2017 IEEE 30th Computer Security Foundations Symposium

Tight Bounds on Information Leakage from Repeated Independent Runs

David M. Smith
Princeton University, Princeton, New Jersey
Email: dms10@princeton.edu

Geoffrey Smith
Florida International University, Miami, Florida
Email: smithg@cis.fiu.edu

***Abstract*—We investigate a problem in quantitative information flow, namely to find the maximum information leakage caused by n repeated independent runs of a channel C with b columns. While this scenario is of general interest, it is particularly motivated by the study of timing attacks on cryptography implemented using the countermeasures known as blinding and bucketing. We measure leakage in terms of multiplicative Bayes capacity (also known as min-capacity) and we prove tight bounds that greatly improve the previously-known ones. To enable efficient computation of our new bounds, we investigate them using techniques of analytic combinatorics, proving that they satisfy a useful recurrence and (when $b = 2$) a close connection to Ramanujan's Q-function.**

I. Introduction

A fundamental problem in computer security is to prevent systems from improperly leaking the sensitive information that they process. Because it is often necessary in practice to tolerate some leakage, the last decade has seen growing interest in theories of *quantitative information flow* [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] which let us talk about "how much" information is leaked and perhaps enable us to tolerate "small" leaks.

A system C taking a secret input X from a finite set $\mathcal{X}$ and (perhaps probabilistically) producing an observable output Y from a finite set $\mathcal{Y}$ can be modeled as an information-theoretic *channel matrix* [11], whose rows give the distribution of outputs corresponding to each possible input. That is, entry $\mathsf{C}_{x,y}$ denotes $p(y|x)$, the conditional probability of getting output y on input x.

In this paper, we study the leakage caused by running a channel C repeatedly, using the same secret input in each run, producing a tuple of observable outputs. Of course, if the channel is *deterministic* then this is a waste of time, since all the outputs in the tuple will be the same. But if the channel is *probabilistic*, then we will see that repeated runs indeed leak more than a single run.

Definition 1.1: For $n \geq 0$, the *repeated independent runs channel* $\mathsf{C}^{(n)}$ is defined by

$$\mathsf{C}^{(n)}_{x,(y_1,\ldots,y_n)} = \prod_{i=1}^{n} \mathsf{C}_{x,y_i} \,.$$

(When $n = 0$ the output is an *empty tuple* $()$, which occurs with probability 1.)

For example, given $\mathcal{X} = \{x_1, x_2, x_3\}$ and $\mathcal{Y} = \{y_1, y_2\}$, if channel C is

C	y_1	y_2
x_1	0	1
x_2	$1/3$	$2/3$
x_3	$1/2$	$1/2$

then the corresponding $\mathsf{C}^{(3)}$ is shown in Figure 1.

Our particular focus in this paper is on bounding the maximum leakage of the repeated independent runs channel $\mathsf{C}^{(n)}$ in the case where C has $b \geq 1$ columns. The leakage measure that we use is *multiplicative Bayes capacity* (also known as *min-capacity*), which is reviewed in Section II.

While this scenario is of general interest, we note that one important motivation is that $\mathsf{C}^{(n)}$ models an n-observation timing attack against a cryptosystem implemented with a countermeasure called *blinding* [12]. And if the cryptosystem is implemented using *bucketing* [13], so that each decryption takes one of b possible times, then we can ensure that C has just b columns. In this setting, we typically expect that b is *small* (say, less than 25) and n is *big* (say, over a million).

For readers unfamiliar with blinding and bucketing, we give a brief review in the following subsection.

A. Motivating application: timing attacks on blinded and bucketed cryptosystems

Starting with Kocher's seminal work [12], it has been known that an adversary able to observe the *times* taken by a large number of decryption operations may be able to recover the secret key, even if the decryption operations are being done remotely, making the timing measurements imprecise [14].

Consider an implementation such that the execution time is given by a function $t : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$, such that $t(K, M)$ gives the time required to decrypt message M using secret key K. Here $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{T}$ are finite sets of possible secret keys, messages, and decryption times, respectively. (Note that a deterministic model like this is not correct for hardware using caching, but it could be made valid by forcing the cache into a fixed initial state before each decryption.)

In a timing attack, the adversary may be able to *choose* a large number of messages M_1, M_2, ..., M_n and observe the time to decrypt each one using the same secret key K. *Blinding* [12] is a countermeasure that decorrelates the messages from the decryption times; it works by *randomizing* each message

DOI 10.1109/CSF.2017.18

$\mathsf{C}^{(3)}$	(y_1,y_1,y_1)	(y_1,y_1,y_2)	(y_1,y_2,y_1)	(y_1,y_2,y_2)	(y_2,y_1,y_1)	(y_2,y_1,y_2)	(y_2,y_2,y_1)	(y_2,y_2,y_2)
x_1	0	0	0	0	0	0	0	1
x_2	$1/27$	$2/27$	$2/27$	$4/27$	$2/27$	$4/27$	$4/27$	$8/27$
x_3	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$	$1/8$

Fig. 1. An example repeated independent runs channel.

M before the decryption, and *derandomizing* afterwards so that the correct decryption is obtained. The result is that the adversary observes not $t(K,M)$ for his chosen M, but instead $t(K,M')$ for a *randomly-chosen* M'.[1]

It follows that a single timing observation is then modeled by a *probabilistic* channel C from $\mathcal{K}$ to $\mathcal{T}$, whose input is a secret key K and whose output is the time to decrypt a randomly-chosen message M' using K. And an n-observation timing attack is then modeled by the repeated independent runs channel $\mathsf{C}^{(n)}$.

Bucketing [13] is an additional countermeasure, which decreases the size of $\mathcal{T}$ by forcing each decryption to take one of b possible times, for some small number b. This is done by choosing a set of b "bucket" times, and then delaying each decryption result until the next bucket time. Experiments in [13] show that on 1024-bit RSA, the performance overhead from using $b = 5$ is less than 0.7% with a careful choice of bucket times. Even using $b = 2$ gives an overhead of under 3%. But using $b = 1$, a constant-time implementation, gives an overhead of over 36%.

Bucketing thus allows us to trade off between security and performance: choosing $b = 1$ eliminates timing attacks but substantially lowers performance, while choosing $b = 5$ gives better performance but allows stronger timing attacks.

B. Our principal contributions

Studying the maximum leakage of a repeated independent runs channel $\mathsf{C}^{(n)}$, where C is an arbitrary channel matrix with b columns, we make the following principal contributions:

1) We greatly improve on the previously-known bounds [15], [16], proving in Theorem 3.1 that the maximum leakage is tightly bounded by a function that we denote $\mathit{cap}_b(n)$, which is only about the *square root* of the previous leakage bound.
2) Because $\mathit{cap}_b(n)$ is expensive to calculate directly, we analyze it using techniques of analytic combinatorics, proving in Theorem 4.1 that it satisfies a remarkable recurrence, which enables $\mathit{cap}_b(n)$, for any b, to be calculated efficiently from $\mathit{cap}_2(n)$.
3) Using a bound asserted by Ramanujan, in Theorem 4.4 we prove a simple upper bound on $\mathit{cap}_2(n)$ that is also asymptotically correct.

These contributions enable tight leakage bounds for $\mathsf{C}^{(n)}$ to be computed accurately and efficiently, for any b and any n.

The rest of the paper is organized as follows. Section II gives background about multiplicative Bayes capacity and about the previously-known bounds on the maximum leakage of $\mathsf{C}^{(n)}$. Section III proves a new, tight bound in terms of $\mathit{cap}_b(n)$. Section IV presents our analytic results about $\mathit{cap}_b(n)$, which enable it to be computed efficiently. Because the proofs of these results require specialized techniques of analytic combinatorics, they are all deferred to Section V. Finally, Section VI discusses related work and Section VII concludes.

II. Background

A. Multiplicative Bayes capacity $\mathcal{ML}^{\times}(\mathsf{C})$

An important insight of quantitative information flow is that there are *many* measures of information leakage, and the question of which measure is most appropriate in a given situation depends on the details of the operational scenario, which comprises both the adversary's *goals* and *capabilities*. It is for this reason that g-leakage [7] parameterizes the leakage measure with a *gain function* g, which models the operational scenario. However, one particularly important measure of the leakage of a channel C is its *multiplicative Bayes capacity* $\mathcal{ML}^{\times}(\mathsf{C})$, which in the previous literature has usually been called *min-capacity*.

A detailed tutorial on quantitative information flow and $\mathcal{ML}^{\times}(\mathsf{C})$ can be found in [10]; here we limit ourselves to three key properties of $\mathcal{ML}^{\times}(\mathsf{C})$ (from [17], [5], [7]) which are sufficient for this paper:

- $\mathcal{ML}^{\times}(\mathsf{C})$ is *easy to compute* from the channel matrix C: it is simply the *sum of* C*'s column maximums*.[2]
- $\mathcal{ML}^{\times}(\mathsf{C})$ is *operationally significant*: it is the maximum factor (over all priors) by which C increases an adversary's probability of guessing the secret input X correctly in one try.
- $\mathcal{ML}^{\times}(\mathsf{C})$ is *robust*: it is an upper bound on multiplicative g-leakage, for any gain function g and any prior.

For example, for the C and $\mathsf{C}^{(3)}$ above we have

$$\mathcal{ML}^{\times}(\mathsf{C}) = 1/2 + 1 = 3/2$$

and

$$\mathcal{ML}^{\times}(\mathsf{C}^{(3)}) = 1/8+1/8+1/8+4/27+1/8+4/27+4/27+1 = 35/18.$$

[1] To see an example, note that the RSA decryption of M using secret key (d,N) is $M^d \bmod N$. If the corresponding public key is (e,N), then M can be randomized by choosing a random r that is relatively prime to N, and setting $M' = M \cdot r^e \bmod M$, which makes M' uniformly distributed. Note that M' decrypts to $(M \cdot r^e)^d = M^d \cdot r \bmod N$, so the decryption can be derandomized by multiplying it by $r^{-1} \bmod N$.

[2] In the previous literature, $\mathcal{ML}^{\times}(\mathsf{C})$ has normally been defined as the *logarithm* of this quantity. Since taking the logarithm just changes the scale, we omit it for simplicity.

B. Previous bounds on $\mathcal{ML}^{\times}(\mathsf{C}^{(n)})$

Turning now to prior work on repeated independent runs channels, it is proved by Köpf and Smith [15] and Espinoza and Smith [16] that, for fixed b, the multiplicative Bayes capacity of $\mathsf{C}^{(n)}$ grows only polynomially in n:

Theorem 2.1: If C has $b \geq 1$ columns and $n \geq 0$, then $\mathcal{ML}^{\times}(\mathsf{C}^{(n)}) \leq \binom{n+b-1}{n}$.

Note that in the particular cases when b is 1, 2, 3, or 4, the bounds are 1, $n+1$, $(n+1)(n+2)/2$, or $(n+1)(n+2)(n+3)/6$, respectively.

The proof of Theorem 2.1 uses the information-theoretic *method of types* [11]. The *type* of an output sequence is the proportion of each of the b possible outputs that it contains. For example, if $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ and $n = 10$, then the output sequence $(y_3, y_2, y_2, y_4, y_2, y_3, y_2, y_2, y_2, y_2)$ has type $(0, {}^{7}/_{10}, {}^{2}/_{10}, {}^{1}/_{10})$. Here the key observation is that if two output sequences have the same type, then their columns in $\mathsf{C}^{(n)}$ are identical, because the probabilities in Definition 1.1 are invariant under permutations of the output sequence. Therefore all columns with the same type can be *merged* without affecting the multiplicative Bayes capacity, since they all have their column maximums in the same position.[3] For example, here is the merged version of $\mathsf{C}^{(3)}$ above, where now the columns are labeled with types:

merged $\mathsf{C}^{(3)}$	$(1,0)$	$({}^{2}/_{3}, {}^{1}/_{3})$	$({}^{1}/_{3}, {}^{2}/_{3})$	$(0,1)$
x_1	0	0	0	1
x_2	${}^{1}/_{27}$	${}^{2}/_{9}$	${}^{4}/_{9}$	${}^{8}/_{27}$
x_3	${}^{1}/_{8}$	${}^{3}/_{8}$	${}^{3}/_{8}$	${}^{1}/_{8}$

As recalled above, the multiplicative Bayes capacity of a channel is the sum of its column maximums. Since each column maximum is at most 1, we see that the capacity of the merged $\mathsf{C}^{(n)}$ (and hence of $\mathsf{C}^{(n)}$ as well) is at most the number of possible types. By simple combinatorics, this is given by the binomial coefficient $\binom{n+b-1}{n}$, and Theorem 2.1 follows.

III. A BETTER LEAKAGE BOUND

The obvious weakness of Theorem 2.1 is that it uses the trivial upper bound 1 for all the column maximums; we might sense that this is quite pessimistic. For a more careful analysis, we need to determine the maximum probability with which $\mathsf{C}^{(n)}$ outputs a sequence of each type t. For example, suppose that $b = 2$ and $n = 1000$ and consider the type $t = ({}^{417}/_{1000}, {}^{583}/_{1000})$. When would a row of C have the greatest probability of generating an output sequence of type t? Intuitively, it would seem best for a row of C to give exactly the same distribution on outputs as in t. This intuition turns out to be correct. Indeed, Theorem 11.1.2 of [11] shows that if a row of C gives distribution q on $\mathcal{Y}$, then the probability that the corresponding row of $\mathsf{C}^{(n)}$ outputs a particular sequence of type t is precisely

$$2^{-n(H(t)+D_{KL}(t||q))},$$

where $H(t)$ is the *Shannon entropy* of t and $D_{KL}(t||q)$ is the *Kullback-Leibler distance* between t and q. Now, *Gibbs' inequality* says that $D_{KL}(t||q) \geq 0$, with equality iff $t = q$. Hence the above probability is maximized (to $2^{-nH(t)}$) when $t = q$.

Turning our attention now to the merged $\mathsf{C}^{(n)}$, we see that to get the maximum possible entry in the column for type t we need to multiply the probability above (of getting a particular sequence of type t) by the number of sequences of type t. But it seems easier to calculate this maximum probability directly. Any type t is of the form

$$\left(\frac{x_1}{n}, \frac{x_2}{n}, \ldots, \frac{x_b}{n}\right),$$

where $x_1, x_2, \ldots, x_b$ are non-negative integers that sum to n. If a row of C matches t, then the corresponding row of the merged $\mathsf{C}^{(n)}$ gives output t with probability

$$\frac{n!}{x_1!x_2!\cdots x_b!}\left(\frac{x_1}{n}\right)^{x_1}\left(\frac{x_2}{n}\right)^{x_2}\cdots\left(\frac{x_b}{n}\right)^{x_b}.$$

To see this, note that the first factor gives the number of sequences of type t, while the second factor gives the probability of each such sequence.

To get the maximum sum of the column maximums of the merged $\mathsf{C}^{(n)}$, we simply sum the above term over all possible choices of x_1, x_2, …, x_b. With some reorganization, this leads to the following function:

Definition 3.1: For integers $b \geq 1$ and $n \geq 0$, define

$$cap_b(n) = \frac{n!}{n^n} \sum_{\substack{x_1,x_2,\ldots,x_b \in \mathbb{N} \\ x_1+x_2+\cdots+x_b=n}} \frac{x_1^{x_1}x_2^{x_2}\cdots x_b^{x_b}}{x_1!x_2!\cdots x_b!}.$$

We have thus proved the following stronger bound[4] on $\mathcal{ML}^{\times}(\mathsf{C}^{(n)})$:

Theorem 3.1: If C has $b \geq 1$ columns and $n \geq 0$, then $\mathcal{ML}^{\times}(\mathsf{C}^{(n)}) \leq cap_b(n)$.

Note moreover that the bound in Theorem 3.1 is *tight*, since for any n it is achieved by any C which has a row matching each possible type t.[5]

To get a sense of the significance of the Theorem 3.1, consider the case when $b = 2$. In that case, Definition 3.1 simplifies to

$$cap_2(n) = \frac{n!}{n^n}\sum_{i=0}^{n}\frac{i^i(n-i)^{n-i}}{i!(n-i)!}. \tag{1}$$

Recall that the old upper bound in Theorem 2.1 uses the trivial upper bound 1 for each column maximum. In fact the true values are far smaller, as shown in Figure 2, which plots the values of the terms of $cap_2(2000)$ for $0 \leq i \leq 2000$. It shows

[3] More deeply, merging such columns does not affect the *abstract channel* (by which we mean the mapping that $\mathsf{C}^{(n)}$ gives from prior distributions to hyper-distributions) [8].

[4] We have learned that this stronger bound was also proved, independently, by Goran Doychev and Boris Köpf (personal communication).

[5] Also, it is plausible that a blinded timing channel might satisfy this assumption, since such a channel has a huge number of rows (one for each possible secret key) but only polynomially-many (in n) types.

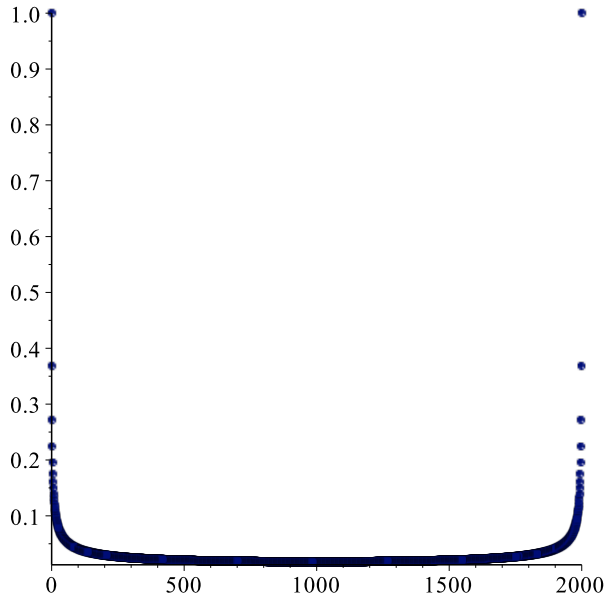

Fig. 2. Plot of the terms of $cap_2(2000)$ for $0 \leq i \leq 2000$.

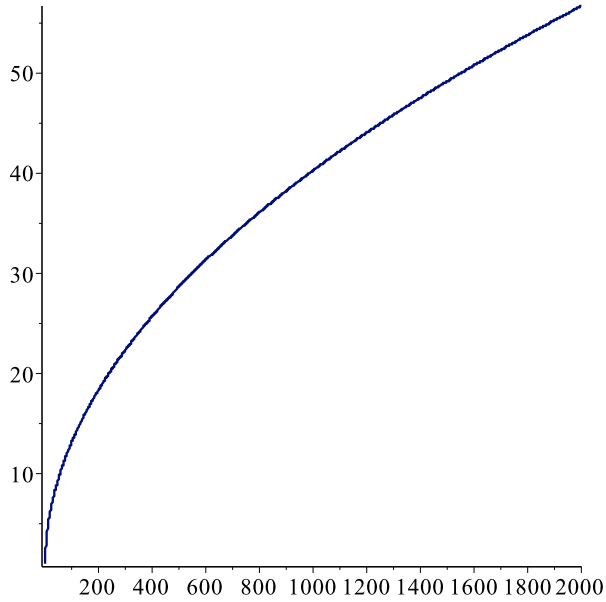

Fig. 3. Plot of $cap_2(n)$ for $0 \leq n \leq 2000$.

that the term is 1 when i is 0 or 2000, and that its value falls sharply as i moves away from those endpoints. In particular, it is less than 0.2 for i between 4 and 1996, and less than 0.041 for i between 100 and 1900. The minimum value, about 0.0178, occurs when $i = 1000$.

We find that $cap_2(2000)$ is about 56.72, showing that $\mathcal{ML}^{\times}(\mathsf{C}^{(2000)}) \leq 56.72$, which is a far better bound than the one given by Theorem 2.1, which is 2001.

Figure 3 shows the value of $cap_2(n)$ for $0 \leq n \leq 2000$. As can be seen, its growth appears to be much slower than linear. In fact, as will be seen in Section IV-C below, $cap_2(n)$ is approximately $\sqrt{\frac{\pi n}{2}}$.

A. Computational challenges

The improved leakage bound in Theorem 3.1 is valuable, but unfortunately it is difficult to apply it in security analysis. The problem is that the formula in Definition 3.1 is very expensive to compute directly, since it involves $b-1$ nested sums. In the case of a 100-observation timing attack with 6 buckets, for instance, we are interested in the value of $cap_6(100)$, but direct computation of this in Maple using exact rational arithmetic already takes about 20 minutes.[6] And we would actually be interested in computing $cap_6(n)$ for a far larger number of timing observations, such as $n = 10^9$ or even $n = 10^{12}$. Note that when $b = 6$, the number of terms in Definition 3.1 is given by the binomial coefficient $\binom{n+5}{n}$, which for large n is about $n^5/120$. This implies that increasing n from 10^2 to 10^9 increases the number of terms by a factor of 10^{35}. Hence, even ignoring the fact that the individual terms become more expensive to compute as n increases, we find that computing $cap_6(10^9)$ would take over 10^{30} years.

It is therefore clear that a better way of computing $cap_b(n)$ is needed before we can make serious use of Theorem 3.1.

[6]The computation was done on an iMac with a 3.2 GHz Intel Core i5 and 16 GB of memory.

$cap_b(n)$	0	1	2	3	4	5	...
1	1	1	1	1	1	1	...
2	1	2	$\frac{5}{2}$	$\frac{26}{9}$	$\frac{103}{32}$	$\frac{2194}{625}$	...
3	1	3	$\frac{9}{2}$	$\frac{53}{9}$	$\frac{231}{32}$	$\frac{5319}{625}$	...
4	1	4	7	$\frac{92}{9}$	$\frac{437}{32}$	$\frac{10804}{625}$	...
5	1	5	10	$\frac{145}{9}$	$\frac{745}{32}$	$\frac{19669}{625}$	...
6	1	6	$\frac{27}{2}$	$\frac{214}{9}$	$\frac{591}{16}$	$\frac{33174}{625}$	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Fig. 4. Some values of $cap_b(n)$, with row = b, column = n.

IV. Analytic results about $cap_b(n)$

Figure 4 gives a table of some of the values of $cap_b(n)$, where the row gives the value of b and the column gives the value of n. The first row suggests that $cap_1(n) = 1$ for all n, and this is indeed straightforward to verify from Definition 3.1. But the behavior of $cap_b(n)$ for larger values of b is not obvious. In the following subsections, we present a number of analytic results that clarify the behavior of $cap_b(n)$ and make it possible to compute it efficiently.

A. A recurrence satisfied by $cap_b(n)$

Through analytic and empirical study, we discovered that $cap_b(n)$ is governed by a remarkable recurrence.

Theorem 4.1: For all $b \geq 3$ and all $n \geq 0$,

$$cap_b(n) = cap_{b-1}(n) + \frac{n}{b-2} cap_{b-2}(n)\,.$$

To illustrate, when $b = 5$ and $n = 4$ we have

$$cap_{5-1}(4) + \frac{4}{5-2} cap_{5-2}(4) = \frac{437}{32} + \frac{4}{3} \cdot \frac{231}{32} = \frac{745}{32} = cap_5(4).$$

The proof of Theorem 4.1 involves specialized techniques of analytic combinatorics, so we defer it until Section V. To give some indication now of the difficulty of the proof, we mention that the special case of Theorem 4.1 when $b = 3$, namely $cap_3(n) = cap_2(n) + n$, was discovered by Lacasse [18, p. 90] in his 2010 dissertation about machine learning but left as a conjecture. Since then, no fewer than four proofs of this special case, using a variety of techniques, have been published by Younsi [19], Chen, Peng, and Yang [20], Sun [21], and Prodinger [22].

B. Using the Theorem 4.1 recurrence to compute $cap_b(n)$

Theorem 4.1 applied repeatedly enables us to express $cap_b(n)$, for any b, in terms of $cap_2(n)$ and $cap_1(n)$. For instance we get

$$\begin{aligned} cap_4(n) &= cap_{4-1}(n) + \frac{n}{4-2} cap_{4-2}(n) \\ &= cap_{3-1}(n) + \frac{n}{3-2} cap_{3-2}(n) + \frac{n}{2} cap_2(n) \\ &= \left(\frac{n}{2} + 1\right) cap_2(n) + n\, cap_1(n)\,. \end{aligned}$$

But since $cap_1(n) = 1$ for all n, this means that every $cap_b(n)$ can be expressed just in terms of $cap_2(n)$, through formulas like the following:

$$\begin{aligned} cap_3(n) &= cap_2(n) + n \\ cap_4(n) &= \left(\frac{n}{2} + 1\right) cap_2(n) + n \\ cap_5(n) &= \left(\frac{5n}{6} + 1\right) cap_2(n) + \frac{n^2}{3} + n \\ cap_6(n) &= \left(\frac{n^2}{8} + \frac{13n}{12} + 1\right) cap_2(n) + \frac{7n^2}{12} + n \end{aligned}$$

Note that computing $cap_b(n)$ using Definition 3.1 requires summing a total of $\binom{n+b-1}{n}$ terms. In particular, calculating $cap_6(100)$ requires summing almost 100 million terms. But, using the equation above, $cap_6(100)$ can be computed by summing just 101 terms for $cap_2(100)$ and evaluating some simple polynomials. Experimentally, we find that the time to compute $cap_6(100)$ in Maple is reduced from 20 minutes to about 5 milliseconds, a speed up of more than a factor of 200 thousand.

In spite of these positive results, we are still faced with the problem that computing $cap_2(n)$ from equation (1) becomes expensive as n becomes large. For instance, direct computation of $cap_2(50000)$ in Maple using exact rational arithmetic takes almost 2 hours.

For this reason, in the next subsection we give results allowing $cap_2(n)$ to be approximated very accurately and efficiently.

C. Results about $cap_2(n)$

It turns out that $cap_2(n)$ has a remarkable relationship to Ramanujan's celebrated Q-function [23], which is defined for $n \geq 1$ by

$$\begin{aligned} Q(n) &= 1 + \frac{n-1}{n} + \frac{n-1}{n}\frac{n-2}{n} + \frac{n-1}{n}\frac{n-2}{n}\frac{n-3}{n} + \cdots \\ &= \sum_{k=1}^{n} \frac{n!}{n^k (n-k)!}\,. \end{aligned}$$

Prodinger [22] proves the following theorem:

Theorem 4.2: For all $n \geq 1$, $cap_2(n) = Q(n) + 1$.

Prodinger's proof uses complex analysis, and we elucidate it in Appendix A. We have also found an easier proof, which we defer to Section V.

An amazing consequence of Theorem 4.2 is that $cap_2(n)$ has a significance unrelated to information leakage: it is also the *expected number of people needed in order to find two having the same birthday, when n is the number of days in a year*. For instance, $cap_2(365) \approx 24.616586$.

Of more practical significance, asymptotic approximations for $Q(n)$ are known, and these (by adding 1 to them) work for $cap_2(n)$ as well. For instance, from equation (2.6) of [24], we have the following theorem:

Theorem 4.3:

$$cap_2(n) = \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} - \frac{4}{135n} + O(n^{-3/2})\,.$$

This theorem allows $cap_2(n)$ (and, using Theorem 4.1, every $cap_b(n)$) to be approximated very efficiently, with a relative error that tends to 0 as n goes to infinity.

For security analysis, however, an *approximation* to $cap_2(n)$ like Theorem 4.3 is not quite satisfactory, since for any particular value of n we do not know how big the $O(n^{-3/2})$ term might be. Hence an *upper bound* would be preferable.

As our final major result (whose proof is again deferred to Section V) we have applied an assertion due to Ramanujan to prove such an upper bound:

Theorem 4.4: For all $n \geq 1$,

$$cap_2(n) < \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}}\,.$$

Note that Theorem 4.4 gives an upper bound that is also asymptotically correct, since it is the result of truncating the expansion in Theorem 4.3 after its third term.

Finally, we mention that techniques similar to those used in proving Theorem 4.4 allow us also to prove an asymptotically-correct *lower bound*:

Theorem 4.5: For all $n \geq 1$,

$$cap_2(n) > \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} - \frac{4}{135n}\,.$$

This lower bound tells us that the upper bound given by Theorem 4.4 on $n = 10^6$, namely

$$cap_2(10^6) < 1253.98090843\,,$$

is correct to 11 significant digits.

D. Applying Theorem 3.1 in security analysis

These analytic results about $cap_b(n)$ now enable us to assess the extent to which Theorem 3.1 improves Theorem 2.1 in security analysis.

First, we can compare the leakage bounds *asymptotically*. If we regard b as a constant, then the old Theorem 2.1 bound of $\binom{n+b-1}{n}$ is asymptotically $O(n^{b-1})$, while the new Theorem 3.1 bound of $cap_b(n)$ is asymptotically $O(n^{\frac{b-1}{2}})$.

To see this, note that

$$cap_1(n) = O(1) = O(n^{\frac{1-1}{2}})$$

and (by Theorem 4.4)

$$cap_2(n) = O(\sqrt{n}) = O(n^{\frac{2-1}{2}})\,.$$

Now (by Theorem 4.1) for $b \geq 3$ we can reason inductively:

$$\begin{aligned} cap_b(n) &= cap_{b-1}(n) + \frac{n}{b-2} cap_{b-2}(n) \\ &= O(n^{\frac{b-2}{2}}) + O(n) \cdot O(n^{\frac{b-3}{2}}) \\ &= O(n^{\frac{b-2}{2}}) + O(n^{\frac{b-1}{2}}) \\ &= O(n^{\frac{b-1}{2}})\,. \end{aligned}$$

Thus Theorem 3.1 tells us that the maximum leakage of $\mathsf{C}^{(n)}$ is asymptotically only the *square root* of the previous Theorem 2.1 bound.

More concretely, consider a scenario similar to the one in [15, p. 50], namely an n-observation timing attack against a blinded cryptosystem implemented with bucketing. Suppose that, based on the key length we are using, we decide that we can tolerate leakage of at most 10^{26}.

If we want to be secure against $n = 10^9$ timing observations, then the bound from Theorem 2.1 suggests that we can use only 3 buckets, because when $b = 4$ it gives maximum leakage somewhat over our limit:

$$\binom{10^9 + 4 - 1}{10^9} > 1.666666 \times 10^{26}\,.$$

In contrast, Theorem 3.1 tells us that $b = 4$ is actually fine:

$$cap_4(10^9) < 1.981797 \times 10^{13}\,.$$

This also illustrates that the new bound is indeed roughly the *square root* of the previous bound.

The improved leakage bound from Theorem 3.1 gives us a number of ways that we can modify b and n, while staying below the leakage limit of 10^{26}. If we just want to increase the performance of the cryptosystem, we can keep $n = 10^9$ and increase the number of buckets to $b = 7$:

$$cap_7(10^9) < 6.667823 \times 10^{25}\,.$$

Or we can increase the number of buckets to $b = 5$ and also increase the number of timing observations to $n = 10^{13}$, allowing us to choose fresh keys much less frequently:

$$cap_5(10^{13}) < 3.333337 \times 10^{25}\,.$$

Or we can use $b = 6$ and $n = 10^{10}$:

$$cap_6(10^{10}) < 1.566710 \times 10^{24}\,.$$

Finally, we note that computing all of these $cap_b(n)$ values using Theorem 4.1 and Theorem 4.4 takes an insignificant amount of time in Maple (just a few milliseconds).

V. Analytic Proofs

In this section, we give proofs of all of the theorems from Section IV.

Our efforts at proving the recurrence of Theorem 4.1 by elementary techniques were unsuccessful. Some progress can be made, however, by focusing on the *columns* of the table in Figure 4, rather than the *rows*. That is, we fix n and view $cap_b(n)$ as a *function of* b. Elementary calculations then let us determine these functions for small values of n:

$$cap_b(0) = 1$$

$$cap_b(1) = b$$

$$cap_b(2) = \frac{b^2 + 3b}{4}$$

$$cap_b(3) = \frac{b^3 + 9b^2 + 17b}{27}$$

$$cap_b(4) = \frac{b^4 + 18b^3 + 95b^2 + 142b}{256}$$

$$cap_b(5) = \frac{b^5 + 30b^4 + 305b^3 + 1220b^2 + 1569b}{3125}$$

$$cap_b(6) = \frac{b^6 + 45b^5 + 745b^4 + 5595b^3 + 18694b^2 + 21576b}{46656}$$

One benefit of these polynomials is that they let us straightforwardly verify Theorem 4.1 for $0 \leq n \leq 6$ and all $b \geq 3$.

More significantly, we can observe a pattern: for fixed n, we find that $cap_b(n)$ can be written as n^{-n} times a degree-n polynomial in b with integer coefficients. Some searching for these coefficients on the *On-Line Encyclopedia of Integer Sequences* leads us to sequence A060281 (https://oeis.org/A060281) and the *tree polynomials* $t_n(b)$ studied by Knuth and Pittel [24]. Remarkably, the first few tree polynomials are

$$t_0(b) = 1$$

$$t_1(b) = b$$

$$t_2(b) = b^2 + 3b$$

$$t_3(b) = b^3 + 9b^2 + 17b$$

$$t_4(b) = b^4 + 18b^3 + 95b^2 + 142b$$

This coincidence suggests that perhaps

$$cap_b(n) = n^{-n} t_n(b) \tag{2}$$

holds for all $n \geq 0$. It also suggests investigating $cap_b(n)$ using techniques of *analytic combinatorics* [25]. We do this in the following subsections.

A. The tree polynomials

Here we review the theory of tree polynomials as developed by Knuth and Pittel [24]. The starting point is *Cayley's tree function* $T(z)$, which is the formal power series[7]

$$T(z) = \sum_{n=1}^{\infty} n^{n-1} \frac{z^n}{n!} \,. \tag{3}$$

$T(z)$ satisfies a fundamental equation, due to Eisenstein:

$$T(z) = z e^{T(z)} \,. \tag{4}$$

Hence, if we take derivatives with respect to z, we get

$$T'(z) = e^{T(z)} \cdot 1 + z \cdot e^{T(z)} \cdot T'(z) \,,$$

which implies

$$T'(z)(1 - z e^{T(z)}) = e^{T(z)} \,.$$

Now, using equation (4) again, we get

$$T'(z) = \frac{T(z)}{z(1 - T(z))} \,, \tag{5}$$

showing that $T'(z)$ can be expressed in terms of $T(z)$.

Next, the *tree polynomials* $t_n(b)$ are defined implicitly from $T(z)$ by the equation

$$\left(\frac{1}{1 - T(z)} \right)^b = \sum_{n=0}^{\infty} t_n(b) \frac{z^n}{n!} \,. \tag{6}$$

If we let $[z^n] f(z)$ denote the coefficient of z^n in the Taylor expansion of $f(z)$, then equation (6) is equivalent to

$$[z^n] \left(\frac{1}{1 - T(z)} \right)^b = \frac{t_n(b)}{n!} \,. \tag{7}$$

In fact, Knuth and Pittel show that the tree polynomials satisfy a recurrence exactly analogous to Theorem 4.1. The argument starts, for $b \geq 3$, by replacing b with $b - 2$ in equation (6) to get

$$\sum_{n=0}^{\infty} t_n(b-2) \frac{z^n}{n!} = \frac{1}{(1 - T(z))^{b-2}} \,.$$

Taking derivatives on both sides with respect to z, we get

$$\sum_{n=0}^{\infty} n t_n(b-2) \frac{z^{n-1}}{n!} = \frac{d}{dz} \frac{1}{(1 - T(z))^{b-2}} \,.$$

[7]A *formal power series* $\sum_{n=0}^{\infty} c_n z^n$ can be understood simply as a notation for an infinite sequence of coefficients $[c_0, c_1, c_2, \ldots]$, where two such series are equal iff their corresponding coefficients are equal. Formal power series are *not* to be evaluated with a numerical value for the formal symbol z, so questions of convergence are irrelevant. But, as shown by Niven [26], formal power series can in fact be manipulated validly as if they were power series, using the usual operations (addition, subtraction, multiplication, division, differentiation, exponentiation, and functional composition).

Hence, multiplying both sides by z, we have

$$\begin{aligned}
\sum_{n=0}^{\infty} n t_n(b-2) \frac{z^n}{n!} &= z \frac{d}{dz} (1 - T(z))^{2-b} \\
&= z(b-2)(1 - T(z))^{1-b} \, T'(z) \\
&= z(b-2)(1 - T(z))^{1-b} \frac{T(z)}{z(1 - T(z))} \\
&= \frac{(b-2) T(z)}{(1 - T(z))^b} \,.
\end{aligned}$$

Hence we get

$$\begin{aligned}
\sum_{n=0}^{\infty} \frac{n}{b-2} t_n(b-2) \frac{z^n}{n!} &= \frac{T(z)}{(1 - T(z))^b} \\
&= \frac{1 - (1 - T(z))}{(1 - T(z))^b} \\
&= \frac{1}{(1 - T(z))^b} - \frac{1}{(1 - T(z))^{b-1}} \,.
\end{aligned}$$

Finally, by equating the corresponding coefficients of the formal power series, we get the desired recurrence for the tree polynomials:

$$t_n(b) = t_n(b-1) + \frac{n}{b-2} t_n(b-2) \,. \tag{8}$$

B. Relating $cap_b(n)$ and $t_n(b)$

Given equation (8), we can prove Theorem 4.1 by proving equation (2). The crucial idea for how to do this is given by Prodinger [22]. It is based on the following expansion of $1/(1 - T(z))$:

$$\begin{aligned}
\frac{1}{1 - T(z)} &= 1 + \frac{T(z)}{1 - T(z)} \\
&= 1 + z T'(z) && \text{(using (5))} \\
&= 1 + z \sum_{n=1}^{\infty} n^n \frac{z^{n-1}}{n!} && \text{(using (3))} \\
&= \sum_{n=0}^{\infty} n^n \frac{z^n}{n!} \,.
\end{aligned}$$

Hence, starting with equation (7), we can reason as follows:

$$\begin{aligned}
t_n(b) &= n! [z^n] \left(\frac{1}{1 - T(z)} \right)^b \\
&= n! [z^n] \left(\sum_{n=0}^{\infty} n^n \frac{z^n}{n!} \right)^b \\
&= n! \sum_{\substack{x_1, x_2, \ldots, x_b \in \mathbb{N} \\ x_1 + x_2 + \cdots + x_b = n}} \frac{x_1^{x_1} x_2^{x_2} \cdots x_b^{x_b}}{x_1! x_2! \cdots x_b!} \\
&= n^n cap_b(n) \,.
\end{aligned}$$

The second-to-last step follows because the z^n term in the product results from selecting terms from each of the b power series and multiplying them; the sum over x_1, x_2, …, x_b corresponds to all the possible ways of selecting these terms.

Hence we can rewrite equation (8) to

$$n^n cap_b(n) = n^n cap_{b-1}(n) + \frac{n}{b-2} n^n cap_{b-2}(n)\,,$$

which (dividing both sides by n^n) finally completes the proof of Theorem 4.1.

C. Relating $cap_2(n)$ and $Q(n)$

Our discovery that $cap_2(n) = Q(n) + 1$ was the result of searching for the sequence $n^n cap_2(n)$ on the *On-Line Encyclopedia of Integer Sequences*, which led us to sequence A063170 (https://oeis.org/A063170) and Prodinger's paper [22], which proves Theorem 4.2 as part of his proof of the $b = 3$ case of Theorem 4.1. Prodinger's proof uses complex analysis and is quite terse, so we explicate it in Appendix A.

We found an easier proof of Theorem 4.2 that starts from equation (2.8) of [24] (which is also equation (2.3) of [23]):

$$\sum_{n=1}^{\infty} n^{n-1} Q(n) \frac{z^n}{n!} = \ln \frac{1}{1-T(z)}\,.$$

Differentiating both sides with respect to z, we get

$$\begin{aligned}\sum_{n=1}^{\infty} n^n Q(n) \frac{z^{n-1}}{n!} &= (1-T(z))(-1)(1-T(z))^{-2}(-1)T'(z) \\ &= \frac{T'(z)}{1-T(z)} \\ &= \frac{T(z)}{z(1-T(z))^2}\,,\end{aligned}$$

where the last step uses equation (5). Hence, multiplying both sides by z we have

$$\begin{aligned}\sum_{n=1}^{\infty} n^n Q(n) \frac{z^n}{n!} &= \frac{T(z)}{(1-T(z))^2} \\ &= \frac{1-(1-T(z))}{(1-T(z))^2} \\ &= \frac{1}{(1-T(z))^2} - \frac{1}{1-T(z)}\,.\end{aligned}$$

Using equation (6) and equating the corresponding coefficients of the formal power series, we have for $n \geq 1$ that

$$n^n Q(n) = t_n(2) - t_n(1) = n^n cap_2(n) - n^n cap_1(n)\,,$$

which, because $cap_1(n) = 1$, implies that

$$cap_2(n) = Q(n) + 1\,,$$

proving Theorem 4.2.

D. Using an assertion by Ramanujan to bound $cap_2(n)$

The fact that $cap_2(n) = Q(n) + 1$ lets us obtain asymptotic approximations for $cap_2(n)$ from known asymptotic approximations for $Q(n)$. In this section, we use an assertion by Ramanujan to establish an *upper bound* on $cap_2(n)$.

In Ramanujan's first letter to Hardy, dated 16 January 1913, he asserted that

$$\frac{1}{2} e^n = 1 + \frac{n}{1!} + \frac{n^2}{2!} + \cdots + \frac{n^n}{n!}\theta(n)\,,$$

where, for all $n \geq 0$,

$$\theta(n) = \frac{1}{3} + \frac{4}{135(n+k(n))},\ \text{where}\ \frac{2}{21} \leq k(n) \leq \frac{8}{45}\,.$$

A complete proof of Ramanujan's assertion was finally given in 1995, by Flajolet, Grabner, Kirschenhofer and Prodinger as Theorem 7 of [23]. Following that paper, let

$$R(n) = 1 + \frac{n}{n+1} + \frac{n^2}{(n+1)(n+2)} + \cdots\,.$$

Since

$$\frac{n!}{n^n} e^n = \sum_{k=0}^{\infty} \frac{n!}{k! n^{n-k}}\,,$$

it is straightforward to verify that

$$Q(n) + R(n) = \frac{n! e^n}{n^n}$$

and

$$\theta(n) = \frac{1}{2}(R(n) - Q(n))\,.$$

Hence we get

$$\begin{aligned}Q(n) &= \frac{n! e^n}{n^n} - R(n) \\ &= \frac{n! e^n}{n^n} - (2\theta(n) + Q(n)) \\ &= \frac{n! e^n}{n^n} - \frac{2}{3} - \frac{8}{135(n+k(n))} - Q(n)\,,\end{aligned}$$

which gives

$$Q(n) = \frac{1}{2}\frac{n! e^n}{n^n} - \frac{1}{3} - \frac{4}{135(n+k(n))}$$

and, using Theorem 4.2,

$$cap_2(n) = \frac{1}{2}\frac{n! e^n}{n^n} + \frac{2}{3} - \frac{4}{135(n+k(n))}\,.$$

Now Ramanujan's remarkable assertion

$$\frac{2}{21} \leq k(n) \leq \frac{8}{45}$$

enables us to establish upper bounds (and also lower bounds) on $cap_2(n)$. For it gives

$$\begin{aligned}cap_2(n) &\leq \frac{1}{2}\frac{n! e^n}{n^n} + \frac{2}{3} - \frac{4}{135(n+\frac{8}{45})} \\ &< \frac{1}{2}\frac{n! e^n}{n^n} + \frac{2}{3} - \frac{4}{135n}\left(1 - \frac{8}{45n}\right) \\ &= \frac{1}{2}\frac{n! e^n}{n^n} + \frac{2}{3} - \frac{4}{135n} + \frac{32}{6075n^2}\,.\end{aligned}$$

Now, Robbins [27] strengthens Stirling's approximation, for $n \geq 1$, to

$$n! < \sqrt{2\pi} n^{n+1/2} e^{-n} e^{1/12n}\,,$$

which gives

$$cap_2(n) < \sqrt{\frac{\pi n}{2}} e^{1/12n} + \frac{2}{3} - \frac{4}{135n} + \frac{32}{6075n^2}\,.$$

Next we obtain bounds for $e^{1/12n}$. For $n \geq 1$ we have

$$\begin{aligned} e^{1/12n} &= \sum_{j=0}^{\infty} \frac{1}{12^j n^j j!} \\ &= 1 + \frac{1}{12n} + \frac{1}{n^2} \sum_{j=2}^{\infty} \frac{1}{12^j n^{j-2} j!} \\ &\leq 1 + \frac{1}{12n} + \frac{1}{n^2} \sum_{j=2}^{\infty} \frac{1}{12^j j!} \\ &= 1 + \frac{1}{12n} + \frac{1}{n^2} \left(e^{1/12} - 1 - \frac{1}{12} \right) , \end{aligned}$$

which gives us

$$\begin{aligned} \mathit{cap}_2(n) < \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} \\ - \frac{4}{135n} + \left(e^{1/12} - \frac{13}{12} \right) \sqrt{\frac{\pi}{2n^3}} + \frac{32}{6075n^2} . \end{aligned}$$

Because for $n \geq 1$ the sum of the last three terms is negative, we get

$$\mathit{cap}_2(n) < \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} ,$$

proving Theorem 4.4. As already mentioned, this upper bound is also asymptotically correct, since it is the result of truncating the expansion in Theorem 4.3 after its third term.

Finally, we can similarly use Ramanujan's assertion that $k(n) \geq \frac{2}{21}$ to get the asymptotically-correct lower bound

$$\mathit{cap}_2(n) > \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \frac{1}{12}\sqrt{\frac{\pi}{2n}} - \frac{4}{135n} ,$$

proving Theorem 4.5. (We omit the details.) Thus $\mathit{cap}_2(n)$ is tightly bounded on both sides.

VI. Related work

Espinoza and Smith [16] establish bounds on the multiplicative Bayes capacity of a number of different channel compositions. In addition to proving Theorem 2.1 for repeated independent runs, they also consider the case where n *different* channels are run on the same input X, producing a tuple of outputs. They show that the leakage in this case can be far greater than with repeated independent runs—indeed n channels, each having 2 columns, can be constructed such that their composition has multiplicative Bayes capacity of 2^n.

Repeated independent runs channels have also been studied by Boreale, Pampaloni, and Paolini [28]. They prove that $\mathcal{ML}^{\times}(\mathsf{C}^{(n)})$ converges asymptotically to the number of distinct rows of C, and they use the information-theoretic method of types to prove that the rate of convergence is exponential. In [29], the same authors prove stronger rate bounds based on the minimum Chernoff Information between the rows of C.

In the context of timing attacks on cryptosystems, Doychev and Köpf [30] establish leakage bounds for *unpredictability entropy*, which allows them to consider resource-bounded adversaries, in contrast with the information-theoretic adversaries considered here.

VII. Conclusion

Our results establish a new, tight bound on the maximum leakage of n repeated independent runs of a channel with b columns. Our new bound shows that the maximum leakage is actually only about the *square root* of the previously-known bound, and our analytic results allow the new bound to be computed accurately and efficiently.

In future work, it would be of great interest to learn whether the Theorem 3.1 leakage bound, which is proved here for information-theoretic adversaries, can also be carried over to computationally-bounded adversaries.

Acknowledgments

The authors are grateful to Nicolás Bordenabe, Boris Köpf, and Helmut Prodinger for helpful discussions of this work, and to the anonymous referees for their comments and suggestions.

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

[4] 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.

[5] 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.

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

[7] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *Proc. 25th IEEE Computer Security Foundations Symposium (CSF 2012)*, Jun. 2012, pp. 265–279.

[8] A. McIver, C. Morgan, G. Smith, B. Espinoza, and L. Meinicke, "Abstract channels and their robust information-leakage ordering," in *Proc. 3rd Conference on Principles of Security and Trust (POST 2014)*, 2014, pp. 83–102.

[9] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Additive and multiplicative notions of leakage, and their capacities," in *Proc. 27th IEEE Computer Security Foundations Symposium (CSF 2014)*, 2014, pp. 308–322.

[10] G. Smith, "Recent developments in quantitative information flow (*Invited Tutorial*)," in *Proc. LICS 2015: 30th ACM/IEEE Symposium on Logic in Computer Science*, 2015, pp. 23–31.

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

[12] P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. Advances in Cryptology (CRYPTO 1996)*, ser. Lecture Notes in Computer Science, vol. 1109. Springer-Verlag, 1996, pp. 104–113.

[13] B. Köpf and M. Dürmuth, "A provably secure and efficient countermeasure against timing attacks," in *Proc. 22nd IEEE Computer Security Foundations Symposium (CSF '09)*, 2009, pp. 324–335.

[14] D. Brumley and D. Boneh, "Remote timing attacks are practical," *Computer Networks*, vol. 48, no. 5, pp. 701–716, 2005.

[15] 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.

[16] B. Espinoza and G. Smith, "Min-entropy as a resource," *Information and Computation (Special Issue on Information Security as a Resource)*, vol. 226, pp. 57–75, Apr. 2013.

[17] 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.
[18] A. Lacasse, "Bornes PAC-Bayes et algorithmes d'apprentissage," Ph.D. dissertation, Université Laval, Québec, 2010.
[19] M. Younsi, "Proof of a combinatorial conjecture coming from the PAC-Bayesian machine learning theory," *arXiv :1209.0824*, 2012.
[20] W. Y. C. Chen, J. F. F. Peng, and H. R. L. Yang, "Decomposition of triply rooted trees," *Electronic Journal of Combinatorics*, vol. 20, no. 2, 2013.
[21] Y. Sun, "A simple proof of an identity of Lacasse," *The Electronic Journal of Combinatorics*, vol. 20, no. 2, 2013.
[22] H. Prodinger, "An identity conjectured by Lacasse via the tree function," *The Electronic Journal of Combinatorics*, vol. 20, no. 3, 2013.
[23] P. Flajolet, P. J. Grabner, P. Kirschenhofer, and H. Prodinger, "On Ramanujan's Q-function," *Journal of Computational and Applied Mathematics*, vol. 58, pp. 103–116, 1995.
[24] D. E. Knuth and B. Pittel, "A recurrence related to trees," *Proceedings of the American Mathematical Society*, vol. 105, no. 2, pp. 335–349, February 1989.
[25] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge University Press, 2009.
[26] I. Niven, "Formal power series," *American Mathematical Monthly*, vol. 76, pp. 871–889, 1969.
[27] H. Robbins, "A remark on Stirling's formula," *American Mathematical Monthly*, vol. 62, no. 1, pp. 26–29, Jan. 1955.
[28] M. Boreale, F. Pampaloni, and M. Paolini, "Asymptotic information leakage under one-try attacks," in *Proc. FOSSACS '11*, 2011, pp. 396–410.
[29] ——, "Quantitative information flow, with a view," in *Proc. ESORICS '11*, 2011, pp. 588–606.
[30] G. Doychev and B. Köpf, "Rational protection against timing attacks," in *Proc. 28th IEEE Computer Security Foundations Symposium (CSF 2015)*, 2015, pp. 526–536.

APPENDIX A
PRODINGER'S PROOF OF THEOREM 4.2

Prodinger [22] uses complex analysis to prove Theorem 4.2. Because his explanation is terse, here we give additional detail to clarify his argument.

The main job is to compute the coefficients of the Taylor expansion of $1/(1-T(z))^b$. Let

$$f(z) = \left(\frac{1}{1-T(z)}\right)^b .$$

By Cauchy's differential formula, around a circle γ containing a we have

$$f^{(n)}(a) = \frac{n!}{2\pi i}\oint_\gamma \frac{f(z)}{(z-a)^{n+1}}dz\,.$$

Since $f^{(n)}(0) = n![z^n]f(z)$, we have, abbreviating $T(z)$ to T,

$$[z^n]\left(\frac{1}{1-T}\right)^b = \frac{1}{2\pi i}\oint_\gamma \frac{dz}{z^{n+1}}\left(\frac{1}{1-T}\right)^b .$$

By equation (4) we have

$$z = Te^{-T}$$

which implies that

$$\frac{dz}{dT} = e^{-T} - Te^{-T} = e^{-T}(1-T)$$

and

$$dz = e^{-T}(1-T)dT\,.$$

Therefore we have

$$\begin{aligned}
[z^n]\left(\frac{1}{1-T}\right)^b &= \frac{1}{2\pi i}\oint_\gamma \frac{e^{-T}(1-T)dT}{T^{n+1}e^{-(n+1)T}}\left(\frac{1}{1-T}\right)^b \\
&= \frac{1}{2\pi i}\oint_\gamma dT\frac{e^{nT}}{T^{n+1}}\left(\frac{1}{1-T}\right)^{b-1} \\
&= \frac{1}{2\pi i}\oint_\gamma dT\frac{e^{nT}}{T^{n+1}}\sum_{k=0}^{\infty}\binom{b+k-2}{k}T^k \\
&= \frac{1}{2\pi i}\oint_\gamma dTe^{nT}\sum_{k=0}^{\infty}\binom{b+k-2}{k}\frac{1}{T^{n-k+1}}\,,
\end{aligned}$$

where the next-to-last step uses the binomial expansion formula for $b > 1$. Now because

$$e^{nT} = \sum_{j=0}^{\infty}\frac{n^jT^j}{j!}$$

we get

$$[z^n]\left(\frac{1}{1-T}\right)^b = \frac{1}{2\pi i}\oint_\gamma dT\sum_{j=0}^{\infty}\sum_{k=0}^{\infty}\binom{b+k-2}{k}\frac{n^j}{j!T^{n-k-j+1}}.$$

Now we apply a result of complex analysis, which says that for integer n,

$$\oint_\gamma \frac{1}{z^n}dz = \begin{cases} 0, & \text{if } n \neq 1 \\ 2\pi i, & \text{if } n = 1. \end{cases}$$

Hence the only terms in the previous expression that contribute are those where $j = n-k$. And, since $j \geq 0$, we also have $k \leq n$. Hence we can simplify to

$$\begin{aligned}
[z^n]\left(\frac{1}{1-T}\right)^b &= \frac{1}{2\pi i}\oint_\gamma \frac{1}{T}dT\sum_{k=0}^{n}\binom{b+k-2}{k}\frac{n^{n-k}}{(n-k)!} \\
&= \frac{1}{2\pi i}(2\pi i)\sum_{k=0}^{n}\binom{b+k-2}{k}\frac{n^{n-k}}{(n-k)!} \\
&= \sum_{k=0}^{n}\binom{b+k-2}{k}\frac{n^{n-k}}{(n-k)!}\,.
\end{aligned}$$

Having done that, we recall that

$$cap_b(n) = n^{-n}t_n(b) = \frac{n!}{n^n}[z^n]\left(\frac{1}{1-T}\right)^b ,$$

thereby obtaining a new expression for $cap_b(n)$ when $b > 1$:

$$cap_b(n) = \sum_{k=0}^{n}\binom{b+k-2}{k}\frac{n!}{n^k(n-k)!}\,. \tag{9}$$

Specializing to the case when $b = 2$, we obtain Theorem 4.2:

$$cap_2(n) = \sum_{k=0}^{n}\frac{n!}{n^k(n-k)!} = 1 + Q(n)\,.$$

Note finally that Equation (9) lets *every* $cap_b(n)$ be computed via a sum of $n+1$ terms, which is useful when b is big and n is small. In fact, Maple recognizes (9) as the *generalized hypergeometric function* ${}_2F_0(b-1,-n;;-\frac{1}{n})$ and can compute it efficiently for quite large n.