Recent Developments in Quantitative Information Flow

(Invited Tutorial)

Geoffrey Smith
School of Computing and Information Sciences
Florida International University
Miami, Florida USA
Email: smithg@cis.fiu.edu

Abstract—In computer security, it is frequently necessary in practice to accept some leakage of confidential information. This motivates the development of theories of Quantitative Information Flow aimed at showing that some leaks are “small” and therefore tolerable. We describe the fundamental view of channels as mappings from prior distributions on secrets to hyper-distributions, which are distributions on posterior distributions, and we show how g-leakage provides a rich family of operationally-significant measures of leakage. We also discuss two approaches to achieving robust judgments about leakage: notions of capacity and a robust leakage ordering called composition refinement.

Keywords—security, confidentiality, information theory.

I. INTRODUCTION

A fundamental and vexing problem in computer security is to control the leakage of sensitive information. In the past year, numerous large-scale compromises (e.g. at Home Depot, Target, and JPMorgan Chase) showed how far we are from being able to solve this problem. When breaches of this kind occur, one important and natural response is to investigate the precise vulnerabilities that enabled the breach, and to seek ways to remedy them. But it appears more and more doubtful that the approach of patching vulnerabilities as they are discovered will ever lead us to a secure and trustworthy cyber-infrastructure. Fundamentally, achieving a real solution requires the development of a true science of information flow, a science that supports the construction and analysis of systems with precise information-flow guarantees.

Of course there are profound challenges in the development of such a science. A first issue concerns the modeling of systems. As Lynch and Fischer wrote long ago [1], “one would like simple mathematical models that exhibit the essential features of these systems while abstracting away irrelevant details”. But what is essential and what is irrelevant? The history of computer security is full of “irrelevant details” that turned out to be essential—indeed, many low-level system details (e.g. timing, power consumption, caching behavior), which a mathematical model would naturally abstract away, have turned out to afford powerful side channels leaking sensitive information. Of course the modeling issue is not one that can be solved in generality; we always need to be mindful of the abstractions that we make and sensitive to the issues that we may be overlooking.

Another key issue is the information flow policies to be enforced. Early work in information flow focused on completely preventing leakage of sensitive information, requiring that the observable output of a system be independent of the secret input; this property is known as noninterference [2], [3]. Noninterference is of course desirable whenever it can be achieved, and it has been found possible to guarantee it through the use of type systems (see, for example, [4], [5]). But it is more typical that noninterference is too strong, because some leakage of sensitive information is unavoidable in practice.

To illustrate, let us consider several motivating examples.

- A password checker stores a secret password, which must not be leaked, but when the checker rejects an incorrect guess it thereby reveals that the secret password differs from the guess.
- The Crowds anonymity protocol [6] enables anonymous communication with a server through randomized forwarding within the “crowd” of users. But if some crowd members are collaborators that report users that send messages to them, then some information about the initiator of the message is leaked.
- In a statistical database, the database entries may be considered to be confidential, but the results of statistical queries (e.g. average salary) are made public, revealing some information about the confidential entries. Noise may be added to query results in order to control the leakage, as in differential privacy [7].
- In typical implementations, the time required to do an RSA decryption depends on the RSA secret key, and these timing leaks can be used to recover the secret key [8]. Notice that this is a case where the choice of system model is crucial, since here we need a system model that includes running time as an observable output.

In each of these examples, noninterference is violated, yet we might sense intuitively that the “amount” of confidential information leaked is small, or perhaps can be made small through some mechanism. This perspective motivates the study of Quantitative Information Flow [QIF], an area that
has seen growing interest over the past decade, including works with a foundational focus [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], works aimed at the development of verification techniques [23], [24], [25], [26], [27], [28], [29], [30], and works analyzing the leakage in real system vulnerabilities [31], [32], [33], [34].

As a first example, if \( X \) and \( Y \) are 32-bit integers, then we would intuitively expect that the C assignment statement

\[
X = Y & \ 0x1ff;
\]

should leak 9 bits out of 32, since the bitwise “and” operation \( & \) masks out the first 23 bits of \( X \). But how should leakage be defined in general? And what security guarantees does leakage analysis offer?

The goal of this tutorial paper is to give a unified presentation of some recent developments in Quantitative Information Flow. It is organized as follows. Section II presents fundamental concepts, including channels as mappings from prior distributions to hyper-distributions, vulnerability, min-entropy leakage, and \( g \)-leakage. Section III then discusses the importance of robust judgments about leakage. Section IV discusses capacity, and Section V discusses robust channel ordering. Finally, Section VI discusses some future directions and concludes.

II. FUNDAMENTAL CONCEPTS OF QIF

Given a confidential value \( X \), its secrecy can be modeled by a prior probability distribution \( \pi \) on its set \( X \) of possible values, which we assume to be finite. For example, a uniform distribution gives \( X \) maximum secrecy, while a point distribution gives it no secrecy at all. We assume that \( \pi \) is known to the adversary \( A \), and it thus reflects the adversary’s prior knowledge about \( X \).

We remark that, philosophically, the distribution \( \pi \) is clear in the case when \( X \) is a randomly-generated string of bits. In contrast, if \( X \) is a secret like my mother’s maiden name, then \( \pi \) can be seen as reflecting \( A \)’s knowledge of the population I come from.

A. Channels are mappings from priors to hyper-distributions

The motivating example mentioned in Section I can all be modeled as information-theoretic channels. A channel \( C \) that (perhaps probabilistically) maps secret input \( X \) to some observable output \( Y \) can be modeled as an information-theoretic channel matrix [35], whose rows show the distribution of outputs corresponding to each possible input. For example,

\[
\begin{array}{cccc}
C & y_1 & y_2 & y_3 & y_4 \\
x_1 & 1/2 & 1/2 & 0 & 0 \\
x_2 & 0 & 1/4 & 1/2 & 1/4 \\
x_3 & 1/2 & 1/3 & 1/6 & 0 \\
\end{array}
\]

A significant special case is a deterministic channel, in which each row contains exactly one 1, with all other entries 0.

Assuming that the adversary \( A \) knows prior \( \pi \) and channel \( C \), what does the channel’s output \( Y \) reveal about \( X \)? It turns out that the effect of \( C \) is to map the prior distribution \( \pi \) to a distribution on posterior distributions, as we now illustrate with an example.

Given prior \( \pi = (1/4, 1/2, 1/4) \), we form the joint matrix \( J \) by multiplying each row of \( C \) by the prior probability:

\[
\begin{array}{cccccc}
J & y_1 & y_2 & y_3 & y_4 \\
x_1 & 1/8 & 1/8 & 0 & 0 \\
x_2 & 0 & 1/8 & 1/4 & 1/8 \\
x_3 & 1/8 & 1/12 & 1/24 & 0 \\
\end{array}
\]

Summing the columns of \( J \) gives the marginal distribution \( p_Y = (1/4, 1/3, 7/24, 1/8) \). Each possible value of \( Y \) gives rise to a posterior distribution on \( X \), by Bayesian updating; these can be calculated by normalizing the columns of \( J \):

\[
\begin{array}{cccc}
\pi, C & X_{1|3} & X_{2|2} & X_{3|13} & X_{4|14} \\
x_1 & 1/2 & 3/8 & 0 & 0 \\
x_2 & 0 & 3/8 & 6/7 & 1 \\
x_3 & 1/2 & 1/4 & 1/7 & 0 \\
\end{array}
\]

These posterior distributions reflect the knowledge about \( X \) obtained from each of the four possible output values. For instance, the posterior distribution \( p_{X|y_4} \) shows that on output \( y_4 \), the adversary learns that \( X \) must be \( x_2 \). Notice that because we have a distribution \( (p_Y) \) on the four output values, we also have a distribution on the posterior distributions.

Finally, we observe that the output labels do not matter; all that matters is the distribution on posterior distributions; we call this a hyper-distribution [16], and denote it by \( [\pi, C] \):

\[
\begin{array}{cccc}
\pi, C & X_{1} & X_{2} & X_{3} \\
x_1 & 1/2 & 3/8 & 0 & 0 \\
x_2 & 0 & 3/8 & 6/7 & 1 \\
x_3 & 1/2 & 1/4 & 1/7 & 0 \\
\end{array}
\]

Figure 1 shows a graphical representation (using barycentric coordinates) of the prior \( \pi \) and the hyper-distribution \( [\pi, C] \), showing how channel \( C \) “explodes” the prior \( \pi \) into four posterior distributions.\(^1\)

In conclusion, the information-theoretic essence of \( C \) is a mapping from priors \( \pi \) to hyper-distributions \( [\pi, C] \).

Note that this implies that classic channel matrices have structural redundancies: labels of outputs, columns that are multiples of one another, and all-zero columns, which are irrelevant with respect to (information-theoretic) leakage.\(^2\) Eliminating these redundancies gives abstract channels, as studied in [20].

\(^1\)Interestingly, if the four posterior distributions are weighted according to their probabilities \((1/4, 1/3, 7/24, 1/8)\), then the center of mass is located at the prior \( \pi \).

\(^2\)All-zero columns represent outputs that never occur, while columns that are multiples of one other give rise to the same posterior distribution.
One might expect that the posterior entropy $H(X|Y)$ of 56 bits would imply that $X$ is safe from discovery by the adversary. Yet $1/8$ of the time the adversary learns $X$ exactly!

C. Vulnerability

For confidentiality, it seems more useful to measure leakage based on $X$’s vulnerability [15] to be guessed correctly by the adversary in one try. The prior vulnerability is simply the largest probability in $π$:

$$V[π] = \max_x π_x$$

and the posterior vulnerability is

$$V[π, C] = \sum_y p(y) V[pX|y]$$

which is the average vulnerability in the hyper-distribution. It is also the complement of the Bayes risk.3

A strength of vulnerability is its clear operational significance with respect to confidentiality. Indeed we can understand $V[π]$ as an optimal adversary $A$’s probability of winning the following game, in which a value $x$ is sampled according to $π$ and $A$ tries to guess it:

$$x \leftarrow π;$$
$$w \leftarrow A(π);$$
if $w = x$ then **win** else **lose**

Similarly, $V[π, C]$ is an optimal adversary $A$’s probability of winning the following game, in which $A$ is also given an output $y$ sampled according to row $x$ of $C$:

$$x \leftarrow π;$$
$$y \leftarrow C_x;$$
$$w \leftarrow A(π, C, y);$$
if $w = x$ then **win** else **lose**

D. Min-entropy leakage

It is natural to quantify leakage in terms of the prior and posterior vulnerabilities, $V[π]$ and $V[π, C]$. However, there are a number of plausible ways that we might define the leakage $L(π, C)$, including

- multiplicative: $V[π, C]/V[π]$  
- “logged” multiplicative: $\log(V[π, C]/V[π])$  

Notice that the two multiplicative definitions focus on the relative difference between the posterior and prior vulnerabilities (with “logged” just changing the scale), while the additive definition focuses on the absolute difference.

3A more conservative definition [37], [38] uses instead worst-case posterior vulnerability, taking the maximum of the vulnerabilities of the distributions in the hyper-distribution. But this has the drawback of being highly sensitive to a channel’s worst output, even if that output is very unlikely. For instance it judges a password checker to be as bad as a channel that always leaks the entire secret.
While it is not clear that any of these definitions is canonical, a pleasant fact is that if we are just interested in comparing the leakage caused by two channels $A$ and $B$, then all of these definitions give the same answer—we always get

$$\mathcal{L}(\pi, A) \leq \mathcal{L}(\pi, B) \iff V[\pi, A] \leq V[\pi, B].$$

In [15], min-entropy leakage is defined using the “logged” multiplicative definition. The reason for the name is that Rényi’s min-entropy [39] is the negative logarithm of the vulnerability:

$$H_\infty(X) = -\log V[\pi].$$

So, since

$$\mathcal{L}(\pi, C) = \log(V[\pi, C]/V[\pi])$$

$$= \log V[\pi, C] - \log V[\pi]$$

$$= (\log V[\pi]) - (\log V[\pi, C])$$

we see that $\mathcal{L}(\pi, C)$ is equivalently the difference between the prior and posterior min-entropy of $X$.

An interesting property of min-entropy leakage is that $\mathcal{L}(\pi, C) = 0$ whenever the adversary’s best guess is unaffected by the output $y$. This can sometimes be surprising, as in the following example, which illustrates the so-called base-rate fallacy. Suppose that $C$ is the channel matrix of a good, but imperfect, test for cancer:

<table>
<thead>
<tr>
<th>$C$</th>
<th>positive</th>
<th>negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>$9/10$</td>
<td>$1/10$</td>
</tr>
<tr>
<td>no cancer</td>
<td>$1/10$</td>
<td>$9/10$</td>
</tr>
</tbody>
</table>

Moreover, suppose that for the population under consideration (say, age 40–50, no symptoms, no family history) the prior $\pi$ is heavily biased towards “no cancer”:

$$\pi = (1/100, 99/100).$$

Then, although the channel might appear to be quite reliable, we find that the min-entropy leakage is 0. For we find that the hyper-distribution $[\pi, C]$ is

<table>
<thead>
<tr>
<th>$[\pi, C]$</th>
<th>$27/250$</th>
<th>$223/250$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancer</td>
<td>$1/12$</td>
<td>$891/892$</td>
</tr>
<tr>
<td>no cancer</td>
<td>$11/12$</td>
<td>$891/892$</td>
</tr>
</tbody>
</table>

We see that a positive test result increases the probability of cancer from $1/100$ to $1/12$, while a negative test result decreases it to $1/892$. Nevertheless, $C$’s output is useless in winning the second game above, since $A$’s best guess is always “no cancer”. And we see that

$$V[\pi, C] = 27/250 \cdot 11/12 + 223/250 \cdot 891/892 = 99/100 = V[\pi],$$

implying that the min-entropy leakage

$$\mathcal{L}(\pi, C) = \log(V[\pi, C]/V[\pi]) = \log 1 = 0.$$

E. Generalizing to $g$-vulnerability and $g$-leakage

While vulnerability is clearly a fundamental security metric, it is certainly not appropriate in all scenarios—for instance, in the context of password checker, the adversary might be allowed five tries before being locked out. For this reason, [18] introduced $g$-vulnerability $V_g$, which parameters vulnerability with a gain function $g$ that can model diverse operational scenarios, including those where the adversary gains by guessing the secret partially, approximately, or in $k$ tries, or where there is a penalty for incorrect guesses.

In each scenario, there will be some set $\mathcal{W}$ of guesses (or actions) that the adversary could make about the secret, and for any guess $w$ and secret value $x$, there will be some gain $g(w, x)$ that the adversary gets by having chosen $w$ when the secret’s actual value was $x$. Formally, $g : \mathcal{W} \times X \rightarrow [0, 1]$, where $\mathcal{W}$ is a finite, non-empty set.

Given a gain function $g$, the prior $g$-vulnerability is defined as the maximum expected gain over all possible guesses:

$$V_g[\pi] = \max_w \sum_x \pi_x g(w, x).$$

The posterior $g$-vulnerability is then defined as before:

$$V_g[\pi, C] = \sum_y p(y) V_g[p_X|y].$$

And then we can define $g$-leakage in terms of the prior and posterior $g$-vulnerabilities, either multiplicatively or additively.

Note that (ordinary) vulnerability is a special case of $g$-vulnerability, corresponding to the identity gain function $g_{id} : X \times X \rightarrow [0, 1]$ given by

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

(Notice that when written as a matrix, $g_{id}$ is the identity matrix.)

But gain functions can do much more. As explained in [18], they can model a wide variety of practical operational 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 some bounded number of tries. They can also model scenarios where there is a penalty for incorrect guesses, or where some parts of the secret are worth more than others.

III. Robustness in QIF Analysis

Using $g$-leakage, we can measure leakage in a rich variety of operational scenarios. But we could worry about the

\footnote{It is, however, of interest to generalize gain functions in various ways, allowing the range to be larger than the interval $[0, 1]$ and allowing the set $\mathcal{W}$ to be countably infinite.}
robustness of our conclusions about leakage. The issue is that the \( g \)-leakage \( L_g(\pi, C) \) depends on both prior \( \pi \), which models the adversary’s prior knowledge about \( X \), and gain function \( g \), which models (among other things) what is valuable to the adversary. Also, we can choose to calculate leakage additively or multiplicatively. How confident can we be about these? Can we minimize our sensitivity to questionable assumptions about \( \pi \) and \( g \)?

There has been considerable interest in achieving robustness in QIF analysis. One important approach is to consider capacity, the maximum leakage over all priors \( \pi \) and/or all gain functions \( g \); we can also consider both multiplicative and additive versions of capacity. A second approach to robustness concerns comparison of channels, aimed at showing that one channel never leaks more than another, regardless of the prior or gain function. We discuss these two approaches in the following sections.

IV. CAPACITY

One important approach to robustness is to abstract away from the prior \( \pi \) and/or the gain function \( g \), considering instead the capacity or maximum leakage over all priors and/or gain functions. We separately consider both multiplicative and additive versions of capacity.

A. Multiplicative capacity

Definition 4.1: The min-capacity of \( C \), denoted \( \mathcal{ML}(C) \), is the maximum min-entropy leakage of \( C \) over all \( \pi \):

\[
\mathcal{ML}(C) = \sup_{\pi} \log(V[\pi, C]/V[\pi]).
\]

Pleasingly, min-capacity satisfies a number of useful theorems.

First, as shown in [40], [31], min-capacity is easy to calculate:

Theorem 4.1: \( \mathcal{ML}(C) \) is the logarithm of the sum of the column maximums of \( C \), and it is always realized on a uniform prior \( \pi \).

Corollary 4.2: \( \mathcal{ML}(C) = 0 \) iff the rows of \( C \) are identical.

Corollary 4.3: For deterministic \( C \), \( \mathcal{ML}(C) \) is the logarithm of the number of feasible output values.

Another important result about min-capacity concerns cascading. Given channel \( A \) from \( X \) to \( Y \) and channel \( B \) from \( Y \) to \( Z \), the cascade [41] is the channel \( AB \) from \( X \) to \( Z \) formed by multiplying channel matrices \( A \) and \( B \). As shown in [38], the min-capacity of a cascade \( AB \) is upper bounded by the min-capacities of \( A \) and \( B \):

Theorem 4.4: \( \mathcal{ML}(AB) \leq \min\{\mathcal{ML}(A), \mathcal{ML}(B)\} \).

Min-capacity is also interesting in terms of its relationship with other leakage measures. Interestingly, min-capacity and Shannon capacity coincide on deterministic channels [42], and min-capacity is an upper bound on Shannon capacity on arbitrary channels [18].

More remarkably, [18] shows that min-capacity is an upper bound on “logged” multiplicative \( g \)-leakage, for every prior and every gain function:

Theorem 4.5 (Miracle): For all priors \( \pi \) and gain functions \( g, \mathcal{ML}(C) \geq \log(V_g[\pi, C]/V_g[\pi]) \).

The Miracle Theorem shows that if we can prove that \( C \)'s min-capacity is small, then we know that its “logged” multiplicative \( g \)-leakage is also small, regardless of the prior \( \pi \) and gain function \( g \).

To illustrate, suppose that the secret is an array \( X \) containing 10-bit, uniformly distributed passwords for 1000 users. Consider the following probabilistic channel \( C \), which leaks some randomly-chosen user’s password:

\[
\begin{align*}
   u &\leftarrow \{0.999\}; \\
   Y &\leftarrow (u, X[u])
\end{align*}
\]

With respect to the uniform prior \( \pi_u \), we have \( V[\pi_u] = 2^{-10000} \) and \( V[\pi_u, C] = 2^{-9990} \), so (using Theorem 4.1) we conclude that \( \mathcal{ML}(C) = 10 \). But this measures the leakage caused by \( C \) in terms of the benefit to an adversary trying to guess the entire array of passwords.

We might instead focus on an adversary \( A \) who simply wants to guess some user’s password, with no preference as to whose it is. To achieve this, we define

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

and

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

The Miracle Theorem assures us that the “logged” multiplicative \( g \)-leakage cannot exceed 10, meaning that the \( g \)-vulnerability cannot increase by more than a factor of \( 2^{10} \). In fact, this is precisely what happens here. We have \( V_g[\pi_u] = 2^{-10} \), since \( A \)'s best choice \textit{a priori} is simply to guess some password \( x \) for some user \( u \), which gives expected gain \( 2^{-10} \). And \( V_g[\pi_u, C] = 1 \), since given the output of \( C \) the adversary can always achieve a gain of 1.

It is important to note, then, that the Miracle Theorem does not give any direct guarantees about the posterior \( g \)-vulnerability; it bounds only the ratio between the posterior and prior \( g \)-vulnerabilities.

B. Additive capacity

In the case of additive capacity under a fixed gain function \( g \) and a universally quantified prior, it is proved in [21] that it challenging to compute the capacity:

Theorem 4.6: When \( g \) is \( g_{id} \) (giving ordinary vulnerability), it is NP-complete to decide whether the additive capacity of \( C \) exceeds a given threshold \( t \).

It should be emphasized that the input here is the channel matrix \( C \), rather than a concise program (which invariably leads to intractability).
Even more interesting is additive capacity when $\pi$ is fixed and $g$ is universally quantified. While one might expect a similar computational hardness result, it is shown in [21] that the additive capacity can be computed efficiently by exploiting some abstract mathematics. For the additive capacity is the supremum over $g$ of $V_0[\pi, C] - V_0[\pi]$, which is equivalent to the supremum over $V_0$ of the difference between the expectations of $V_0$ over $[\pi, C]$ and over $[\pi]$, where $[\pi]$ is understood as a point hyper-distribution. This is strikingly similar to the Kantorovich distance between $[\pi, C]$ and $[\pi]$, except that the latter takes the supremum over all 1-Lipschitz functions from distributions on $X$ to reals. But, with $g$ restricted to 1-spanning gain functions, it $V_0$ is always 1-Lipschitz. Moreover, gain functions are expressive enough that we actually achieve equality:

Theorem 4.7: The additive capacity (over all 1-spanning gain functions) of $C$ under prior $\pi$ is equal to the Kantorovich distance between $[\pi, C]$ and $[\pi]$. This result is exceedingly useful, because by the 1958 Kantorovich-Rubinstein theorem [43], it implies that the additive capacity is also the earth-moving distance between $[\pi, C]$ and $[\pi]$. And that can be computed straightforwardly in time linear in the size of $C$.

As an illustration, consider the channel $C$ discussed in Section II-A. With respect to ordinary vulnerability (which corresponds to $g_{id}$), we can easily calculate the additive leakage:

$$V[\pi] = 1/2$$

and

$$V[\pi, C] = 1/4 \cdot 1/2 + 1/3 \cdot 3/8 + 7/24 \cdot 6/7 + 1/8 \cdot 1 = 5/8,$$

giving additive leakage of $5/8 - 1/2 = 1/8$.

If we are interested in calculating the additive capacity over all 1-spanning gain functions, then by Theorem 4.7 it suffices to calculate the earth-moving distance between $[\pi]$ and $[\pi, C]$. To do this, we first calculate the earth-moving distance between $\pi$ and each of the posterior distributions. For instance, the distance between $\pi$ and $p_{X|Y|g}$ is $5/14$, since we must move $1/4$ units of “earth” from $x_1$ to $x_2$, and $3/28$ units from $x_3$ to $x_2$. Overall we get an additive capacity of

$$1/4 \cdot 1/2 + 1/3 \cdot 3/8 + 7/24 \cdot 5/14 + 1/8 \cdot 1/2 = 1/3.$$ 

Notice that $1/3$ exceeds $1/8$, the additive leakage obtained with $g_{id}$. This is as expected, since $g_{id}$ is a 1-spanning gain function.

V. ROBUST CHANNEL ORDERING

Given channels $A$ and $B$, both taking input $X$, the question of which leaks more will ordinarily depend on the prior and gain function used. As an example, assume that $X$ is a 64-bit unsigned integer, and consider the following channels:

A. $Y = X \oplus Y = -1$
B. $Y = X \mid 0 \times 7$

Both have a min-capacity of 61.0 bits out of 64, since both have a posterior vulnerability of about $1/8$. But we can distinguish them using gain functions.

We can define a gain function $g_3$ that allows the adversary 3 tries. Its guesses $w$ are size-3 subsets of $X$, and

$$g_3(w, x) = \begin{cases} 1, & \text{if } x \in w, \\ 0, & \text{if } x \notin w. \end{cases}$$

Observe that the posterior $g_3$-vulnerability of $B$ is $3/8$, since $B$ reveals all but the last three bits of $X$. In contrast, the posterior $g_3$-vulnerability of $A$ remains about $1/8$, because when $Y \neq -1$, the adversary already knows $X$ exactly, and when $Y = -1$, the adversary can just make 3 stabs in the dark. Hence $g_3$ makes $B$ leak more than $A$.

Alternatively, we can define a gain function $g_{tiger}$ for which making a wrong guess triggers a penalty (say, opening a trap door to a pit of tigers), and which allows the adversary to use the special value $\perp$ to indicate that it chooses not to make a guess. With suitable gain and penalty values, the $g_{tiger}$-leakage of $B$ is 0, because the output of $B$ gives the adversary only a $1/8$ probability of guessing $X$ correctly, making $\perp$ the best choice. In contrast, the $g_{tiger}$-leakage of $A$ is greater than 0, because when $Y \neq -1$, the adversary knows $X$ exactly and can make a guess without fear of any penalty. Hence $g_{tiger}$ makes $A$ leak more than $B$.

The question is then whether there is a robust leakage ordering, allowing us to conclude that one channel never leaks more than another, regardless of the prior or gain function. Such a robust ordering could support a stepwise refinement methodology for constructing secure systems.

It turns out that there is such an ordering [18], [44]:

Definition 5.1: A composition refines $A$, written $A \subseteq_o B$, if there exists a channel $C$ such that $B = AC$.

Note that composition refinement requires that $B$ can be expressed as the cascade of $A$ and $C$, for some “post-processing” channel $C$.

The main interest in composition refinement is its relation to $g$-leakage. First, composition refinement implies a strong $g$-leakage ordering; this can be seen as an analogue of the classic data-processing inequality.9

Theorem 5.1 (Data-processing inequality): If $A \subseteq_o B$ then the $g$-leakage of $B$ never exceeds that of $A$, for any prior $\pi$ and any gain function $g$.

More interestingly, the converse implication holds as well:

8In [18], note that the order of the two channels was reversed.
9To see that this is indeed the $g$-leakage version of the data-processing inequality [35], note that if $B = AC$, where $B$ goes from $X$ to $Z$, $A$ goes from $X$ to $Y$, and $C$ goes from $Y$ to $Z$, then for any prior $\pi$ we have a Markov chain $X \rightarrow Y \rightarrow Z$. The data-processing inequality says that in this case $I(X; Z) \leq I(X; Y)$.
Theorem 5.2 (“Coriaceous”): If the $g$-leakage of $B$ never exceeds that of $A$, for any prior $\pi$ and any gain function $g$, then $A \sqsubseteq_\pi B$.

We hence see that composition refinement is an ordering on channels with both structural and leakage-testing significance.

These fundamental theorems about composition refinement were proved in [18] and [20], using the separating hyperplane lemma [45] to construct the $g$ needed in the contrapositive form of the “coriaceous” direction. However, we later learned that these theorems were in fact discovered already in 1951, by statistician David Blackwell [46].

We conclude this section with some discussion of the structure of channels under composition refinement. We first observe that composition refinement is only a pre-order on channel matrices. For example, consider the following channel matrices:

$$
\begin{array}{ccc}
C & y_1 & y_2 & y_3 \\
x_1 & 1 & 0 & 0 \\
x_2 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
x_3 & \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\
\end{array}
\quad
\begin{array}{ccc}
D & z_1 & z_2 & z_3 \\
x_1 & \frac{1}{5} & 0 & \frac{3}{5} \\
x_2 & \frac{1}{10} & \frac{3}{4} & \frac{3}{20} \\
x_3 & \frac{1}{5} & \frac{1}{2} & \frac{3}{10} \\
\end{array}
$$

It turns out that $C \sqsubseteq_\pi D$ and also $D \sqsubseteq_\pi C$. But this is less significant than it might first appear, because in fact $C$ and $D$ are actually the same abstract channel, since they denote the same mapping from priors to hyper-distributions. (To see this, note that the second and third columns of $C$ are multiples of one another, as are the first and third columns of $D$; because such “similar” columns always produce the same posterior distribution, they may as well be merged.)

In [20], it is shown that composition refinement is anti-symmetric on abstract channels, and hence a partial order.

VI. FUTURE DIRECTIONS AND CONCLUSION

We conclude with a brief discussion of some important future directions.

One important direction is the automated analysis of leakage in systems. A variety of static-analysis techniques have been explored in the literature, based on type systems [23], statistical sampling [27], [28], model checking [24], [25], [32], [30], and abstract interpretation [37], [34]. While considerable progress has been achieved, it remains unclear whether such analyses can be scaled up successfully to large-scale systems.

Differential privacy [7] is a popular approach to controlling the leakage of sensitive information caused by queries to statistical databases. The relationship between differential privacy and QIF has received some study [47], [48], but it would be valuable to achieve a better unification of these two approaches.

Finally, our perspective here has been information theoretic, concerned only with a channel’s mapping from priors to hyper-distributions, and abstracting from details like the names of outputs. These choices are appropriate if we are interested only in the information that a channel provides to the adversary with unbounded computational resources. But of course the resulting conclusions may be overly pessimistic with respect to a resource-constrained adversary.

For example, consider a channel $C$ that takes as input an $n$-bit prime, assumed uniformly distributed. $C$ then randomly chooses an $(n+1)$-bit prime $q$ and outputs $pq$. Notice then that each column of the channel matrix of $C$ contains exactly one non-zero entry—each column is labeled with the product $pq$ of an $n$-bit prime $p$ and an $(n+1)$-bit prime $q$, and has a non-zero entry only in row $p$. This means that the hyper-distribution $[\pi, C]$ contains only point posterior distributions. Hence $V[\pi, C] = 1$, meaning that $C$ leaks $p$ completely. But of course adversary $A$, seeing output label $pq$, needs to factor $pq$ in order to recover $p$. And, under standard assumptions about the difficulty of factorization, a polynomial-time adversary has a negligible probability of succeeding in doing this.

In [20], some preliminary ideas are given about a computational version of posterior $g$-vulnerability. The idea is that the maximization done by $V_g[\pi, C]$ can be described concretely as a strategy $S$ that chooses a best guess $w$ given each channel output $y$; it is then natural to demand that $S$ should be efficiently computable. Developing a theory of computational $g$-leakage would be challenging but interesting, as it might allow QIF to address cryptographic constructions.

ACKNOWLEDGEMENTS

I am deeply grateful to my collaborators, especially Mário Alvim, Miguel Andrés, Nico Bordenabe, Costas Chatzikokolakis, Barbara Espinoza, Boris Köpf, Piotr Mardziel, Annabelle McIver, Zhiyuan Meng, Carroll Morgan, Catuscia Palamidessi, and Marco Stronati, for their many profound contributions to quantitative information flow. I am also grateful for support by the National Science Foundation under grant CNS-1116318, from Digiteo, and from the INRIA équipe associée Princess.

REFERENCES


