Statistical Measurement of Information Leakage

Konstantinos Chatzikokolakis[1], Tom Chothia[2], and Apratim Guha[3]

[1] Department of Mathematics and Computer Science, Technische Universiteit Eindhoven,
[2] School of Computer Science, University of Birmingham,
[3] School of Mathimatics, University of Birmingham,

Abstract. Information theory provides a range of useful methods to analyse probability distributions and these techniques have been successfully applied to measure information flow and the loss of anonymity in secure systems. However, previous work has tended to assume that the exact probabilities of every action are known, or that the system is non-deterministic. In this paper, we show that measures of information leakage based on mutual information and capacity can be calculated, automatically, from trial runs of a system alone. We find a confidence interval for this estimate based on the number of possible inputs, observations and samples. We have developed a tool to automatically perform this analysis and we demonstrate our method by analysing a Mixminon anonymous remailer node.

1 Introduction

Information theory provides powerful techniques to measure the relation between different probability distributions and so has proved useful for defining anonymity [26, 15, 24, 29, 9, 11] and quantitative information flow [22, 21, 12, 19, 2]. Typically, secret user inputs or users identities are looked on as *inputs* to an information-theoretic *channel* and the publicly observable actions of the system are looked on as the *outputs* of the channel. The information theoretic notion of *mutual information* measures the amount of information that can be sent across this channel, under a particular usage pattern, and therefore measures the amount of information that leaks out about the secret inputs. *Capacity* is defined as the maximum possible mutual information for any input distribution and so equals the worst case leakage.

Previous work using capacity and mutual information to measure probabilistic information leakage has assumed that the exact behaviour of the system, that is the probability of each observation under any user, is known. Typically, one has to construct a model of the system and use a model checker to compute the actual probabilities. Even then, calculating the leakage is not straight forward, requiring specific assumptions about the system [9] or requiring the user to solve a set of equations [18, 11]. In this paper we show that it is possible to quickly and accurately find these measures of information leakage from trial runs of an implemented system. Basing our method on sampled data, rather than say the output of a formal model, has the advantage of removing the need to create an accurate model of the system, which may be very difficult. It also helps to avoid the problem of the state space of the model becoming too big to be handled by model checking tools (a problem even harder for probabilistic

model checking). Finally it is often the case that an information leakage attack exploits implementation faults and so only appears in the implementation itself.

The user of our method defines the inputs of the system, which correspond to the values that we wish to keep secret, and the possible observations an attacker might make, which corresponds to defining the appropriate attacker model. The system under test is then run a number of times until an estimated probability transition matrix can be built up. For our results to hold, the outcome of each trial of the system must be independent of the previous runs. We apply the Blahut-Arimoto algorithm [1, 5] to this matrix in order to estimate the capacity and hence the information leakage of the system.

Running a numerical process on sampled data does not necessarily produce meaningful results so we prove that our estimate converges to the true information leakage. To provide accurate bounds on the information leakage we find the distribution that our estimate comes from. This turns out to be a χ^2 distribution in the case that the capacity is zero and a normal distribution if the capacity is non-zero. In the latter case, the best estimate of capacity is the mean of the distribution minus a small correction. In finding this result we solve the more general problem of finding the distribution of mutual information between two random variables, when the probability distribution of one is known and the other is not. This result also makes it possible to estimate the mutual information of a system for uniform usage, or any other given prior.

The variance of the estimate is dominated by the number of inputs times the number of outputs, divided by the number of samples. Therefore a statistical estimate will be accurate if there are significantly more samples than the product of the number of inputs and all observable outputs. The ability to generate this many samples, in a reasonable amount of time, acts as a guide to which systems can and cannot be analysed statistically. This can be much more efficient than model-checking; complex systems can have many "internal" states, but generate few observations. In this case, generating samples is easier than constructing the state space of the system. If the number of observations is too big, concentrating on some of them may still lead to a useful analysis of the system.

Work outside the field of computer science has dealt with estimating mutual information (e.g. [25, 6]). To the best of our knowledge ours is the first work to deal with estimating capacity. The contributions of this paper are: First, showing that information leakage, as defined by capacity and mutual information, can be automatically calculated from sampled data. Second, proving bounds on the error of the estimate, and so establishing what types of systems can and cannot be meaningfully analysed using a statistical approach. Third, defining a statistical test to detect when there is zero information leakage from a system. We demonstrate our method by analyzing a Mixminion remailer node. We collect data from a running node, using a packet sniffer, and we analyse this data to see if the timing and size of messages leaving a node leaks any information about their destination.

In the next section we motivate our approach and Sections 3 describes our system model. Section 4 describes how we can calculate an estimate of information leakage from sampled data. We find the distribution that our estimate is drawn from in Section 5 and in Section 6 we analyse a mix node. All the proofs and futher examples are given in a technical report [8]. Our toolset and futher examples are available at `www.cs.bham.ac.uk/~tpc/AE`.

Message orderings	out A,B,C	out A,C,B	out B,A,C	out B,C,A	out C,A,B	out C,B,A
in 1,2,3	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666
in 1,3,2	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666
in 2,1,3	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666
in 2,3,1	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666
in 3,1,2	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666
in 3,2,1	0.1666	0.1666	0.1666	0.1666	0.1666	0.1666

(a) Probabilites of outputs for each input for a perfect mix node

Message orderings	out A,B,C	out A,C,B	out B,A,C	out B,C,A	out C,A,B	out C,B,A
in 1,2,3	0	0.3333	0.3333	0	0	0.3333
in 1,3,2	0.3333	0	0	0.3333	0.3333	0
in 2,1,3	0.3333	0	0	0.3333	0.3333	0
in 2,3,1	0	0.3333	0.3333	0	0	0.3333
in 3,1,2	0	0.3333	0.3333	0	0	0.3333
in 3,2,1	0.3333	0	0	0.3333	0.3333	0

(b) Probabilites of outputs for each input for a flawed mix node

Fig. 1. Probabilities of the Message Ordering for Theoretical Mix Nodes

2 Information-Theoretic Measures of Information Leakage

Information theory reasons about the uncertainty of random variables. Given two random variables X, Y we write $p(x) = P[X = x]$ and $p(y) = P[y = Y]$ for their probability distributions and $\mathcal{X}, \mathcal{Y}$ for their sets of values. The *entropy* of X is defined as: $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$ and, when the logs are base 2, measures the uncertainty about its outcome in bits. It takes the minimum value 0 when X is constant and the maximum value $\log |\mathcal{X}|$ when its distribution is uniform. The *conditional entropy* is defined as: $H(X|Y) = -\sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log p(x|y)$ and measures the uncertainty about X that "remains" when we know Y. It takes its minimum value 0 when Y completely determines the value of X and its maximum value $H(X)$ when X, Y are independent.

The *mutual information* between X, Y, defined as $I(X;Y) = H(X) - H(X|Y)$ measures the information that we learn about X if we observe Y. It is symmetric ($I(X;Y) = I(Y;X)$) and ranges between 0 (when X, Y are independent) and $H(X)$ (when X, Y are totally dependent). Finally, the *relative entropy* between distributions p, q is defined as $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$.

A *channel* consists of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$ and a probability matrix W where $W(y|x) = p(y|x)$ gives the probability of output y when x is the input. Given a channel and an input distribution on $\mathcal{X}$, we can define two random variables X, Y representing the input and output of the channel, and with a slight abuse of notation we write $I(X, W)$ for $I(X;Y)$. The *capacity* of the channel is defined as the mutual information between the input and the output, maximised over all input distributions: $C(W) = \max_{p(x)} I(X, W)$.

The analysis of information leakage aims to quantify the amount of information that an attacker can learn from observing a system. Many authors have pointed out the natural parallel between the amount of information an attacker can learn about secret

Message orderings	out A,B,C	out A,C,B	out B,A,C	out B,C,A	out C,A,B	out C,B,A
in 1,2,3	0.0	0.0118	0.0473	0.0118	0.0059	0.9231
in 1,3,2	0.0117	0.0	0.0351	0.0292	0.0	0.924
in 2,1,3	0.005	0.0222	0.0278	0.0444	0.0056	0.8944
in 2,3,1	0.0060	0.012	0.0301	0.0361	0.0060	0.9096
in 3,1,2	0.0067	0.0133	0.04	0.02	0.0067	0.9133
in 3,2,1	0.0061	0.0122	0.0549	0.0244	0.0061	0.8963

Fig. 2. Probabilities of the Message Ordering from Mixminion Experiments

inputs to a system from its public outputs and the amount of information that can be sent over a channel as measured by mutual information and capacity [22, 24, 29, 12, 9, 11]. In such a framework, we have a set $\mathcal{X}$ of events that we wish to keep hidden and a set $\mathcal{Y}$ of observable events which model what the attacker can observe about the protocol. We assume that on each execution, exactly one $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ will happen and that the output of the protocol is chosen probabilistically. The capacity of this system measures the most an attacker can learn about the secret inputs from observing the public outputs, that is the maximum possible information leakage of the system.

As an example we consider one of the basic building blocks of anonymous systems: a mix node [13]. These nodes will listen for encrypted messages and then reorder and forward all of the messages at the same time. The aim of such a node is to make it difficult for an observer to link a sender and a receiver. If we take the example in which 1 sends a message to A, 2 sends to B and 3 to C, then Figure 1(a) shows the behaviour of a perfect mix node. Here we consider if an attacker observing the order of the messages leave the mix could deduce anything about the order in which the messages entered (if an attacker could link these orders then they could work out who is communicating with whom). Each row shows the order in which the messages enter the mix, each column gives the order in which the messages are forwarded, and each cell gives the conditional probability of a particular output resulting from a particular input. The *capacity* of this matrix is zero, meaning that the idealised mix node leaks no information.

In Figure 1(b) we consider a flawed mix node that just reorders a pair of incoming messages. In this case an observer can deduce more about the order of the inputs from the observed outputs and this is reflected by a much higher capacity of 1 bit. For a full discussion of the use and merits of this metric for measuring information leakage we refer the reader to the papers cited above. The aim of this paper is to show how the leakage may be calculated for real systems. Other work in this area uses the entropy [26, 15], conditional entropy [21] and relative entropy [14]. Our methods for calculating mutual information and capacity could also be adapted to compute these measures.

To apply this kind of analysis to a real system we ran a Mixminion remailer node and sent messages across it to three different e-mail addresses. We used a packet sniffer to detect the order in which messages left the node and the results are shown in Figure 2 (a full description of our tests are given in Section 6). In the general case, there is no analytical formula for capacity, we can find the capacity of the matrices in Figure 1 because they are so simple, matrixes such as Figure 2 pose more of a problem. Recently Malacaria et al. [18, 11], showed that the capacity could be found by solving a series of equations, possibly in matlab. However, we wish to fully automate our analysis so

instead we use the iterative Blahut-Arimoto algorithm [1, 5], which can compute the capacity of an arbitrary channel to within a given precision. To explain this algorithm we first observe that mutual information can be written in terms of relative entropy D:

$$\begin{aligned} I(Q,W) = H(Q) - H(Q|Y) &= \textstyle\sum_x \sum_y Q(x)W(y|x) \log \left(\frac{W(y|x)}{\sum_{x'} Q(x')W(y|x')} \right) \\ &= \textstyle\sum_x Q(x) D(W(\cdot|x) \parallel \sum_{x'} Q(x')W(\cdot|x')) \end{aligned}$$

We write $D_x(W \parallel QW)$ as short hand for $D(W(\cdot|x) \parallel \sum_{x'} Q(x')W(\cdot|x'))$. This leads to an upper bound for capacity; by observing that, for any set of numbers $\{n_1, \ldots, n_m\}$ and any probability distribution $\{p_1, \ldots, p_m\}$ it holds that $\sum_i p_i n_i \leq \max_i n_i$, we find that, for all probability distributions Q:

$$\sum_x Q(x) D_x(W \| QW) \leq C(W) \leq \max_x D_x(W \| QW) \tag{1}$$

It can be shown [5] that these inequalities become equalities when Q is the input distribution that achieves capacity.

The term $D_x(W \parallel QW)$ can be thought of as a measure of the effect that choosing the input x has on the output. Blahut and Arimoto showed that the maximising input distribution could be found by repeatingly increasing this measure. Given a channel W, the algorithm starts from an initial input distribution Q^0 (we start from a uniform one, if no better one is known) and in each step k we obtain a new distribution Q^{k+1} by updating the current Q^k for each input x as follows:

$$Q^{k+1}(x) = Q^k(x) \frac{\exp(D_x(W \parallel Q^k W))}{\sum_{x'} Q^k(x') \exp(D_{x'}(W \parallel Q^k W))}$$

The algorithm is guaranteed to converge to the capacity achieving distribution Q. Furthermore, (1) can be used as a stopping criterion, as for any $\epsilon \geq 0$, terminating the iterations when $\max_x D_x(W \parallel Q^k W) - I(Q^k, W) \leq \epsilon$ ensures that the estimate is within ϵ of the true capacity, with equality when the capacity has been found (i.e., $Q^k = Q$). Matz and Duhamel [20] propose an accelerated algorithm. They demonstrate super-linear convergence for this algorithm, and prove linear convergence in the general case.

Applying the Blahut-Arimoto algorithm to the matrix in Figure 2 finds the capacity to be 0.023, however it would be wrong to take this as evidence that there exists a small information leak from a Mixminion remailer node. As our data is from trial runs of the system, we must find a way to distinguish between true information leakage and noise in the results, which we do in the rest of this paper.

3 System Model and Assumptions

As in other work on information theoretic analysis of information leakage [22, 24, 29, 12, 9, 11] a system in our framework consists of a set of secret inputs $\mathcal{X}$, a set of observable output actions $\mathcal{Y}$ and a probability transition matrix W that describes the behaviour of the system. We require that, given one particular secret input, the system behaves probabilistically. This means that if we run the system W with input x then

there must be a fixed probability of seeing each observable output. In statistical terms, given a configuration of the system x the trial runs of the system must be independent and identically distributed: factors other than the input x, that are not accounted for by the probabilities of the outputs, must not have a statistically significant effect on the observed actions.

We consider a passive attacker that observes the outputs of the system and may try to make deductions from these outputs, but does not interact with the system directly. Capacity measures the most information that can be sent over a channel, no matter how it is used, so we do not require anything about the distribution of secret inputs. As long as the attacker does not have any prior knowledge about how the system is being used, there is no sequence of inputs, or clever processing of the observations, that can lead to a higher information leakage.

Given these assumptions, our analysis estimates the information leakage as the information-theoretic capacity of W. This is the maximum amount of information, in bits, that can be passed over W when it is regarded as a communication channel. In terms of anonymity, for instance, it is the maximum number of bits that the attacker can learn about which event took place, on average, from observing the system. An information leakage of $log_2(\#\mathcal{X})$ means that the system offers no anonymity at all, whereas an information leakage of 0 means that the system is perfectly anonymous. A capacity in between these values indicates a partial loss of information. As with any information theoretic measure of anonymity, we do not distinguish between a small chance of a total loss of anonymity and a high probability of a partial loss, rather our figure represents the average case for the average user. We also note that a statistical approach is ill suited to any measure that rates a tiny probability of a total loss of information as much worse than no loss of information because such a measure would not be continuous as the probability tended to zero and so would not allow for accurate confidence intervals to be found.

Our analysis method makes no assumptions about the distribution on secret inputs and assesses the whole system; this means that our results are valid no matter how the system is used but they cannot say anything about a particular observed run of the system. To do so would require one to make assumptions about the prior distribution as part of, for instance, a Bayesian analysis [3]. Such an analysis (e.g. [10, 27]) gives the probability of identifying the culprit from given observations, but would not be valid if the assumptions are wrong or the users' behaviour changes.

4 Estimating Information Leakage

In this paper we focus on capacity as our measure of information leakage, we now describe how it can be calculated. There are two main obstacles to finding the capacity of a real system: firstly we must find a probability transition matrix that reflects the system under test and gives the conditional probabilities of any observable action (the outputs) given a particular usage of the system (the inputs). Secondly we must calculate capacity from this estimated matrix.

To find the probability transition matrix we start by defining the inputs (the events that we wish to keep secret) and the outputs (the actions observable to an attacker).

W	: the true probability transition matrix for the system
$\hat{W}_n$	: estimated probability transition matrix from n samples
Q	: the input distribution that maximises mutual information for W
$Q(\hat{W}_n)$	: the input distribution that maximises mutual information for a $\hat{W}_n$
$C(W) = I(Q, W)$	: the true capacity of W
$C(\hat{W}_n) = I(Q(\hat{W}_n), \hat{W}_n)$	: the true capacity of the matrix found by sampling
$\hat{Q}_m(\hat{W}_n)$	: the result of running the Blahut-Arimoto on $\hat{W}_n$ for m iterations
$\hat{C}(\hat{W}_n) = I(\hat{Q}_m(\hat{W}_n), \hat{W}_n)$	: our estimate of the capacity of W

Fig. 3. Key values for estimating capacity

The latter corresponds to defining an attacker model. Some level of abstraction must be used; the user of our method, depending on the needs of the analysis, should make this choice. Our method requires many more samples than the number of observations so the more fine grained the attacker's observations are, the more samples we require; we quantify this in Section 5 where we calculate the variance of our results in terms of the number of inputs, outputs and samples. Defining the input and output of the channel is a challenging task and should be approached with some care, as it greatly influences the result of an information theoretic analysis. The data processing inequality states that for all functions F and G we have that $I(F(X); G(Y)) \leq I(X, Y)$ and picking a particular set of output actions can be looked on as picking the function G, therefore if we ignore some possible observations the attacker might make we obtain a lower bound for the true leakage. This paper primarily deals with the step after picking the inputs and outputs i.e., how to compute the leakage in a fully automated way.

Once the inputs and outputs are identified we may run trials of the system for each of the inputs and record the observable outcomes. We use these observations to construct an estimated matrix. Note that the approximate matrix can be generated using any probability distribution on the inputs, without having to making any assumptions about how the system is used. Calculating the capacity then finds the input distribution that leaks the most information. So we can collect our data for any usage of the system and then calculate the worst-case scenario.

There are two sources of error in the method we propose. The first comes from estimating the probability transition matrix for the system and the second from the approximation of capacity based on this matrix. Running a numerical approximation on inaccurate data does not necessarily lead to meaningful results, but we prove below that running the Blahut-Arimoto algorithm on an approximate matrix does return a result that tends to the true capacity as the sample size and the number of iterations increase.

The values and distributions used in our results are summarised in Figure 3. Our analysis of a system is based on the probability transition matrix W that gives conditional probabilities of each input given each output, $W(o|a) = p(o|a)$, i.e., the probability of the attacker seeing observation o given that the system is started in configuration a. We will estimate W by running the system n times with a uniform random input each time. This leads to an estimate $\hat{W}_n$, which is a matrix drawn from a normal distribution with mean W and a variance that decreases as n increases.

Next we have the input distribution that maximises the mutual information for W, which we label Q. The true capacity of the system C is given by the mutual information for input Q, denoted by $C(W) = I(Q, W)$. There is no direct formula to find Q exactly, so we estimate Q using the Blahut-Arimoto algorithm for m iterations; we write $\hat{Q}_m(W)$ for this distribution. We may also apply the Blahut-Arimoto algorithm to our estimated matrix to get $\hat{Q}_m(\hat{W}_n)$ which converges to the input distribution that maximises mutual information for the estimated matrix $\hat{W}_n$. This leads to our estimate of capacity for the system: $\hat{C}(\hat{W}_n) = I(\hat{Q}_m(\hat{W}_n), \hat{W}_n)$.

Our proposed method of analysing systems for information leakage is to use a value based on $\hat{C}(\hat{W}_n)$ in place of the true value $C(W)$. The estimated value can be automatically calculated from sampled alone, and the following theorem tells us that this estimate is good, i.e., with enough samples and iterations of the Blahut-Arimoto algorithm our estimate of capacity almost surely converges to the true value:

Theorem 1. *For any probability $p_e > 0$ and any real number $e > 0$ there exists integers n', m' such that for all $n > n'$ and $m > m'$ and for an estimated probability transition matrix found using n samples $\hat{W}_n$ it holds that*

$$p(|I(\hat{Q}_m(\hat{W}_n), \hat{W}_n) - I(Q, W)| > e) < p_e$$

Proof Sketch: Our proof is by contradiction. We assume that $\hat{C}$ does not almost surely converge to C. Mutual information is continuous and finite for a fixed number of inputs therefore our assumptions imply that there must also be a difference between $I(Q(\hat{W}_n), W)$ and $I(Q, W)$ or between $I(Q, \hat{W}_n)$ and $I(Q(\hat{W}_n), \hat{W}_n)$, however if these differences exist then either $Q(\hat{W}_n)$ does not maximise mutual information for $\hat{W}_n$ or Q does not maximise mutual information for W, leading to a contradiction.

5 Bounds on the possible error

To be sure of our results we need to know how close our estimate of capacity is to the real value. There are two ways in which we can find such a bound. We can estimate the error in each of the matrix entries and then calculate the maximum effect that all of these errors might cause on our final result. This method is relatively simple but leads to wide confidence intervals for the final results, we examine this method further in the technical report version of this paper [8]. A second method is to calculate the distribution that our results come from, in terms of the value we are trying to estimate. This method provides much tighter bounds but, due to the maximising nature of capacity, we must relate our results to a lower bound for capacity: $I(\hat{Q}_m(\hat{W}_n), W)$, rather than the true capacity $I(Q, W)$. While this is a lower bound, it is also zero if, and only if, the true capacity is zero:

Lemma 1. *Let $\hat{W}_n$ be a randomly sampled matrix from n samples and $\hat{Q}_m(\hat{W}_n)$ be the result of m iterations of the Blahut-Arimoto algorithm applied to this matrix, starting from a uniform distribution. Then $I(\hat{Q}_m(\hat{W}_n), W)$ is zero if and only if $C(W)$ is zero.*

The process of finding our estimation of capacity can be looked on as drawing a value from a distribution. In this section we show that the value comes from a χ^2 distribution if and only if the true capacity is zero and we also find the mean and variance of the distribution if the capacity is non-zero. This lets us calculate confidence intervals for a bound on the true capacity in terms of our estimated value.

The mean and variance of sampled mutual information has been found in the case that both distributions are unknown [17, 23, 25]. In our case we know the input distribution and only sample to find the outputs. Therefore we first solve the general problem of finding the mutual information when the input distribution is known and the matrix is sampled, then we describe how we use this result to calculate capacity.

5.1 The distribution of mutual information

Let us denote the input distribution by X and the output distribution by Y. Suppose there are I inputs and J outputs. A slight abuse of notions lets us write the proofs in a more readable way, so we write $p_i = Q(i) = P(X = i), i = 0, \cdots, I-1, p_j = P(Y = j), j = 0, \cdots, J-1$, and $p_{ij} = P(X = i, Y = j)$, where the particular distribution (X or Y) is clear from the context. For the estimated values we write: $\hat{p}_{j|i} = \hat{W}_n(j|i)$=the estimated transition probability from input i to output j, $\hat{p}_{ij} = p_i \times \hat{p}_{j|i}$ =the estimated probability of seeing i and j, and $\hat{p}_j = \Sigma_i Q(i)W(j|i)$=the estimated probability of seeing j.

The mutual information can then be written:

$$I(X;Y) = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} p_{ij} \log \left(\frac{p_{ij}}{p_i p_j} \right),$$

and when both inputs and outputs are sampled the mutual information can be estimated as $\hat{I}'(X;Y) = \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \hat{p}_{ij} \log \left(\frac{\hat{p}_{ij}}{\hat{p}_i \hat{p}_j} \right)$, where the $\hat{p}$'s are the relative frequencies of the corresponding states, based on n samples. We also have that: $\hat{p}_i = \sum_{j=0}^{J-1} \hat{p}_{ij}$ and $\hat{p}_j = \sum_{i=0}^{I-1} \hat{p}_{ij}$.

It may be shown that when the inputs have no relation with the outputs, i.e. $I(X;Y) = 0$, then for large n $2n\hat{I}(X;Y)$ has an approximate χ^2 distribution with $(I-1)(J-1)$ degrees of freedom, see [6]. From that, one may say that $\hat{I}'(X;Y)$ has an approximate bias $(I-1)(J-1)/2n$ and approximate variance $(I-1)(J-1)/2n^2$. When $I(X;Y) > 0$, then it may be shown that $\hat{I}'(X;Y)$ has mean $I(X;Y) + (I-1)(J-1)/2n + O\left(\frac{1}{n^2}\right)$ and variance

$$\frac{1}{n} \left(\sum_{i,j} p_{ij} \log^2 \left(\frac{p_{ij}}{p_i p_j} \right) - \left(\sum_{i,j} p_{ij} \log \left(\frac{p_{ij}}{p_i p_j} \right) \right)^2 \right) + O\left(\frac{1}{n^2} \right),$$

see Moddemejer [23]. Brillinger [6, 7] states that this distribution is approximately normal.

In our case the situation is slightly different in that the input distribution is completely known. Hence, the estimate of $I(X;Y)$ is modified to

$$\hat{I}(X;Y) = \sum_{i=0}^{I-1}\sum_{j=0}^{J-1} \hat{p}_{ij} \log\left(\frac{\hat{p}_{ij}}{p_i \hat{p}_j}\right)$$

There exists no known result that deals with the asymptotic behaviour of the mutual information estimates in this situation. In this paper, we develop a distribution of the mutual information estimate for known input distribution when the output is independent of the input, i.e., the mutual information is zero, and then proceed to compute the asymptotic expectation and variance of the mutual information estimate when its actual value is non-zero.

Firstly, for $I(X;Y) = 0$, i.e. X and Y are independent, we have following;

Theorem 2. *When X and Y are independent with distribution of X known, for large n, $2n\hat{I}(X;Y)$ has an approximate χ^2 distribution with $(I-1)(J-1)$ degrees of freedom.*

We note that this theorem implies that if $I(X;Y) = 0$ then $\hat{I}(X;Y)$ is drawn from a distribution with mean $(I-1)(J-1)/2n$ and variance $(I-1)(J-1)/2n^2$.

When $I(X;Y) > 0$, the distribution is no longer χ^2. In this case, we have the following result:

Theorem 3. *When $I(X;Y) > 0$, $\hat{I}(X;Y)$ has mean $I(X;Y) + (I-1)(J-1)/2n + O\left(\frac{1}{n^2}\right)$ and variance*

$$\frac{1}{n}\sum_i p_i \left(\sum_j p_{j|i} \log^2\left(\frac{p_{ij}}{p_j}\right) - \left(\sum_j p_{j|i} \log\left(\frac{p_{ij}}{p_j}\right)\right)^2 \right) + O\left(\frac{1}{n^2}\right)$$

To prove this we rewrite our estimate as: $\hat{I}(X,Y) = H(X) + \hat{H}(Y) - \hat{H}(X,Y)$, where $\hat{H}$ is the entropy calculated from the sampled data. As the distribution X is known we know $H(X)$ exactly. We proceed by taking the Taylor expansion of $\hat{H}(Y)$ and $\hat{H}(X,Y)$ to the order of $O(n^{-2})$. This gives us their expected values in terms of the powers of the expected difference between the entries of the probability transition matrix and their true values. As the rows of the matrix are multinomials we know these expectations (see e.g. [23]). Then, from the expected values of $\hat{H}(Y)$ and $\hat{H}(X,Y)$, we find the expected value of $\hat{I}(X,Y)$.

To find the variance we observe that:

$$V(\hat{I}_{XY}) = V(\hat{H}(X,Y)) + V(\hat{H}(Y)) - 2Cov(\hat{H}(X,Y), \hat{H}(Y))$$

As above we find the variance of $\hat{H}_{XY}$ and $\hat{H}_Y$, and their co-variance from the Taylor expansion and the expectations of the rows of the matrix. As suggested by Brillinger [6, 7] we have verified experimentaly that this distribution is approximately normal.

It may be noted that the expression of the primary ($O(n^{-1})$) part of the variance above reduces to zero when X and Y are independent, which is consistent with variance of the estimate in the case that $I(X;Y) = 0$.

Comparing our result with that of Moddemejer [23], one point of interest is that the distribution of the estimate of the mutual information under independence of the input and the output (i.e. $C(W) = 0$) does not change whether we know the input distribution or not, and the expectation always remains the same, but the variance reduces when there is some information contained about the output in the input (i.e., C(W)¿0).

In both the zero and the non-zero cases we have a bound on the variance:

Lemma 2. *The variance of the estimates of mutual information in Theorem 2 and 3 are bound above by IJ/n where I and J are the sizes of the distributions domains and n is the number of samples used to find the estimate.*

This means that taking more samples than the product of the number of inputs and outputs ensures that the variance will be low and the results accurate. As running the Blahut-Arimoto algorithm on the data we collect can be done in linear time [20] the time taken to collect the sampled data will be the limiting factor of our method. The ability to generate more samples than the product of the inputs and outputs, in a reasonable amount of time, acts as a guide to which systems can and cannot be analysed statistically. We note, however that the variance can actually be much smaller than IJ/n therefore it may also be possible to get a low variance and accurate results with a smaller number of samples.

5.2 Using the distributions for information leakage

Our results on the distribution of mutual information show that the mutual information is zero if, and only if, the distribution of the estimates has mean $(I-1)(J-1)/2n$ and variance $(I-1)(J-1)/2n^2$ (where I is the number of inputs and J the number of outputs). Whereas the mutual information is non-zero if, and only if, the mean is the true value plus $(I-1)(J-1)/2n$ and the variance is the value given in Theorem 3. Therefore our point estimate of information leakage is:

$$max(0, I(\hat{Q}_m(\hat{W}_n), \hat{W}_n) - (I-1)(J-1)/2n).$$

If a single test falls outside the confidence interval for zero mutual information then we may take it as evidence that the capacity is non-zero and calculate the confidence interval accordingly[4]. However a single test cannot distinguish between zero leakage and a very small amount. If the result is consistent with the χ^2 distribution then we may conclude that the result is between zero and the upper bound of the confidence interval for non-zero mutual information. This leads to the following testing procedure:

A test to estimate information leakage

1. Fix the secret inputs and observable outputs of the system under test. Ensure that each run of the system is independent.

[4] Here we follow Brillinger and take the non-zero distribution to be normal.

2. Run n tests of the system with a random input and calculate an estimated matrix $\hat{W}_n$ (to be sure of good results pick $n >> IJ$).
3. Calculate $e = I(\hat{Q}_m(\hat{W}_n), \hat{W}_n)$ and the point estimate for anonymity $pe = max(0, e - (I-1)(J-1)/2n)$, using enough iterations of the Blahut-Arimoto algorithm to make the error in capacity of the estimated matrix much smaller than the accuracy required by the user.
4. If $2n$ times e is inside the 95% confidence interval of the $\chi^2((I-1)(J-1))$ distribution then the confidence interval for the capacity is: 0 to $pe + 1.65\sqrt{v}$ where v is the variance as given in Theorem 3
5. If $2n$ times e is outside the 95% confidence interval of the $\chi^2((I-1)(J-1))$ distribution then the confidence interval for the capacity is: $pe - 1.96\sqrt{v}$ to $pe + 1.96\sqrt{v}$ where v is the variance as given in Theorem 3.

In many situations a very small leakage would be acceptable, however if we want to be sure of zero leakage then we have to run multiple tests and check the goodness of fit of the variance against the zero and non-zero predictions (tests based on the mean will not be able to distinguish zero and very small mutual information). To check compatibility of the variances we use the test that the observed variance divided by the true variance should be approximately χ^2 with mean one and variance two over the sample size minus one [4]. For very small values of mutual information the variance might be consistent with both predictions, however as the variance of the estimate of values that are truly zero is $O(n^{-2})$ and the variance of the estimate of values that are truly non-zero is $O(n^{-1})$ it will always be possible to distinguish these cases with a large enough n. Therefore, even though for large degrees of freedom a χ^2 distribution will start to resemble a normal distribution, a large enough sample size will always be able to tell the zero and non-zero distributions apart, due to the different orders of magnitude of the variances. This leads to the following test:

A test for zero information leakage

1. Fix the secret inputs and observable outputs of the system under test. Ensure that each run of the system is independent.
2. Run 40[5] analyses with sample size n (as described above), to find $\hat{W}1, \ldots, \hat{W}40$.
3. Calculate an estimate of the maximising input distribution $Q_e = Q_m(\hat{W}_1)$, then calculate $I(Q_e, \hat{W}1), \ldots, I(Q_e, \hat{W}40)$ and find the variance of these results: v.
4. Calculate the variance predicted by Theorem 2 v_{zero} and by Theorem 3 $v_{notZero}$.
5. If v/v_{zero} is inside the confidence interval for $\chi^2(2/n)$ and $v/v_{notZero}$ is outside the confidence interval then conclude that the information leakage is zero.
6. If v/v_{zero} is outside the confidence interval for $\chi^2(2/n)$ and $v/v_{notZero}$ is inside the confidence interval then conclude that the information leakage is non-zero.
7. If v is consistent with both predictions then repeat this process with a larger sample size n.

We note that, due to the differences in magnitude of the two variance predictions, this test is guaranteed to terminate.

[5] We use a sample size of 40 as this should be more than enough to accurately find the variance, see e.g. ([28], page 153)

6 Application to the Mixminion remailer

Returning to the Mixminion remailer mix node from Section 2, we can now analyse the data properly. In this experiment we test whether an observer can learn anything about the order in which three short messages entered a mix node by observing messages coming out. Any link between the order of the inputs and outputs would help an attacker tell who was sending a message to whom, which is exactly what the mix is trying to hide. The messages we sent were of different lengths and sent to different e-mail addresses. In the different tests we alternated the order in which the messages entered the mix. So the secret inputs are the orders in which the three test messages arrive.

To find the observable outputs of the node we ran the WireShark packet sniffer on our test machine. This program recorded all incoming and outgoing packets sent to and from the mix node. To ensure that the observations of the packets leaving our mix were authentic we sent our messages to their destination via real nodes of the Mixminion network[6]. Once all the packets had been collected we recorded the size and number of packets sent to each of the destination mix nodes and the ordering of the packets to each node. These digests of the outgoing streams became the outputs of our channel.

In threshold mode the mix strategy is completely independent between firings. While background network traffic and other programs running on the computer may have an effect on the output, we avoid this affecting our results by randomising the order in which the different input messages orderings are tested. Therefore outside conditions will effect all the results equally, and so our experiments fit the requirement of independent and identically-distributed as described in Section 3. To gather our test data we ran our own Mixminion node. We set the mix time limit to be 2 minutes and in each interval sent three known test messages into the node, effectively running it as a threshold mix (we found that the mixes would occasionally take longer than the specified interval, so that if we set the interval for less than 2 minutes our test messages would occasionally straddle the boundary between mix firings and so invalidate our results).

We first ran 1000 tests looking only at the ordering of the packets entering and leaving the mix. The results are shown in Figure 2. Here message 1 was being sent to address A, 2 to B and 3 to C. It was clear that Mixminion usually sent the messages out in a fixed order (C then B then A), however occasionally a different order was observed. Was this unusual ordering, or anything else, leaking information on the order of the incoming messages? Or was it unrelated to the Mixminion software and due to the computer's network card, or network conditions? A quick run of our software finds that the capacity of this matrix is 0.023, which is well within the 95% upper confidence limit for zero leakage (0.0355), therefore there is no evidence of any loss of anonymity.

Next we ran 10000 tests, in batches of a few hundred, over the course of three weeks and, along with the ordering, also recorded the size and number of packets sent. We disregarded the results when there were large amounts of packet loss due to network disruption; we note that this may be a possible way to attack a mix network. We observed 436 different observable outputs in total. The most common observation by far was 33301 bytes in 32 to 34 packets send to each of the other nodes, with overlap-

[6] We only sent messages via nodes where we had received permission from the person running the node, as our test traffic could easily have looked like an attack on the network.

ping streams starting in a fixed order. Occasionally the streams would start in a different order and different numbers of packets, payload size and timings would be observed.

Our software calculated the point estimated of capacity as 0.0249, which is well within the 95% confidence intervals for the χ^2 distribution for the zero case. Leading to a 95% confidence interval for the information leakage as between 0 and 0.0414. Therefore our result is consistent with a capacity of zero and we may conclude that, in this instance, there is no evidence of any loss of anonymity due to the order that messages arrive and leave a Mixminion. There are known attacks that target more complicated aspects of networks of Mixminion nodes; we plan to investigate whether our method can scale up to detect such attacks in the future.

7 Conclusion

The capacity of a channel with discrete inputs and outputs has been proposed as a metric in a number of areas of computer security. We have shown that such measures of information leakage can be calculated from sampled data and so made it possible to apply this theory to real systems. Our calculation of the variance of the estimates can also be used to tell when systems are, or are not, too complex to successfully analyse statistically.

As further work plan to use our tool to look for information leaks from real systems. We also intend to find the distribution of estimates of conditional mutual information and an upper bound for capacity. For this, we can proceed in the same way as finding the lower bound; for conditional mutual information we can find the Taylor expansions of $H(X|Y)$ and $H(X|Y,Z)$ and for an upper bound on capacity we can find the expansion of $D_x(W \parallel XW)$. This would lead to the mean and variance in terms of the expected differences of the matrix entries, which are known. For conditional mutual information we can use the appropriate adaptation of the Blahut-Arimoto algorithm to find our approximation of the maximising input distributions [16].

References

1. S. Arimoto. An algorithm for computing the capacity of arbitrary memoryless channels. *IEEE Trans. on Inform. Theory*, IT-18(1):14–20, 1972.
2. M. Backes and B. Köpf. Formally bounding the side-channel leakage in unknown-message attacks. In *ESORICS*, pages 517–532, 2008.
3. T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philo. Trans. of the Royal Society of London*, 53:370–418, 1774.
4. P. J. Bickel and K. A. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall, 2006.
5. R. E. Blahut. Computation of channel capacity and rate distortion functions. *IEEE Trans. on Inform. Theory*, IT-18(4):460–473, 1972.
6. D. R. Brillinger. Some data analysis using mutual information. *Brazilian Journal of Probability and Statistics*, 18(6):163–183, 2004.
7. D. R. Brillinger. Personal correspondence, April 2009.
8. K. Chatzikokolakis, T. Chothia, and A. Guha. Calculating probabilistic anonymity from sampled data. Technical report, University of Birmingham, 2009.

9. K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206:378–401, 2008.
10. K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. On the bayes risk in information-hiding protocols. *J. Comput. Secur.*, 16(5):531–571, 2008.
11. H. Chen and P. Malacaria. Quantifying maximal loss of anonymity in protocols. In *ASIACCS*, pages 206–217, 2009.
12. D. Clark, S. Hunt, and P. Malacaria. A static analysis for quantifying information flow in a simple imperative language. *J. Comput. Secur.*, 15(3):321–371, 2007.
13. G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *In Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15, 2003.
14. Y. Deng, J. Pang, and P. Wu. Measuring anonymity with relative entropy. In Theodosis Dimitrakos, Fabio Martinelli, Peter Y. A. Ryan, and Steve A. Schneider, editors, *FAST '06*, volume 4691 of *LNCS*, pages 65–79. Springer, 2006.
15. C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *PET '02*, volume 2482 of *LNCS*, pages 54–68. Springer, 2002.
16. F. Dupuis, W. Yu, and F. M. J. Willems. Blahut-arimoto algorithms for computing channel capacity and rate-distortion with side information. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, pages 179+, 2004.
17. M. Hutter. Distribution of mutual information. In *Advances in Neural Information Processing Systems 14*, pages 399–406. MIT Press, 2002.
18. P. Malacaria and H. Chen. Lagrange multipliers and maximum information leakage in different observational models. In *PLAS '08: Proceedings of the third ACM SIGPLAN workshop on Programming languages and analysis for security*, pages 135–146. ACM, 2008.
19. H. Mantel and H. Sudbrock. Information-theoretic modeling and analysis of interrupt-related covert channels. In *Pre-Proceedings of the Workshop on Formal Aspects in Security and Trust (FAST)*, 2008.
20. G. Matz and P. Duhamel. Information geometric formulation and interpretation of accelerated blahut-arimoto-type algorithms. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, pages 66–70, 2004.
21. A. McIver and C. Morgan. *A probabilistic approach to information hiding in Programming methodology*, pages 441–460. Springer-Verlag New York, Inc., New York, NY, USA, 2003.
22. J. K. Millen. Covert channel capacity. In *IEEE Symposium on Security and Privacy*, pages 60–66, 1987.
23. R. Moddemejer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16:233–248, 1989.
24. I. S. Moskowitz, R. E. Newman, and P. F. Syverson. Quasi-anonymous channels. In *IASTED CNIS*, pages 126–131, 2003.
25. L. Paninski. Estimation of entropy and mutual information. *Neural Comp.*, 15(6):1191–1253, June 2003.
26. A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *PET '02*, volume 2482 of *LNCS*, pages 41–53. Springer, 2002.
27. C. Troncoso and G. Danezis. The bayesian traffic analysis of mix networks. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 369–379, 2009.
28. A. J. Wheeler and A. R. Ganji. *Introduction to Engineering Experimentation*. Prentice Hall, 3rd edition, 2009.
29. Y. Zhu and R. Bettati. Anonymity vs. information leakage in anonymity systems. In *Proc. of ICDCS*, pages 514–524. IEEE Computer Society, 2005.