Measuring Information Leakage using Generalized Gain Functions

Mário S. Alvim*, Konstantinos Chatzikokolakis†, Catuscia Palamidessi†, and Geoffrey Smith‡
*University of Pennsylvania, USA, msalvim@sas.upenn.edu
†INRIA, CNRS and LIX, École Polytechnique, France, {kostas,catuscia}@lix.polytechnique.fr
‡Florida International University, USA, smithg@cis.fiu.edu

***Abstract*—This paper introduces g-leakage, a rich generalization of the min-entropy model of quantitative information flow. In g-leakage, the benefit that an adversary derives from a certain guess about a secret is specified using a *gain function* g. Gain functions allow a wide variety of operational scenarios to be modeled, including those where the adversary benefits from guessing a value *close* to the secret, guessing a *part* of the secret, guessing a *property* of the secret, or guessing the secret within some number of *tries*. We prove important properties of g-leakage, including bounds between min-capacity, g-capacity, and Shannon capacity. We also show a deep connection between a strong leakage ordering on two channels, C_1 and C_2, and the possibility of factoring C_1 into C_2C_3, for some C_3. Based on this connection, we propose a generalization of the *Lattice of Information* from deterministic to probabilistic channels.**

I. Introduction

A fundamental concern in computer security is to control *information flow*, whether to protect confidential information from being leaked, or to protect trusted information from being tainted. In view of the pragmatic difficulty of preventing undesirable flows completely, there is now much interest in theories that allow information flow to be *quantified*, so that "small" leaks can be tolerated. (See, for example, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].) For any leakage measure, a key challenge is to establish its *operational significance*, so that a certain amount of leakage implies a definite security guarantee.

Min-entropy leakage [10], [13] is a leakage measure based on the amount by which a channel increases the *vulnerability* of a secret to being guessed correctly in one try by an adversary.[1] This clear operational significance is a strength of min-entropy, but it also leads to questions about whether min-entropy leakage is relevant across the wide range of possible application scenarios. For instance, what if the adversary is allowed to make *multiple* guesses? Or what if the adversary could gain some benefit by guessing the secret only *partially* or *approximately*?

With respect to guessing the secret *partially*, we can note that we could in fact analyze a sub-channel that models the processing of whatever piece of a larger secret that we wish to consider. While this can be useful, it is clumsy to need to analyze multiple sub-channels of the same channel. Also, such an analysis is misleading in the case of a channel that poses little threat to any *particular* piece of the secret, yet is very likely to leak *some* piece of the secret. To illustrate, suppose that the secret is an array X containing 10-bit, uniformly-distributed passwords for 1000 users. Now consider the following probabilistic channel, which leaks *some* randomly-chosen user's password:

$$\begin{array}{l} u \xleftarrow{?} \{0..999\}; \\ Y = (u, X[u]); \end{array} \qquad \text{(Ex1)}$$

If we analyze the min-entropy leakage of (Ex1), we find that the prior vulnerability is 2^{-10000}, since there are 10000 completely unknown bits, while the posterior vulnerability is 2^{-9990}, since Y reveals 10 of the bits. The min-entropy leakage is the logarithm of the ratio of the posterior and prior vulnerabilities:

$$\mathcal{L} = \log \frac{2^{-9990}}{2^{-10000}} = \log 2^{10} = 10 \text{ bits.}$$

If we instead analyze the sub-channel focused on any particular user i's password, the prior vulnerability is 2^{-10}, and the posterior vulnerability is $0.001 \cdot 1 + 0.999 \cdot 2^{-10} \approx 0.00198$, since with probability 0.001, the adversary learns user i's password from Y, and with probability 0.999, he must still make a blind guess. Thus the min-entropy leakage of the sub-channel is $\log 2.023 \approx 1.016$ bits. Hence we see that the threat of (Ex1) is not well described by min-entropy leakage—the whole channel leaks just 10 bits out of 10000, and the sub-channel just 1.016 bits out of 10, even though *some* user's password is always leaked completely.

In light of the wide range of possible operational threat scenarios, there is growing appreciation that no single leakage measure is likely to be appropriate in all cases. For this reason, in this paper we introduce a generalization of min-entropy leakage, called *g-leakage*. The key idea is to generalize the notion of vulnerability to incorporate what we call a *gain function* g that models the benefit that the adversary gains by making a certain guess about the secret. If the adversary makes guess w when the secret's actual value is x, then $g(w, x)$ models the benefit that the adversary gains from this guess, ranging from 0 (if w has no value at all)

[1] The precise definition is reviewed in Section II.

This work has been partially supported by the European Union Seventh Framework Programme under grant agreement no. 295261 (MEALS), and by the Inria large scale initiative CAPPRIS: Collaborative Action for the Protection of Privacy Rights in the Information Society.

to 1 (if w is ideal). Given gain function g, g-vulnerability is defined as the maximum expected gain over all guesses.

As we will see in Section III, gain functions let us model a wide variety of scenarios, including those where the adversary benefits from guessing a value *close* to the secret, guessing a *part* of the secret, guessing a *property* of the secret, or guessing the secret within k *tries*. We can also model the case when there is a *penalty* for incorrect guesses. Thus g-leakage seems fruitful in addressing a great number of practical situations.

In addition to introducing the new concept of g-leakage, we also make significant technical contributions, principally in Sections V and VI.

In Section V, we establish important bounds on *capacity*, the maximum leakage over all prior distributions. We prove that min-capacity is an upper bound on g-capacity, for *any* gain function g—this means that a channel with small min-capacity is (in a sense) *safe* in every possible scenario. Moreover, we prove that min-capacity is also an upper bound on *Shannon capacity*, settling a conjecture in [14].

In Section VI, we consider the problem of *comparing* two channels, C_1 and C_2, asking whether on *every* prior the leakage of C_1 is less than or equal to that of C_2. Yasuoka and Terauchi [15] and Malacaria [16] recently explored this strong ordering in the case where C_1 and C_2 are *deterministic*, focusing on the fact that deterministic channels induce *partitions* on the space of secrets. They showed that the orderings produced by min-entropy leakage and Shannon leakage are the same and, moreover, they coincide with the *partition refinement* ordering $\sqsubseteq$ in the *Lattice of Information* [17]. Since partition refinement applies only to deterministic channels but leakage ordering makes sense for *any* channels, this equivalence suggests an approach to generalizing the Lattice of Information to probabilistic channels.

Our first result in Section VI identifies a promising generalization of partition refinement $\sqsubseteq$. We show that on deterministic channels, $C_1 \sqsubseteq C_2$ iff there exists a *factorization* of C_1 into a *cascade*: $C_1 = C_2C_3$, for some channel C_3. In this case we say that C_1 is *composition refined* by C_2, written $C_1 \sqsubseteq_\circ C_2$. In the most technically challenging part of our paper, we show a deep connection between $\sqsubseteq_\circ$ and leakage ordering. We show first in Theorem 6.2 that $C_1 \sqsubseteq_\circ C_2$ implies that C_1's g-leakage is less than or equal to C_2's, for *every* prior and *every* g; we denote this by $C_1 \leq_\mathcal{G} C_2$. We conjecture that the converse implication, $C_1 \leq_\mathcal{G} C_2$ implies $C_1 \sqsubseteq_\circ C_2$, is also true, but it turns out to be extremely subtle and we have been unable so far to prove it in full generality. We have proved it in important special cases (e.g. when C_2's columns are linearly independent) even limiting to a very restricted kind of gain function; we have also shown that the unproved case is inherently harder, in that much richer gain functions are required.

The rest of the paper is structured as follows. Sections II, III, and IV present preliminaries, define g-leakage, and show its basic properties. Sections V and VI present our results on capacity and on comparing channels. Finally, Sections VII and VIII discuss related work and conclude.

II. Preliminaries

In this section, we briefly recall the basic definitions of information-theoretic channels [18], vulnerability, and min-entropy leakage [10], introducing the non-standard notation that we will use.

A *channel* is a triple $(\mathcal{X}, \mathcal{Y}, C)$, where $\mathcal{X}$ and $\mathcal{Y}$ are finite sets (of secret input values and observable output values) and C is a *channel matrix*, an $|\mathcal{X}| \times |\mathcal{Y}|$ matrix whose entries are between 0 and 1 and whose rows each sum to 1; the intent is that $C[x, y]$ is the probability of getting output y when the input is x. Channel C is *deterministic* if each entry of C is either 0 or 1, implying that each row contains exactly one 1, which means that each input produces a unique output.

Given a *prior distribution* π on $\mathcal{X}$, we can define a *joint distribution* p on $\mathcal{X} \times \mathcal{Y}$ by $p(x, y) = \pi[x]C[x, y]$. This gives jointly distributed random variables X and Y with marginal probabilities $p(x) = \sum_y p(x, y)$, conditional probabilities $p(y|x) = \frac{p(x,y)}{p(x)}$ (if $p(x)$ is nonzero), and similarly $p(y)$ and $p(x|y)$. As shown in [19], p is the *unique* joint distribution that recovers π and C, in that $p(x) = \pi[x]$ and $p(y|x) = C[x, y]$ (if $p(x)$ is nonzero).

We now define vulnerability, introducing a new notation.[2]

Definition 2.1: Given prior π and channel C, the *prior vulnerability* is given by

$$V(\pi) = \max_{x \in \mathcal{X}} \pi[x],$$

and the *posterior vulnerability* is given by

$$V(\pi, C) = \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} \pi[x]C[x, y].$$

We assume in this paper that the prior distribution π and channel C are known to the adversary $\mathcal{A}$. Then $V(\pi)$ is the prior probability that $\mathcal{A}$ could guess the value of X correctly in one try. To understand posterior vulnerability, note that

$$\begin{aligned} V(\pi, C) &= \textstyle\sum_y \max_x p(x, y) \\ &= \textstyle\sum_y p(y) \max_x p(x|y) \\ &= \textstyle\sum_y p(y) V(p_{X|y}) \end{aligned}$$

making it the (weighted) average of the vulnerabilities of the posterior distributions $p_{X|y}$.

We convert from vulnerability to *min-entropy* by taking the negative logarithm (to base 2):

Definition 2.2:

$$\begin{aligned} H_\infty(\pi) &= -\log V(\pi) \\ H_\infty(\pi, C) &= -\log V(\pi, C). \end{aligned}$$

[2]We deviate from the standard notation $V(X)$ and $V(X|Y)$ used in [14] and elsewhere, because we wish to express explicitly the dependence on X's prior distribution.

Note that vulnerability is a *probability*, while min-entropy is a measure of *bits of uncertainty*.

Next we define *min-entropy leakage* $\mathcal{L}(\pi, C)$ and *min-capacity* $\mathcal{ML}(C)$:

Definition 2.3:

$$\mathcal{L}(\pi, C) = H_\infty(\pi) - H_\infty(\pi, C) = \log \frac{V(\pi, C)}{V(\pi)}$$
$$\mathcal{ML}(C) = \sup_\pi \mathcal{L}(\pi, C).$$

The min-entropy leakage $\mathcal{L}(\pi, C)$ is the amount by which channel C decreases the uncertainty about the secret; equivalently, it is the logarithm of the factor by which C increases the vulnerability. The min-capacity $\mathcal{ML}(C)$ is the maximum min-entropy leakage over all priors π; it can be seen as the worst-case leakage of C.

Finally, we recall [13] that the min-capacity of C is easy to calculate, as it is simply the logarithm of the sum of the column maximums of C:

Theorem 2.1: $\mathcal{ML}(C) = \log \sum_y \max_x C[x, y]$, and it is realized on a uniform prior π.

III. Gain functions, g-vulnerability, and g-leakage

We now develop the theory of gain functions and the leakage measures that they give.

Implicit in the definition of prior and posterior vulnerability $V(\pi)$ and $V(\pi, C)$ is the assumption that the adversary benefits only by guessing the *entire* secret *exactly*. But, as motivated in Section I, there are certainly situations where this assumption is not appropriate. This leads us to introduce what we call *gain functions* as abstract models of the particular operational scenario. The idea is that in any such scenario, there will be some set of *guesses* that the adversary could make about the secret, and for any guess w and secret value x, there will be some *gain* that the adversary gets by choosing w when the secret's actual value is x. A gain function g will specify this gain as $g(w, x)$, using scores that range from 0 to 1.

A first question, however, is what should be the set of allowable guesses. One might be tempted to assume that this should just be $\mathcal{X}$, the set of possible values of the secret. But given our desire to model scenarios where the adversary gains by guessing a *piece* of the secret, or a value *close* to the secret, or some *property* of the secret, we instead let a gain function use an arbitrary set $\mathcal{W}$ of allowable guesses.

Definition 3.1: Given a set $\mathcal{X}$ of possible secrets and a finite, nonempty set $\mathcal{W}$ of allowable guesses, a *gain function* is a function $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$.

Sometimes it is convenient to represent a gain function g as a $|\mathcal{W}| \times |\mathcal{X}|$ matrix G, where $G[w, x] = g(w, x)$; the rows of G correspond to guesses and the columns to secrets.

We now adapt the definition of vulnerability to take account of the gain function:

Definition 3.2: Given gain function g and prior π, the *prior g-vulnerability* is

$$V_g(\pi) = \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi[x] g(w, x).$$

The idea is that adversary $\mathcal{A}$ should make a guess w that maximizes the expected gain; we therefore take the weighted average of $g(w, x)$, for every possible value x of X.[3]

Definition 3.3: Given gain function g, prior π, and channel C, the *posterior g-vulnerability* is

$$\begin{aligned} V_g(\pi, C) &= \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi[x] C[x, y] g(w, x) \\ &= \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} p(x, y) g(w, x) \\ &= \sum_{y \in \mathcal{Y}} p(y) V_g(p_{X|y}) \end{aligned}$$

Now we define g-entropy, g-leakage, and g-capacity in exactly the same way as in Section II:

Definition 3.4:

$$\begin{aligned} H_g(\pi) &= -\log V_g(\pi) \\ H_g(\pi, C) &= -\log V_g(\pi, C) \\ \mathcal{L}_g(\pi, C) &= H_g(\pi) - H_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)} \\ \mathcal{ML}_g(C) &= \sup_\pi \mathcal{L}_g(\pi, C) \end{aligned}$$

In Section IV, we will explore the mathematical properties of g-leakage. But first we present a number of example gain functions that illustrate the usefulness of g-leakage.

A. The identity gain function

One obvious (and often appropriate) gain function is the one that says that a correct guess is worth 1 and an incorrect guess is worth 0:

Definition 3.5: The *identity gain function* $g_{id} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ is given by

$$g_{id}(w, x) = \begin{cases} 1, & \text{if } w = x, \\ 0, & \text{if } w \neq x. \end{cases}$$

Note that for g_{id} we assume that $\mathcal{W} = \mathcal{X}$, since there is no gain to be had from a guess outside of $\mathcal{X}$. In terms of representing a gain function as a matrix, g_{id} corresponds to the identity matrix $I_{|\mathcal{X}|}$. Also notice that g_{id} is the *Kronecker delta*, since $g_{id}(w, x) = \delta_{wx}$.

Now we can show that g-vulnerability is a generalization of ordinary vulnerability:

Proposition 3.1: Vulnerability under g_{id} coincides with vulnerability:

$$V_{g_{id}}(\pi) = V(\pi).$$

[3] We remark that our assumption that gain values are between 0 and 1 is unimportant. Allowing g to return a value in $[0, a]$, for some constant a, just scales all g-vulnerabilities by a factor of a and therefore has no effect on g-leakage.

Proof: Note for any w, $\sum_x \pi[x] g_{id}(w,x) = \pi[w]$. So $V_{g_{id}}(\pi) = \max_w \pi[w] = V(\pi)$. ∎

This means that g_{id}-leakage coincides with min-entropy leakage.

B. Gain functions induced from metrics or other distance functions

Exploring other gain functions, one quite natural kind of structure that $\mathcal{X}$ may exhibit is a notion of *distance* between secrets. That is, there may be a *metric* d on $\mathcal{X}$, which is a function

$$d : \mathcal{X} \times \mathcal{X} \to [0, \infty)$$

satisfying the properties

- (identity of indiscernibles) $d(x_1, x_2) = 0$ iff $x_1 = x_2$,
- (symmetry) $d(x_1, x_2) = d(x_2, x_1)$, and
- (triangle inequality) $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$.

Given a metric d, we can first form a *normalized* metric $\bar{d}$ by dividing all distances by the maximum value of d, and then we can define a gain function g_d by

$$g_d(w, x) = 1 - \bar{d}(w, x).$$

(Note that here we are taking $\mathcal{W} = \mathcal{X}$.) In this case we say that g_d is the gain function *induced* from metric d.[4]

Metrics induce a large class of gain functions—note in particular that the identity gain function is induced by the *discrete metric*, which assigns distance 1 to any two distinct values. However, there are several reasons why it is useful to allow more generality.

For one thing, it may make sense to generalize to a metric on a set $\mathcal{W}$ that is a *superset* of $\mathcal{X}$. To see why, suppose that the space of secrets is the set of corner points of a unit square: $\mathcal{X} = \{(0,0), (0,1), (1,0), (1,1)\}$. Suppose that we use the gain function $g(w, x) = 1 - \bar{d}(w, x)$, where the metric $\bar{d}$ is the normalized Euclidean distance:

$$\bar{d}((x_1, y_1), (x_2, y_2)) = \sqrt{\frac{(x_1 - x_2)^2 + (y_1 - y_2)^2}{2}}$$

Now,

$$V_{g_d}(\pi) = \max_w \sum_x \pi[x](1 - \bar{d}(w, x))$$

and if π is uniform, then it is easy to see that any of the four corner points are equally-good guesses, giving

$$V_{g_d}(\pi) = \frac{1}{4}(1 + 2(1 - \frac{1}{\sqrt{2}}) + 0) \approx 0.3964$$

But the adversary could actually do better by guessing $(\frac{1}{2}, \frac{1}{2})$, a value that is *not* in $\mathcal{X}$, since that guess has normalized distance $\frac{1}{2}$ from each of the four corner points, giving $V_{g_d}(\pi) = \frac{1}{2}$, which is larger than the previous vulnerability.

[4]However, it is also rather natural to define a gain function from a metric by $g(w, x) = e^{-d(w,x)}$; note that here we would actually want d to be an extended metric, so that a gain of 0 becomes possible.

Moreover, the assumption of *symmetry* is sometimes inappropriate. Suppose that the secret is the time (rounded to the nearest minute) that the last RER B train will depart from Lozère back to Paris.[5] The adversary (i.e. the weary traveler) wants to guess this time as accurately as possible, but note that guessing 23:44 when the actual time is 23:47 is completely different from guessing 23:47 when the actual time is 23:44! If we normalize so that a wait of an hour or more is considered intolerable, then we would want the distance function

$$d(w, x) = \begin{cases} \frac{x-w}{60} & \text{if } x - 60 < w \leq x \\ 1 & \text{otherwise} \end{cases}$$

and the gain function

$$g(w, x) = 1 - d(w, x).$$

But $d(w, x)$ is not a metric, because it is not symmetric.[6]

C. Binary gain functions

The family of gain functions that return either 0 or 1 (and no values in between) are of particular interest, since we can characterize them concretely. For given such a gain function, each guess exactly corresponds to the *subset* of $\mathcal{X}$ for which that guess gives gain 1. (Moreover we can assume without loss of generality that no two guesses correspond to the *same* subset of $\mathcal{X}$, since such guesses may as well be merged into one.) Hence we can use the subsets *themselves* as the guesses, leading to the following definition:

Definition 3.6: Given $\mathcal{W} \subseteq 2^{\mathcal{X}}$, $\mathcal{W}$ nonempty, the *binary gain function* $g_{\mathcal{W}}$ is given by

$$g_{\mathcal{W}}(W, x) = \begin{cases} 1, & \text{if } x \in W \\ 0, & \text{otherwise.} \end{cases}$$

Now we can identify a number of interesting gain functions by considering different choices of $\mathcal{W}$.

1) 2-block gain functions: If $\mathcal{W} = \{W, \mathcal{X} \setminus W\}$ then we can see W as a *property* that the secret X might or might not satisfy, and $g_{\mathcal{W}}$ is the gain function corresponding to an adversary that just wants to decide whether or not X satisfies that property.

Such 2-block gain functions are reminiscent of the cryptographic notion of *indistinguishability*, which demands that from a ciphertext an adversary should not be able to decide any *property* of the corresponding plaintext.

2) Partition gain functions: More generally, $\mathcal{W}$ could be any *partition* of $\mathcal{X}$ into one or more disjoint *blocks*, where the adversary just wants to determine which block the secret belongs to.

This is equivalent to saying that $\mathcal{W} = \mathcal{X}/\sim$, where $\sim$ is an equivalence relation on $\mathcal{X}$.

[5]It is well known that RATP uses sophisticated techniques, such as the *droit de retrait*, to make this time as unpredictable as possible.

[6]Such a function is sometimes called a *quasimetric*.

There are two extremes. If $\sim$ is the identity relation, then the elements of $\mathcal{W}$ are all singletons, which means that $g_\sim = g_{id}$. And if $\sim$ is the universal relation, then $\mathcal{W}$ consists of a single block, $\mathcal{W} = \{\mathcal{X}\}$, and $g_\sim = g_\smile$, the "happy" gain function such that $g_\smile(\mathcal{X}, x) = 1$, for every x.

3) The k-tries gain function: Interestingly, we can subsume the theory of *k-tries vulnerability*, in which the adversary is allowed to make k guesses, rather than just 1. For if we define

$$\mathcal{W}_k = \{W \in 2^{\mathcal{X}} \mid |W| \leq k\}$$

then $V_{g_{\mathcal{W}_k}}(\pi)$ is exactly the probability that the adversary could guess the value of X correctly within k tries. This gain function is particularly appropriate for a login program that allows the user only k tries before locking him out.

Notice that $g_{\mathcal{W}_k}$ is not a partition gain function for $k > 1$, since its blocks overlap.

4) General binary gain functions: In general, $\mathcal{W}$ can be an arbitrary nonempty subset of $2^{\mathcal{X}}$. In this case, each element of $\mathcal{W}$ can be understood as a property that X might satisfy, and $g_{\mathcal{W}}$ is the gain function of an adversary that wants to guess *any* of those properties that X satisfies.

Given an arbitrary gain function g, we can define the *complement* gain function g^c by $g^c(w, x) = 1 - g(w, x)$. It is interesting to notice that if $\mathcal{W} \subseteq 2^{\mathcal{X}}$, then $g^c_{\mathcal{W}}$ is essentially the same[7] as $g_{\mathcal{W}'}$, where $\mathcal{W}' = \{W^c \mid W \in \mathcal{W}\}$. For example, for g^c_{id} we have that $\mathcal{W}'$ is the set of complements of singletons. This means that g^c_{id} is the gain function of an adversary that just wants to guess some value *different* from the actual value of the secret; in the context of anonymity, this corresponds to wanting to guess an *innocent* user.

D. A gain function and the g-leakage of the password database example

We now show how we can craft a gain function g appropriate for the password database example (Ex1) from Section I. The intuition that g will implement is that the adversary $\mathcal{A}$ simply wants to guess *some* user's password, with no preference as to whose it is. So we will take

$$\mathcal{W} = \{(u, x) \mid 0 \leq u \leq 999 \text{ and } 0 \leq x \leq 1023\}$$

and define

$$g((u, x), X) = \begin{cases} 1, & \text{if } X[u] = x \\ 0, & \text{otherwise.} \end{cases}$$

How does our analysis of channel (Ex1) change when we use gain function g (rather than g_{id}, used implicitly in min-entropy leakage)?

We first see that the prior vulnerability is vastly higher than before. Under the uniform prior π, it is easy to see that the expected gain of every element (u, x) of $\mathcal{W}$ is 2^{-10}, since for every u, $X[u]$ is uniformly distributed on $[0..1023]$. Hence $V_g(\pi) = 2^{-10}$, compared with $V_{g_{id}}(\pi) = 2^{-10000}$. These values match our intuition that under g the adversary just needs to guess any single 10-bit password; under g_{id}, in contrast, the adversary needs to guess 1000 such passwords.[8]

Turning now to the posterior g-vulnerability, we have $V_g(\pi, \text{Ex1}) = 1$, since given $Y = (u, X[u])$, $\mathcal{A}$ can guess $(u, X[u])$ and be sure of getting gain 1.

Hence we have

$$\mathcal{L}_g(\pi, \text{Ex1}) = \log \frac{V_g(\pi, \text{Ex1})}{V_g(\pi)} = \log \frac{1}{2^{-10}} = 10 \text{ bits.}$$

Curiously, the g-leakage is the *same* as the standard min-entropy leakage, namely 10 bits. But the significance of leaking 10 bits is completely different under g and under g_{id}. If we convert from vulnerability to entropy (see Definition 3.4), we see that $H_g(\pi) = 10$, while $H_{g_{id}}(\pi) = 10000$. In other words, channel (Ex1) leaks 10 bits out of 10 under g, as compared with 10 bits out of 10000 under g_{id}. In conclusion, g-leakage under gain function g allows us to model accurately the threat to a *structured* secret (like a password database), composed of "pieces" that are individually valuable; as we saw in Section I, such threats are not well modeled using min-entropy leakage.

Finally, it is interesting to consider a variant of channel (Ex1) that selects 10 random users and leaks just the *last* bit of each of their passwords. Because the variant still reveals 10 bits to the adversary, the min-entropy leakage remains 10 bits. But the g-leakage is now only 1 bit: the posterior g-vulnerability is now 2^{-9} since (at least) 9 bits of each user's password remain unknown. In other words, gain function g captures the *structure* of the password database, where certain sets of bits are worth more than others.

E. Gain functions that distinguish two channels

We conclude this section by revisiting two example channels from [10]:

$$\textbf{if } (X \,\%\, 8 == 0)\ Y = X;\ \textbf{else } Y = 1; \qquad \text{(Ex2)}$$

$$Z = X \mid 07; \qquad \text{(Ex3)}$$

Assuming that X is a uniformly-distributed 64-bit unsigned integer, both channels have min-entropy leakage of 61.000 bits, even though they present quite different threats: (Ex2) leaks all of X one-eighth of the time and leaks almost nothing seven-eights of the time, while (Ex3) always leaks all but the last three bits of X.

[7] They don't actually have the same set of guesses.

[8] It is interesting to notice that we would get a much bigger prior vulnerability if we used a gain function g' that allows $\mathcal{A}$ to guess just a password x, without specifying *whose* it is, and which gives a gain of 1 if x is correct for *any* of the users. For then we would have

$$V_{g'}(\pi) = 1 - \left(\frac{1023}{1024}\right)^{1000} \approx 0.6236$$

But g' is not such a reasonable gain function, since really a password is valuable only with respect to a particular user.

We now show how these two channels can be distinguished by gain functions that model different attack scenarios, showing that each channel can sometimes be worse than the other.

Consider first the 3-try gain function from Section III-C3. Because $g_{\mathcal{W}_3}$ gives a gain of 1 if the adversary can guess X within 3 tries, the prior vulnerability is tripled:

$$V_{g_{\mathcal{W}_3}}(\pi) = 3 \cdot 2^{-64}.$$

Allowing 3 tries also triples the posterior vulnerability for (Ex3):

$$V_{g_{\mathcal{W}_3}}(\pi, \text{Ex3}) = \frac{3}{8},$$

so $\mathcal{L}_{g_{\mathcal{W}_3}}(\pi, \text{Ex3})$ remains 61 bits. But allowing 3 tries hardly helps (Ex2):

$$V_{g_{\mathcal{W}_3}}(\pi, \text{Ex2}) = \frac{1}{8} \cdot 1 + \frac{7}{8} \cdot 3 \cdot 2^{-64} \approx \frac{1}{8},$$

so the $g_{\mathcal{W}_3}$-leakage of (Ex2) becomes smaller:

$$\mathcal{L}_{g_{\mathcal{W}_3}}(\pi, \text{Ex2}) \approx \log \frac{2^{-3}}{3 \cdot 2^{-64}} \approx 59.4$$

Thus (Ex3) is worse than (Ex2) under $g_{\mathcal{W}_3}$.

But now suppose that making a wrong guess triggers a penalty (say, opening a trap door to a pit of tigers). This scenario can be modeled through a gain function $g_{\textit{tiger}}$ using $\mathcal{W} = \mathcal{X} \cup \{\bot\}$, where the special value $\bot$ is used to opt not to make a guess:

$$g_{\textit{tiger}}(w, x) = \begin{cases} 1, & \text{if } w = x \\ \frac{1}{2}, & \text{if } w = \bot \\ 0, & \text{otherwise.} \end{cases}$$

Now we get

$$V_{g_{\textit{tiger}}}(\pi) = V_{g_{\textit{tiger}}}(\pi, \text{Ex3}) = \frac{1}{2}$$

since $\mathcal{A}$'s best choice is $\bot$ both *a priori* and also given Z, since knowing Z gives only a $\frac{1}{8}$ probability of guessing X, and $\frac{1}{8} < \frac{1}{2}$. In contrast,

$$V_{g_{\textit{tiger}}}(\pi, \text{Ex2}) = \frac{1}{8} \cdot 1 + \frac{7}{8} \cdot \frac{1}{2} = \frac{9}{16}.$$

Hence the $g_{\textit{tiger}}$-leakage of (Ex3) is 0, while that of (Ex2) is $\log 1.125 \approx 0.17$, showing that (Ex2) is worse than (Ex3) under $g_{\textit{tiger}}$.

Having shown some examples of the usefulness of gain functions, we turn in the next section to a study of the mathematical properties of g-leakage.

IV. Mathematical properties of g-vulnerability and g-leakage

We now establish some mathematical properties of g-leakage. First, because gain functions return values in $[0, 1]$, it is easy to see that g-vulnerabilities are also in $[0, 1]$. Also, prior vulnerability cannot exceed posterior vulnerability:

Theorem 4.1: For any π and C, $V_g(\pi, C) \geq V_g(\pi)$, which implies that $\mathcal{L}_g(\pi, C) \geq 0$.

Proof:

$$\begin{aligned} V_g(\pi, C) &= \textstyle\sum_y \max_w \sum_x p(x, y) g(w, x) \\ &\geq \max_w \textstyle\sum_y \sum_x p(x, y) g(w, x) \\ &= \max_w \textstyle\sum_x \sum_y p(x, y) g(w, x) \\ &= \max_w \textstyle\sum_x p(x) g(w, x) \\ &= V_g(\pi) \end{aligned}$$

Hence $\mathcal{L}_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)} \geq \log 1 \geq 0$. ∎

A mathematical issue, however, is that we could have $V_g(\pi) = 0$, since we could have $g(w, x) = 0$ for every w and every x with $\pi[x] > 0$. But in such a case it is easy to see that we must also have $V_g(\pi, C) = 0$, for any C, so we could simply define the g-leakage to be 0 in that case. In this paper, however, we will instead rule out such "pathological" gain functions, by insisting that for every secret x there exist some guess w such that $g(w, x) > 0$.

A. Comparing g-leakage and min-entropy leakage

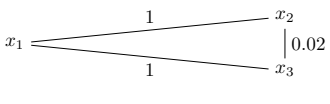
How can the g-leakage and min-entropy leakage of a channel compare? We can first observe that g-leakage can be arbitrarily smaller than min-entropy leakage. A trivial way that this can happen is to use the "happy" gain function $g_\smile$ from Section III-C2. With this gain function, the prior vulnerability is always 1, so the $g_\smile$-leakage is always 0, no matter the channel.

More interestingly, consider the channel

	y_1	y_2
x_1	$\frac{1}{2}$	$\frac{1}{2}$
x_2	1	0
x_3	0	1

(Ex4)

If we assume a uniform prior $\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, then the min-entropy leakage is $\log 2 = 1$. Now suppose we use the following metric-induced gain function g_d:



g_d	x_1	x_2	x_3
x_1	1	0	0
x_2	0	1	0.98
x_3	0	0.98	1

We find that

$$\begin{aligned} V_{g_d}(\pi) &= \frac{1}{3} \max\{1 + 0 + 0, 0 + 1 + 0.98, 0 + 0.98 + 1\} \\ &= 0.66 \end{aligned}$$

reflecting the fact that x_2 and x_3 behave almost like a single secret value. Turning now to $V_{g_d}(\pi, \text{Ex4})$, we calculate that the posterior distribution $p_{X|y_1} = (\frac{1}{3}, \frac{2}{3}, 0)$. Hence

$$V_{g_d}(p_{X|y_1}) = \max \left\{ \begin{array}{l} \frac{1}{3} \cdot 1 + \frac{2}{3} \cdot 0 + 0 \cdot 0, \\ \frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 1 + 0 \cdot 0.98, \\ \frac{1}{3} \cdot 0 + \frac{2}{3} \cdot 0.98 + 0 \cdot 1 \end{array} \right\} = \frac{2}{3}$$

Similarly, we can calculate that $V_{g_d}(p_{X|y_2}) = \frac{2}{3}$. So, since $p_Y = (\frac{1}{2}, \frac{1}{2})$, we get $V_{g_d}(\pi, \text{Ex4}) = \frac{2}{3}$, giving

$$\mathcal{L}_{g_d}(\pi, \text{Ex4}) = \log \frac{\frac{2}{3}}{0.66} \approx \log 1.0101 \approx 0.01450$$

Here the min-entropy leakage is about 70 times the g_d-leakage. Intuitively, (Ex4) lets us distinguish between x_2 and x_3, but since these are so close together under d, this hardly increases the g_d-vulnerability.

We may wonder if g-leakage can ever exceed min-entropy leakage. Indeed it can, as the following example shows:

	y_1	y_2
x_1	0.6	0.4
x_2	0	1
x_3	0	1

(Ex5)

Under prior $\pi = (0.6, 0.2, 0.2)$, the min-entropy leakage is 0, because $\mathcal{A}$'s best guess is unaffected by Y; indeed $p_{X|y_1} = (1, 0, 0)$ and $p_{X|y_2} = (0.375, 0.3125, 0.3125)$, so the best guess is always x_1.

In contrast, we find that the g_d-leakage is positive, where g_d is the same as in (Ex4) above. First, $V_{g_d}(\pi) = 0.6$ because the combined probabilities of x_2 and x_3 are only 0.4. Next we find that $V_{g_d}(p_{X|y_1}) = 1$, $V_{g_d}(p_{X|y_2}) = 0.61875$, and $p_Y = (0.36, 0.64)$, so

$$V_{g_d}(\pi, \text{Ex5}) = 0.36 \cdot 1 + 0.64 \cdot 0.61875 = 0.756,$$

giving g_d-leakage of $\log \frac{0.756}{0.6} = \log 1.26 \approx 0.3334$.

B. On g-leakage of 0

We can characterize precisely when g-leakage is 0. As in the case of min-entropy leakage, we find that a channel's g-leakage is 0 iff the adversary's best guess about the secret is not affected by channel's output. Before stating this property formally, we first introduce some notion, given prior π and channel C:

$$E_g(w) = \sum_x \pi[x] g(w, x)$$
$$E_g(w, y) = \sum_x \pi[x] C[x, y] g(w, x)$$

$E_g(w)$ is the expected gain of guess w *a priori*, while $E_g(w, y)$ is the expected gain for w given output y. These satisfy the following properties:

$$V_g(\pi) = \max_w E_g(w)$$
$$V_g(\pi, C) = \sum_y \max_w E_g(w, y)$$
$$E_g(w) = \sum_y E_g(w, y)$$

and, in the case of min-entropy, they reduce to

$$E_{g_{id}}(x) = \pi[x]$$
$$E_{g_{id}}(x, y) = \pi[x] C[x, y].$$

Theorem 4.2: Given channel C, gain function g, and prior π, the g-leakage is 0 iff there exists a guess $w^\star$ that gives the best expected gain for all outputs:

$$\forall w, y : E_g(w^\star, y) \geq E_g(w, y).$$

If such a guess exists then it also gives the best prior gain:

$$\forall w : E_g(w^\star) \geq E_g(w).$$

Proof: Assuming that such a guess $w^\star$ exists, we first show that it gives the best prior gain. We have

$$V_g(\pi, C) = \textstyle\sum_y \max_w E_g(w, y) = \sum_y E_g(w^\star, y) = E_g(w^\star).$$

Since $E_g(w^\star) = \max_w E_g(w) = V_g(\pi)$, the g-leakage is 0.

Now assume that such a guess does not exist, and let $w^\star$ be a guess giving the best prior gain. Then there exist w', y such that $E_g(w', y) > E_g(w^\star, y)$. Now we have

$$V_g(\pi, C) = \textstyle\sum_y \max_w E_g(w, y) > \sum_y E_g(w^\star, y) = V_g(\pi)$$

which implies that the g-leakage is greater than 0. ■

C. On g-vulnerability as a linear optimization problem

It is sometimes useful to think of g-vulnerability as the solution to an optimization problem, where the adversary assigns guesses to channel outputs, with the goal of maximizing his gain. Let C be a channel from $\mathcal{X}$ to $\mathcal{Y}$ and let $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$ be a gain function. A function $s : \mathcal{Y} \to \mathcal{W}$ is called a *strategy*. Intuitively, $s(y)$ is the attacker's guess when he sees the output y. It is also possible to write s as a deterministic channel S from $\mathcal{Y}$ to $\mathcal{W}$ (i.e. $S[y, w] = 1$ iff $s(y) = w$).

Now consider the definition of posterior g-vulnerability (Def. 3.3) and let s be an *optimal* strategy, i.e. such that $s(y)$ is a w giving the $\max_w$ for each y in the definition. Viewing s, g as matrices S, G, we have $g(s(y), x) = SG[y, x]$, so we can write $V_g(\pi, C)$ as:

$$\begin{aligned} V_g(\pi, C) &= \textstyle\sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi[x] C[x, y] g(w, x) \\ &= \textstyle\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \pi[x] C[x, y] SG[y, x] \\ &= tr(D_\pi CSG) \end{aligned}$$

where D_π is the matrix having π in the diagonal and 0 elsewhere (note that $D_\pi C[x, y] = \pi[x] C[x, y]$), and $tr(A)$ denotes the trace of A, i.e. the sum of its diagonal elements.

So, seeing now the elements of S as variables, $V_g(\pi, C)$ is the solution to the linear optimization problem (for fixed π, C, G):

maximize $tr(D_\pi CSG)$
subject to S being a channel:

$$S[y, w] \geq 0 \quad \forall y \in \mathcal{Y}, w \in \mathcal{W}$$
$$\textstyle\sum_w S[y, w] = 1 \quad \forall y \in \mathcal{Y}$$

Note that in the optimization problem S is not necessarily deterministic, meaning that the choice of guess can be made probabilistically. However, from linear programming theory

we know that there is always an optimal solution on a *vertex*, which here corresponds to S being deterministic.

We should clarify that this linear programming formulation is *not* particularly useful if we just wish to compute $V_g(\pi, C)$, since we can directly evaluate the formula in Definition 3.3. The value of these linear programming insights will be demonstrated later, especially in the proof of Theorem 6.2 and in the challenging algorithmic problems considered in Section VI-F.

V. Results on channel capacity

In this section, we compare min-capacity with g-capacity and Shannon capacity, proving some important relationships.

A. Min-capacity and g-capacity

When we compare the g-leakage and min-entropy leakage of a channel under some particular prior π, we saw in Section IV that each may exceed the other greatly.

Remarkably, when we turn our attention to capacity, we find that a definite order must hold: min-capacity is an upper bound on g-capacity, for *every* gain function g.

Theorem 5.1 ("Miracle"): For any channel C and gain function g, $\mathcal{ML}_g(C) \leq \mathcal{ML}(C)$.

Proof: For any C, g, and π, we have

$$\begin{aligned} V_g(\pi, C) &= \textstyle\sum_y \max_w \sum_x C[x,y]\pi[x]g(w,x) \\ &\leq \textstyle\sum_y \max_w \sum_x \big(\max_x C[x,y]\big)\pi[x]g(w,x) \\ &= \big(\textstyle\sum_y \max_x C[x,y]\big)\big(\max_w \sum_x \pi[x]g(w,x)\big) \\ &= 2^{\mathcal{ML}(C)}\, V_g(\pi), \end{aligned}$$

using Theorem 2.1 in the last step. Hence

$$\mathcal{L}_g(\pi, C) = \log \frac{V_g(\pi, C)}{V_g(\pi)} \leq \log 2^{\mathcal{ML}(C)} = \mathcal{ML}(C),$$

which implies that $\mathcal{ML}_g(C) \leq \mathcal{ML}(C)$. ■

This gives a nice corollary about k-tries leakage:

Corollary 5.2: The capacity of a channel C under the k-tries scenario is no greater than its capacity under the 1-try scenario (i.e. its min-capacity).

Proof: Follows from Theorem 5.1 and the fact that the k-tries scenario is given by the $g_{\mathcal{W}_k}$-leakage, where $g_{\mathcal{W}_k}$ is the gain function from Section III-C3. ■

So, while allowing more than one guess obviously increases both the prior and posterior vulnerabilities, it cannot increase the capacity.

B. Min-capacity and Shannon capacity

The significance of min-capacity as an upper bound on leakage is further attested by another result that we have achieved—we have been able to show that min-capacity is also an upper bound on *Shannon capacity* (i.e. the maximum mutual information $I(X;Y)$ over all priors π [18]), confirming the conjecture made in [14]:

Theorem 5.3: For any channel C, C's min-capacity is at least as great as its Shannon capacity.

Proof: Our argument makes crucial use of *Jensen's inequality*, which says that if f is a concave ($\frown$) function, $\lambda_1, \lambda_2, \ldots, \lambda_n$ are convex coefficients, and $x_1, x_2, \ldots, x_n$ are arbitrary, then

$$\textstyle\sum_i \lambda_i f(x_i) \leq f(\sum_i \lambda_i x_i).$$

Let prior π on $\mathcal{X}$ be arbitrary. We reason as follows:

$$\begin{aligned} & I(X;Y) \\ &= H(Y) - H(Y|X) \\ &= -\textstyle\sum_y p(y)\log p(y) + \sum_x p(x) \sum_y p(y|x)\log p(y|x) \\ &= \textstyle\sum_{x,y} p(x,y)\log \frac{p(y|x)}{p(y)} \\ &\leq \text{[by Jensen's inequality and the concavity of } \log] \\ &\quad \log \textstyle\sum_{x,y} p(x,y)\frac{p(y|x)}{p(y)} \\ &= \log \textstyle\sum_y \sum_x p(x|y)p(y|x) \\ &\leq \log \textstyle\sum_y \sum_x p(x|y)(\max_x p(y|x)) \\ &= \log \textstyle\sum_y (\max_x p(y|x)) \sum_x p(x|y) \\ &= \log \textstyle\sum_y \max_x C[x,y] \\ &= \mathcal{ML}(C) \end{aligned}$$

Because this inequality holds for every π, it follows that

$$\text{Shannon capacity of } C = \sup_\pi I(X;Y) \leq \mathcal{ML}(C).$$

■

C. Practical implications of capacity bounds

When we consider the risk to confidentiality caused by a system C, different leakage measures may be appropriate in different scenarios, depending on factors like the structure of the set of secrets, the design of the system, and the adversary's strategy or power. For this reason, Theorems 5.1 and 5.3 can be very useful in simplifying our security analysis. For they tell us that if we can show that the min-capacity of C is small, then we are guaranteed that the leakage under *any* gain function g and under *any* prior π is also small, as is the Shannon leakage. In such a case, the multitude of possible gain functions g need not burden us.

This is not to say that we can simply forget about the gain function g, since a particular g can make the prior vulnerability much larger (as in (Ex1), for example). Indeed, we could say that leakage bounds address the *conservation* of confidentiality, while prior vulnerability addresses its *creation*, involving parameters like the sizes of passwords and their prior distribution.

Moreover, when we *compare* two channels, we may find that one has worse min-capacity than the other, even though the opposite ordering holds under the gain function and prior relevant for the scenario of interest.

To illustrate, recall (Ex1) from Section I, which leaks a randomly-chosen user's 10-bit password, giving it a min-capacity of 10 bits. Compare that channel with

$$\begin{array}{l} n \xleftarrow{?} \{0..9\}; \\ \textbf{if } n = 0 \textbf{ then} \\ \quad Y = (762, X[762]) \\ \textbf{else} \\ \quad Y := (0, 0) \end{array} \qquad \text{(Ex6)}$$

Since (Ex6) has only a $\frac{1}{10}$ probability of leaking anything, it is easy to see that its min-capacity is less than 10 bits.[9] So, with respect to min-capacity, (Ex1) is worse than (Ex6).

But suppose that it turns out that user 762 is Bill Gates, whose password is vastly more valuable than all the other passwords. In this scenario, it would make sense to replace g from Section III-D with a gain function like

$$g'((u,x),X) = \begin{cases} 1, & \text{if } u = 762 \text{ and } x = X[762] \\ 0.01, & \text{if } u \neq 762 \text{ and } x = X[u] \\ 0 & \text{otherwise} \end{cases}$$

Under g', the min-capacity ordering is reversed: now we find that (Ex6) is worse, since it has a $\frac{1}{10}$ probability of revealing Bill Gates's password, which under g' is worth 100 times as much as every other password.[10]

D. The prior that realizes g-capacity

A property of min-capacity that makes it easy to calculate is that it is always realized on a uniform prior. We have found, however, that this does *not* hold for g-capacity.

Consider channel (Ex5) above and its gain function g_d. Under a uniform prior π, we calculate that $V_{g_d}(\pi) = 0.66$, $p_Y = (0.2, 0.8)$, $V_{g_d}(p_{X|y_1}) = 1$, $V_{g_d}(p_{X|y_2}) = 0.825$, and $V_{g_d}(\pi, \text{Ex5}) = 0.86$, giving $\mathcal{L}_{g_d}(\pi, \text{Ex5}) = 0.3819$.

Now if we consider the prior $\pi' = (0.5, 0.5, 0)$, we find that $V_{g_d}(\pi') = 0.5$, $p_Y = (0.3, 0.7)$, $V_{g_d}(p_{X|y_1}) = 1$, $V_{g_d}(p_{X|y_2}) = \frac{5}{7}$, and $V_{g_d}(\pi', \text{Ex5}) = 0.8$, which gives $\mathcal{L}_{g_d}(\pi', \text{Ex5}) = \log 1.6 \approx 0.6781$. Hence the g_d-capacity of (Ex5) is *not* realized on a uniform distribution.

Notice here that $\log 1.6$ is also (Ex5)'s min-capacity. Hence, by Theorem 5.1, we know that $\log 1.6$ must in fact be its g_d-capacity, realized on π'.[11] But, so far, we have not found a general technique for calculating g-capacity; this remains an area for future study.

VI. Comparing channels

Given any leakage measure m (for example, Shannon leakage, min-entropy leakage, or g-leakage for some g), an interesting question that can be asked about two channels C_1 and C_2 is whether the leakage of C_1 is less than or equal to that of C_2, on *every* prior. For this question to make sense, both channels need to have the same input space $\mathcal{X}$, but they need not have the same output space.

Definition 6.1: Given channels C_1 from $\mathcal{X}$ to $\mathcal{Z}$ and C_2 from $\mathcal{X}$ to $\mathcal{Y}$, and a leakage measure m, write $C_1 \leq_m C_2$ if the m-leakage of C_1 never exceeds that of C_2, on any prior.

Notice that $C_1 \leq_m C_2$ implies that the m-capacity of C_1 is less than or equal to that of C_2, but not conversely.

One would expect that $\leq_m$ will depend on the particular choice of leakage measure m. Interestingly, Yasuoka and Terauchi [15] and Malacaria [16] show that on *deterministic* channels, we get the *same* ordering $\leq_m$ when m is either Shannon, min-entropy, or guessing entropy leakage. They show moreover a connection to the *Lattice of Information*.

Recall (e.g. [17], [7]) that a deterministic channel C from $\mathcal{X}$ to $\mathcal{Y}$ gives rise to an equivalence relation (or partition) on $\mathcal{X}$, given by $x_1 \sim_C x_2$ iff $C(x_1) = C(x_2)$. (By $C(x)$ we denote the unique y such that $C[x, y] = 1$.) In the Lattice of Information, we order these equivalence relations by *partition refinement*:

Definition 6.2: Given deterministic channels C_1 and C_2, write $C_1 \sqsubseteq C_2$ if the partition of C_1 is refined by the partition of C_2, in that each equivalence class of $\sim_{C_2}$ is contained within some equivalence class of $\sim_{C_1}$:

$$x_1 \sim_{C_2} x_2 \text{ implies } x_1 \sim_{C_1} x_2.$$

Yasuoka and Terauchi [15] and Malacaria [16] show that on deterministic channels, $\leq_m$ (for m being Shannon, min-entropy, or guessing entropy leakage) all coincide with $\sqsubseteq$.

The Lattice of Information applies only to deterministic channels, since probabilistic channels do not give partitions of $\mathcal{X}$. On the other hand, $\leq_m$ does make sense for probabilistic channels, so a natural question is: how can we generalize $\sqsubseteq$ to probabilistic channels, and what leakage ordering would characterize it? This is what we explore in this section.

Our first result (already observed in [17]) is that partition refinement on deterministic channels coincides with the existence of a channel factorization:

Theorem 6.1: Let C_1 from $\mathcal{X}$ to $\mathcal{Z}$ and C_2 from $\mathcal{X}$ to $\mathcal{Y}$ be deterministic channels. Then $C_1 \sqsubseteq C_2$ iff there exists deterministic C_3 from $\mathcal{Y}$ to $\mathcal{Z}$ such that $C_1 = C_2 C_3$. ($C_2 C_3$ denotes the *cascade* of C_2 and C_3, corresponding to multiplication of the channel matrices.)

Proof: If $C_1 = C_2 C_3$, for some deterministic C_3, then $C_2(x_1) = C_2(x_2)$ implies that $C_1(x_1) = C_3(C_2(x_1)) = C_3(C_2(x_2)) = C_1(x_2)$. Hence $C_1 \sqsubseteq C_2$.

Conversely, if $C_1 \sqsubseteq C_2$, then for every $y \in \mathcal{Y}$, C_1 maps all $x \in C_2^{-1}(y)$ to the same value, say z_y. If we define deterministic C_3 that maps each $y \in \mathcal{Y}$ to z_y, then it is easy to see that $C_1 = C_2 C_3$. ∎

Given this theorem, it seems promising to generalize partition refinement to probabilistic channels by introducing what we call *composition refinement*:

[9] In fact, its min-capacity turns out to be about 6.6907 bits.

[10] Under g', the prior vulnerability is 2^{-10}. Under (Ex1), the posterior vulnerability is 0.01099, giving g'-leakage of 3.492 bits. Under (Ex6), the posterior vulnerability is 0.10088, giving g'-leakage of 6.6907 bits.

[11] Curiously, π' also realizes (Ex5)'s *min-capacity*.

Definition 6.3: Given channel C_1 from $\mathcal{X}$ to $\mathcal{Z}$ and C_2 from $\mathcal{X}$ to $\mathcal{Y}$, we say that C_1 is *composition refined* by C_2, denoted $C_1 \sqsubseteq_\circ C_2$, if there exists a channel C_3 from $\mathcal{Y}$ to $\mathcal{Z}$ such that $C_1 = C_2 C_3$.

In terms of notation, we use $\leq_g$ to denote $\leq_m$ when m is g-leakage for a specific gain function g; that is, $C_1 \leq_g C_2$ iff $\forall \pi : \mathcal{L}_g(\pi, C_1) \leq \mathcal{L}_g(\pi, C_2)$. Note that this is equivalent to $\forall \pi : V_g(\pi, C_1) \leq V_g(\pi, C_2)$. We also use $\leq_{\mathcal{S}}$, where $\mathcal{S}$ is a set of gain functions, to denote the ordering under all gain functions in $\mathcal{S}$ (i.e. $\leq_{\mathcal{S}} = \cap_{g \in \mathcal{S}} \leq_g$). In particular, we use $\leq_{\mathcal{G}}, \leq_{\mathcal{G}_2}, \leq_{\mathcal{G}_\sim}$ to denote the ordering under *all*, *2-block*, and *partition* gain functions, respectively.

The key question, then, is whether the previous equivalence between $\sqsubseteq$ and $\leq_m$ carries over somehow to $\sqsubseteq_\circ$ and $\leq_g$ or $\leq_{\mathcal{G}}$.

In fact, a recent result in Espinoza and Smith [19] shows that $C_1 \sqsubseteq_\circ C_2$ implies $C_1 \leq_{\text{min-entropy}} C_2$.[12] We now show that we can generalize this implication to g-leakage under *any* gain function:

Theorem 6.2: If $C_1 \sqsubseteq_\circ C_2$, then $C_1 \leq_{\mathcal{G}} C_2$.

Proof: A direct proof is given in the Appendix. We here discuss a more intuitive proof in terms of viewing $V_g(\pi, C_1)$ as the solution to a linear optimization problem. Recall from Section IV-C that $V_g(\pi, C_1)$ is the solution to the problem of maximizing $tr(D_\pi C_1 S G)$ subject to S being a channel matrix. Let S_1 be any feasible solution to this problem (i.e. any channel matrix) and assume $C_1 = C_2 C_3$. Then $S_2 = C_3 S_1$ is a feasible solution to the optimization problem for C_2, giving gain

$$tr(D_\pi C_2 S_2 G) = tr(D_\pi C_2 C_3 S_1 G) = tr(D_\pi C_1 S_1 G)$$

That is, for any feasible solution of C_1's problem, there is a feasible solution for C_2's problem, giving the same gain. Thus, the optimal solution for C_2 (i.e. $V_g(\pi, C_2)$) can be no smaller than the optimal solution for C_1 (i.e. $V_g(\pi, C_1)$). ■

Now it is natural to wonder about *converses* to Theorem 6.2. We might first wonder whether (as in the deterministic case) $C_1 \leq_g C_2$ for a particular g is sufficient to imply that $C_1 \sqsubseteq_\circ C_2$. This turns out not to be true for g_{id} (i.e. min-entropy leakage) and the following channel matrices:

$$C_1 = \begin{pmatrix} 1/4 & 3/4 \\ 1/4 & 3/4 \\ 3/5 & 2/5 \end{pmatrix} \qquad C_2 = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \\ 1/2 & 1/2 & 0 \end{pmatrix}$$

It can be verified (using the decision procedure of Section VI-F) that $C_1 \leq_{g_{id}} C_2$ but $C_1 \not\sqsubseteq_\circ C_2$, so $\leq_{g_{id}}$ by itself does not imply composition refinement.[13]

But what if the g-leakage ordering holds for *all* gain functions? We conjecture that this is sufficient to imply composition refinement:

Conjecture 6.3 ("Coriaceous"): If $C_1 \leq_{\mathcal{G}} C_2$, then $C_1 \sqsubseteq_\circ C_2$.

If the conjecture holds, then $\leq_{\mathcal{G}}$ and $\sqsubseteq_\circ$ coincide, providing an extension of Yasuoka, Terauchi, and Malacaria's equivalence to the probabilistic case (the only difference being that we need to consider the ordering under all gain functions). The conjecture, however, turns out to be remarkably subtle, and we have not yet been able to prove it in full generality. But we have been able to prove it in substantial special cases, using techniques that we now describe.

A. The case of invertible C_2

We begin with a useful tool for showing that a leakage ordering does *not* hold.

Definition 6.4: Vector v is a *cat-vector* for C_1 and C_2 if the inner product of v with *each* column of C_2 is non-negative, and the inner product of v with *some* column of C_1 is negative.

Lemma 6.4: If there exists a cat-vector v for C_1 and C_2, then there is a 2-block gain function g such that $C_1 \not\leq_g C_2$.

Proof: Assume that C_1 goes from $\mathcal{X}$ to $\mathcal{Z}$ and C_2 from $\mathcal{X}$ to $\mathcal{Y}$. Given cat-vector v indexed by $\mathcal{X}$, let $z^\star$ be (the index of) a column of C_1 whose inner product with v is negative:

$$\textstyle\sum_{x \in \mathcal{X}} v[x] C_1[x, z^\star] < 0. \tag{1}$$

In contrast, for every column y of C_2, we have

$$\textstyle\sum_{x \in \mathcal{X}} v[x] C_2[x, y] \geq 0. \tag{2}$$

It follows from these two facts that v must contain both *positive* and *negative* entries. This lets us split set $\mathcal{X}$ into two nonempty parts:

$$\mathcal{X}^+ = \{x \in \mathcal{X} \mid v[x] \geq 0\}$$

and

$$\mathcal{X}^- = \{x \in \mathcal{X} \mid v[x] < 0\}.$$

Now let us define prior π using the absolute values of the entries in v:

$$\pi[x] = \frac{1}{\gamma} |v[x]|$$

where normalizing factor γ is defined as $\gamma = \sum_{x \in \mathcal{X}} |v[x]|$.

The intuition behind this choice of π is that because of (2), we know that under C_2, the *a posteriori* probability of $\mathcal{X}^-$ never exceeds that of $\mathcal{X}^+$, for any output y. In contrast, because of (1), we know that under C_1, the *a posteriori* probability of $\mathcal{X}^-$ *does* exceed that of $\mathcal{X}^+$ on output $z^\star$.

We can define a 2-block gain function to exploit this difference. Define $\mathcal{W} = \{\mathcal{X}^+, \mathcal{X}^-\}$ and

$$g(W, x) = \begin{cases} 1, & \text{if } x \in W \\ 0, & \text{if } x \notin W \end{cases}$$

In other words, g cares only about whether we correctly guess whether x belongs to $\mathcal{X}^+$ or to $\mathcal{X}^-$.

[12] Also, the classic *data-processing inequality* [18] shows (essentially) the same implication for Shannon leakage.

[13] We also mention that we have experimental evidence (but no proof) that $C_1 \leq_{\text{Shannon}} C_2$, so the Shannon leakage order also appears insufficient to imply composition refinement.

Now we will argue that under g and π, C_1 has greater leakage than C_2; in fact we show that C_1's g-leakage is positive, while C_2's is zero.

Looking at C_2's leakage, we have

$$\begin{aligned}
&V_g(\pi, C_2)\\
&= \sum_{y\in\mathcal{Y}} \max_{W\in\mathcal{W}} \sum_{x\in\mathcal{X}} \pi[x]C_2[x,y]g(W,x)\\
&= \frac{1}{\gamma}\sum_{y\in\mathcal{Y}} \max_{W\in\mathcal{W}} \sum_{x\in\mathcal{X}} |v[x]|C_2[x,y]g(W,x)\\
&= \frac{1}{\gamma}\sum_{y\in\mathcal{Y}} \max_{W\in\mathcal{W}} \begin{pmatrix} \sum_{x\in\mathcal{X}^+} v[x]C_2[x,y]g(W,x)\\ -\sum_{x\in\mathcal{X}^-} v[x]C_2[x,y]g(W,x)\end{pmatrix}\\
&= \frac{1}{\gamma}\sum_{y\in\mathcal{Y}} \max\left\{\sum_{x\in\mathcal{X}^+} v[x]C_2[x,y], -\sum_{x\in\mathcal{X}^-} v[x]C_2[x,y]\right\}
\end{aligned}$$

Now, in light of equation (2) we can see that in the final "max", the left sum is greater than or equal to the right sum, for every y. Hence $\mathcal{X}^+$ is the best guess under every y, which implies by Theorem 4.2 that $\mathcal{X}^+$ is also the best guess *a priori*, and that $\mathcal{L}_g(\pi, C_2) = 0$.

When we consider C_1's leakage, in contrast, we can show by a similar calculation that equation (1) implies that the best guess under output $z^\star$ is $\mathcal{X}^-$. But, since $\mathcal{X}^+$ is the best guess *a priori*, we conclude by Theorem 4.2 that $\mathcal{L}_g(\pi, C_2) > 0$. ■

Lemma 6.4 allows us to prove some significant special cases of Conjecture 6.3, as we now show.

Theorem 6.5: If C_2 is invertible and $C_1 \leq_{\mathcal{G}_2} C_2$, then $C_1 \sqsubseteq_\circ C_2$.

Proof: We argue the contrapositive. Suppose that C_2 is invertible and $C_1 \not\sqsubseteq_\circ C_2$. Then there does not exist a channel matrix C_3 such that $C_1 = C_2C_3$. But, assuming that C_2 is invertible, we do have $C_1 = C_2(C_2^{-1}C_1)$, so it must be that $C_2^{-1}C_1$ is *not* a channel matrix.

Now, a basic property of matrix multiplication is that multiplication on the right by a channel matrix preserves row sums. Since $I = C_2^{-1}C_2$, it follows that each row of C_2^{-1} sums to 1. And this implies that each row of $C_2^{-1}C_1$ also sums to 1. Hence for $C_2^{-1}C_1$ to not be a channel matrix, it must contain a negative entry, say at position $[y^\star, z^\star]$. This is equivalent to saying that the inner product of row $y^\star$ of C_2^{-1} and column $z^\star$ of C_1 is negative. Moreover, since $C_2^{-1}C_2 = I$ we know that the inner product of row $y^\star$ of C_2^{-1} and any column y of C_2 is non-negative (in fact the inner product is always either 0 or 1). Hence we see that row $y^\star$ of C_2^{-1} is a cat-vector for C_1 and C_2, and the result follows from Lemma 6.4. ■

Note that $\leq_{\mathcal{G}} \subseteq \leq_{\mathcal{G}_2}$; the above theorem shows that in the case when C_2 is invertible, the conjecture holds even if we restrict to 2-block gain functions.

B. The case of "skinny", full-rank C_2

We now strengthen Theorem 6.5 to the case when C_2's columns are linearly independent, dropping the assumption that its rows are linearly independent; this is the case of a full-rank C_2 that is "skinny", with at least as many rows as columns.

Theorem 6.6: If C_2's columns are linearly independent and $C_1 \leq_{\mathcal{G}_2} C_2$, then $C_1 \sqsubseteq_\circ C_2$.

Proof: The key idea is that if $C_1 \leq_{\mathcal{G}_2} C_2$ and the rows of C_2 are linearly dependent, then the rows of C_1 must be linearly dependent with the *same* coefficients. For if there is a vector v whose inner product with each column of C_2 is 0 but whose inner product with some column of C_1 is nonzero, then either v or $-v$ is a cat-vector for C_1 and C_2. Hence a factorization exists iff there is a factorization for the linearly independent rows of C_2 and the corresponding rows of C_1. On the assumption that C_2's columns are linearly independent, the linearly independent rows of C_2 form an invertible matrix, and so we are done by Theorem 6.5. ■

C. The case of "fat" C_2

As we discussed in the previous sections, in the case when C_2 is invertible or "skinny" the conjecture can be shown to hold (i.e. leakage ordering implies factorability), even if we consider only *2-block gain functions*. But when we consider the case of a full-rank C_2 that is "fat", with more columns than rows, the situation becomes far more difficult. It turns out then that neither 2-block gain functions nor even *general binary gain functions* (see Section III-C4) are sufficient.

Consider the following channels (note that C_2 is "fat"):

$$C_1 = \begin{pmatrix} .2 & .22 & .58\\ .2 & .4 & .4\\ .35 & .4 & .25\end{pmatrix} \qquad C_2 = \begin{pmatrix} .1 & .4 & .1 & .4\\ .2 & .2 & .3 & .3\\ .5 & .1 & .1 & .3\end{pmatrix}$$

It can be verified (using the decision procedure of Section VI-F) that $C_1 \leq_g C_2$ for *all* general binary gain functions g, but $C_1 \not\sqsubseteq_\circ C_2$. Nevertheless, these channels are *not* a counterexample to Conjecture 6.3, because the following gain function g (again computed using the techniques of Section VI-F) makes C_1 leak more than C_2:

g	x_1	x_2	x_3
w_1	$153/296$	0	$1/2$
w_2	0	$289/296$	$63/296$
w_3	$21/148$	1	0

For this gain function we have

$$V_g(\pi_u, C_1) = 0.412117 \qquad V_g(\pi_u, C_2) = 0.409797$$

which implies that $C_1 \not\leq_g C_2$.

D. The case of deterministic channels

Another special case in which we are able to settle our conjecture is the one when C_1 is deterministic (without any

restriction on C_2, which could be "fat"). In fact, the conjecture holds even if we restrict to *partition* gain functions.

Theorem 6.7: If C_1 is deterministic and $C_1 \leq_{\mathcal{G}_\sim} C_2$, then $C_1 \sqsubseteq_\circ C_2$.

The main idea of the proof is to construct a partition gain function using the partition $\sim_{C_1}$ induced by C_1. In terms of representing gain functions by matrices, this corresponds to taking $G = C_1^T$ (the transpose of C_1).

We next focus on the purely deterministic case, i.e. when both C_1 and C_2 are deterministic. In this case we can prove a stronger result, namely that the ordering induced by a *single* gain function is enough to imply factorability. This is in fact expected, since we already know by the result of Yasuoka, Terauchi, and Malacaria, together with Theorem 6.1, that $\leq_{g_{id}}$ implies factorability. We generalize this to the class of *single-optimal* guess functions:

Definition 6.5: A gain function $g : \mathcal{W} \times \mathcal{X} \to [0,1]$ is called *single-optimal* iff

$$\forall x \in \mathcal{X} \exists w \in \mathcal{W} : g(w,x) = 1$$
$$\forall x, x' \in \mathcal{X}, w \in \mathcal{W} : g(w,x) = 1 \wedge x \neq x' \Rightarrow g(w,x') < 1$$

Intuitively, a gain function is single-optimal if for every secret there is an optimal guess (giving gain 1) and each guess can be optimal for at most one secret. Note that g_{id} and all gain functions induced by metrics are single optimal. However partition gain functions are not (except for g_{id}).

Theorem 6.8: If C_1, C_2 are deterministic, g is a single-optimal gain function, and $C_1 \leq_g C_2$, then $C_1 \sqsubseteq_\circ C_2$.

Note that the above result does not always hold for non-single-optimal gain functions. A trivial example is the "happy" gain function since $\mathcal{L}_{g_\supset}(\pi, C) = 0$ for all π, C. Moreover, even for non-trivial gain functions, such as the 2-try gain function $g_{\mathcal{W}_2}$, the result might not hold. Let C_1 be the identity channel and C_2 be the deterministic channel $C_2(1) = C_2(2) = 1$ and $C_2(x) = x, x \in \{3, \ldots, n\}$. Thus $C_1 \not\sqsubseteq_\circ C_2$. In the case of C_1 the gain is always 1 since the input can be completely inferred from the output. In the case of C_2, seeing the output 1 the attacker is confused between 1 and 2, but having 2 tries, he can guess $\{1,2\}$ and still get gain 1. So $\forall \pi : \mathcal{L}_{g_{\mathcal{W}_2}}(\pi, C_1) = \mathcal{L}_{g_{\mathcal{W}_2}}(\pi, C_2)$, thus $C_1 \leq_{g_{\mathcal{W}_2}} C_2$.

E. Other results on leakage ordering

We present two other general results about $\leq_\mathcal{G}$. First, note that $C_1 \leq_\mathcal{G} C_2$ contains a double quantification: it requires that the g-leakage of C_1 does not exceed that of C_2 for all priors and all gain functions. It turns out that quantifying over gain functions is powerful enough that we can limit ourselves to *uniform* priors π_u without weakening the ordering.

Theorem 6.9: If $\mathcal{L}_g(\pi_u, C_1) \leq \mathcal{L}_g(\pi_u, C_2)$ for all gain functions g, then $C_1 \leq_\mathcal{G} C_2$.

Also, $\leq_\mathcal{G}$ is preserved under left multiplication.

Theorem 6.10: For all channels C, C_1, C_2, if $C_1 \leq_\mathcal{G} C_2$ then $CC_1 \leq_\mathcal{G} CC_2$.

F. Decision procedures for comparing channels

In this section, we discuss algorithms for two decision problems related to the leakage orderings. Note that our goal is not to develop efficient algorithms that can be used in practice, but rather to be able to obtain the examples and counter-examples presented in the previous sections. Still, the problems we tackle are of interest on their own.

Problem 6.1: Given C_1, C_2, g, decide whether $C_1 \leq_g C_2$.

The challenge is clearly the quantification over all priors. Recall from Section IV-C that $V_g(\pi, C_1) = \max_S tr(D_\pi C_1 S G)$ subject to S being a channel matrix. To decide $\leq_g$ we can solve the following optimization problem, with π, S_1, S_2 being variables.

$$\max_\pi \big(\max_{S_1} tr(D_\pi C_1 S_1 G) - \max_{S_2} tr(D_\pi C_2 S_2 G) \big)$$

subject to π being a probability distribution and S_1, S_2 being channel matrices. Note that $C_1 \leq_g C_2$ holds iff the solution is non-positive.

There are however two issues with this problem: first, it is quadratic and second, it contains nested optimizations. To cope with these issues, we notice that there is a finite number of deterministic strategies S_1 (in particular, there are $|\mathcal{W}|^{|\mathcal{Z}|}$ such strategies) and we know that V_g can always be given by a deterministic strategy. Moreover, for a fixed S_1, the property "π is a prior such that S_1 is optimal" can be expressed by a set of linear constraints (the variables being π):

$$E_g(S_1(z), z) \geq E_g(w, z) \quad \forall z \in \mathcal{Z}, w \in \mathcal{W}$$

(using the notation of Sec IV-B; note that E_g depends on π and C_1). Intuitively, the constraints require that the guess chosen by S_1 for each output z is no worse than any other guess. We refer to these constraints as $opt(S_1)$.

Then, the solution to the above (non-linear) optimization problem will be the maximum of the solutions to the following linear problems:

$$\max_\pi \big(tr(D_\pi C_1 S_1 G) - tr(D_\pi C_2 S_2 G) \big) \quad (3)$$
subject to π being a distribution and $opt(S_1), opt(S_2)$

for each S_1, S_2, i.e. $|\mathcal{W}|^{|\mathcal{Z}|+|\mathcal{Y}|}$ systems in total. In case $C_1 \not\leq_g C_2$ the solution also provides a counter-example π.

Problem 6.2: Given C_1, C_2, n, decide whether $C_1 \leq_{\mathcal{G}_n} C_2$, where $\mathcal{G}_n$ denotes the set of all gain functions with n possible guesses (i.e. where $|\mathcal{W}| = n$).

Note that the exact set $\mathcal{W}$ of guesses is not important, as any gain function with n possible guesses can be represented by a $n \times |\mathcal{X}|$ matrix G. First, from Theorem 6.9 (adapted to $\leq_{\mathcal{G}_n}$ instead of $\leq_\mathcal{G}$), we know that $C_1 \leq_{\mathcal{G}_n} C_2$ iff $\mathcal{L}_g(\pi_u, C_1) \leq \mathcal{L}_g(\pi_u, C_2)$ for all $g \in \mathcal{G}_n$. We can then decide this problem by solving the same finite number of linear optimization

problems as in Problem 6.1, the only difference being that π is now fixed to a uniform one, while the variables are the elements of G (with the constraints $G[w, x] \in [0, 1]$).

VII. Related Work

The converse of gain functions, usually called *loss functions*, have been used for a long time in fields such as decision theory [20], [21], economics [22], [23], [24], and machine learning [25], [26], to cite a few. In the domain of information flow, Ghosh et al. [27] explore the utility of randomization mechanisms for queries to statistical databases subject to differential privacy guarantees. Their work, which inspired our linear programming formulation of vulnerability, assumes that the utility of the mechanism for a particular user depends on his prior on secrets, and on a loss function corresponding to how much this particular user loses by guessing an answer j when the actual answer is i. Their approach is close to ours in spirit, but they impose restrictions on the loss functions (symmetry and monotonicity) that we do not, which provides our approach with much more flexibility.

As discussed in Section VI, Yasuoka and Terauchi [15] and Malacaria [16] explore the relation between leakage ordering and partition refinement in the Lattice of Information. Their works, however, consider deterministic channels only, while in this paper we address the more general case of probabilistic channels.

Boreale et al. [28] extend the information flow scenario with the notion of *views*, which are essentially partitions of the space of secrets (possibly probabilistic partitions). They derive bounds on the probability that the adversary correctly guesses which block the secret belongs to, as the number of observations tends to infinity. Their bounds, however, only consider the a posteriori probability of success, whereas our approach considers the *leakage*, i.e. the relation between the probability of success *a posteriori* and *a priori*.

McIver et al. [29], [30] consider a refinement order that is preserved under composition and that is a partial order on programs. This order is sound and complete for Bayes risk, and they show that Bayes risk is maximally discerning, if contexts are taken into account, when compared to the alternative elementary tests of marginal guesswork, guessing entropy and Shannon entropy. Again, McIver et al. consider only the a posteriori probability of success, whereas we consider leakage.

VIII. Conclusion

In this paper we introduced g-leakage, a generalization of min-entropy leakage that makes use of gain functions to allow for the accurate quantification of leakage in a rich variety of operational scenarios. We also proved important mathematical properties of g-leakage that further attest to the significance of our framework.

As future work we intend to identify algorithms to calculate g-capacity, possibly using linear programming. Also, it would be interesting to extend g-leakage to the scenario where the adversary does not *know* the prior π, but instead has (possibly incorrect) *beliefs* about it, as in the works of Clarkson, Myers, and Scheider [31] and Hamadou, Sassone, and Palamidessi [32]. Finally, we also want to investigate the applicability of g-leakage to the problem of privacy and utility in differential privacy.

Acknowledgments: We are grateful to Miguel E. Andrés for discussions of this work, and to the anonymous referees for their comments and suggestions. Mário S. Alvim was partially supported by the MURI program under AFOSR Grant No: FA9550-08-1-0352, and by INRIA and LIX. Geoffrey Smith was partially supported by the National Science Foundation under grants CNS-0831114 and CNS-1116318 and by LIX and Digiteo.

References

[1] J. K. Millen, "Covert channel capacity," in *IEEE Symposium on Security and Privacy*, 1987, pp. 60–66.

[2] J. McLean, "Security models and information flow," in *IEEE Symposium on Security and Privacy*, 1990, pp. 180–189.

[3] J. W. Gray, III, "Toward a mathematical foundation for information flow security," in *IEEE Symposium on Security and Privacy*, 1991, pp. 21–35.

[4] D. Clark, S. Hunt, and P. Malacaria, "Quantitative analysis of the leakage of confidential data," in *Proc. Workshop on Quantitative Aspects of Programming Languages*, ser. Electr. Notes Theor. Comput. Sci, vol. 59 (3), 2001, pp. 238–251.

[5] M. Boreale, "Quantifying information leakage in process calculi," in *Proc. ICALP '06*, 2006, pp. 119–131.

[6] P. Malacaria, "Assessing security threats of looping constructs," in *Proc. 34th Symposium on Principles of Programming Languages (POPL '07)*, 2007, pp. 225–235.

[7] B. Köpf and D. Basin, "An information-theoretic model for adaptive side-channel attacks," in *Proc. 14th ACM Conference on Computer and Communications Security (CCS '07)*, 2007, pp. 286–296.

[8] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden, "Anonymity protocols as noisy channels," *Information and Computation*, vol. 206, pp. 378–401, 2008.

[9] ——, "On the Bayes risk in information-hiding protocols," *Journal of Computer Security*, vol. 16, no. 5, pp. 531–571, 2008.

[10] G. Smith, "On the foundations of quantitative information flow," in *Proc. 12th International Conference on Foundations of Software Science and Computational Structures (FoSSaCS '09)*, ser. Lecture Notes in Computer Science, L. de Alfaro, Ed., vol. 5504, 2009, pp. 288–302.

[11] M. R. Clarkson and F. B. Schneider, "Quantification of integrity," in *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, 2010, pp. 28–43.

[12] M. Boreale, F. Pampaloni, and M. Paolini, "Asymptotic information leakage under one-try attacks," in *Proc. FOSSACS '11*, 2011, pp. 396–410.

[13] C. Braun, K. Chatzikokolakis, and C. Palamidessi, "Quantitative notions of leakage for one-try attacks," in *Proc. 25th Conference on Mathematical Foundations of Programming Semantics (MFPS 2009)*, ser. ENTCS, vol. 249, 2009, pp. 75–91.

[14] G. Smith, "Quantifying information flow using min-entropy," in *Proc. QEST 2011: 8th International Conference on Quantitative Evaluation of SysTems*, 2011, pp. 159–167.

[15] H. Yasuoka and T. Terauchi, "Quantitative information flow — verification hardness and possibilities," in *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, 2010, pp. 15–27.

[16] P. Malacaria, "Algebraic foundations for information theoretical, probabilistic and guessability measures of information flow," *CoRR*, vol. abs/1101.3453, 2011.

[17] J. Landauer and T. Redmond, "A lattice of information," in *Proc. Computer Security Foundations Workshop VI*, Jun. 1993, pp. 65 –70.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.

[19] B. Espinoza and G. Smith, "Min-entropy leakage of channels in cascade," in *Proc. Formal Aspects of Security and Trust (FAST 2011)*, ser. Lecture Notes in Computer Science, G. Barthe, A. Datta, and S. Etalle, Eds., 2011, to appear.

[20] G. Parmigiani and L. Inoue, *Decision Theory: Principles and Approaches*. John Wiley & Sons, 2009.

[21] A. Zellner, "Bayesian estimation and prediction using asymmetric loss functions," *Journal of the American Statistical Association*, vol. 81, no. 394, pp. 446–451, 1986.

[22] G. Koop, *Bayesian Econometrics*. John Wiley & Sons, 2003.

[23] F. Schorfheide, "Loss function-based evaluation of dsge models," *Journal of Applied Econometrics*, vol. 15, no. 6, pp. 645–670, 2000.

[24] J.-N. Pan and J. Pan, "A comparative study of various loss functions in the economic tolerance design," in *Management of Innovation and Technology, 2006 IEEE International Conference on*, vol. 2, june 2006, pp. 783 –787.

[25] M. A. H. Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kauffman Publishers, 2011.

[26] T. G. Dietterich, "Machine learning for sequential data: A review," in *Structural, Syntactic, and Statistical Pattern Recognition*. Springer-Verlag, 2002, pp. 15–30.

[27] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," in *Proc. 41st ACM Symposium on Theory of Computing (STOC'09)*, 2009, pp. 351–360.

[28] M. Boreale, F. Pampaloni, and M. Paolini, "Quantitative information flow, with a view," in *Proc. ESORICS '11*, 2011.

[29] A. McIver, L. Meinicke, and C. Morgan, "Compositional closure for Bayes risk in probabilistic noninterference," in *Proc. ICALP'10*, 2010, pp. 223–235.

[30] ——, "Compositional closure for Bayes Risk in probabilistic noninterference," *CoRR*, vol. abs/1007.1054, 2010.

[31] M. Clarkson, A. Myers, and F. Schneider, "Belief in information flow," in *Proc. 18th IEEE Computer Security Foundations Workshop (CSFW '05)*, 2005, pp. 31–45.

[32] S. Hamadou, V. Sassone, and C. Palamidessi, "Reconciling belief and vulnerability in information flow," in *Proc. 31st IEEE Symposium on Security and Privacy*, 2010, pp. 79–92.

APPENDIX

We provide here the proofs omitted from Section VI.

Theorem 6.2: If $C_1 \sqsubseteq_\circ C_2$, then $C_1 \leq_{\mathcal{G}} C_2$.

Proof: Assuming that C_1 goes from $\mathcal{X}$ to $\mathcal{Z}$ and C_2 goes from $\mathcal{X}$ to $\mathcal{Y}$, by hypothesis we have C_3 from $\mathcal{Y}$ to $\mathcal{Z}$ such that $C_1 = C_2C_3$. Given any prior π and gain function $g : \mathcal{W} \times \mathcal{X} \to [0,1]$, we have

$$
\begin{aligned}
&V_g(\pi, C_1)\\
&= \sum_{z\in\mathcal{Z}} \max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} (C_2C_3)[x,z]\pi[x]g(w,x)\\
&= \sum_{z\in\mathcal{Z}} \max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} C_2[x,y]C_3[y,z]\pi[x]g(w,x)\\
&\leq \text{[moving } \max \text{ inside a non-negative sum]}\\
&\quad \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Y}} \max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} C_2[x,y]C_3[y,z]\pi[x]g(w,x)\\
&= \sum_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} C_3[y,z] \max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} C_2[x,y]\pi[x]g(w,x)\\
&= \text{[the max does not depend on } z]\\
&\quad \sum_{y\in\mathcal{Y}} \big(\sum_{z\in\mathcal{Z}} C_3[y,z]\big)\big(\max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} C_2[x,y]\pi[x]g(w,x)\big)\\
&= \text{[summing over a row of } C_3]\\
&\quad \sum_{y\in\mathcal{Y}} \max_{w\in\mathcal{W}} \sum_{x\in\mathcal{X}} C_2[x,y]\pi[x]g(w,x)\\
&= V_g(\pi, C_2)
\end{aligned}
$$

It follows that $C_1 \leq_g C_2$. ∎

Theorem 6.7: If C_1 is deterministic and $C_1 \leq_{\mathcal{G}_\sim} C_2$, then $C_1 \sqsubseteq_\circ C_2$.

Proof: The key idea is that in the case when C_1 is a deterministic channel from $\mathcal{X}$ to $\mathcal{Z}$, then there is always a *partition* gain function g that makes C_1 a perfect channel.

We simply use the columns of C_1 to determine the partition, interpreting each column as a subset of $\mathcal{X}$—note that there can be no overlap among the columns of a deterministic channel. Seeing g as a matrix G, this means that G is just the *transpose* of C_1, and that the set $\mathcal{W}$ of guesses is $\mathcal{Z}$.

Clearly, $V_g(\pi, C_1) = 1$ for any π, since given output z, the adversary can guess z and be guaranteed of getting gain 1. That is, the optimal strategy for C_1 is just the identity function (a "direct" strategy, to use the terminology of [27]).

Now suppose that C_1's leakage with respect to g does not exceed C_2's, on some prior π, which we can assume to have *full support* (meaning that $\pi[x] > 0$ for every x). Using the linear programming formulation of vulnerability from Section IV-C, this means that C_2 must have a strategy S_2 such that $tr(D_\pi C_2 S_2 G) = 1$. Since $\sum_x \pi[x] = 1$, this means that the diagonal elements of $C_2 S_2 G$ must all be 1. Hence for every x we have

$$1 = (C_2 S_2 G)[x, x] = \sum_z (C_2 S_2)[x, z] G[z, x].$$

Now notice that column x of G is all zero, except for a single 1 at the unique $z^\star$ such that $C_1[x, z^\star] = 1$. Hence

$$1 = (C_2 S_2)[x, z^\star] = C_1[x, z^\star]$$

But we also know that $C_2 S_2$ is a channel matrix, which means that all its other entries in row x have to be 0. Hence $C_2 S_2 = C_1$, in other words, C_2's optimal strategy S_2 is *exactly* the C_3 that we are seeking! ∎

The following lemma is used in Theorem 6.8; its proof is straightforward and is omitted due to space constraints.

Lemma A.1: Let C be a deterministic channel, $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$ a gain function and π a prior. Then

$$V_g(\pi, C) = \sum_{A \in \mathcal{X}/\sim_C} \max_{w \in \mathcal{W}} \sum_{x \in A} \pi[x] g(w, x)$$

Theorem 6.8: If C_1, C_2 are deterministic, g is a single-optimal gain function and $C_1 \leq_g C_2$, then $C_1 \sqsubseteq_\circ C_2$.

Proof: Assume that $f \not\sqsubseteq_\circ h$, thus there exist $x_1, x_2 \in \mathcal{X}$ such that $C_1(x_1) \neq C_1(x_2)$ and $h(x_1) = h(x_2)$. Note that x_1, x_2 belong to different equivalent classes of $\sim_{C_1}$ (let A_1, A_1' be those classes) but to the same equivalent class A_2 of $\sim_{C_2}$. We define a prior π as $\pi[x_1] = \pi[x_2] = \frac{1}{2}$ and 0 elsewhere. We have:

$$\begin{aligned}
&V_g(\pi, C_1) \\
&= \sum_{A \in \mathcal{X}/\sim_{C_1}} \max_{w \in \mathcal{W}} \sum_{x \in A} \pi[x] g(w, x) \qquad \text{[Lemma A.1]} \\
&= \max_{w \in \mathcal{W}} \sum_{x \in A_1} \pi[x] g(w, x) + \max_{w \in \mathcal{W}} \sum_{x \in A_1'} \pi[x] g(w, x) \\
&= \max_{w \in \mathcal{W}} \pi[x_1] g(w, x_1) + \max_{w \in \mathcal{W}} \pi[x_2] g(w, x_2) \\
&= 1 \qquad [g(w, x_i) = 1 \text{ for some } w]
\end{aligned}$$

$$\begin{aligned}
&V_g(\pi, C_2) \\
&= \sum_{A \in \mathcal{X}/\sim_{C_2}} \max_{w \in \mathcal{W}} \sum_{x \in A} \pi[x] g(w, x) \qquad \text{[Lemma A.1]} \\
&= \max_{w \in \mathcal{W}} \sum_{x \in A_2} \pi[x] g(w, x) \\
&= \max_{w \in \mathcal{W}} \big(\pi[x_1] g(w, x_1) + \pi[x_2] g(w, x_2)\big) \\
&= \frac{1}{2} \max_{w \in \mathcal{W}} \big(g(w, x_1) + g(w, x_2)\big) \\
&< 1 \qquad [g(w, x_1) < 1 \text{ or } g(w, x_2) < 1]
\end{aligned}$$

Thus $V_g(\pi, C_1) > V_g(\pi, C_2)$ and $\mathcal{L}_g(\pi, C_1) > \mathcal{L}_g(\pi, C_2)$ which is a contradiction. ∎

Theorem 6.9 If $\mathcal{L}_g(\pi_u, C_1) \leq \mathcal{L}_g(\pi_u, C_2)$ for all gain functions g, then $C_1 \leq_\mathcal{G} C_2$.

Proof: Let C_1 and C_2 be channels from $\mathcal{X}$ to $\mathcal{Z}$ and $\mathcal{Y}$ respectively, let π be a prior, and let $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$ be a gain function. We show that $\mathcal{L}_g(\pi, C_1) \leq \mathcal{L}_g(\pi, C_2)$.

We define a gain function $g' : \mathcal{W} \times \mathcal{X} \to [0, 1]$ as $g'(w, x) = \pi[x] g(w, x)$. The idea is that π is "hard-coded" inside g'. By hypothesis we have $\mathcal{L}_{g'}(\pi_u, C_1) \leq \mathcal{L}_{g'}(\pi_u, C_2)$ which implies

$$\begin{aligned}
&\textstyle\sum_{z \in \mathcal{Z}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} C_1[x, z] \frac{1}{|\mathcal{X}|} g'(w, x) \\
&\leq \textstyle\sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} C_2[x, y] \frac{1}{|\mathcal{X}|} g'(w, x)
\end{aligned}$$

which implies $\mathcal{L}_g(\pi, C_1) \leq \mathcal{L}_g(\pi, C_2)$. ∎

Theorem 6.10: For all channels C, C_1, C_2, if $C_1 \leq_\mathcal{G} C_2$ then $CC_1 \leq_\mathcal{G} CC_2$.

Proof: Let C be a channel from $\mathcal{X}$ to $\mathcal{Y}$, C_1, C_2 be channels from $\mathcal{Y}$ to $\mathcal{Z}, \mathcal{Z}'$ respectively and let G be a gain function (in matrix form). We first show that

$$V_G(\pi_u, CC_i) = \frac{|\mathcal{Y}|}{|\mathcal{X}|} V_{GC}(\pi_u, C_i) \quad i \in 1, 2$$

i.e. we "hard-code" C inside the gain function. Note that GC is a valid gain function since C is stochastic. We have:

$$\begin{aligned}
&V_G(\pi_u, CC_1) \\
&= \sum_{z \in \mathcal{Z}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} (CC_1)[x, z] \frac{1}{|\mathcal{X}|} G[w, x] \\
&= \sum_{z \in \mathcal{Z}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} C[x, y] C_1[y, z] \frac{1}{|\mathcal{X}|} G[w, x] \\
&= \sum_{z \in \mathcal{Z}} \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} C_1[y, z] \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} G[w, x] C[x, y] \\
&= \frac{|\mathcal{Y}|}{|\mathcal{X}|} \sum_{z \in \mathcal{Z}} \max_{w \in \mathcal{W}} \sum_{y \in \mathcal{Y}} C_1[y, z] \frac{1}{|\mathcal{Y}|} GC[w, y] \\
&= \frac{|\mathcal{Y}|}{|\mathcal{X}|} V_{GC}(\pi_u, C_1)
\end{aligned}$$

and similarly for $V_G(\pi_u, CC_2)$. By hypothesis we have that $V_{GC}(\pi_u, C_1) \leq V_{GC}(\pi_u, C_2)$ which implies $V_G(\pi_u, CC_1) \leq V_G(\pi_u, CC_2)$, thus, by Theorem 6.9, we get $CC_1 \leq_\mathcal{G} CC_2$. ∎