# Quantifying Information Flow Using Min-Entropy

*(Invited Paper)*

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

***Abstract*—Quantitative theories of information flow are of growing interest, due to the fundamental importance of protecting confidential information from improper disclosure, together with the unavoidability of "small" leaks in practical systems. But while it is tempting to measure leakage using classic information-theoretic concepts like Shannon entropy and mutual information, these turn out not to provide very satisfactory security guarantees. As a result, several researchers have developed an alternative theory based on Rényi's min-entropy. In this theory, uncertainty is measured in terms of a random variable's vulnerability to being guessed in one try by an adversary; note that this is the complement of the Bayes Risk. In this paper, we survey the main theory of min-entropy leakage in deterministic and probabilistic systems, including comparisons with mutual information leakage, results on min-capacity, results on channels in cascade, and techniques for calculating min-entropy leakage in systems.**

## I. Introduction

One of the most fundamental challenges in computer security is to control the *flow of information*, whether to prevent confidential information from being *leaked*, or to prevent trusted information from being *tainted*. But while it is sometimes possible to stop undesirable information flows completely, it is perhaps more typical that some undesirable flows are unavoidable. For instance an ATM machine that rejects an incorrect PIN thereby reveals that the secret PIN differs from the one that was entered. Similarly, revealing the tally of votes in an election reveals some information about the secret ballots that were cast. More subtly, the amount of *time* taken by a cryptographic operation may be observable by an adversary, and may inadvertently reveal information about the secret key. As a result, the last decade has seen growing interest in *quantitative* theories of information flow [1], [2], [3], [4], [5], which allow us to talk about "how much" information is leaked and (perhaps) allow us to tolerate "small" leaks.

*Information theory* [6], [7] offers a very general setting for such theories in its notion of a *channel*. Channels do not rely on any explicit notion of "messages"; instead they capture relationships between system *inputs* and *outputs* through a *channel matrix*, which gives the conditional probability of each possible output, given each possible input. (Note that "output" should be understood to encompass any aspects of the system's behavior that are observable to an adversary, possibly including time.) Within this framework, it is natural to quantify *leakage* of confidential information based on the extent to which a channel's output helps an adversary to determine the secret input.

We might also wish to quantify *integrity*. But while there is important recent work [8], [9] that approaches quantitative integrity by considering the extent to which trusted outputs are "influenced" or "tainted" by untrusted inputs, appropriate metrics for integrity seem less clear than for confidentiality. For this reason, we restrict our attention to quantitative confidentiality in the rest of this paper.

Given the stature of information theory, it is tempting to measure leakage of confidential information using classic concepts like *Shannon entropy* and *mutual information*. But its celebrated results, like the noisy channel coding theorem [6], really address a different sort of question than what concerns us here. They address the question of how quickly a channel can *reliably* transmit a stream of inputs—with respect to confidentiality, in contrast, the crucial question is whether there is a significant risk that the secret input *might* be guessed from the output. In 1956, Claude Shannon himself warned of the danger of applying information theory indiscriminately:

> While we feel that information theory is indeed a valuable tool in providing fundamental insights into the nature of communication problems and will continue to grow in importance, it is certainly no panacea for the communications engineer or, *a fortiori*, for anyone else. Seldom do more than a few of nature's secrets give way at one time. [10]

Indeed we will see that, in the context of confidentiality against an adversary trying to guess the secret, measuring leakage using mutual information does not result in very satisfactory security guarantees. For this reason, a number of researchers have developed an alternative theory of quantitative information flow based on Rényi's *min-entropy* [11]. In this theory, uncertainty is measured in terms of a random variable's *vulnerability* to being guessed in one try by an adversary; note that this is the complement of the Bayes Risk.

The goal of this paper is to present the main theory of min-entropy leakage, collecting results from the recent literature, and also to point out some directions for future research. Specifically, Section II presents the basic definitions of mutual information and min-entropy leakage in deterministic and probabilistic channels, comparing their suitability with respect to confidentiality. Section III presents basic results about min-capacity, which is the maximum min-entropy leakage over all *a priori* distributions. Section IV gives a more speculative discussion of some deeper questions about the relationship between min-entropy leakage and mutual information. Section V describes results on the min-entropy leakage of a cascade of channels. Section VI surveys some recent work on calculating (or approximating) the min-entropy leakage of probabilistic or deterministic systems. Finally, Section VII concludes.

## II. Basic Definitions

In this section, we begin by recalling important concepts of information theory [6], [12], [13], [14], [15], [7], such as Shannon entropy and mutual information, and then present the motivation and basic definitions of the min-entropy measure of information leakage.

A *channel* is a triple $(\mathcal{S}, \mathcal{O}, C_{\mathcal{SO}})$, where $\mathcal{S}$ is a finite set of secret input values, $\mathcal{O}$ is a finite set of observable output values, and $C_{\mathcal{SO}}$ is an $|\mathcal{S}| \times |\mathcal{O}|$ matrix, called the *channel matrix*, such that $C_{\mathcal{SO}}[s, o]$ is the conditional probability of obtaining output $o$ given that the input is $s$. Formally, each entry of $C_{\mathcal{SO}}$ is a real number between 0 and 1, and each row sums to 1. An important special case is a *deterministic channel*, in which each input produces a unique output. In terms of $C_{\mathcal{SO}}$, this means that each entry is either 0 or 1, and each row contains exactly one 1.

Any *a priori* distribution $P_S$ on $\mathcal{S}$ determines a random variable $S$. Moreover, $P_S$ and $C_{\mathcal{SO}}$ determine a joint probability matrix $P^\star_{SO}$ on $\mathcal{S} \times \mathcal{O}$, where

$$P^\star_{SO}[s, o] = P_S[s] C_{\mathcal{SO}}[s, o]. \tag{1}$$

It can be shown that $P^\star_{SO}$ is the unique joint distribution that recovers the *a priori* $P_S$ by marginalization:

$$P^\star_S[s] = \sum_{o \in \mathcal{O}} P^\star_{SO}[s] = P_S[s]$$

and recovers the conditional probabilities in $C_{\mathcal{SO}}$, whenever they are defined:

$$P^\star_{O|S}[o|s] = \frac{P^\star_{SO}[s, o]}{P^\star_S[s]} = C_{\mathcal{SO}}[s, o].$$

Also $P^\star_{SO}$ gives a marginal distribution on $\mathcal{O}$:

$$P^\star_O[o] = \sum_{s \in \mathcal{S}} P^\star_{SO}[s, o]$$

giving a random variable $O$.

We *quantify* the amount of information that flows from $S$ to $O$ by considering an adversary $\mathcal{A}$ who wishes to find out the value of $S$. It is natural to measure information leakage by comparing $\mathcal{A}$'s "uncertainty" about $S$ before and after seeing the value of $O$, using the equation

leakage = initial uncertainty – remaining uncertainty.

### A. Measuring leakage using mutual information

Until recently, the literature on quantitative information flow (for example, [1], [16], [3], [17]) generally defined "initial uncertainty" using *Shannon entropy* [6]:

$$H(S) = -\sum_{s \in \mathcal{S}} P_S[s] \log P_S[s]$$

and "remaining uncertainty" using *conditional Shannon entropy*:

$$H(S|O) = \sum_{o \in \mathcal{O}} P_O[o] H(S|o).$$

This leads to defining leakage as *mutual information*:

$$\text{leakage} = H(S) - H(S|O) = I(S; O).$$

In the special case of a *deterministic* channel, note that we have $H(O|S) = 0$, since the value of $O$ is determined by the value of $S$. Using the fact that mutual information is symmetric, this gives a simpler formula for leakage in a deterministic channel:

$$I(S; O) = I(O; S) = H(O) - H(O|S) = H(O).$$

A critical question about any leakage measure, however, is whether it gives good operational security guarantees. In particular we would like to know whether the measure of remaining uncertainty accurately reflects the threat to $S$, given $O$. Of course, answering this question depends on our model of what the adversary can do. Here we focus on an adversary $\mathcal{A}$ who tries to *guess* the value of $S$ in a brute-force manner, using the value of $O$ as a "clue".

For $H(S|O)$, Massey's *guessing entropy* bound [18] shows that $G(S|O)$, the expected number of guesses required to guess $S$ given $O$, grows exponentially with $H(S|O)$. This is the sort of operational security guarantee that we seek in trying to justify a measure of quantitative information flow. There is, however, a serious weakness to this particular bound; it turns out that $G(S|O)$ can be arbitrarily high even when $S$ is highly vulnerable to being guessed by $\mathcal{A}$ in *one* try.

To illustrate, consider the following example from [19]:

```
if (S % 8 == 0)
    O = S;
else
    O = 1;
```
(2)

This program copies $S$ to $O$ if $S$ is a multiple of 8, and otherwise sets $O$ to 1. Assume that $S$ is a uniformly-distributed 64-bit unsigned integer, $0 \le S < 2^{64}$, so that

the initial uncertainty $H(S) = 64$. For this program, the mutual information leakage is

$$I(S;O) = H(O) = 2^{61}2^{-64}\log 2^{64} + \frac{7}{8}\log\frac{8}{7} \approx 8.17,$$

which means that the remaining uncertainty $H(S|O) \approx 55.83$. Here we see that adversary $\mathcal{A}$'s expected probability of guessing $S$ in one try exceeds $\frac{1}{8}$, since $S$ is leaked completely whenever $O \neq 1$. But nevertheless the guessing entropy is high, since nothing is leaked when $O = 1$ (except the fact that the last three bits are not all 0):

$$G(S|O) = \frac{1}{8}\cdot 1 + \frac{7}{8}\cdot\frac{1}{2}\cdot(\frac{7}{8}2^{64}+1) \approx 2^{62.6}.$$

It is instructive to compare program (2) with

$$O = S \;\&\; 0777; \tag{3}$$

which simply copies the last 9 bits of $S$ into $O$. The mutual information leakage of program (3) is 9, making it *worse* than program (2), even though it gives $\mathcal{A}$ a probability of guessing $S$ in one try of only $2^{-55}$, since the first 55 bits of $S$ remain completely unknown.

### B. Measuring leakage using min-entropy

In view of the unsatisfactory security guarantees given by mutual information leakage, it was proposed in [19] to define "uncertainty" in terms of the *vulnerability* of $S$ to being guessed correctly *in one try* by $\mathcal{A}$. If we make the worst-case assumption that $\mathcal{A}$ knows $P_S$ and $C_{\mathcal{SO}}$, then the *a priori* vulnerability is

$$V(S) = \max_{s\in\mathcal{S}} P_S[s]$$

and the *a posteriori* vulnerability is

$$\begin{aligned} V(S|O) &= \sum_{o\in\mathcal{O}} P^\star_O[o] \max_{s\in\mathcal{S}} P^\star_{S|O}[s|o] \\ &= \sum_{o\in\mathcal{O}} \max_{s\in\mathcal{S}} P^\star_{SO}[s,o] \\ &= \sum_{o\in\mathcal{O}} \max_{s\in\mathcal{S}} (P_S[s]C_{\mathcal{SO}}[s,o]). \end{aligned}$$

Note that $V(S|O)$ is the complement of *Bayes risk*, which was used as an anonymity metric in [20].

Adversary $\mathcal{A}$ can achieve these vulnerabilities with a simple guessing strategy. *A priori*, $\mathcal{A}$ should guess some $s$ that maximizes $P_S[s]$. *A posteriori*, $\mathcal{A}$ should guess some $s$ that maximizes $P^\star_{SO}[s,o]$; this corresponds to looking down column $o$ of the joint matrix $P^\star_{SO}$. Notice that the time required by $\mathcal{A}$ is linear in the size of $C_{\mathcal{SO}}$. Of course, one can imagine scenarios where $C_{\mathcal{SO}}$ is extremely large. Consider for instance a channel whose input is a uniformly-distributed 100-digit prime $p$, and whose output is $pq$, where $q$ is a uniformly-distributed 101-digit prime.[1] In this case, $V(S|O) = 1$, since each column of $P^\star_{SO}$ has a unique nonzero entry—but it is not easy for $\mathcal{A}$ to find it! Thus we emphasize that vulnerability is *information theoretic*, rather than *computational*.

We convert from vulnerability to uncertainty by taking the negative logarithm, giving Rényi's *min-entropy* [11]. Our definitions, then, are

- initial uncertainty: $H_\infty(S) = -\log V(S)$
- remaining uncertainty: $H_\infty(S|O) = -\log V(S|O)$

Finally, we define the *min-entropy leakage from $S$ to $O$*, denoted $\mathcal{L}_{SO}$, to be

$$\begin{aligned} \mathcal{L}_{SO} &= H_\infty(S) - H_\infty(S|O) \\ &= -\log V(S) - (-\log V(S|O)) \\ &= \log\frac{V(S|O)}{V(S)}. \end{aligned}$$

Thus min-entropy leakage is the logarithm of the factor by which knowledge of $O$ increases the one-guess vulnerability of $S$.

The last formula for $\mathcal{L}_{SO}$ can be rewritten directly in terms of the joint matrix $P^\star_{SO}$:

$$\mathcal{L}_{SO} = \log\frac{\sum_o \max_s P^\star_{SO}[s,o]}{\max_s \sum_o P^\star_{SO}[s,o]} \tag{4}$$

in which the numerator and denominator differ only in the order of the sum and maximum operations. Notice that the numerator is the sum of the column maximums, while the denominator is the maximum of the row sums.

Revisiting program (2), we find that its min-entropy leakage is 61.00, reflecting the fact that $V(S|O) \approx \frac{1}{8}$. In contrast, for program (3) the min-entropy leakage is 9, reflecting the fact that $V(S|O) = 2^{-55}$. Thus min-entropy leakage judges program (2) to have far greater leakage than program (3).

We remark that conditional min-entropy $H_\infty(S|O)$ was not defined by Rényi [11], and there is no universally agreed-upon definition [21, p. 16], [22, section 2.4]. Our definition above is equivalent to the following definition from [22]:

$$H_\infty(S|O) = -\log\sum_{o\in\mathcal{O}} P^\star_O[o]\max_{s\in\mathcal{S}} P^\star_{S|O}[s|o].$$

In contrast, the definition from [21] is equivalent to

$$H_\infty(S|O) = -\sum_{o\in\mathcal{O}} P^\star_O[o]\log\max_{s\in\mathcal{S}} P^\star_{S|O}[s|o].$$

Repositioning the "log" in the definition makes a big difference—in fact, if we used the latter definition of $H_\infty(S|O)$, it turns out that the min-entropy leakage of program (2) would be 8.17 bits, the *same* as its mutual information leakage.

Because min-entropy leakage is defined by

$$\mathcal{L}_{SO} = H_\infty(S) - H_\infty(S|O)$$

[1] Note that, by the prime number theorem, the number of rows in the channel matrix exceeds $10^{97}$.

it might seem tempting to denote it as $I_\infty(S;O)$, by analogy with mutual information. We do not adopt this notation, however, because of the crucial difference that mutual information is *symmetric*:

$$I(S;O) = I(O;S)$$

while min-entropy leakage is not:

$$\mathcal{L}_{SO} \neq \mathcal{L}_{OS}, \text{ in general.}$$

In thinking about the definition of min-entropy leakage, the fact that it is defined based on *one-guess* vulnerability may seem questionable, since there are certainly scenarios in which $\mathcal{A}$ would be able to make *multiple* guesses. It is instructive to compare program (2) with

$$O = S \mid 07; \tag{5}$$

which copies the first 61 bits of $S$ into $O$, masking out the last 3 bits. The min-entropy leakage of program (5) is 61 bits, which is the same as that of program (2). Yet one might feel that program (5) is clearly more dangerous, since it always allows $\mathcal{A}$ to determine $S$ within 8 guesses, while program (2) reveals almost nothing about $S$ seven-eighths of the time. But suppose that $\mathcal{A}$ is in a scenario where it is reluctant to guess, because making a *wrong* guess would trigger an alarm. In that case, program (2) might reasonably be judged to be worse because, whenever $O \neq 1$, $\mathcal{A}$ *knows* the value of $S$ exactly.

Perhaps the best conclusion is that the appropriateness of any measure of information leakage always needs to be assessed with respect to a specific adversary model. And, following Shannon's warning, we should certainly not view min-entropy leakage as a panacea. Still, one-guess vulnerability seems to be a basic enough concern to support useful conclusions in a wide variety of scenarios. For instance, useful bounds can often be obtained simply by observing that allowing $i$ guesses at most increases the vulnerability by a factor of $i$. Thus if we write $V_i$ for $i$-guess vulnerability, we have

$$V_i(S) \leq iV(S)$$

and

$$V_i(S|O) \leq iV(S|O).$$

Hence if the one-guess vulnerability is negligible, then the $i$-guess vulnerability will also be negligible, assuming that $i$ is not too large.

## III. Min-capacity

An important notion in information theory is *channel capacity*, which is the maximum leakage over all possible *a priori* distributions. When we measure leakage using mutual information, we will use the name *Shannon capacity*, and when we measure leakage using min-entropy, we will use the name *min-capacity* and the notation $\mathcal{ML}(C_{\mathcal{SO}})$. While calculating the Shannon capacity of a channel matrix is in general difficult, calculating the min-capacity is easy, as it is just the logarithm of the sum of the column maximums of $C_{\mathcal{SO}}$ [23], [24]:

*Theorem 3.1:* For any channel matrix $C_{\mathcal{SO}}$,

$$\mathcal{ML}(C_{\mathcal{SO}}) = \log \sum_{o \in \mathcal{O}} \max_{s \in \mathcal{S}} C_{\mathcal{SO}}[s,o]$$

and it is realized on a uniform distribution on $\mathcal{S}$ (and possibly on other distributions as well).

*Proof:* Using the formulas in Section II we have

$$\begin{aligned}
\mathcal{L}_{SO} &= \log \frac{V(S|O)}{V(S)} \\
&= \log \frac{\sum_{o \in \mathcal{O}} \max_{s \in \mathcal{S}} (P_S[s] C_{\mathcal{SO}}[s,o])}{\max_{s \in \mathcal{S}} P_S[s]} \\
&\leq \log \frac{\sum_{o \in \mathcal{O}} \max_{s \in \mathcal{S}} C_{\mathcal{SO}}[s,o] (\max_{s \in \mathcal{S}} P_S[s])}{\max_{s \in \mathcal{S}} P_S[s]} \\
&= \log \textstyle\sum_{o \in \mathcal{O}} \max_{s \in \mathcal{S}} C_{\mathcal{SO}}[s,o]
\end{aligned}$$

The upper bound is realized when $S$ is uniformly distributed. It can also be realized on nonuniform distributions, provided that some proper subset of the rows of $C_{\mathcal{SO}}$ includes at least one maximum from each column. ■

Min-capacity is of interest because it frees us from the need to know the *a priori* $P_S$, focusing more simply on the "worst-case" leakage of the channel. Indeed, min-entropy leakage can sometimes be surprisingly dependent on the *a priori* distribution. Consider the following example, which illustrates the so-called *base-rate fallacy*. Suppose that $C_{\mathcal{SO}}$ is the channel matrix of an imperfect test for cancer:

| | *positive* | *negative* |
|---|---|---|
| *cancer* | 0.90 | 0.10 |
| *no cancer* | 0.07 | 0.93 |

Moreover, suppose that for the population under consideration (say, age 40–50, no symptoms, no family history) the *a priori* distribution is

$$P_S[\textit{cancer}] = 0.008 \quad P_S[\textit{no cancer}] = 0.992$$

Then, although the channel might appear to be quite reliable, we find that the min-entropy leakage is 0. For if we calculate the min-entropy leakage $\mathcal{L}_{SO}$ using equation (4), we find that the joint matrix $P^\star_{SO}$ is

| | *positive* | *negative* |
|---|---|---|
| *cancer* | 0.00720 | 0.00080 |
| *no cancer* | 0.06944 | 0.92256 |

Hence the sum of the column maximums coincides with the maximum of the row sums, since both column maximums occur in the *no cancer* row. Operationally, this reflects the fact that $\mathcal{A}$ should guess *no cancer*, regardless

of whether the test was *positive* or *negative*. (In particular, $P^\star[\textit{cancer}|\textit{positive}] \approx 0.094$, which is much greater than $P_S[\textit{cancer}] = 0.008$, but still much less than 0.500.) In general, even if some $o$ increases or decreases the probability of some $s$, there is no min-entropy leakage unless the changed probability causes $\mathcal{A}$ to make a different guess.[2]

In contrast, min-capacity behaves more straightforwardly. For instance, it is an easy corollary to Theorem 3.1 that the min-capacity of $C_{\mathcal{SO}}$ is 0 iff $C_{\mathcal{SO}}$ has no leakage at all [24]:

*Corollary 3.2:* $\mathcal{ML}(C_{\mathcal{SO}}) = 0$ iff the rows of $C_{\mathcal{SO}}$ are identical.

As another corollary, the min-capacity of a *deterministic* channel is just the logarithm of the number of feasible outputs. Interestingly, this is also the Shannon capacity [19]:

*Theorem 3.3:* If $C_{\mathcal{SO}}$ is deterministic, then its min-capacity and Shannon capacity coincide, with both equal to $\log|\mathcal{O}|$ (assuming that every element of $\mathcal{O}$ is feasible).

*Proof:* Assume that every element of $\mathcal{O}$ is feasible. By Theorem 3.1, the min-capacity of $C_{\mathcal{SO}}$ is the logarithm of the sum of its column maximums. Since each entry of $C_{\mathcal{SO}}$ is 0 or 1, this is just $\log|\mathcal{O}|$.

Moreover, the Shannon capacity of a deterministic channel is the maximum value of $H(O)$ over all *a priori* distributions $P_S$. This maximum is $\log|\mathcal{O}|$, since $O$ has $|\mathcal{O}|$ feasible values and we can construct a $P_S$ that makes them all equally likely. (Note that this will typically not be a uniform distribution on $\mathcal{S}$.) ∎

Hence it does not matter for deterministic channels whether we measure capacity using min-capacity or Shannon capacity. However, as we will see in Section IV, this coincidence does not carry over to the general case of probabilistic channels.

One useful way to bound min-capacity is by finding a *factorization* of the channel matrix, which corresponds to decomposing a channel into the *cascade* [12] of two channels.

*Theorem 3.4:* If channel matrix $C_{\mathcal{SO}} = C_{\mathcal{ST}}C_{\mathcal{TO}}$, where $C_{\mathcal{ST}}$ and $C_{\mathcal{TO}}$ are channel matrices, then the min-capacity of $C_{\mathcal{SO}}$ is at most $\log|\mathcal{T}|$, the logarithm of the "inner dimension" of the matrix product.

Intuitively, viewing $C_{\mathcal{SO}}$ as a pipe, the number of rows of $C_{\mathcal{SO}}$ is the size of the input end, and the number of columns of $C_{\mathcal{SO}}$ is the size of the output end. In the case where $C_{\mathcal{SO}} = C_{\mathcal{ST}}C_{\mathcal{TO}}$, where the inner dimension is small, we can view the pipe as being narrow in the middle, which prevents it from leaking very much. Theorem 3.4 is proved in [24], where it is used to bound the min-entropy leakage of timing attacks against public-key cryptosystems implemented using the defenses of *blinding* and *bucketing*.

[2]This example also illustrates the fact that min-entropy leakage is not symmetric. For if we view $P^\star_{SO}$ as a channel from $O$ to $S$, we find that $\mathcal{L}_{OS} \approx 0.01 > 0 = \mathcal{L}_{SO}$. Intuitively, $S$ helps in guessing $O$ because, in the rare case when $S$ is *cancer*, $\mathcal{A}$'s best guess for $O$ changes from *negative* to *positive*.

## IV. The Relationship Between Min-Entropy Leakage and Mutual Information

It is valuable to study the mathematical properties of min-entropy leakage, in particular by studying the relationship between min-entropy leakage, $\mathcal{L}_{SO}$, and leakage measured by mutual information, $I(S;O)$, or by other measures. Understanding such relationships is important because (as Shannon warned) it is misguided to expect *any* single measure to be ideal in all situations.[3]

One question of particular interest concerns the relationship between Shannon capacity and min-capacity. As shown in Theorem 3.3, these capacities *coincide* for deterministic channels, but this coincidence fails for probabilistic channels. To demonstrate, let $C_{\mathcal{SO}}$ be a square channel matrix with $2^{64} - 2^{54} + 1$ rows and columns, whose entries are all $2^{-64}$ except on the main diagonal, where the entries are all $2^{-10}$:

$$C_{\mathcal{SO}} = \begin{pmatrix} 2^{-10} & 2^{-64} & 2^{-64} & \dots & 2^{-64} \\ 2^{-64} & 2^{-10} & 2^{-64} & \dots & 2^{-64} \\ 2^{-64} & 2^{-64} & 2^{-10} & \dots & 2^{-64} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2^{-64} & 2^{-64} & 2^{-64} & \dots & 2^{-10} \end{pmatrix}$$

Note that this is a valid channel matrix, since its row sums are $2^{-10}+(2^{64}-2^{54})\cdot 2^{-64} = 1$. Because $C_{\mathcal{SO}}$ is symmetric, we can easily calculate its Shannon capacity using a formula in [7]:

$$\begin{aligned} \text{capacity} &= \log|\mathcal{O}| - H(\text{row}) \\ &= \log(2^{64} - 2^{54} + 1) \\ &\quad -(2^{-10}\cdot 10 + (2^{64} - 2^{54})\cdot 2^{-64}\cdot 64) \\ &\approx 0.05132 \end{aligned}$$

Thus the Shannon capacity of $C_{\mathcal{SO}}$ is about one-twentieth of a bit.

Calculating the min-capacity is even easier, as it is just the log of the sum of the column maximums, all of which are $2^{-10}$:

$$\text{min-capacity} = \log(2^{-10}\cdot(2^{64} - 2^{54} + 1)) \approx 53.99859$$

The min-capacity of approximately 54 bits reflects the fact that, under a uniform *a priori* distribution, $C_{\mathcal{SO}}$ increases the vulnerability from about $2^{-64}$ to $2^{-10}$. (Notice that $P^\star[s_i|o_i] = 2^{-10}$.) That is, $C_{\mathcal{SO}}$ enables $\mathcal{A}$ to guess a (roughly) 64-bit secret with probability $1/1024$. Thus this example shows how misleading Shannon capacity can be as a measure of security risk.

[3]For instance, while min-entropy leakage has clear operational significance for quantitative *confidentiality*, it is not obvious that it is a useful measure of quantitative *integrity*.

Generalizing from this example, one can show that for any $k \geq 3$, there exists a $k \times k$ channel matrix whose min-capacity exceeds its Shannon capacity by a factor of $k/2$.

Another important property comes from the Santhi-Vardy bound [25]:

$$P_e \leq 1 - 2^{-H(S|O)}.$$

Here $P_e$ is the adversary's *probability of error* in guessing $S$ given $O$; it is thus the same as $1 - V(S|O)$. Hence we can rewrite the Santhi-Vardy bound to $2^{-H(S|O)} \leq V(S|O)$. From this it follows that

$$H(S|O) \geq H_\infty(S|O), \tag{6}$$

showing that the remaining uncertainty under Shannon entropy is at most the remaining uncertainty under min-entropy.

Turning to leakage, assume first a uniform *a priori* distribution on $\mathcal{S}$. Then $H(S) = H_\infty(S)$, so

$$I(S;O) = H(S) - H(S|O) \leq H_\infty(S) - H_\infty(S|O) = \mathcal{L}_{SO},$$

so $I(S;O)$ cannot exceed $\mathcal{L}_{SO}$ in that case. But if we consider an arbitrary *a priori* distribution then when we compare $H(S) - H(S|O)$ to $H_\infty(S) - H_\infty(S|O)$, we have both $H(S) \geq H_\infty(S)$ and $H(S|O) \geq H_\infty(S|O)$, so it is not immediate that any ordering holds in general.

Indeed, there are cases where $I(S;O)$ exceeds $\mathcal{L}_{SO}$. Consider the channel matrix

$$C_{\mathcal{SO}} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \end{pmatrix} \tag{7}$$

Figure 1 plots the conditional vulnerability $V(S|O)$, the vulnerability $V(S)$, the min-entropy leakage $\mathcal{L}_{SO}$, and the mutual information leakage $I(S;O)$ of this channel, each shown as a function of the *a priori* distribution $P_S = (x, 1-x)$. Notice that if $x \leq 1/4$ or $x \geq 2/3$, then the min-entropy leakage is 0. The reason is that min-entropy leakage measures the improvement in $\mathcal{A}$'s ability to guess $S$, given $O$. But when $x \leq 1/4$, $\mathcal{A}$'s best guess is that $S$ is $s_1$, regardless of whether the output is $o_1$, $o_2$, or $o_3$. (Similarly, when $x \geq 2/3$, $\mathcal{A}$'s best guess is always that $S$ is $s_2$.) In contrast, $I(S;O)$ is greater than 0, except when $x$ is 0 or 1.

While mutual information leakage can exceed min-leakage, we conjecture that Shannon capacity cannot exceed min-capacity. To try to prove this conjecture, it appears useful to explore concepts relating to convexity and concavity. Recall that a real-valued function $f$ on a vector space is *concave* if for all vectors $v_1$, $v_2$ and $0 \leq \lambda \leq 1$, $\lambda f(v_1) + (1-\lambda) f(v_2) \leq f(\lambda v_1 + (1-\lambda) v_2)$. (For *convex*, replace $\leq$ with $\geq$.) There are classic results about mutual information [14]:

- For any $C_{\mathcal{SO}}$, $I(S;O)$ is a concave function of $P_S$.
- For any $P_S$, $I(S;O)$ is a convex function of $C_{\mathcal{SO}}$.
- Shannon capacity is a convex function of $C_{\mathcal{SO}}$.

When we turn our attention to min-entropy leakage, we can see from Figure 1 that $\mathcal{L}_{SO}$ is neither convex nor concave. However, it can be seen to be *piecewise* convex on the two regions $x \leq 0.5$ and $x \geq 0.5$. Following [20], we can see that

- $V(S)$ and $V(S|O)$ are convex and piecewise linear functions of $P_S$.

From this, it may be possible to show that $\mathcal{L}_{SO}$ is piecewise convex in $P_S$, building on [26]. Moreover, it appears possible to show that $\mathcal{L}_{SO}$ is a piecewise concave function of $C_{\mathcal{SO}}$; possibly these properties could be used to show that Shannon capacity cannot exceed min-capacity.

## V. Channel Matrix Factorization

As stated in Theorem 3.4, factorization of a channel matrix $C_{\mathcal{SO}} = C_{\mathcal{ST}} C_{\mathcal{TO}}$ yields an upper bound, $\log |\mathcal{T}|$, on the min-capacity. In this situation, we can view channels $(\mathcal{S}, \mathcal{T}, C_{\mathcal{ST}})$ and $(\mathcal{T}, \mathcal{O}, C_{\mathcal{TO}})$ as being composed into a *cascade* [12].[4] We can strengthen Theorem 3.4 to get bounds on the min-entropy leakage with respect to a given *a priori* $P_S$:

*Theorem 5.1:* Let $(\mathcal{S}, \mathcal{O}, C_{\mathcal{SO}})$ be the cascade of $(\mathcal{S}, \mathcal{T}, C_{\mathcal{ST}})$ and $(\mathcal{T}, \mathcal{O}, C_{\mathcal{TO}})$. Then for any *a priori* distribution $P_S$, we have $\mathcal{L}_{SO} \leq \mathcal{L}_{ST}$.

This theorem is proved in [27]; note that it can be seen as the min-entropy analogue of the classic *data-processing inequality* [7].

Curiously, when we consider the relation between $\mathcal{L}_{SO}$ and $\mathcal{L}_{TO}$, we find that we do not get a comparable result, as $\mathcal{L}_{SO}$ can actually exceed $\mathcal{L}_{TO}$. But, turning to channel capacity, we do find that the min-capacity of a cascade cannot exceed the min-capacity of either link.

Finally, results on the min-entropy leakage of different forms of channel composition are important in several recent studies [28], [29] that consider the relationship between min-entropy leakage and *differential privacy* [30].

## VI. Techniques for Computing Min-Entropy Leakage

Another important direction is the development of techniques for computing (or approximating) the min-entropy leakage of programs or systems, to verify whether they conform to a given quantitative flow policy. This is a challenging problem, as shown by the negative computational complexity results given in [31]. That paper shows that the problem of *comparing* the min-entropy leakage of two loop-free boolean programs is #P-hard; they give a reduction showing that one

[4] One subtle point here is that having a factorization $C_{\mathcal{SO}} = C_{\mathcal{ST}} C_{\mathcal{TO}}$ together with an *a priori* distribution $P_S$ is not enough to uniquely determine the joint distribution on $S$, $T$, and $O$. The problem is that $C_{\mathcal{TO}}$ could be dependent on $S$. But if $C_{\mathcal{TO}}$ is invariant with respect to $S$, then a unique joint distribution $P^\star_{STO}[s,t,o] = P_S[s] C_{\mathcal{ST}}[s,t] C_{\mathcal{TO}}[t,o]$ is obtained.

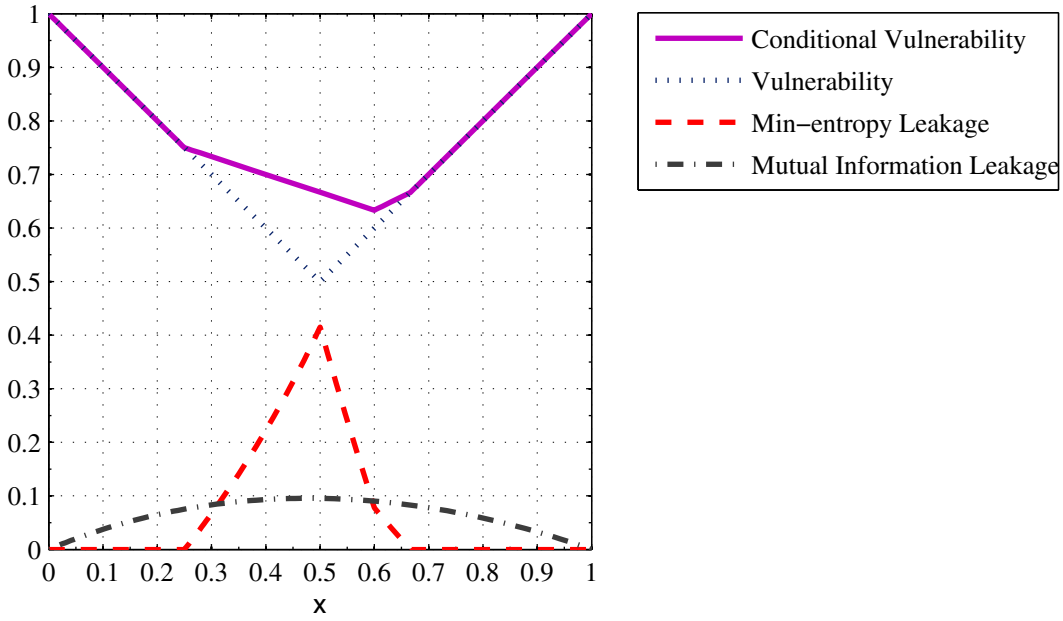


Figure 1. Vulnerability, min-entropy and mutual information leakage for channel (7), as a function of *a priori* $(x, 1-x)$.

can count the number of satisfying assignments of a boolean proposition (which is #P-complete) via a polynomial number of such comparison queries. More simply, suppose that $\varphi$ is a boolean proposition and let $\mathcal{S}$ and $\mathcal{O}$ be the set of encodings of truth assignments for $\varphi$, together with at least one "bad" encoding $b$. Consider the following program, which tests whether $S$ encodes a truth assignment that satisfies $\varphi$:

$$\begin{array}{l} \textbf{if } (S \text{ satisfies } \varphi) \\ \quad O = S; \\ \textbf{else} \\ \quad O = b; \end{array} \tag{8}$$

Since this program is deterministic, its min-capacity is the logarithm of the number of feasible outputs, which can be seen to be one more than the number of satisfying assignments of $\varphi$. Hence we see that the problem of counting the number of satisfying assignments of $\varphi$ (which is #P-complete) reduces to the problem of calculating the min-capacity of a deterministic, loop-free boolean program.

Nevertheless, this is an area that is now seeing a great deal of work, both in the context of probabilistic systems [32], [33] and deterministic imperative programs [8], [34], [35], [36].

For systems represented as probabilistic automata, Andrés et al. [32] show how to calculate the channel matrix using established model-checking techniques, including Gaussian elimination on linear equations, iterative methods involving regular expressions and strongly-connected components, and quantitative counterexample generation. In contrast, Chatzikokolakis et al. [33] estimate the channel matrix through statistical sampling, obtaining strong statistical bounds on the Shannon capacity; it is not clear whether similar bounds could be achieved for the min-capacity.

Turning to deterministic imperative programs, several recent works have used model-checking techniques to calculate the capacity; recall that, in the deterministic setting, min-capacity and Shannon capacity coincide, with both being equal to the logarithm of the number of feasible outputs.

Rather than actually trying to determine the capacity, Heusser and Malacaria [35] test whether a program $P$ has capacity at least $\log b$, for a given $b$, by testing whether it can produce at least $b$ different outputs. To test this, they form a new program $P'$ that runs $P$ independently $b$ times on nondeterministically-chosen inputs, and then check (using the bounded model checker CBMC) whether there is a path to a state where all $b$ outputs are distinct. While the technique yields interesting results on leakage in real Linux kernel vulnerabilities, it is important to note that the time taken by this method grows very quickly with $b$. Based on their experimental timings, it seems that one cannot go very much above $b = 128$; checking with $b = 2^{20}$ (corresponding to a 20-bit capacity) would appear infeasible.

Newsome, McCamant, and Song [8] estimate the capacity of x86 binaries. Interestingly, their motivation is quantitative *integrity*, looking at the amount of *influence* the untrusted input can have on the trusted output, as measured by Shannon capacity. But this again amounts to counting the number of feasible output values. They estimate this through various heuristics, using the decision procedure STP to check whether a particular output is feasible or not, and

whether an *interval* contains any feasible outputs. Using binary search, they try to find which intervals in the range of $O$ contain feasible outputs and which do not. When they find that an interval contains at least one feasible output, they use random sampling to estimate the *density* of feasible outputs within it. To deal with programs whose feasible outputs are sparse and scattered, they rely complementarily on a more expensive probabilistic #SAT algorithm to estimate directly the number of feasible output values.

Meng and Smith [36] consider another approach to bounding the number of feasible output values based on *two-bit patterns*. Suppose that the output $O$ is $n$ bits long. They use STP to determine, for every *pair* $(i, j)$ of bit positions, which of the four combinations $(0, 0)$, $(0, 1)$, $(1, 0)$, and $(1, 1)$ are feasible values for bits $i$ and $j$. As an example, consider the following program from [8]:

```
O = ((S >> 16) ^ S) & 0xffff;
O = O | O << 16;
```

where `S` and `O` are 32-bit unsigned integers. Here there are $32 \cdot 31/2 = 496$ two-bit patterns on `O` to determine; the only interesting ones are that bits $i$ and $i + 16$ of `O` must be equal, for $0 \leq i \leq 15$. That is, bits $i$ and $i + 16$ can be $(0, 0)$ or $(1, 1)$, but not $(0, 1)$ or $(1, 0)$. If we count the number of solutions to the two-bit patterns, we get an *upper bound* on the number of feasible output values. Here there are $2^{16}$ solutions to the two-bit patterns, giving a min-capacity of at most $\log 2^{16} = 16$ bits, which is exact in this case. Sometimes two-bit patterns greatly overestimate the capacity, but experimentally they often seem to give quite accurate bounds.

A quite different approach to approximating leakage is given in the recent work of Köpf and Rybalchenko [34], which uses statistical sampling to estimate the *mutual-information leakage* of a deterministic imperative program from input $S$ to output $O$, under a uniform *a priori* distribution. While they present the technique in terms of estimating $H(S|O)$, it is clearer to remember that the mutual-information leakage is just $H(O)$. They assume that for each feasible output value $o$, we can estimate its probability (by estimating the number of values of $S$ that lead to $o$). Then they observe that $H(O)$ is the expected value of $log \frac{1}{P(o)}$, where $o$ is a sampled output value:

$$\text{mutual information leakage} = H(O) = E\left(log \frac{1}{P(o)}\right).$$

With $n$ samples, $o_1$, $o_2$, $\ldots o_n$, we find that $H(O)$ is also the expected value of $\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{P(o_i)}$. Crucially, the *variance* of this last random variable is small relative to the number of possible inputs, which means that the Chebyshev inequality can be used to give good bounds on the accuracy of the estimate for not-too-large values of $n$. However, it is not clear whether a similar technique could be used to calculate min-entropy leakage.

While the surveyed research efforts have made considerable progress, the automatic calculation of leakage has so far been done only for small programs; scaling the analyses to large systems remains a challenge.

## VII. Conclusion

Quantitative information flow has become a vibrant research area, with rapid theoretical and practical advances. The theory of min-entropy leakage seems to be a particularly attractive framework for analyzing confidentiality properties of systems, showing promise as a useful foundation for computer security.

## Acknowledgments

There are many people who have contributed to my understanding of min-entropy leakage. For insightful discussions, I am particularly grateful to Mário Alvim, Miguel Andrés, Christelle Braun, Kostas Chatzikokolakis, Barbara Espinoza, Boris Köpf, Pasquale Malacaria, Ziyuan Meng, Catuscia Palamidessi, and Leonard Reisman. This work was partially supported by the National Science Foundation under grant CNS-0831114.

## References

[1] D. Clark, S. Hunt, and P. Malacaria, "Quantitative information flow, relations and polymorphic types," *Journal of Logic and Computation*, vol. 18, no. 2, pp. 181–199, 2005.

[2] 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.

[3] 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.

[4] M. Alvim, M. Andrés, and C. Palamidessi, "Probabilistic information flow," in *Proc. 25th IEEE Symposium on Logic in Computer Science (LICS 2010)*, 2010, pp. 314–321.

[5] 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.

[6] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

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

[8] J. Newsome, S. McCamant, and D. Song, "Measuring channel capacity to distinguish undue influence," in *Proc. Fourth Workshop on Programming Languages and Analysis for Security (PLAS '09)*, 2009, pp. 73–85.

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

[10] C. E. Shannon, "The bandwagon," *IRE Transactions on Information Theory*, vol. 2, no. 1, p. 3, 1956.

[11] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960*, 1961, pp. 547–561.

[12] N. Abramson, *Information Theory and Coding*. McGraw-Hill, 1963.

[13] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. John Wiley & Sons, Inc., 1968, vol. I.

[14] R. G. Gallager, *Information Theory and Reliable Communication*. John Wiley and Sons, Inc., 1968.

[15] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

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

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

[18] J. L. Massey, "Guessing and entropy," in *Proc. 1994 IEEE International Symposium on Information Theory*, 1994, p. 204.

[19] 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.

[20] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden, "On the Bayes risk in information-hiding protocols," *Journal of Computer Security*, vol. 16, no. 5, pp. 531–571, 2008.

[21] C. Cachin, "Entropy measures and unconditional security in cryptography," Ph.D. dissertation, Swiss Federal Institute of Technology, 1997.

[22] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal of Computing*, vol. 38, no. 1, pp. 97–139, 2008.

[23] 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.

[24] B. Köpf and G. Smith, "Vulnerability bounds and leakage resilience of blinded cryptography under timing attacks," in *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, 2010, pp. 44–56.

[25] N. Santhi and A. Vardy, "On an improvement over Rényi's equivocation bound," in *44th Annual Allerton Conference on Communication, Control, and Computing*, 2006.

[26] K. Chatzikokolakis and K. Martin, "A monotonicity principle for information theory." in *Proc. 24th Conference on the Mathematical Foundations of Programming Semantics*, ser. Electronic Notes in Theoretical Computer Science, vol. 218, 2008, pp. 111–129.

[27] B. Espinoza and G. Smith, "Min-entropy leakage of channels in cascade," 2011, submitted for publication.

[28] M. Alvim, M. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "Differential privacy: on the trade-off between utility and information leakage," LIX, Ecole Polytechnique, Tech. Rep., 2011.

[29] G. Barthe and B. Köpf, "Information-theoretic bounds for differentially private mechanisms," in *Proc. 24th IEEE Computer Security Foundations Symposium (CSF 2011)*, 2011, to appear.

[30] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, 2011.

[31] 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.

[32] M. Andrés, C. Palamidessi, P. van Rossum, and G. Smith, "Computing the leakage of information-hiding systems," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '10)*, ser. Lecture Notes in Computer Science, J. Esparza and R. Majumdar, Eds., vol. 6015, 2010, pp. 373–389.

[33] K. Chatzikokolakis, T. Chothia, and A. Guha, "Statistical measurement of information leakage," in *Tools and Algorithms for the Construction and Analysis of Systems (TACAS '10)*, ser. Lecture Notes in Computer Science, J. Esparza and R. Majumdar, Eds., vol. 6015, 2010, pp. 390–404.

[34] B. Köpf and A. Rybalchenko, "Approximation and randomization for quantitative information-flow analysis," in *Proc. 23nd IEEE Computer Security Foundations Symposium (CSF '10)*, 2010, pp. 3–14.

[35] J. Heusser and P. Malacaria, "Quantifying information leaks in software," in *Proc. ACSAC '10*, 2010.

[36] Z. Meng and G. Smith, "Calculating bounds on information leakage using two-bit patterns," in *Proc. Sixth Workshop on Programming Languages and Analysis for Security (PLAS '11)*, 2011.