an information leakage game the defender tries to minimize the leakage of information from the system, while the attacker tries to maximize it. In this basic scenario, their goals are just opposite (zero-sum). Both of them can influence the execution and the observable behavior of the system via a specific set of actions. We assume players to be rational (i.e., they are able to figure out what is the best strategy to maximize their expected payoff), and that the set of actions and the payoff function are common knowledge.

Players choose their own strategy, which in general may be mixed (i.e. probabilistic), and choose their action by a random draw according to that strategy. After both players have performed their actions, the system runs and produces some output value which is visible to the attacker and may leak some information about the secret. The amount of leakage constitutes the attacker's gain, and the defender's loss.

To quantify the leakage we model the system as an information-theoretic channel (cf. Section 2.2). We recall that leakage is defined as the difference (additive leakage) or the ratio (multiplicative leakage) between posterior and prior vulnerability. Since we are only interested in comparing the leakage of different channels for a given prior, *we will define the payoff just as the posterior vulnerability*, as the value of prior vulnerability will be the same for every channel.

5.1 Defining information leakage games

An (*information*) *leakage game* consists of: (1) two nonempty sets $\mathcal{D}$, $\mathcal{A}$ of defender's and attacker's actions respectively, (2) a function $C : \mathcal{D} \times \mathcal{A} \to (\mathcal{X} \times \mathcal{Y} \to \mathbb{R})$ that associates to each pair of actions $(d, a) \in \mathcal{D} \times \mathcal{A}$ a channel $C_{da} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, (3) a prior $\pi \in \mathbb{D}\mathcal{X}$ on secrets, and (4) a vulnerability measure $\mathbb{V}$. The payoff function $u : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$ for pure strategies is defined as $u(d, a) \stackrel{\text{def}}{=} \mathbb{V}[\pi, C_{da}]$. We have only one payoff function because the game is zero-sum.

Like in traditional game theory, the order of actions and the extent by which a player knows the move performed by the opponent play a critical role in deciding strategies and determining the payoff. In security, however, knowledge of the opponent's move affects the game in yet another way: the effectiveness of the attack, i.e., the amount of leakage, depends crucially on whether or not the attacker knows what channel is being used. It is therefore convenient to distinguish two phases in the leakage game:

Phase 1: Each player determines the most convenient strategy (which in general is mixed) for himself, and draws his action accordingly. One of the players may commit first to his action, and his choice may or may not be

revealed to the follower. In general, knowledge of the leader's action may help the follower choose a more advantageous strategy.

Phase 2: The attacker observes the output of the selected channel C_{da} and performs his attack on the secret. In case he knows the defender's action, he is able to determine the exact channel C_{da} being used (since, of course, the attacker knows his own action), and his payoff will be the posterior vulnerability $\mathbb{V}[\pi, C_{da}]$. However, if the attacker does not know exactly which channel has been used, then his payoff will be smaller.

Note that the issues raised in Phase 2 are typical of leakage games; they do not have a correspondence (to the best of our knowledge) in traditional game theory. On the other hand, these issues are central to security, as they reflect the principle of preventing the attacker from inferring the secret by obfuscating the link between secret and observables.

Following the above discussion, we consider various possible scenarios for games, along two lines of classification. First, there are three possible orders for the two players' actions.

Simultaneous: The players choose (draw) their actions in parallel, each without knowing the choice of the other.

Sequential, defender-first: The defender draws an action, and commits to it, before the attacker does.

Sequential, attacker-first: The attacker draws an action, and commits to it, before the defender does.

Note that these sequential games may present imperfect information (i.e., the follower may not know the leader's action).

Second, the visibility of the defender's action during the attack may vary:

Visible choice: The attacker knows the defender's action when he observes the output of the channel, and therefore he knows which channel is being used. Visible choice is modeled by the operator $\lfloor \cdot \rfloor$.

Hidden choice: The attacker does not know the defender's action when he observes the output of the channel, and therefore in general he does not exactly know which channel is used (although in some special cases he may infer it from the output). Hidden choice is modeled by the operator $\sum$.

Note that the distinction between sequential and simultaneous games is orthogonal to that between visible and hidden choice. Sequential and simultaneous games model whether or not, respectively, the follower's choice can be affected by knowledge of the leader's action. This dichotomy captures how knowledge about the other player's actions can *help a player choose his own action*. On the other hand, visible and hidden choice capture whether or not, respectively, the attacker is able to fully determine the channel representing the system, *once defender and attacker's actions have already been fixed*. This dichotomy reflects the different *amounts of information leaked* by the system as viewed by the adversary. For instance, in a simultaneous game neither player can choose his action

15

		Order of action		
		simultaneous	**defender 1$^{\text{st}}$**	**attacker 1$^{\text{st}}$**
Defender's	**visible** $\lfloor\cdot\rceil$	Game I	Game II	Game III
choice	**hidden** $\displaystyle\sum$	Game IV	Game V	Game VI

Table 2. Kinds of games we consider. All sequential games have perfect information, except for game V.

based on the choice of the other. However, depending on whether or not the defender's choice is visible, the adversary will or will not, respectively, be able to completely recover the channel used, which will affect the amount of leakage.

If we consider also the subdivision of sequential games into perfect and imperfect information, there are 10 possible different combinations. Some, however, make little sense. For instance, defender-first sequential game with perfect information (by the attacker) does not combine naturally with hidden choice $\sum$, since that would mean that the attacker knows the action of the defender and choses his strategy accordingly, but forgets it at the moment of the attack. (We assume *perfect recall*, i.e., the players never forget what they have learned.) Yet other combinations are not interesting, such as the attacker-first sequential game with (totally) imperfect information (by the defender), since it coincides with the simultaneous-game case. Note that attacker and defender are not symmetric with respect to hiding/revealing their actions a and d, since the knowledge of a affects the game only in the usual sense of game theory, while the knowledge of d also affects the computation of the payoff (cf. "Phase 2" above).

Table 2 lists the meaningful and interesting combinations. In Game V we assume imperfect information: the attacker does not know the action chosen by the defender. In all the other sequential games we assume that the follower has perfect information. In the remaining of this section, we discuss each game individually, using the example of Section 3 as running example.

Game I (simultaneous with visible choice) This simultaneous game can be represented by a tuple $(\mathcal{D}, \mathcal{A}, u)$. As in all games with visible choice $\lfloor\cdot\rceil$, the expected payoff U of a mixed strategy profile (δ, α) is defined to be the expected value of u, as in traditional game theory: $U(\delta, \alpha) \stackrel{\text{def}}{=} \mathbb{E}_{\substack{d\leftarrow\delta \\ a\leftarrow\alpha}} u(d, a) = \sum_{\substack{d\in\mathcal{D} \\ a\in\mathcal{A}}} \delta(d)\,\alpha(a)\,u(d, a)$, where we recall that $u(d, a) = \mathbb{V}[\pi, C_{da}]$.

From Theorem 7(b) we derive: $U(\delta, \alpha) = \mathbb{V}\left[\pi, \lfloor\cdot\rceil_{\substack{d\leftarrow\delta \\ a\leftarrow\alpha}} C_{da}\right]$. hence the whole system can be equivalently regarded as the channel $\lfloor\cdot\rceil_{\substack{d\leftarrow\delta \\ a\leftarrow\alpha}} C_{da}$. Still from Theorem 7(b) we can derive that $U(\delta, \alpha)$ is linear in δ and α. Therefore the Nash equilibrium can be computed using the minimax method (cf. Section 2.1).

Example 9 *Consider the example of Section 3 in the setting of Game I. The Nash equilibrium (δ^*, α^*) can be obtained using the closed formula from Section 2.1, and it is given by $\delta^*(0) = \alpha^*(0) = (2/3-1)/(1/2-1-1+2/3) = 2/5$. The corresponding payoff is $U(\delta^*, \alpha^*) = 2/5\,2/5\,1/2 + 2/5\,3/5 + 3/5\,2/5 + 3/5\,3/5\,2/3 = 4/5$.*

Game II (defender 1$^{\text{st}}$ with visible choice) This defender-first sequential game can be represented by a tuple $(\mathcal{D}, \mathcal{D} \to \mathcal{A}, u)$. A mixed strategy profile is of the form $(\delta, \sigma_{\mathsf{a}})$, with $\delta \in \mathbb{D}\mathcal{D}$ and $\sigma_{\mathsf{a}} \in \mathbb{D}(\mathcal{D} \to \mathcal{A})$, and the corresponding payoff is $U(\delta, \sigma_{\mathsf{a}}) \overset{\text{def}}{=} \mathbb{E}_{\substack{d \leftarrow \delta \\ s_{\mathsf{a}} \leftarrow \sigma_{\mathsf{a}}}} u(d, s_{\mathsf{a}}(d)) = \sum_{\substack{d \in \mathcal{D} \\ s_{\mathsf{a}}: \mathcal{D} \to \mathcal{A}}} \delta(d)\, \sigma_{\mathsf{a}}(s_{\mathsf{a}})\, u(d, s_{\mathsf{a}}(d))$, where $u(d, s_{\mathsf{a}}(d)) = \mathbb{V}\big[\pi, C_{d s_{\mathsf{a}}(d)}\big]$.

Again, from Theorem 7(b) we derive: $U(\delta, \sigma_{\mathsf{a}}) = \mathbb{V}\Big[\pi, \big\lfloor \cdot \big\rfloor_{\substack{d \leftarrow \delta \\ s_{\mathsf{a}} \leftarrow \sigma_{\mathsf{a}}}} C_{d s_{\mathsf{a}}(d)}\Big]$ and hence the system can be expressed as channel $\big\lfloor \cdot \big\rfloor_{\substack{d \leftarrow \delta \\ s_{\mathsf{a}} \leftarrow \sigma_{\mathsf{a}}}} C_{d s_{\mathsf{a}}(d)}$. From the same Theorem we also derive that $U(\delta, \sigma_{\mathsf{a}})$ is linear in δ and σ_{a}, so the mutually optimal strategies can be obtained again by solving the minimax problem. In this case, however, the solution is particularly simple, because it is known that there are optimal strategies which are deterministic. Hence it is sufficient for the defender to find the action d which minimizes $\max_a u(d, a)$.

Example 10 *Consider the example of Section 3 in the setting of Game II. If the defender chooses 0 then the attacker chooses 1. If the defender chooses 1 then the attacker chooses 0. In both cases, the payoff is 1. The game has therefore two solutions, $(0, 1)$ and $(1, 0)$.*

Game III (attacker 1$^{\text{st}}$ with visible choice) This game is also a sequential game, but with the attacker as the leader. Therefore it can be represented as tuple of the form $(\mathcal{A} \to \mathcal{D}, \mathcal{A}, u)$. It is the same as Game II, except that the roles of the attacker and the defender are inverted. In particular, the payoff of a mixed strategy profile $(\sigma_{\mathsf{d}}, \alpha) \in \mathbb{D}(\mathcal{A} \to \mathcal{D}) \times \mathbb{D}\mathcal{A}$ is given by $U(\sigma_{\mathsf{d}}, \alpha) \overset{\text{def}}{=} \mathbb{E}_{\substack{s_{\mathsf{d}} \leftarrow \sigma_{\mathsf{d}} \\ a \leftarrow \alpha}} u(s_{\mathsf{d}}(a), a) = \sum_{\substack{s_{\mathsf{d}}: \mathcal{A} \to \mathcal{D} \\ a \in \mathcal{A}}} \sigma_{\mathsf{d}}(s_{\mathsf{d}})\, \alpha(a)\, u(s_{\mathsf{d}}(a), a) = \mathbb{V}\Big[\pi, \big\lfloor \cdot \big\rfloor_{\substack{s_{\mathsf{d}} \leftarrow \sigma_{\mathsf{d}} \\ a \leftarrow \alpha}} C_{s_{\mathsf{d}}(a) a}\Big]$, and the whole system can be equivalently regarded as channel $\big\lfloor \cdot \big\rfloor_{\substack{s_{\mathsf{d}} \leftarrow \sigma_{\mathsf{d}} \\ a \leftarrow \alpha}} C_{s_{\mathsf{d}}(a) a}$. Obviously, also in this case the minimax problem has a deterministic solution.

In summary, in the sequential case, whether the leader is the defender or the attacker (Games II and III, respectively), the minimax problem has always a deterministic solution [24].

Theorem 11 *In a defender-first sequential game with visible choice, there exist $d \in \mathcal{D}$ and $a \in \mathcal{A}$ such that, for every $\delta \in \mathbb{D}\mathcal{D}$ and $\sigma_{\mathsf{a}} \in \mathbb{D}(\mathcal{D} \to \mathcal{A})$ we have: $U(d, \sigma_{\mathsf{a}}) \leq u(d, a) \leq U(\delta, a)$. Similarly, in an attacker-first sequential game with visible choice, there exist $d \in \mathcal{D}$ and $a \in \mathcal{A}$ such that, for every $\sigma_{\mathsf{d}} \in \mathbb{D}(\mathcal{A} \to \mathcal{D})$ and $\alpha \in \mathbb{D}\mathcal{A}$ we have: $U(d, \alpha) \leq u(d, a) \leq U(\sigma_{\mathsf{d}}, a)$.*

Example 12 *Consider now the example of Section 3 in the setting of Game III. If the attacker chooses 0 then the defender chooses 0 and the payoff is $^1/_2$. If the attacker chooses 1 then the defender chooses 1 and the payoff is $^2/_3$. The latter case is more convenient for the attacker, hence the solution of the game is the strategy profile $(1, 1)$.*

Game IV (simultaneous with hidden choice) This game is a tuple $(\mathcal{D}, \mathcal{A}, u)$. However, *it is not an ordinary game* in the sense that *the payoff a mixed strategy profile cannot be defined by averaging the payoff of the corresponding pure strategies.* More precisely, the payoff of a mixed profile is defined by averaging on the strategy of the attacker, but not on that of the defender. In fact, when hidden choice is used, there is an additional level of uncertainty in the relation between the observables and the secret from the point of view of the attacker, since he is not sure about which channel is producing those observables. A mixed strategy δ for the defender produces a convex combination of channels (the channels associated to the pure strategies) with the same coefficients, and we know from previous sections that the vulnerability is a convex function of the channel, and in general is not linear.

In order to define the payoff of a mixed strategy profile (δ, α), we need therefore to consider the channel that the attacker perceives given his limited knowledge. Let us assume that the action that the attacker draws from α is a. He does not know the action of the defender, but we can assume that he knows his strategy (each player can derive the optimal strategy of the opponent, under the assumption of common knowledge and rational players).

The channel the attacker will see is $\sum_{d \leftarrow \delta} C_{da}$, obtaining a corresponding payoff of $\mathbb{V}\big[\pi, \sum_{d \leftarrow \delta} C_{da}\big]$. By averaging on the strategy of the attacker we obtain $U(\delta, \alpha) \overset{\text{def}}{=} \mathbb{E}_{a \leftarrow \alpha} \mathbb{V}\big[\pi, \sum_{d \leftarrow \delta} C_{da}\big] = \sum_{a \in \mathcal{A}} \alpha(a) \mathbb{V}\big[\pi, \sum_{d \leftarrow \delta} C_{da}\big]$. From Theorem 7(b) we derive: $U(\delta, \alpha) = \mathbb{V}\big[\pi, \big[\cdot\big]_{a \leftarrow \alpha} \sum_{d \leftarrow \delta} C_{da}\big]$ and hence the whole system can be equivalently regarded as channel $\big[\cdot\big]_{a \leftarrow \alpha} \sum_{d \leftarrow \delta} C_{da}$. Note that, by Proposition 6(c), the order of the operators is interchangeable, and the system can be equivalently regarded as $\sum_{d \leftarrow \delta} \big[\cdot\big]_{a \leftarrow \alpha} C_{da}$. This shows the robustness of this model.

From Corollary 8 we derive that $U(\delta, \alpha)$ is convex in δ and linear in η, hence we can compute the Nash equilibrium by the minimax method.

Example 13 *Consider now the example of Section 3 in the setting of Game IV. For $\delta \in \mathbb{D}\mathcal{D}$ and $\alpha \in \mathbb{D}\mathcal{A}$, let $p = \delta(0)$ and $q = \alpha(0)$. The system can be represented by the channel $(C_{00\ p} \oplus C_{10})\ _q \sqcup (C_{01\ p} \oplus C_{11})$ represented below.*

$C_{00\ p} \oplus C_{10}$	$y = 0$	$y = 1$
$x = 0$	p	$\bar{p}$
$x = 1$	1	0

$_q \sqcup$

$C_{01\ p} \oplus C_{11}$	$y = 0$	$y = 1$
$x = 0$	$1/3 + 2/3\,p$	$2/3 - 2/3\,p$
$x = 1$	$2/3 - 2/3\,p$	$1/3 + 2/3\,p$

For uniform π, we have $\mathbb{V}\big[\pi, C_{00\ p} \oplus C_{10}\big] = 1 - 1/2$; and $\mathbb{V}\big[\pi, C_{10\ p} \oplus C_{11}\big]$ is equal to $2/3 - 2/3\,p$ if $p \le 1/4$, and equal to $1/3 + 2/3\,p$ if $p > 1/4$. Hence the payoff, expressed in terms of p and q, is $U(p, q) = q(1 - 1/2) + \bar{q}(2/3 - 2/3\,p)$ if $p \le 1/4$, and $U(p, q) = q(1 - 1/2) + \bar{q}(1/3 + 2/3\,p)$ if $p > 1/4$. The Nash equilibrium (p^*, q^*) is given by $p^* = \operatorname{argmin}_p \max_q U(p, q)$ and $q^* = \operatorname{argmax}_q \min_p U(p, q)$, and by solving the above, we obtain $p^* = q^* = 4/7$.

Game V (defender 1$^{\text{st}}$ with hidden choice) This is a defender-first sequential game with imperfect information, hence it can be represented as a tuple of

18

the form $(\mathcal{D}, K_\mathsf{a} \to \mathcal{A}, u_\mathsf{d}, u_\mathsf{a})$, where K_a is a partition of $\mathcal{D}$. Since we are assuming perfect recall, and the attacker does not know anything about the action chosen by the defender in Phase 2, i.e., at the moment of the attack (except the probability distribution determined by his strategy), we must assume that the attacker does not know anything in Phase 1 either. Hence the indistinguishability relation must be total, i.e., $K_\mathsf{a} = \{\mathcal{D}\}$. But $\{\mathcal{D}\} \to \mathcal{A}$ is equivalent to $\mathcal{A}$, hence this kind of game is equivalent to Game IV.

It is also a well known fact in Game theory that when in a sequential game the follower does not know the leader's move before making his choice, the game is equivalent to a simultaneous game.[11]

Game VI (attacker 1$^{\text{st}}$ with hidden choice) This game is also a sequential game with the attacker as the leader, hence it is a tuple of the form $(\mathcal{A} \to \mathcal{D}, \mathcal{A}, u)$. It is similar to Game III, except that the payoff is convex on the strategy of the defender, instead of linear. The payoff of the mixed strategy profile $(\sigma_\mathsf{d}, \alpha) \in \mathbb{D}(\mathcal{A} \to \mathcal{D}) \times \mathbb{D}\mathcal{A}$ is $U(\sigma_\mathsf{d}, \alpha) \stackrel{\text{def}}{=} \mathbb{E}_{a \leftarrow \alpha} \mathbb{V}\left[\pi, \sum_{s_\mathsf{d} \leftarrow \sigma_\mathsf{d}} C_{s_\mathsf{d}(a)a}\right] = \mathbb{V}\left[\pi, \sum_{a \leftarrow \alpha} \bigsqcup_{s_\mathsf{d} \leftarrow \sigma_\mathsf{d}} C_{s_\mathsf{d}(a)a}\right]$, so the whole system can be equivalently regarded as channel $\sum_{a \leftarrow \alpha} \bigsqcup_{s_\mathsf{d} \leftarrow \sigma_\mathsf{d}} C_{s_\mathsf{d}(a)a}$. Also in this case the minimax problem has a deterministic solution, but only for the attacker.

Theorem 14 *In an attacker-first sequential game with hidden choice, there exist $a \in \mathcal{A}$ and $\delta \in \mathbb{D}\mathcal{D}$ such that, for every $\alpha \in \mathbb{D}\mathcal{A}$ and $\sigma_\mathsf{d} \in \mathbb{D}(\mathcal{A} \to \mathcal{D})$ we have that $U(\delta, \alpha) \le U(\delta, a) \le U(\sigma_\mathsf{d}, a)$.*

Example 15 *Consider again the example of Section 3, this time in the setting of Game VI. Consider also the calculations made in Example 13, we will use the same results and notation here. In this setting, the attacker is obliged to make its choice first. If he chooses 0, which corresponds to committing to the system $C_{00}\ _p\oplus\ C_{10}$, then the defender will choose $p = 1/4$, which minimizes its vulnerability. If he chooses 1, which corresponds to committing to the system $C_{01}\ _p\oplus\ C_{11}$, the defender will choose $p = 1$, which minimizes its vulnerability of the above channel. In both cases, the leakage is $p = 1/2$, hence both these strategies are solutions to the minimax. Note that in the first case the strategy of the defender is mixed, while that of the attacker is always pure.*

5.2 Comparing the games

If we look at the various payoffs obtained for the running example in the various games, we obtain the following values (listed in decreasing order): II : 1; I : 4/5; III : 2/3; IV : 4/7; V : 4/7; VI : 1/2.

---

[11] However, one could argue that, since the defender has already committed, the attacker does not need to perform the action corresponding to the Nash equilibrium, any payoff-maximizing solution would be equally good for him.

This order is not accidental: for any vulnerability function, and for any prior, the various games are ordered, with respect to the payoff, as shown in Figure 2. The relations between II, I, and III, and between IV-V and VI come from the fact that, in any zero-sum sequential game the leader's payoff will be less or equal to his payoff in the corresponding simultaneous game. We think this result is well-known in game theory, but we give the hint of the proof nevertheless, for the sake of clarity.
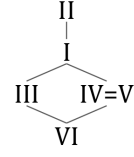
Fig. 2. Order of games w.r.t. payoff. Games higher in the lattice have larger payoff.

Theorem 16 *It is the case that:*

(a) $\min_\delta \max_{\sigma_a} \mathbb{V}\left[\pi, \bigsqcup_{\substack{d \leftarrow \delta \\ s_a \leftarrow \sigma_a}} C_{ds_a(d)}\right] \geq \min_\delta \max_\alpha \mathbb{V}\left[\pi, \bigsqcup_{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}} C_{da}\right]$

$$\geq \max_\alpha \min_{\sigma_d} \mathbb{V}\left[\pi, \bigsqcup_{\substack{s_d \leftarrow \sigma_d \\ a \leftarrow \alpha}} C_{s_d(a)a}\right]$$

(b) $\min_\delta \max_\alpha \mathbb{V}\left[\pi, \bigsqcup_{a \leftarrow \alpha} \sum_{d \leftarrow \delta} C_{da}\right] \geq \max_\alpha \min_{\sigma_d} \mathbb{V}\left[\pi, \sum_{a \leftarrow \alpha} \bigsqcup_{s_d \leftarrow \sigma_d} C_{s_d(a)a}\right]$

Proof. We prove the first inequality in (a). Independently of δ, consider the attacker strategy τ_a that assigns probability 1 to the function s_a defined as $s_a(d) = \operatorname{argmax}_a \mathbb{V}[\pi, C_{da}]$. Then we have that

$$\min_\delta \max_{\sigma_a} \mathbb{V}\left[\pi, \bigsqcup_{\substack{d \leftarrow \delta \\ s_a \leftarrow \sigma_a}} C_{ds_a(d)}\right] \geq \min_\delta \mathbb{V}\left[\pi, \bigsqcup_{\substack{d \leftarrow \delta \\ s_a \leftarrow \tau_a}} C_{ds_a(d)}\right]$$

$$\geq \min_\delta \max_\alpha \mathbb{V}\left[\pi, \bigsqcup_{\substack{d \leftarrow \delta \\ a \leftarrow \alpha}} C_{da}\right]$$

Note that the strategy τ_a is optimal for the adversary, so the first of the above inequalities is actually an equality. All other cases can be proved with an analogous reasoning. □

Concerning III and IV-V: these are not related. In the running example the payoff for III is higher than for IV-V, but it is easy to find other cases in which the situation is reversed. For instance, if in the running example we set C_{11} to be the same as C_{00}, the payoff for III will be $1/2$, and that for IV-V will be $2/3$.

Finally, the relation between III and VI comes from the fact that they are both attacker-first sequential games, and the only difference is the way in which the payoff is defined. Then, just observe that in general we have, for every $a \in \mathcal{A}$ and every $\delta \in \mathbb{DD}$: $\mathbb{V}\left[\pi, \sum_{d \leftarrow \delta} C_{da}\right] \leq \mathbb{V}\left[\pi, \bigsqcup_{d \leftarrow \delta} C_{da}\right]$.

The relations in Figure 2 can be used by the defender as guidelines to better protect the system, if he has some control over the rules of the game. Obviously, for the defender the games lower in the ordering are to be preferred.

6 Case study: a safer, faster password-checker

In this section we apply our game-theoretic, compositional approach to show how a defender can mitigate an attacker's typical timing side-channel attack while avoiding the usual burden imposed on the password-checker's efficiency.

Consider the password-checker $\mathtt{PWD}_{123}$ of Figure 3, which performs a bitwise-check of a 3-bit low-input $a=a_1a_2a_3$, provided by the attacker, against a 3-bit secret password $x=x_1x_2x_3$. The low-input is rejected as soon as it mismatches the secret, and is accepted otherwise.

```
Program PWD₁₂₃

High Input: x ∈ {000, 001, . . . , 111}
Low Input: a ∈ {000, 001, . . . , 111}
Output: y ∈ {T, F}
accept := T
for i = 1, 2, 3 do
    if aᵢ ≠ xᵢ then
        accept := F
        break
    end if
end for
return accept
```

Fig. 3. Password-checker algorithm.

The attacker can choose low-inputs to try to gain information about the password. Obviously, in case $\mathtt{PWD}_{123}$ accepts the low-input, the attacker learns the password value is $a=x$. Yet, even when the low-input is rejected, there is some leakage of information: from the duration of the execution the attacker can estimate how many iterations have been performed before the low-input was rejected, thus inferring a prefix of the secret password.

To model this scenario, let $\mathcal{X} = \{000, 001, \ldots, 111\}$ be the set of all possible 3-bit passwords, and $\mathcal{Y} = \{(F, 1), (F, 2), (F, 3), (T, 3)\}$ be the set of observables produced by the system. Each observable is an ordered pair whose first element indicates whether the password was accepted (T or F), and the second element indicates the duration of the computation (1, 2, or 3 iterations). For instance, channel $C_{123,101}$ in Figure 4 models $\mathtt{PWD}_{123}$'s behavior when the attacker provides low-input $a=101$.

We will adopt as a measure of information *Bayes vulnerability* [27]. The *prior Bayes vulnerability* of a distribution $\pi \in \mathbb{D}\mathcal{X}$ is defined as $V_g[\pi] = \max_{x \in \mathcal{X}} \pi_x$, and represents the probability that the attacker guesses correctly the password in one try. For instance, if the distribution on all possible 3-bit passwords is $\hat{\pi} = (0.0137, 0.0548, 0.2191, 0.4382, 0.0002, 0.0002, 0.0548, 0.2191)$, its prior Bayes vulnerability is $\mathbb{V}[\hat{\pi}] = 0.4382$.

The *posterior Bayes vulnerability* of a prior π and a channel $C : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is defined as $\mathbb{V}[\pi, C] = \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} \pi_x C(x, y)$, and it represents the probability that the attacker guesses correctly the password in one try, after he observes the output of the channel (i.e., after he has measured the time needed for the checker to accept or reject the low-input). For prior $\hat{\pi}$ above, the posterior Bayes vulnerability of channel $C_{123,101}$ is $\mathbb{V}[\hat{\pi}, C_{123,101}] = 0.6577$ (which represents an increase in Bayes vulnerability of about 50%), and the expected running time for this checker is of 1.2747 iterations.

A way to mitigate this timing side-channel is to make the checker's execution time independent of the secret. Channel $C_{\text{cons},101}$ from Figure 4 models a checker that does that (by eliminating the **break** command within the loop in $\mathtt{PWD}_{123}$) when the attacker's low-input is $a=101$. This channel's posterior Bayes vulner-

ability is $\mathbb{V}\big[\hat\pi, C_{123,101}\big]$=0.4384, which brings the multiplicative Bayes leakage down to an increase of only about 0.05%. However, the expected running time goes up to 3 iterations (an increase of about 135% w.r.t. that of $C_{123,101}$).

Seeking some compromise between security and efficiency, assume that the defender can employ password-checkers that perform the bitwise comparison among low-input a and secret password x in different orders. More precisely, there is one version of the checker for ev-

$C_{123,101}$	$y=$ $(F,1)$	$y=$ $(F,2)$	$y=$ $(F,3)$	$y=$ $(T,3)$
x=000	1	0	0	0
x=001	1	0	0	0
x=010	1	0	0	0
x=011	1	0	0	0
x=100	0	0	1	0
x=101	0	0	0	1
x=110	0	1	0	0
x=111	0	1	0	0

$C_{\mathrm{cons},101}$	$y=$ $(F,3)$	$y=$ $(T,3)$
x=000	1	0
x=001	1	0
x=010	1	0
x=011	1	0
x=100	1	0
x=101	0	1
x=110	1	0
x=111	1	0

Fig. 4. Channels C_{da} modeling the password checker for defender's action d and attacker's action a.

ery possible order in which the index i ranges in the control of the loop. For instance, while PWD$_{123}$ checks the bits in the order $1, 2, 3$, the alternative algorithm PWD$_{231}$ uses the order $2, 3, 1$.

To determine a defender's best choice of which versions of the checker to run, we model this problem as game. The attacker's actions $\mathcal{A} = \{000, 001, \ldots, 111\}$ are all possible low-inputs to the checker, and the defender's $\mathcal{D} = \{123, 132, 213, 231, 312, 321\}$ are all orders to perform the comparison. Hence, there is a total of 48 possible channels $C_{ad}:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}$, one for each combination of $d\in\mathcal{D}$, $a\in\mathcal{A}$.

In our framework, the utility of a mixed strategy profile (δ, α) is given by $U(\delta, \alpha) = \mathbb{E}_{a\leftarrow\alpha}\mathbb{V}\big[\pi, \sum_{d\leftarrow\delta}C_{da}\big]$. For each pure strategy profile (d, a), the payoff of the game will be the posterior Bayes vulnerability of the resulting channel C_{da} (since, if we measuring leakage, the prior vulnerability is the same for every channel once the prior is fixed). Table 3 depicts such payoffs. Note that the attacker's and defender's actions substantially affect the effectiveness of the attack: vulnerability ranges between 0.4934 and 0.9311 (and so multiplicative leakage is in the range between an increase of 12% and one of 112%). Using techniques from [4], we can compute the best (mixed) strategy for the defender in this game, which turns out to be $\delta^* = (0.1667, 0.1667, 0.1667, 0.1667, 0.1667, 0.1667)$. This strategy is part of an equilibrium and guarantees that for any choice of the attacker the posterior Bayes vulnerability is at most 0.6573 (so the multiplicative leakage is bounded by 50%, an intermediate value between the minimum of about 12% and the maximum of about 112%). It is interesting to note that the expected running time, for any action of the attacker, is bounded by at most 2.3922 iterations (an increase of only 87% w.r.t. the channel PWD$_{123}$), which is below the worst possible expected 3 iterations of the constant-time password checker.

	Attacker's action a							
$U(d,a)$	000	001	010	011	100	101	110	111
123	0.7257	0.7257	0.9311	0.9311	0.6577	0.6577	0.7122	0.7122
132	0.8900	0.9311	0.8900	0.9311	0.7122	0.7122	0.7122	0.7122
213	0.5068	0.5068	0.9311	0.9311	0.4934	0.4934	0.7668	0.7668
231	0.5068	0.5068	0.7668	0.9311	0.5068	0.5068	0.7668	0.9311
312	0.7257	0.9311	0.7257	0.9311	0.7122	0.8766	0.7122	0.8766
321	0.6712	0.7122	0.7257	0.9311	0.6712	0.7122	0.7257	0.9311

(Defender's action d)

Table 3. Utility for each pure strategy profile.

7   Related work

Many studies have applied game theory to analyses of security and privacy in networks [7,14,3], cryptography [15], anonymity [1], location privacy [13], and intrusion detection [30], to cite a few. See [20] for a survey.

In the context of quantitative information flow, most works consider only passive attackers. Boreale and Pampaloni [8] consider adaptive attackers, but not adaptive defenders, and show that in this case the adversary's optimal strategy can be always deterministic. Mardziel et al. [21] propose a model for both adaptive attackers and defenders, but in none of their extensive case-studies the attacker needs a probabilistic strategy to maximize leakage. In this paper we characterize when randomization is necessary, for either attacker or defender, to achieve optimality in our general information leakage games.

Security games have been employed to model and analyze payoffs between interacting agents, especially between a defender and an attacker. Korzhyk et al. [19] theoretically analyze security games and study the relationships between Stackelberg and Nash Equilibria under various forms of imperfect information. Khouzani and Malacaria [18] study leakage properties when perfect secrecy is not achievable due to constraints on the allowable size of the conflating sets, and provide universally optimal strategies for a wide class of entropy measures, and for g-entropies. These works, contrarily to ours, do not consider games with hidden choice, in which optimal strategies differ from traditional game-theory.

Several security games have modeled leakage when the sensitive information are the defender's choices themselves, rather than a system's high input. For instance, Alon et al. [2] propose zero-sum games in which a defender chooses probabilities of secrets and an attacker chooses and learns some of the defender's secrets. Then they present how the leakage on the defender's secrets gives influences on the defender's optimal strategy. More recently, Xu et al. [29] show zero-sum games in which the attacker obtains partial knowledge on the security resources that the defender protects, and provide the defender's optimal strategy under the attacker's such knowledge.

Regarding channel operators, sequential and parallel composition of channels have been studied (e.g., [17]), but we are unaware of any explicit definition and investigation of hidden and visible choice operators. Although Kawamoto et al. [16] implicitly use the hidden choice to model a probabilistic system as the weighted sum of systems, they do not derive the set of algebraic properties we do for this operator, and for its interaction with the visible choice operator.

8   Conclusion and future work

In this paper we used protocol composition to model the introduction of noise performed by the defender to prevent leakage of sensitive information. More precisely, we formalized visible and hidden probabilistic choices of different protocols. We then formalized the interplay between defender and adversary in a game-theoretic framework adapted to the specific issues of QIF, where the payoff

is information leakage. We considered various kinds of leakage games, depending on whether players act simultaneously or sequentially, and whether the choices of the defender are visible or not to the adversary. We established a hierarchy of these games, and provided methods for finding the optimal strategies (at the points of equilibrium) in the various cases.

As future research, we would like to extend leakage games to the case of repeated observations, i.e., when the attacker can observe the outcomes of the system in successive runs, under the assumption that both attacker and defender may change the channel in each run. We would also like to extend our framework to non zero-sum games, in which the costs of attack and defense are not equivalent, and to analyze differentially-private mechanisms.

References

1. Alessandro Acquisti, Roger Dingledine, and Paul F. Syverson. On the economics of anonymity. In *Proc. of FC*, pages 84–102, 2003.
2. Noga Alon, Yuval Emek, Michal Feldman, and Moshe Tennenholtz. Adversarial leakage in games. *SIAM J. Discrete Math.*, 27(1):363–385, 2013.
3. Tansu Alpcan and Sonja Buchegger. Security games for vehicular networks. *IEEE Trans. Mob. Comput.*, 10(2):280–290, 2011.
4. Mário S. Alvim, Konstantinos Chatzikokolakis, Yusuke Kawamoto, and Catuscia Palamidessi. Information leakage games. In *Proc. of GameSec*, volume 10575 of *LNCS*, pages 437–457. Springer, 2017.
5. Mário S. Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. Axioms for information leakage. In *Proc. of CSF*, pages 77–92, 2016.
6. Mário S. Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *Proc. of CSF*, pages 265–279, 2012.
7. Tamer Basar. The gaussian test channel with an intelligent jammer. *IEEE Trans. Information Theory*, 29(1):152–157, 1983.
8. Michele Boreale and Francesca Pampaloni. Quantitative information flow under generic leakage functions and adaptive adversaries. *Logical Methods in Computer Science*, 11(4), 2015.
9. Christelle Braun, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proc. of MFPS*, volume 249 of *ENTCS*, pages 75–91. Elsevier, 2009.
10. Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Prakash Panangaden. On the Bayes risk in information-hiding protocols. *J. of Comp. Security*, 16(5):531–571, 2008.

11. David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.
12. Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 265–284. Springer, 2006.
13. Julien Freudiger, Mohammad Hossein Manshaei, Jean-Pierre Hubaux, and David C. Parkes. On non-cooperative location privacy: A game-theoretic analysis. In *Proc. of CCS*, pages 324–337, 2009.
14. Jens Grossklags, Nicolas Christin, and John Chuang. Secure or insure?: A game-theoretic analysis of information security games. In *Proc. of WWW*, pages 209–218, 2008.
15. Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. In *Proc. of TCC*, pages 251–272, 2008.
16. Yusuke Kawamoto, Fabrizio Biondi, and Axel Legay. Hybrid statistical estimation of mutual information for quantifying information flow. In *Proc. of FM 2016*, pages 406–425, 2016.
17. Yusuke Kawamoto, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. On the compositionality of quantitative information flow. *Logical Methods in Computer Science*, 13(3:11):1–31, 2017.
18. M. H. R. Khouzani and P. Malacaria. Relative perfect secrecy: Universally optimal strategies and channel design. In *Proc. of CSF*, pages 61–76, 2016.
19. Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Intell. Res.*, 41:297–327, 2011.
20. Mohammad Hossein Manshaei, Quanyan Zhu, Tansu Alpcan, Tamer Bacşar, and Jean-Pierre Hubaux. Game theory meets network security and privacy. *ACM Comput. Surv.*, 45(3):25:1–25:39, 2013.
21. Piotr Mardziel, Mário S. Alvim, Michael W. Hicks, and Michael R. Clarkson. Quantifying information flow for dynamic secrets. In *Proc. of S&P*, pages 540–555, 2014.
22. Massey. Guessing and entropy. In *Proceedings of the IEEE Int. Symposium on Information Theory*, page 204. IEEE, 1994.
23. Annabelle McIver, Carroll Morgan, Geoffrey Smith, Barbara Espinoza, and Larissa Meinicke. Abstract channels and their robust information-leakage ordering. In *Proc. of POST*, volume 8414 of *LNCS*, pages 83–102. Springer, 2014.
24. Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.
25. Juliano Rizzo and Thai Duong. The CRIME attack, 2012.
26. Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 625–56, 1948.
27. Geoffrey Smith. On the foundations of quantitative information flow. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 288–302. Springer, 2009.
28. Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Proc. of S&P*, pages 19–30. IEEE, 2002.
29. Haifeng Xu, Albert Xin Jiang, Arunesh Sinha, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe. Security games with information leakage: Modeling and computation. In *Proc. of IJCAI*, pages 674–680, 2015.
30. Quanyan Zhu, Carol J. Fung, Raouf Boutaba, and Tamer Basar. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *Proc. of GAMENETS*, pages 384–392. IEEE, 2009.

A Proofs of Technical Results

In this section we provide proofs for our technical results.

A.1 Preliminaries for proofs

We start by providing some necessary background for the subsequent technical proofs.

Definition 4 states that two compatible channels (i.e., with the same input space) are equivalent if yield the same value of vulnerability for every prior and every vulnerability function. The result below, from the literature, provides necessary and sufficient conditions for two channels being equivalent. The result employs the extension of channels with an all-zero column as follows. For any channel C of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, its *zero-column extension* C^0 is the channel of type $\mathcal{X} \times (\mathcal{Y} \cup \{y_0\}) \to \mathbb{R}$, with $y_0 \notin \mathcal{Y}$, s.t. $C^0(x,y) = C(x,y)$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $C^0(x,y_0) = 0$ for all $x \in \mathcal{X}$.

Lemma 17 (Characterization of channel equivalence [6,23]) *Two channels C_1 and C_2 are equivalent iff every column of C_1 is a convex combination of columns of C_2^0, and every column of C_2 is a convex combination of columns of C_1^0.*

Note that the result above implies that for being equivalent, any two channels must be compatible.

A.2 Proofs of Section 4

Proposition 2 *Given a family $\{C_i\}_{i \in \mathcal{I}}$ of channels of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, and a distribution μ on $\mathcal{I}$, the hidden choice $\sideset{}{}\sum_{i \leftarrow \mu} C_i$ is a channel of type $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.*

Proof. Since hidden choice is defined as a summation of matrices, the type of $\sideset{}{}\sum_{i \leftarrow \mu} C_i$ is the same as the type of every C_i in the family.

To see that $\sideset{}{}\sum_{i \leftarrow \mu} C_i$ is a channel (i.e., all of its entries are non-negative, and all of its rows sum up to 1), first note that, since each C_i in the family is a channel matrix, $C_i(x,y)$ lies in the interval $[0,1]$ for all $x \in \mathcal{X}$, $y \in \mathcal{Y}$. Since μ is a set of convex coefficients, from the definition of hidden choice it follows that also $\sum_{i \in \mathcal{I}} \mu(i) C_i(x,y)$ must lie in the interval $[0,1]$ for every x, y.

Second, note that for all $x \in \mathcal{X}$:

$$
\begin{aligned}
\sum_{y \in \mathcal{Y}} \left(\sideset{}{}\sum_{i \leftarrow \mu} C_i \right)(x,y) &= \sum_{y \in \mathcal{Y}} \sum_{i \in \mathcal{I}} \mu(i) C_i(x,y) && \text{(def. of hidden choice)} \\
&= \sum_{i \in \mathcal{I}} \mu(i) \sum_{y \in \mathcal{Y}} C_i(x,y) \\
&= \sum_{i \in \mathcal{I}} \mu(i) \cdot 1 && (C_i{:}\mathcal{X}{\times}\mathcal{Y}{\to}\mathbb{R} \text{ are channels}) \\
&= 1 && (\mu \text{ is a prob. dist.})
\end{aligned}
$$

Proposition 3 *Given a family $\{C_i\}_{i\in\mathcal{I}}$ of compatible channels s.t. each C_i has type $\mathcal{X}\times\mathcal{Y}_i\to\mathbb{R}$, and a distribution μ on $\mathcal{I}$, the result of the visible choice $\bigsqcup_{i\leftarrow\mu}C_i$ is a channel of type $\mathcal{X}\times\left(\bigsqcup_{i\in\mathcal{I}}\mathcal{Y}_i\right)\to\mathbb{R}$.*

Proof. Visible choice applied to a family $\{C_i\}$ of channels scales each matrix C_i by a factor $\mu(i)$, which preserves the type $\mathcal{X}\times\mathcal{Y}_i\to\mathbb{R}$ of each matrix, and then concatenates all the matrices so produced, yielding a result of type $\mathcal{X}\times\left(\bigsqcup_{i\in\mathcal{I}}\mathcal{Y}_i\right)\to\mathbb{R}$.

To see that $\bigsqcup_{i\leftarrow\mu}C_i$ is a channel (i.e., that all of its entries are non-negative, and that all rows sum-up to 1), note that each element of the visible choice on the family $\{C_i\}$ can be denoted by $\left(\bigsqcup_{i\leftarrow\mu}C_i\right)(x,(y,j))$, where $x\in\mathcal{X}$, $j\in\mathcal{I}$, and $y\in\mathcal{Y}_j$. Then, note that for all $x\in\mathcal{X}$, $j\in\mathcal{I}$, and $y\in\mathcal{Y}_j$:

$$\left(\bigsqcup_{i\leftarrow\mu}C_i\right)(x,(y,j)) = \left(\diamondsuit_{i\in\mathcal{I}}\mu(i)C_i\right)(x,(y,j)) \qquad \text{(def. of visible choice)}$$

$$= \left(\mu(j)C_j\right)(x,y) \qquad\qquad \text{(def. of concatenation)}$$

$$= \mu(j)C_j(x,y) \qquad\qquad\quad \text{(def. of scalar mult.)} \quad (1)$$

which is a non-negative value, since, both $\mu(j)$ and $C_j(x,y)$ are non-negative.

Finally, note that for all $x\in\mathcal{X}$:

$$\sum_{\substack{j\in\mathcal{I}\\y\in\mathcal{Y}_j}}\left(\bigsqcup_{i\leftarrow\mu}C_i\right)(x,(y,j)) = \sum_{\substack{j\in\mathcal{I}\\y\in\mathcal{Y}_j}}\mu(j)C_j(x,y) \qquad\qquad \text{(by Eq. (1))}$$

$$= \sum_{j\in\mathcal{I}}\mu(j)\sum_{y\in\mathcal{Y}_j}C_j(x,y)$$

$$= \sum_{j\in\mathcal{I}}\mu(j)\cdot 1 \qquad\quad (C_j\colon\mathcal{X}\times\mathcal{Y}_j\to\mathbb{R}\text{ are channels})$$

$$= 1 \qquad\qquad\qquad\qquad\quad (\mu\text{ is a prob. dist.})$$

Proposition 5 (Idempotency) *Given a family $\{C_i\}_{i\in\mathcal{I}}$ of channels s.t. $C_i = C$ for all $i\in\mathcal{I}$, and a distribution μ on $\mathcal{I}$, then: (a) $\underset{i\leftarrow\mu}{\bigvee}C_i = C$; and (b) $\bigsqcup_{i\leftarrow\mu}C_i \approx C$.*

Proof. a) Idempotency of hidden choice:

$$\underset{i\leftarrow\mu}{\bigvee}C_i = \sum_i\mu(i)C_i \qquad\qquad \text{(def. of hidden choice)}$$

$$= \sum_i\mu(i)C \qquad\qquad \text{(since every }C_i = C\text{)}$$

$$= C\sum_i\mu(i)$$

$$= C \qquad\qquad\qquad\quad (\text{since }\mu\text{ is a prob. dist.})$$

27

b) Idempotency of visible choice:

$$\left\lfloor \cdot \right\rfloor_{i\leftarrow\mu} C_i = \diamondsuit_i\,\mu(i)C_i \qquad\qquad \text{(def. of visible choice)}$$

$$= \diamondsuit_i\,\mu(i)C \qquad\qquad \text{(since every } C_i = C\text{)}$$

$$\approx C \qquad\qquad \text{(by Lemma 17)}$$

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

Proposition 6 ("Reorganization of operators") *Given a family $\{C_{ij}\}_{i\in\mathcal{I},j\in\mathcal{J}}$ of channels indexed by sets $\mathcal{I}$ and $\mathcal{J}$, a distribution μ on $\mathcal{I}$, and a distribution η on $\mathcal{J}$:*

(a) $\sum_{i\leftarrow\mu} \sum_{j\leftarrow\eta} C_{ij} = \sum_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij}$, if all C_i's have the same type;

(b) $\left\lfloor \cdot \right\rfloor_{i\leftarrow\mu} \left\lfloor \cdot \right\rfloor_{j\leftarrow\eta} C_{ij} \approx \left\lfloor \cdot \right\rfloor_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij}$, if all C_i's are compatible; and

(c) $\sum_{i\leftarrow\mu} \left\lfloor \cdot \right\rfloor_{j\leftarrow\eta} C_{ij} \approx \left\lfloor \cdot \right\rfloor_{j\leftarrow\eta} \sum_{i\leftarrow\mu} C_{ij}$, if, for each i, all C_{ij}'s have same type $\mathcal{X}\times\mathcal{Y}_j\to\mathbb{R}$.

Proof. a)

$$\sum_{i\leftarrow\mu\,j\leftarrow\eta} C_{ij} = \sum_{i\leftarrow\mu}\left(\sum_j \eta(j)C_{ij}\right) \qquad\qquad \text{(def. of hidden choice)}$$

$$= \sum_i \mu(i)\left(\sum_j \eta(j)C_{ij}\right) \qquad\qquad \text{(def. of hidden choice)}$$

$$= \sum_{i,j} \eta(i)\eta(j)C_{ij} \qquad\qquad \text{(reorganizing the sums)}$$

$$= \sum_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij} \qquad\qquad \text{(def. of hidden choice)}$$

b)

$$\left\lfloor \cdot \right\rfloor_{i\leftarrow\mu\,j\leftarrow\eta} C_{ij} = \left\lfloor \cdot \right\rfloor_{i\leftarrow\mu}\left(\diamondsuit_j C_{ij}\right) \qquad\qquad \text{(def. of visible choice)}$$

$$= \diamondsuit_i\,\mu(i)\left(\diamondsuit_j\eta(j)C_{ij}\right) \qquad\qquad \text{(def. of visible choice)}$$

$$\approx \diamondsuit_{ij}\,\mu(i)\eta(j)C_{ij} \qquad\qquad \text{(by Lemma 17)}$$

$$= \left\lfloor \cdot \right\rfloor_{\substack{i\leftarrow\mu\\j\leftarrow\eta}} C_{ij} \qquad\qquad \text{(def. of visible choice)}$$

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

c)

$$\sum_{i \leftarrow \mu} \lfloor \cdot \rfloor_{j \leftarrow \eta} C_{ij} = \sum_{i \leftarrow \mu} \left(\Diamond_j C_{ij} \right) \qquad \text{(def. of visible choice)}$$

$$= \sum_i \mu(i) \left(\Diamond_j \eta(j) C_{ij} \right) \qquad \text{(def. of hidden choice)}$$

$$\approx \Diamond_j \, \eta(j) \left(\sum_i \eta(i) C_{ij} \right) \qquad \text{(by Lemma 17)}$$

$$= \lfloor \cdot \rfloor_{j \leftarrow \eta} \sum_{i \leftarrow \mu} C_{ij} \qquad \text{(def. of operators)}$$

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

Theorem 7 *Let $\{C_i\}_{i \in \mathcal{I}}$ be a family of channels, and μ be a distribution on $\mathcal{I}$. Then, for every distribution π on $\mathcal{X}$, and every vulnerability $\mathbb{V}$:*

(a) posterior vulnerability is convex w.r.t. to hidden choice: $\mathbb{V}\left[\pi, \sum_{i \leftarrow \mu} C_i\right] \leq \sum_{i \in \mathcal{I}} \mu(i) \, \mathbb{V}\left[\pi, C_i\right]$ *if all C_i's have the same type.*

(b) posterior vulnerability is linear w.r.t. to visible choice: $\mathbb{V}\left[\pi, \lfloor \cdot \rfloor_{i \leftarrow \mu} C_i\right] = \sum_{i \in \mathcal{I}} \mu(i) \, \mathbb{V}\left[\pi, C_i\right]$ *if all C_i's are compatible.*

Proof. a) Let us call $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ the type of each channel C_i in the family $\{C_i\}$. Then:

$$\mathbb{V}\left[\pi, \sum_{i \leftarrow \mu} C_i\right] = \mathbb{V}\left[\pi, \sum_i \mu(i) C_i\right] \qquad \text{(def. of hidden choice)}$$

$$= \sum_{y \in \mathcal{Y}} p(y) \cdot \mathbb{V}\left[\frac{\pi(\cdot) \sum_i \mu(i) C_i(\cdot, y)}{p(y)}\right] \qquad \text{(def. of posterior } \mathbb{V})$$

$$= \sum_{y \in \mathcal{Y}} p(y) \cdot \mathbb{V}\left[\sum_i \mu(i) \frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right]$$

$$\leq \sum_{y \in \mathcal{Y}} p(y) \cdot \sum_i \mu(i) \mathbb{V}\left[\frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right] \qquad \text{(by convexity of } \mathbb{V})$$

$$= \sum_i \mu(i) \sum_{y \in \mathcal{Y}} p(y) \mathbb{V}\left[\frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right]$$

$$= \sum_i \mu(i) \mathbb{V}\left[\pi, C_i\right]$$

where $p(y) = \sum_{x \in \mathcal{X}} \pi(x) \sum_i \mu(i) C_i(x, y)$.

b) Let us call $\mathcal{X} \times \mathcal{Y}_i \to \mathbb{R}$ the type of each channel C_i in the family $\{C_i\}$. Then:

$$\mathbb{V}\left[\pi, \underset{i \leftarrow \mu}{\lfloor \cdot \rfloor} C_i\right] = \mathbb{V}\left[\pi, \diamondsuit_i \mu(i) C_i\right] \qquad \text{(def. of visible choice)}$$

$$= \sum_{y \in \mathcal{Y}} p(y) \cdot \mathbb{V}\left[\frac{\pi(\cdot) \diamondsuit_i \mu(i) C_i(\cdot, y)}{p(y)}\right] \qquad \text{(def. of posterior } \mathbb{V})$$

$$= \sum_{y \in \mathcal{Y}} p(y) \cdot \mathbb{V}\left[\diamondsuit_i \mu(i)\frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right]$$

$$= \sum_{y \in \mathcal{Y}} p(y) \cdot \sum_i \mu(i) \mathbb{V}\left[\frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right] \qquad \text{(see (*) below)}$$

$$= \sum_i \mu(i) \sum_{y \in \mathcal{Y}} p(y) \mathbb{V}\left[\frac{\pi(\cdot) C_i(\cdot, y)}{p(y)}\right]$$

$$= \sum_i \mu(i) \mathbb{V}\left[\pi, C_i\right]$$

where $p(y) = \sum_{x \in \mathcal{X}} \pi(x) \sum_i \mu(i) C_i(x, y)$, and step (*) holds because in the vulnerability of a concatenation of matrices every column will contribute to the vulnerability in proportion to its weight in the concatenation, and hence it is possible to break the vulnerability of a concatenated matrix as the weighted sum of the vulnerabilities of its sub-matrices.

Corollary 8 *Let $\{C_{ij}\}_{i \in \mathcal{I}, j \in \mathcal{J}}$ be a family of channels, all with domain $\mathcal{X}$ and with the same type, and let $\pi \in \mathbb{D}\mathcal{X}$, and $\mathbb{V}$ be any vulnerability. Define U : $\mathbb{D}\mathcal{I} \times \mathbb{D}\mathcal{J} \to \mathbb{R}$ as follows: $U(\mu, \eta) \stackrel{\text{def}}{=} \mathbb{V}\left[\pi, \sum_{i \leftarrow \mu} \lfloor \cdot \rfloor_{j \leftarrow \eta} C_{ij}\right]$. Then U is convex on μ and linear on η.*

Proof. To see that $U(\mu, \eta)$ is convex on μ, note that:

$$U(\mu, \eta) = \mathbb{V}\left[\pi, \underset{i \leftarrow \mu}{\textstyle\sum} \underset{j \leftarrow \eta}{\lfloor \cdot \rfloor} C_{ij}\right] \qquad \text{(by definition)}$$

$$\leq \sum_i \mu(i) \, \mathbb{V}\left[\pi, \underset{j \leftarrow \eta}{\lfloor \cdot \rfloor} C_{ij}\right] \qquad \text{(by Theorem 7)}$$

To see that $U(\mu, \eta)$ is linear on η, note that:

$$U(\mu, \eta) = \mathbb{V}\left[\pi, \underset{i \leftarrow \mu}{\textstyle\sum} \underset{j \leftarrow \eta}{\lfloor \cdot \rfloor} C_{ij}\right] \qquad \text{(by definition)}$$

$$= \mathbb{V}\left[\pi, \underset{j \leftarrow \eta}{\lfloor \cdot \rfloor} \underset{i \leftarrow \mu}{\textstyle\sum} C_{ij}\right] \qquad \text{(by Prop. 6)}$$

$$= \sum_j \eta(j) \, \mathbb{V}\left[\pi, \underset{i \leftarrow \mu}{\textstyle\sum} C_{ij}\right] \qquad \text{(by Theorem. 7)}$$

B Properties of binary versions of channel operators

In this section we derive some relevant properties of the binary versions of the hidden and visible choice operators. We start with results regarding each operator individually.

Proposition 18 (Properties of the binary hidden choice) *For any channels C_1 and C_2 of the same type, and any values $0 \le p, q \le 1$, the binary hidden choice operator satisfies the following properties:*

a) Idempotency: $C_1 \;_p\!\bigoplus C_1 = C_1$
b) Commutativity: $C_1 \;_p\!\bigoplus C_2 = C_2 \;_{\bar{p}}\!\bigoplus C_1$
c) Associativity:

$$C_1 \;_p\!\bigoplus (C_2 \;_q\!\bigoplus C_3) = (^1\!/_q \cdot C_1 \;_p\!\bigoplus C_2) \;_q\!\bigoplus \bar{p} \cdot C_3$$

if $q \ne 0$.
d) Absorption:

$$(C_1 \;_p\!\bigoplus C_2) \;_q\!\bigoplus (C_1 \;_r\!\bigoplus C_2) = C_1 \;_{(pq+\bar{q}r)}\!\bigoplus C_2.$$

Proof. We will prove each property separately.

a) *Idempotency*:

$$
\begin{aligned}
C_1 \;_p\!\bigoplus C_1 &= p \cdot C_1 + \bar{p} \cdot C_1 && \text{(def. of hidden choice)} \\
&= (p + \bar{p}) \cdot C_1 \\
&= C_1 && (p + \bar{p} = 1)
\end{aligned}
$$

b) *Commutativity*:

$$
\begin{aligned}
C_1 \;_p\!\bigoplus C_2 &= p \cdot C_1 + \bar{p} \cdot C_2 && \text{(def. of hidden choice)} \\
&= \bar{p} \cdot C_2 + p \cdot C_1 \\
&= C_2 \;_{\bar{p}}\!\bigoplus C_1 && \text{(def. of hidden choice)}
\end{aligned}
$$

c) *Associativity*:

$$
\begin{aligned}
&C_1 \;_p\!\bigoplus (C_2 \;_q\!\bigoplus C_3) \\
&= p \cdot C_1 + \bar{p}(q \cdot C_2 + \bar{q} \cdot C_3) && \text{(def. of hidden choice)} \\
&= p \cdot C_1 + \bar{p}q \cdot C_2 + \bar{p}\bar{q} \cdot C_3 \\
&= q(p(^1\!/_q \cdot C_1) + \bar{p} \cdot C_2) + \bar{q}(\bar{p} \cdot C_3) \\
&= (^1\!/_q \cdot C_1 \;_p\!\bigoplus C_2) \;_q\!\bigoplus \bar{p} \cdot C_3 && \text{(def. of hidden choice)}
\end{aligned}
$$

d) *Absorption*: First note that:

$$(C_1 \;_p\!\bigoplus C_2) \;_q\!\bigoplus (C_1 \;_r\!\bigoplus C_2)$$

$$= q(pC_1 + \bar{p}C_2) + \bar{q}(rC_1 + \bar{r}C_2) \qquad \text{(def. of hidden choice)}$$

$$= pqC_1 + \bar{p}qC_2 + \bar{q}rC_1 + \bar{q}\bar{r}C_2$$

$$= (pq + \bar{q}r)C_1 + (\bar{p}q + \bar{q}\bar{r})C_2$$

$$= C_1 \;_{(pq+\bar{q}r)}\!\bigoplus C_2 \qquad\qquad\qquad\qquad (*)$$

To complete the proof, note that in step (*) above, $pq + \bar{q}r$ and $\bar{p}q + \bar{q}\bar{r}$ form a valid binary probability distribution (they are both non-negative, and they add up to 1), then apply the definition of hidden choice.

Proposition 19 (Properties of binary visible choice) *For any compatible channels C_1 and C_2, and any values $0 \le p, q \le 1$, the visible choice operator satisfies the following properties:*

a) Idempotency*: $C_1 \;_p\!\sqcup C_1 \approx C_1$*
b) Commutativity*: $C_1 \;_p\!\sqcup C_2 \approx C_2 \;_{\bar{p}}\!\sqcup C_1$*
c) Associativity*: $C_1 \;_p\!\sqcup (C_2 \;_q\!\sqcup C_3) \approx (1/q \cdot C_1 \;_p\!\sqcup C_2) \;_q\!\sqcup \bar{p} \cdot C_3$ if $q \ne 0$.*

Proof. We will prove each property separately.

a) *Idempotency*: $C_1 \;_p\!\sqcup C_1 \approx C_1$, by immediate application of Lemma 17, since every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

b) *Commutativity*:

$$C_1 \;_p\!\sqcup C_2 = p \cdot C_1 \diamond \bar{p} \cdot C_2 \qquad \text{(def. of visible choice)}$$

$$\approx \bar{p} \cdot C_2 \diamond p \cdot C_1 \qquad\quad \text{(by Lemma 17)}$$

$$= C_2 \;_{\bar{p}}\!\sqcup C_1 \qquad\qquad \text{(def. of visible choice).}$$

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

c) *Associativity*:

$$C_1 \;_p\!\sqcup (C_2 \;_q\!\sqcup C_3) = p \cdot C_1 \diamond \bar{p}(q \cdot C_2 \diamond \bar{q} \cdot C_3) \qquad \text{(def. of visible choice)}$$

$$\approx p \cdot C_1 \diamond (\bar{p}q \cdot C_2 \diamond \bar{p}\bar{q} \cdot C_3) \qquad\quad \text{(by Lemma 17)}$$

$$= q(p(1/q \cdot C_1) \diamond \bar{p} \cdot C_2) \diamond \bar{q}(\bar{p} \cdot C_3)$$

$$= (1/q \cdot C_1 \;_p\!\sqcup C_2) \;_q\!\sqcup \bar{p} \cdot C_3 \qquad \text{(def. of visible choice)}$$

32

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.

Now we turn our attention to the interaction between hidden and visible choice operators.

A first result is that hidden choice does not distribute over visible choice. To see why, note that $C_1 \; _p\!\bigoplus (C_2 \; _q\!\sqcup C_3)$ and $(C_1 \; _p\!\bigoplus C_2) \; _q\!\sqcup (C_1 \; _p\!\bigoplus C_3)$ cannot be both defined: if the former is defined, then C_1 must have the same type as $C_2 \; _q\!\sqcup C_3$, whereas if the latter is defined, C_1 must have the same type as C_2, but $C_2 \; _q\!\sqcup C_3$ and C_2 do not have the same type (they have different output sets).

However, as the next result shows, visible choice distributes over hidden choice.

Proposition 20 (Distributivity of $_p\!\sqcup$ over $_q\!\bigoplus$) *Let C_1, C_2 and C_3 be compatible channels, and C_2 and C_3 have the same type. Then, for any values $0 \le p, q \le 1$:*

$$C_1 \; _p\!\sqcup (C_2 \; _q\!\bigoplus C_3) \approx (C_1 \; _p\!\sqcup C_2) \; _q\!\bigoplus (C_1 \; _p\!\sqcup C_3).$$

Proof.

$$
\begin{aligned}
& C_1 \; _p\!\sqcup (C_2 \; _q\!\bigoplus C_3) \\
=\; & (C_1 \; _q\!\bigoplus C_1) \; _p\!\sqcup (C_2 \; _q\!\bigoplus C_3) && \text{(idempotency of visible choice)} \\
=\; & p(q \cdot C_1 + \bar{q} \cdot C_1) \diamond \bar{p}(q \cdot C_2 + \bar{q} \cdot C_3) && \text{(def. of operators)} \\
=\; & (pq \cdot C_1 + p\bar{q} \cdot C_1) \diamond (\bar{p}q \cdot C_2 + \bar{p}\bar{q} \cdot C_3) && \\
\approx\; & (pq \cdot C_1 \diamond \bar{p}q \cdot C_2) + (p\bar{q} \cdot C_1 \diamond \bar{p}\bar{q} \cdot C_3) && \text{(by Lemma 17)} \\
=\; & q(p \cdot C_1 \diamond \bar{p} \cdot C_2) + \bar{q}(p \cdot C_1 \diamond \bar{p} \cdot C_3) && \\
=\; & q(C_1 \; _p\!\sqcup C_2) + \bar{q}(C_1 \; _p\!\sqcup C_3) && \text{(def. of visible choice)} \\
=\; & (C_1 \; _p\!\sqcup C_2) \; _q\!\bigoplus (C_1 \; _p\!\sqcup C_3) && \text{(def. of hidden choice)}
\end{aligned}
$$

In the above derivation, we can apply Lemma 17 because every column of the channel on each side of the equivalence can be written as a convex combination of the zero-column extension of the channel on the other side.