A Statistical Test for Information Leaks Using Continuous Mutual Information

Tom Chothia
School of Computer Science, University of Birmingham, UK
t.chothia@cs.bham.ac.uk

Apratim Guha
School of Mathematics, University of Birmingham, UK
guhaa@for.mat.bham.ac.uk

***Abstract*—We present a statistical test for detecting information leaks in systems with continuous outputs. We use continuous mutual information to detect the information leakage from trial runs of a probabilistic system. It has been shown that there is no universal rate of convergence for sampled mutual information, however when the leakage is zero, and under some reasonable conditions, we establish a rate for the sampled estimate, and show that it can converge to zero very quickly. We use this result to develop a statistical test for information leakage, and we use our new test to analyse a number of possible fixes for a time-based information leak in e-passports. We compare our new test with existing statistical methods, and we find that our test outperforms these other tests in almost all cases, and in one case in particular, ours is the only statistical test that can detect an information leak.**

I. Introduction

Security faults come in all shapes and sizes; it would be misleading to think of computer systems as either perfectly secure, or entirely broken and open to abuse. For example, some systems can be broken by brute force but still provide some protection against a casual observer, whereas other systems might leak a small amount of information that could, over time, be exploited by an attacker. Understanding and measuring the different levels of security that a system might offer is vital if we are going to develop a safe, efficient digital world.

The information theory measurement, *entropy* $H(X)$, measures the amount of uncertainty in a probability distribution. This tells us how hard it would be to guess a value from a probability distribution. Therefore, the entropy of a distribution on some secret values X, provides a measure of the initial security of those values:

$$H(X) = -\sum_X p(x)\log(p(x))$$

If the results of observing a run of a system are given by a probability distribution Y, then the remaining uncertainty of the secret values, after observing the system, is given by the *conditional entropy* of X given Y, written $H(X|Y)$.

Mutual information, $I(X;Y)$ is defined as the difference between the initial uncertainty $H(X)$ and the uncertainty that remains after observing the system $H(X|Y)$. When X is a distribution on some secret values in a system, and Y are the observable actions of a system, the mutual information between X and Y equals the amount of information about the secret values that is leaked by the running a system, and is a popular choice as a measure of the security of a system e.g. [24], [25], [36], [12], [9]. Simplifying $H(X) - H(X|Y)$ we get our measure of the information leakage:

$$I(X;Y) = \sum_X \sum_Y p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \qquad (1)$$

The observations that lead to an information leak often come from a continuous domain (such as time or power measurements). One way to deal with this would be to split the continuous domain into a finite number of bins, and treat two values in the same bin as identical, so making our data discrete. However, in doing so, we would lose a lot of the information that these observations hold and we might possibly miss evidence of an information leak, as we show in Section V.

In this paper we show that leaks from systems with continuous outputs can be effectively analysed by using a continuous[1] version of mutual information, given by the equation:

$$I(X;Y) = \sum_X \int_Y p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)\, dy \qquad (2)$$

The most prudent choice of bin size to use when making continuous observations of a system Y discrete would be the highest possible resolution with which any attacker can measure Y. In general, such a bound on the power of the attacker will not be known, and may not even exist. If we make a continuous Y discrete, by placing it into bins, the value of discrete mutual information (Equation 1) tends to the value of continuous mutual information (Equation 2) as the bin size with which Y is measured tends to zero. This means that continuous mutual information is equal to the information leaked by the system when the attacker can make arbitrarily accurate observations of the Y values, and therefore it is a safe

[1] This is technically a hybrid, continuous/discrete version of mutual information, however we will refer to it as continuous to stress the difference between it and the wholly discrete version.

estimate of the true information leakage from a continuous system.

We use this continuous version of mutual information to develop a statistical test to decide when the results of trial runs of a probabilistic system do or do not indicate the presence of an information leak. Answering this question is a key step in finding many practical attacks; often the response times to particular messages leak some information about what is going on inside the system (e.g. [26], [11]), and this can be enough information to completely break a key [17] or reveal the identity of a user [27]. Such leaks are often immediately obvious from plots of the sampled data; when the date is more ambiguous some statistical tests can be used, the particular test is often picked on an ad-hoc basis, as most tests are quite specific in the kinds of systems they can be applied to.

The test we propose has the advantage of being general and based on the theoretically meaningful notion of mutual information. We show in sections V and VI that our test outperforms two sample tests and the process of test the data by placing it into bins and using the discrete version of mutual information. In particular, in some of our examples all these other methods fail to detect an information leak that our new test finds. This shows that the test we propose is not just of theoretical interest; it is also a practical tool that can immediately be applied to find information leaks.

It has been proven that there is no universal bound on the convergence rate for sampled mutual information. However, we prove that, under some reasonable conditions, the estimate of leakage from a single bit to continuous values with finite support[2] does converge. This result tells us that we can accurately approximate mutual information in a reasonable number of samples.

We use this result to develop a statistical test to detect the presence of an information leak. To test a particular system, we perform a number of trial runs and for each run record the secret value x and the observable action that the system produces y. If no better distribution of the possible secrets is known, we use the uniform distribution on the secrets to generate these samples. We then use kernel density estimation to estimate the probability density function $p(y|x_i)$ for each of the possible secret values x_i. We then calculate an estimate of the mutual information using the composite rectangle method and Equation 2. We prove that, if the true leakage is zero, there is a positive bias in this result. Therefore, we need a method of telling when a very small result does or does not indicate the presence of an information leak.

To find out what zero information leakage should look like, we return to our sampled data and for each pair of secret and observable actions (x, y) we replace the secret value with another secret value chosen at random. The mutual information for this shuffled data must be zero, because now the secret values are unrelated to the observations. Calculating the estimate of mutual information for this shuffled data and then repeating this process a large number of times gives us a baseline for what zero information leakage will look like. We then compare the original estimate of the mutual information with this baseline for zero leakage, and if it is sufficiently different, we decide that the data provides evidence of an information leak. If it is not sufficiently different, we decide that there is no evidence of an information leak in our data.

The "two-sample" case, where we want to measure leakage from one of two possible options to a continuous value, is a common situation in many practical attacks (e.g. [26], [27], [11]). While our convergence result only holds for these cases, our test can be used with an arbitrary number of discrete secret values and observations from a possibly infinite support. We perform some simulations to show that we get a reasonable convergence rate and we find that our test outperforms other statistical two sample tests in most situations. This provides evidence that our test is the best statistical test to use when looking for information leaks.

We note that the test we propose is a statistical test and, as with all statistical methods, the results of the test do not prove that there is or is not an information leak in a system. Rather the test we propose is a practical tool to help analysts decide if some collected data indicates the presence of an information leak. Likewise, we also do not prove that our test is better than other existing statistical tests instead, using a number of simulations and examples, we show that in almost all cases our test performs better and can find information leaks that other tests miss.

We use our test to examine some possible fixes to an information leak in RFID e-passports. It has been shown that it is possible to trace a given passport by replaying a particular message and monitoring the response times [11], i.e. there is an information leak from the identity of the passport to the response time of the message. The obvious way to fix this information leak is to pad the response time for error messages. We examine the effect of padding the response time by a fixed interval, and by making the RFID chip perform extra calculations. We show that the fixed interval method does not completely fix the leak, whereas performing extra calculations does.

The main contributions of this paper are:

- Showing that continuous mutual information can be used to measure information leakage.
- Proving that, in some cases, there is an upper bound on the convergence rate of estimates of mutual information.
- A test for the presence of an information leak, which uses kernel density estimation to find an estimation of the mutual information, and then shuffles the data to find a baseline for zero leakage.
- Using this test to find a fix for an information leak in e-passports.

In the next section, we describe background and related work. In Section III we present our convergence result and in Section IV we develop a test for information leakage. In Section V we use our test to find a fix for the information leak in e-passports, and in Section VI we use simulated data

[2] The support of a distribution is the smallest closed interval whose complement has probability zero. The support is finite if it has a finite upper and lower bound.

to compare our test with other statistical tests. We conclude in Section VII. Proofs and further comparisons with other statistical tests can be found in a technical report [10].

II. Background

We use a standard system model for information theoretic security [24], [12], [9], [3], [20], a system in our framework consists of a set of secret inputs $\mathcal{X}$, a set of observable output actions $\mathcal{Y}$, the system is run with a single secret input x and produces a single observable output y. We require that, given one particular secret input, the system behaves probabilistically. This means that if we run the system 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 note that we only measure the possible information leaks from the distribution X to the distribution Y. If we find no evidence of an information leak from X to Y, that should not be taken to mean the system is completely safe, as other ways of interacting with a system might produce different observable actions which are unrelated to Y.

When $\mathcal{X}$ and $\mathcal{Y}$ are both discrete sets, mutual information is a useful metric to measure the amount of information that leaks from the secrets of the system to the observable actions. Given two random variables X, Y, we write $p_X(x) = P[X = x]$ and $p_Y(y) = P[y = Y]$ for their probability mass functions, and we write $p_{XY}(x, y)$ for the joint distribution of X and Y. We will drop the subscripts when they are clear from the context, and we will use X to stand for both a random variable and that variable's distribution. The mutual information of X and Y tells us how much we can learn about one of these variables from observing the other, and is given by the equation:

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{3}$$

We take the logs base two, therefore the value of mutual information tells us the number of bits of information we learn about X from observing Y. It takes its minimum possible value 0 when X and Y are independent, and there is no leakage from X to Y. The maximum possible value is equal to the entropy of X, $(H(X))$, and this indicates that X and Y are totally dependant.

As described in the introduction, mutual information is equal to the difference between the uncertainty in X and the uncertainty in X after observing Y. One way to understand this is to note that:

$$\frac{p(x,y)}{p(x)p(y)} = \frac{p(x|y)}{p(x)}$$

i.e., the ratio of the *posteriori* to the *proiori* probability. Therefore, the log term in Equation 3 tells us the information provided about a particular secret x by a particular observation y. The average information provided by Y about X is then found by summing these individual terms for x and y values times by the probably that those particular x and y occur.

There are a number of ways of calculating mutual information for a particular system. If the exact behaviour of the system is known, it can be calculated by hand. When the system is simple enough, it may be possible to construct a formal model in the probabilistic model checker such as PRISM and use this to calculate the possible information leakage [7]. Backes and Köpf [3] provide an algorithm for computing conditional entropy as a measure of unknown-message side-channel attacks. McCamant and Ernst [23] calculate information flow for C-like programs using a combination of static and dynamic analysis. Their framework is powerful enough to find information flow leaks in C programs, however it can be tripped up by programs with a complicated internal state space and may require the source code to be annotated. Malacaria and Heusser have shown that the leakage of a deterministic program can be found using the CBMC model checker [15] and Newsome et al. [28] quantify the information flow along a single path of x86 binaries.

In a recent paper [8] we introduced a method of calculating the information leakage of a system from trial runs of an implementation alone. This has the advantage of being applicable to systems that are probabilistic and where the exact internal behaviour is not known. It can also capture implementation faults, which could easily be left out of a formal model. We found that for accurate results it was necessary to collect a number of samples that was larger than the product of the size of the domain of secrets and the possible number of observations. This leads to better performance than many model checking approaches, but means that it is still not possible to check systems with particularly large domains of secrets or observations.

Gierlichs et al. [14] have suggested the use of mutual information for power analysis of smart cards, and used kernel density estimation to calculate continuous mutual information [4]. They look at smart cards that are known to have an information leak and use mutual information as a distinguisher to find the best guess at the key. In their framework, a smart card uses a secret key k to process plain texts drawn uniformly from some distribution P. The observable behavior of the device can be denoted by $Y_{k,P}$. The attacker wishes to guess either the whole key k, or some part of it, and has a model of how the leakage occurs $X_{j,P}$ for a particular (sub)key j. To discover the device's secret key the attacker looks for the j which maximizes the estimation of $I(X_{j,P}; Y_{k,P})$. This is a powerful method, which can extract cryptographic keys from supposedly secure smart cards, using a model of how the leakage occurs. In contrast, our work presents a general-purpose statistical test to detect the presence of a leak, and shows that the estimate used for this test converges quickly.

Other information theory measures have also been proposed for computer security examples include capacity (e.g [9]), which is the worst case mutual information, and min-entropy (e.g. [34]), which gives the difficultly of breaking a system in a single guess. The methods we present in this paper could be adapted to these measures. Another popular measure is the conditional entropy of the secret values and observable actions (e.g. [3], [20]). This value gives the difficulty of finding the secret value, rather than the amount of information leaked. Conditional entropy is equal to the mutual information minus the entropy of the secret values. Therefore, the test we present here could easily be transformed into a test of conditional entropy.

Köpf and Dürmuth [19] and Köpf and Smith [21] use information theory to analyse time-based side channel attacks. They treat time as discrete, and use conditional entropy as their measure of security. They found bounds on the possible leakage and used these to design effective counter measures that can guarantee a low level of leakage.

There are a number of results that show that, in the most general case, estimates of information theoretical values may converge arbitrarily slowly when calculated from sampled data [5], [2], [29]. These results might suggest that trying to calculate mutual information from sampled data would not be effective. However, we prove that, under some reasonable conditions, an estimate of mutual information converges quickly, therefore suggesting that our test will be effective with a reasonable number of samples.

The statistical test we propose is based on two hypotheses, the first $\mathbb{H}_0$ is that there is no information leak in the system, and the second $\mathbb{H}_1$is that an information leak is present. We take $\mathbb{H}_0$ to be our null hypothesis and we test the sampled data, at a given confidence level (typically 95%) to see if it is consistent with this hypothesis. If it is, we conclude that the sample data does not provide any evidence of an information leak. If the test fails, then we reject the null hypothesis and conclude that there is an information leak in the system. We note that as with any statistical test, we do not provide proof of truth or falsehood of the hypothesis, rather we aim to provide a practical method with which to analyse systems for information leaks.

Our test is nonparametric, i.e., it does not assume an underlying distribution on the data. When looking at the two-sample case there are a number of other nonparametric statistical tests that can be used in order to tell if two samples came from the same distribution. The most popular of these are the Kolmogorov-Smirnov (KS) test [13], the BWS test [6], the Anderson-Darling (AD) test [30] and the Cramér-von Mises (CVM) test [1]. We compare our new test to each of these and find that in most cases it has much better performance. We refer the reader to the cited papers for details of these tests.

III. A Convergence Result for Mutual Information

A. Motivation

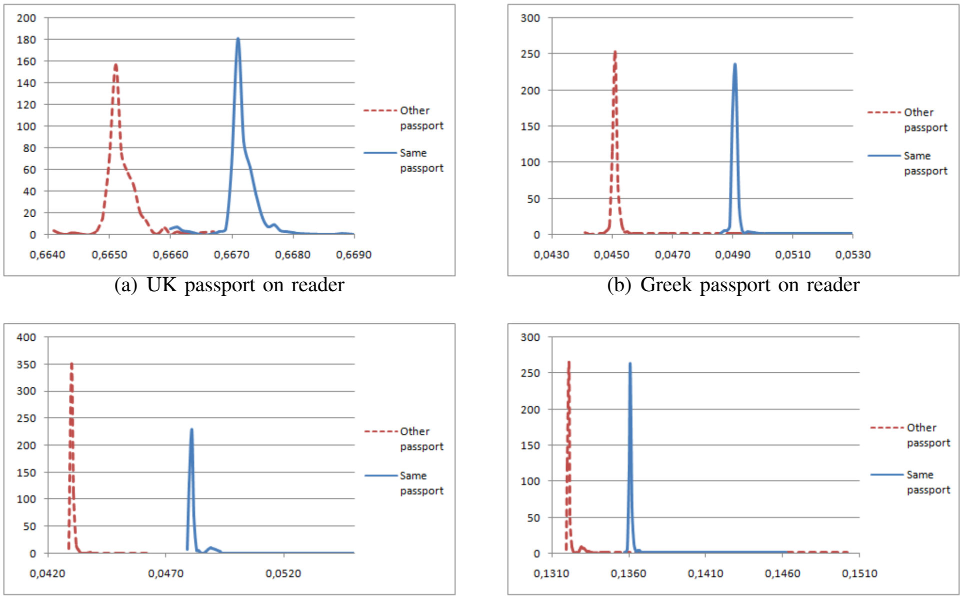
We initially developed our statistical test in order to be able to verify possible fixes for an information leak in e-passports, so we use this as a motivating example. E-passports contain an RFID chip that will broadcast the information printed on the passport, a JPEG copy of the picture, and possibly the passport owner's fingerprints, all this data is signed by the issuing country. All e-passports conform to the same protocols, but each of the countries we have looked at use different hardware and software. The access to this information is protected by a cryptographic protocol that, amongst other goals, aims to make the passport untraceable [16] i.e., after observing a run of a particular passport, it should not be possible to detect the presence of that particular passport in the future.

Each passport has a unique encryption and MAC key, and one of the initial messages to the passport is encrypted and MACed using these keys. This message contains a random nonce, so cannot be used directly to trace a passport. However, it is possible to trace a particular passport by recording this message, and replaying it when we want to check for the presence of that passport in the future. If the passport we replayed the message to was a different passport, the MAC check would fail and an error message would be issued quickly. If, on the other hand, it was the same passport again, the MAC check would pass (because the message was formed using that passport's unique MAC key). The passport would then decrypt the message and check the nonce, which would not match, and only then would the passport send an error message.

The extra time it takes for the passport to do the decryption and check the nonce is clearly noticeable. Therefore this time delay can be used to tell if a particular passport was the one used in the session than generated the message. So there is an information leak from the identity of the passport to the time it takes to receive the error message. The exact response times vary between runs, due to difference in the power supplied to the chip, interference, etc. We plot the times it takes the British, German, Greek and Irish passports to reject the replayed messages in Figure 1. The dashed, red line shows the time it takes to reject the message when it is sent to the passport we are trying to trace, and the solid, blue line shows the rejection times when it is some other passport. We refer the reader to a previous paper [11] for full details of the attack.

In terms of our information theoretic model we take our secret value X to be 1 if we are replaying the message to the same passport, and 0 if we are replaying it to a different passport from the same country (passports from different countries can be distinguished by other means [32]). We take Y to be the response time of the error message. The information leakage from the passport can then be measured as $I(X;Y)$.

Looking at the plotted data in Figure 1 the information leak is clear; there is no particular need to use mutual information to measure this leak. However, we are interested in finding a simple fix for this leak, and when trying to fix the leak, it would be easy to get the plots to look similar but for the leak to still to exist and be detectable with the right analysis. So when trying to fix this leak we do need a statistical test that can tell us with some certainty if a leak exists.



(a) UK passport on reader

(b) Greek passport on reader

(c) Irish passport on reader

(d) German passport on reader

Fig. 1. Sampled Times from Replaying a Message to the Same or a Different Passport

B. Continuous Mutual Information

We measure the response time of the passport error message by adding a clock to the Python program that we use to interact with the passport. This gives us time measurements, in seconds, to 9 decimal places. Given our measuring framework only about the first 5 decimal places of the measurement could contain meaningful data, but this still presents us with a problem, when treating this as discrete data. For discrete sampling, the number of samples needed should be proportional to the number of possible observations, so using the data to 5 decimal places would require tens of thousands of samples, taking days to collect. Rounding the data further, would mean we would lose much of the information in the measurements, so while fine for detecting leaks, we could not rely on this to show the absence of an information leak.

The problem here is that discrete mutual information treats each observation as separate and unique; there is no notion of a particular observation being more or less similar to any other. Therefore, when treating time measurements as discrete values we lose some of the information in the measurements, and may miss an information leak or need more samples that could be collected in a reasonable amount of time.

We solve this problem by treating the time measurements as continuous values. Instead of estimating a discrete probability mass function (pmf) from the observations of Y, we estimate a continuous probability density function (pdf). From this we calculate an estimate of the mutual information for the system:

$$\hat{I}(X;Y) = \sum_{x\in\mathcal{X}} \int_{y:\hat{p}(x,y)>0} \hat{p}(x)\hat{p}(y|x)\log\left(\frac{\hat{p}(y|x)}{\Sigma_{x}\hat{p}(x')\hat{p}(y|x')}\right) dy \qquad (4)$$

which is derived from Equation 2. This equation represents the most general case, when we are estimating both the X and the Y distributions. If we do not have a source from which to estimate X, we can generate our samples using a uniform distribution and either estimate X from that or plug in the exact values for $\hat{p}(x)$.

If we take N samples $(X_1, Y_1), \ldots, (X_N, Y_N)$, we estimate the pmf $\hat{p}(x)$ using a simple proportion:

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N}\chi_{\{X_i=x\}}$$

where χ_B is the indicator function of a set B (ie., $\chi_B(x) = 1$ if $x \in B$ and otherwise $\chi_B(x) = 0$).

We write $(x, Y^x_1), \ldots, (x, Y^x_{Nx})$ for the N_x samples that used the secret value x. The mixed conditional distribution of continuous y given discrete x is estimated using the kernel density estimate (see e.g. [33]):

$$\hat{p}(y|x) = \frac{1}{N_x h}\sum_{i=1}^{Nx} K\left(\frac{Y^x_i - y}{h_N}\right)$$

This equation estimates the probability of a particular value of y given some x by looking through every observation that resulted from x, i.e., $Y^x_1, \ldots, Y^x_{Nx}$. For each of these observed values, we look at how far it is from the value we want to estimate y and apply the kernel function K to decide how that particular observation should affect our estimate at y. Here h is the *bandwidth*, this value controls the distance from y at which an observation will have an effect on our estimate of $p(y|x)$.

There are a number of possible choices for the bandwidth and the kernel; many of these are particularly good when the observations are known to follow a given pattern, (e.g. the sum of normal distributions). As we do not want to make particular assumptions about the observations, we use a general-purpose bandwidth and kernel that will work reasonably well in all situations.

Silverman [33] provides a comparison of a number of different bandwidths and kernel functions. He suggest the following general purpose bandwidth:

$$h_{\text{OPT}} = 1.06\text{SD}(Y)N^{-1/5}, \tag{5}$$

where $\text{SD}(Y)$ is the standard deviation of Y. and We use the Epanechnikov kernel, which lets any observation, within h of y, have an affect of the estimate on y that tails off quadratically:

$$K(u) = \frac{3}{4}(1-u^2)\chi_{\{|u|\leq 1\}}$$

If the user of our test has particular knowledge about the distributions of Y, e.g. it is approximately a sum of normal distributions, then other kernels and bandwidths may be used. Once we have we have an estimate of $p(y|x)$ for every possible value of y we can use the composite rectangle method to compute the mutual information estimate from Equation 4.

C. A Convergence Result for Zero Mutual Information

To make accurate statements about our system using the value $\hat{I}(X;Y)$ from Equation 4, it is useful to know how it relates to the true value of mutual information, and how it may vary.

It is known that there is no universal rate at which the error in estimation of mutual information goes to zero, no matter what estimator we pick, see [2] and [29]. A better, and more positive, result can be obtained by looking at the convergence rate in the case of zero leakage, under reasonable regularity conditions.

We assume that the pairs $\{X_i, Y_i\}, 1 \leq i \leq N$ are independent and identically distributed (IID) satisfying

A1. Y_i's are bounded continuous real-valued random variables with finite support.
A2. For $u = 0, 1$, $p(u, y)$ has a continuous bounded second derivative in y;
A3. K has a finite support symmetric around zero, and integrates to 1.
A4. $h \to 0$, $Nh_N^2 \to \infty$ and $Nh_N^4 \to 0$ as $N \to \infty$.

The first condition states that the observations must be continuous and they must have finite support, i.e., the smallest closed interval whose complement has probability zero must have finite upper and lower bounds. This condition is important for our convergence proof, but as we will see in Section VI our test works well in a number of cases where the observations have infinite support.

Condition A2 restricts our proof to measuring the leakage of a single bit. We hope to be able to remove this restriction in further work, and we show that our test works on some multi-input cases in Section VI. Condition A2 also restricts the observation's distribution to have a bounded second derivative, meaning that it must be reasonably smooth. We would expect this from any observations collected from a real system, and this condition can easily be checked by a visual inspection of a plot of the data.

Conditions A3 and A4 restrict the possible kernels and bandwidths we may choose. These conditions are easy to fulfil, and are met by the choice of kernel and bandwidth we suggest above.

Under the null hypothesis that there is no information leak, i.e.

$$\mathbb{H}_0 : X \text{ and } Y \text{ are independent},$$

a large sample distribution for $\hat{I}_{XY}$ is given by the following theorem:

Theorem 1: Under $\mathbb{H}_0$ and the assumptions A1-A4, we have that $Nh^{-1/2}(\hat{I}_{XY}/\log(e) - C_1/(Nh))$ converges to a normal distribution with mean 0 and variance $C_2 = 0.5\int(\int K(w)K(v+w)dw)^2 dv \int \chi_{p(y)>0}\ dy$ as $N \to \infty$, where $C_1 = 0.5\int K^2(v)\ dv \int \chi_{p(y)>0}\ dy$.

The proof of this theorem is available in a technical report [10]. We note that C_1 and C_2 only depend on the choice of kernel function K and the support of Y. Therefore, they will be constant for any set of sampled data. The value C_1/Nh is the bias in the result, therefore, in the case of zero leakage we would expect $\hat{I}(X;Y)$ to be close to this value. The $log(e)$ term is required because we are using logs base 2.

The $\int \chi_{p(y)>0}\ dy$ term in C_1 and C_2 equals the total length of the regions for which the probability of y is not zero. Our theorem only applies in the case that this is not infinite, hence the requirement A1 that the support is finite.

By rearranging the terms in Theorem 1 we can find the distribution of $\hat{I}_{XY}$ when X and Y are independent. When the sample size is large enough, $\hat{I}_{XY}$ approximates the following normal distribution:

$$\hat{I}_{XY} \sim \mathcal{N}\left(\frac{C_1 \log(e)}{Nh}, \frac{C_2(\log(e))^2}{N^2h^{-1}}\right)$$

We note that the standard error (the standard deviation above) converges to 0 faster than the bias (the mean). So for a large number of samples the bias will be the dominating factor. Looking at just the mean we can observe that:

$$\hat{I}_{XY} \approx \frac{C_1 \log(e)}{Nh} \tag{6}$$

These results tell us that the convergence rate depends on the bandwidth h. Condition A4 requires h to be strictly between $N^{-1/2}$ and $N^{-1/4}$. If we take $h = N^{-(1/4+\delta)}$ for a small positive δ, Equation 6 tells us that:

$$\hat{I}_{XY} \approx \frac{C_1 \log(e)}{N.N^{-(1/4+\delta)}} \approx O(\frac{1}{N^{(3/4-\delta)}})$$

This tells us the rate of convergence for our estimate of mutual information under the conditions A1 to A4 and the independence of X and Y:

Corollary 1: Setting $h = N^{-(1/4+\delta)}$ for a small $\delta, 0 < \delta < 1/4$, a rate of convergence of $N^{-(3/4-\delta)}$ for the MI estimate $\hat{I}(X;Y)$ can be achieved.

Proof Sketch: Theorem 1 tells us that $\hat{I}(X;Y)$ converges to 0 at the same rate as C_1/Nh, which is $C_1 N^{-(3/4-\delta)}$ for our choice of h. Hence the result follows.

This corollary implies that when conditions A1 to A4 are satisfied the mutual information estimate is an efficient one due to its high rate of convergence, and hence the performance of the mutual information test statistic may be expected to improve quickly with larger sample sizes. In turn, this means that the test we propose below will produce good results in a reasonable number of samples.

We conjecture that we will get good, accurate results in a much wider range of situations, including all those that we would expect to see when looking for information leaks in a computer system. In Section VI we perform a number of simulations that show good results when looking at systems with many secret values (relaxing Condition A2) and some distributions with infinite support (relexing Condition A1), such as the normal and t-distribution. As future work we would like to relax the conditions on our system and prove a good convergence rate in a wider range of situations.

IV. A Statistical Test for Information Leakage

We test a system by collecting samples from a number of trial runs. For each trial we randomly pick a secret value x and then run the system and record the public observable output. This gives us a pair of values (x, y), we write Y_x for the distribution of y for the given x. We then run enough tests to obtain a smooth line for y, for all possible x values. There are a number of statistical tests that can tell us when we have enough data, e.g. cross validation (see e.g. [18]) however, it is usually clear from a plot of the data alone.

For the two-sample case, with known, finite support, we could use Theorem 1 to calculate the bias and construct the confidence interval for values that would be compatible with zero information leakage. We choose not to make this the basis of our test because, first, this interval is not trivial to calculate and so this does not lend itself to automation, and second we want to develop a test that can also be applied to systems with large secret domains and possibly infinite support for the observations.

From our sampled data, we construct new data sets that are guaranteed to have a mutual information of zero: For each sampled pair (x, y) we replace x with another value chosen randomly from the distribution X. This gives us a new set of data in which the x value has no relation to the y and so we know that the mutual information for this data set is 0. We now repeat this shuffling process a large number of times[3] and calculate the $\hat{I}(X';Y)$ statistic for each data set to give us $I_1, \ldots I_K$. As these new data sets will have the same support as the true data set, in the two-sample case, the mean of these values will approximate the bias, giving us C_1 from Theorem 1, and the variance of these results will approximate C_2. Therefore these $I_1, \ldots, I_K$ values tell us exactly what zero information leakage should look like for the system we are currently testing.

We now compare the mutual information estimate $I = \hat{I}(X;Y)$ to the $I_1, \ldots I_K$ baseline. To perform the test at a level α we then check that that the I value is greater than the $100(1-\alpha)$ percentile of the baseline. If it is then we reject the null hypothesis and conclude that there is an information leak. Alternatively, we can estimate the *p-value* for this test statistic [31], which is equal to the proportion of the $I_1, \ldots I_K$, which are larger than I. A p-value approximates the probability of observing a test statistic at least as extreme as the actual result, assuming the null hypothesis is true. Therefore, a p-value close to 0 may be taken as evidence of an information leak. More formally, our test consists of the following steps:

Test 1: Given a system with observable actions $\mathcal{Y}$ and secret values $\mathcal{X}$, with the distribution X. To test for an information leak, under the null hypothesis that there is no leak:

1 Picking x_i from the distribution X, collect samples Ys_i to estimate $p(y, x_i)$, check that enough samples have been collected to correctly estimate $p(y, x_i)$, (e.g. by cross validation, or visual inspection of the results).

2 As described in Section III.B, calculate the mutual information test statistic I for this data using kernel density estimation, the composite rectangle method and Equation 4.

3 Combine all the sampled observations $Ys_0, \ldots Ys_K$ in one single sample which we denote by Ys.

4 For each element of Ys, simulate a new random x using the distribution X. Denote the value of the estimated mutual information by I_1. We repeat this step a large number of times[2], say K, and obtain the “bootstrapping” samples $I_j; j = 1, 2 \cdots, K$.

5a If the test statistic I is above the $100(1-\alpha)$th percentile of the sampling distribution of $I_1, \cdots, I_K$, reject the null hypothesis and conclude that there is a leak.

5b Alternatively, an estimated p-value of the test statistic can be computed as the percentage of $I_1, \cdots, I_K$ exceeding the observed MI for the test sample, say I.

We provide software support for this test in the form of a Java jar file and an R program. The Java jar file can be run on a text file that contents sampled data in the form of `(secret value, observed action)`. The program then follows the steps described above and computes the p-value. As the R language is the de-facto standard language amongst statisticians, we provide an implementation in R as a way of making our test more accessible to the statistics community.

[3] 100 or more shuffles will be enough to get a useful baseline for zero leakage, but, if practical, a few thousand shuffles will make the test more powerful.

V. Fixing e-Passports

We now use out test to compare two possible fixes for the e-passport information leak. We first examine the effect of simply padding the response time, and then look at rewriting the passport code to remove the leak. We also compare our test with other statistical tests from the literature, and we find that our test performs the best, detecting attacks that all of the other statistical tests miss.

For this test, we replay a message to a passport and look for any relationship between the time it takes a passport to respond and whether or not the message came from that particular passport. In our scenario, X is 1 if the passport we replay the message to is the same one used in the session where the message was recorded, and X is 0 if the message did not come from this particular passport. The continuous variable Y in this example is the time it takes to reject the message. The passport is considered to be secure if, and only if, there is no evidence of dependence between X and Y, i.e., the mutual information is zero.

Each country has its own implementation of the e-passport, and we found that the time taken for passports to communicate with a reader has the same distribution when they have the same nationality. We therefore tested passports from four different countries: Germany, Greece, Ireland and the UK. For each of these we first calculated the passport's cryptographic key from the date of birth, date of expiry and passport number. Then, using a basic RFID reader, we ran the access protocol and recorded the message we needed to replay. For the German, Greek and Irish passports, we replayed the message to the passport 500 times, and then sent the message 500 times to a different passport from the same country. For the British passport, we replayed the message to the passports 1000 times. We added a clock to our computer program to exactly measure the time between when the replayed message was sent and when the passport's error message was received by the reader. A plot of this data is given in Figure 1.

The first two columns of Table I presents the values of the mutual information test statistics computed following the methods described in Section IV and the corresponding p-value estimates based on 10000 bootstrapped samples. It is observed that the mutual information estimates are very near to 1 for all four passports considered and hence it is obvious that the passports can be traced.

A. Testing a time delay based fix

A quick and easy solution to this problem would be to simply add a fixed time delay in the case that the passport's MAC check fails. This solution has the advantage of being easy to test, and easy to implement.

To test if this solution would fix the information leak we experimented with adding various constants to the response times. Adding the difference of medians to the short set of response times seemed to work best in terms of reduction of the mutual information estimates. The corresponding MI test statistics are presented in the third column of Table I. All the MI values show significant reduction, and hence the fix may seem to be working. However, if we look at the estimated p-values presented in the last column of Table I, we can see that this is not the case. Only for the Greek passport the p-value increases from 0; hence the problem is not solved for any of the other 3 passports. For the Greek passport, at a 5% level of significance we would not reject the null hypothesis. A p-value this low clearly indicates that an information leak is a high possibility.

Nationality	MI (no padding)	p-value	MI (padded)	p-value
British	0.9542736	0	0.09446402	0
Irish	0.9999755	0	0.04872853	0
Greek	0.9795026	0	0.01775579	0.075
German	0.983794	0	0.03101871	0

TABLE I

A comparison of the mutual information estimates obtained for different passports before and after applying the time padding based on difference of medians.

It seems that a time padding fix is not very efficient, although it does make the attack an order of magnitude harder to perform. It is possible to accurately identify e-passports with three messages [11]. If the response time was padded as we describe here the attacker would need dozens of messages to achieve a similar level of accuracy. However, to devise a completely leak-proof passport, a better fix is obviously required.

B. A comparison with other tests

There are a number of existing other nonparametric tests that can be used to test if two samples came from the same distribution. The most popular of these are the Kolmogorov-Smirnov (KS) test [13], the (BWS) test [6], the Anderson-Darling (AD) test [30] and the Cramér-von Mises (CVM) test [1].

Table II compares the p-values of these other nonparametric tests applied to the simple time padding fix for each of the passport types we examined. They agree with the conclusions of our mutual information test for the British, Irish and Greek passports. However, every other test failed to detect a leak in the padded German passport samples. To check that this really was an information leak from the time-padded German passport data, we collected two more sets of data; in each case, our test could correctly trace a passport after time padding, but all of the other tests failed to do so. To ensure against false positives we generated two sets of timed data from the same passport, i.e., with no information leak, and, as expected, none of the tests found false evidence of a leak.

This result clearly demonstrates the superior sensitivity of our test, and justifies its use to test for information leaks in this situation. We present further evidence of the superiority of our test, using a number of simulations in Section VI.

A comparison with discrete mutual information: Another approach to analysing this data would have been to truncate the times, treat them as discrete values and use discrete mutual

Nationality	MI test	KS test	CVM test	AD test	BWS test
British	0	0	0	0	0
Irish	0	0.001	0	0	0
Greek	0.075	0.718	0.544	0.367	0.408
German	0	0.257	0.743	0.302	0.271

TABLE II
A COMPARISON OF THE P-VALUES OF DIFFERENT TEST STATISTICS OBTAINED FOR DIFFERENT PASSPORTS AFTER APPLYING THE TIME PADDING BASED ON DIFFERENCE OF MEDIANS.

information to analyse the data (as described in [8]). We tried this approach on the padded times, rounding the time measurements to 3, 4 and 5 decimal places, with 500 samples. None of these tests detected the information leak; rounding to 3 decimal places removed the leak. When rounding to 4 or 5 decimal places the estimate of mutual information was larger. However, with just 500 samples these estimates were still inside the confidence interval for values that were compatible with zero leakage, and so did not constitute evidence of a leak.

Collecting many more samples would have led to the detection of a leak. In general, the discrete approach requires more samples than the product of the number of possible secrets and observations. Collecting enough data to ensure a correct analysis when measuring time to 5 decimal places could have taken many hours.

This shows that using continuous mutual information does make better use of the data than the purely discrete test. In this case, using a purely discrete approach failed to detect the information leak, whereas continuous mutual information did detect the leak. Making the continuous data artificially discrete using the histogram approach and putting it into bins does not work.

C. Building a better passport

Our test shows that adding a fixed time delay will not fix the information leak in e-passports. So instead, we tried changing the passport program to decrypt the message sent to it, even if the MAC check failed. While this may seem like an obvious solution, it is more complicated to check, as it requires a reimplementation of the e-passport protocols, and it requires the passport to try to decrypt messages that may have been corrupted in transit.

None of the countries that have issued e-passports have released the source code or implementational details of the RFID chips. However, the Digital Security group at the University of Nijmegen have released the source for a full implementation of the e-passport, called the "JMRTD" package[4]. This implementation uses "Java Card", which is a platform that makes it possible to run programs written in a subset of Java on a JCOP compatible RFID card.

We tested this JMRTD version of the e-passport and found that the information leak occurred. To fix the problem we rewrote the code that handled the authentication process so that even if the MAC check failed, the message body would still be decrypted. The only file that needed to be changed was the JMRTD's PassportService.java file; we note that we also had to disable a check of the data format of the decrypted message, which may also have led to an information leak.

Once the initial authentication protocol has been run, all of the signed data can be read and copied off the passport. We used the data read from a genuine passport, and our rewritten authentication protocol to make a working e-passport. We collected 500 samples from this passport and we applied our test to investigate if our rewrite fixed the leak. We found a p-value of 0.486, indicating that there was no evidence of an information leak. While other information leaks may still exist, and more sensitive measuring equipment might turn up different results, these measurements suggest that this particular leak has been fixed.

As our implementation follows the specification [16] exactly and the data is copied from a real e-passport, and is therefore signed by the UK government, our new passport chip should be completely functional. We tried out our new e-passport with a number of publicly available e-passport reader implementations and it functioned perfectly as a passport. This gives us, perhaps, the only untraceable e-passport chip in the world. We had hoped to try it out on the automated immigration machines used at Birmingham International Airport, however the university lawyers we spoke to insisted that we did not attempt this without written permission from immigration staff at the airport, and so far airport staff have not responded to our requests.

VI. Simulations and Comparisons with other Tests

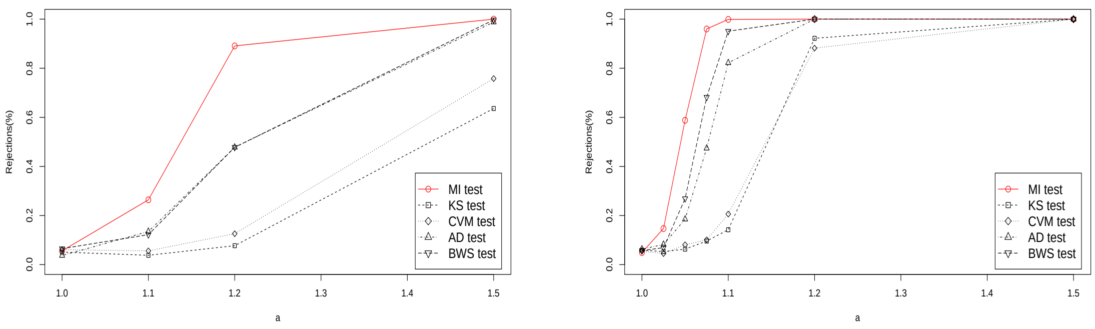
The comparison of our test with other statistical tests in the last section showed that our new test was the best at analysing the information leak from the passport data. In this section, we generate some data from known distributions and make some more comparisons. We find that our new test is better at distinguishing two distributions in all situations, apart from the case of two distributions that differ only by their mean.

We first compare samples taken from the different uniform distributions. As these distributions have finite support, Theorem 1 applies and guarantees a good convergence rate. To examine how our test works when the sampled data comes from distributions with infinite support we compare the results taken from a number of normal and t-distributions. We then run our test on some multiple input cases. In all of these simulations, we get a good rate of convergence.

For our simulations, we take m samples from one distribution F_0 and n samples from another distribution F_1 and then test the null hypothesis that $F_0 = F_1$. We use the Epanechnikov kernel based estimate and the bandwidth chosen according to Equation (5) to obtain the MI test statistics. We then use the following algorithm to compute the power of the mutual information test:

1) Obtain samples $\mathbf{Y}_0$ of length m from F_0 and $\mathbf{Y}_1$ of length n from F_1. Combine them in one single vector

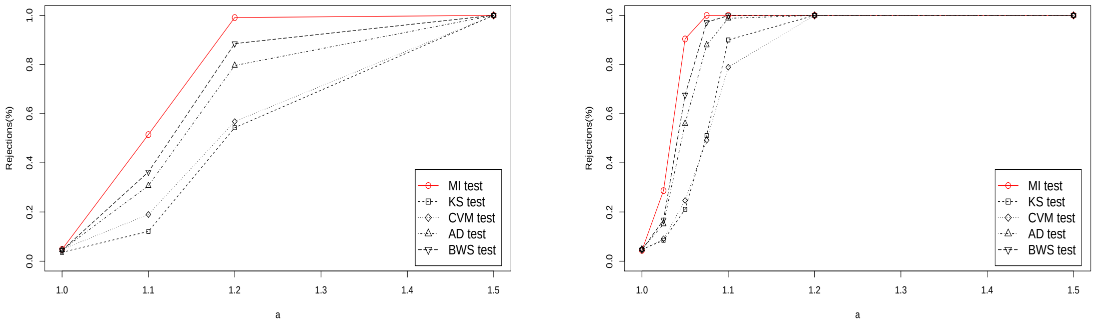
[4] http://jmrtd.org/



(a) Sample size m=n=100

(b) Sample size m=n=500

Fig. 2. Estimated power of tests based on 1000 replications comparing $U(-1,1)$ with $U(-a,a)$ for different values of a.



(a) Sample size m=n=100

(b) Sample size m=n=500

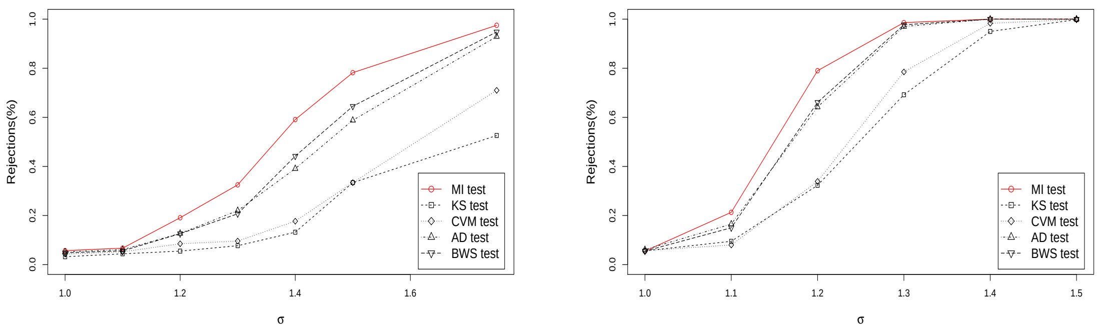
Fig. 3. Estimated power of tests based on 1000 replications comparing $U(0,1)$ samples with samples from $U(0,a)$ distributions for different values of a.

which we denote by $\mathbf{Y}$.

2) We simulate a random sample of size $N := m+n$, say $\mathbf{X}_1$ from the Bernoulli distribution with $P(1) = m/N$ and compute its mutual information with $\mathbf{Y}$. Let us denote the value of the estimated mutual information by I_1. We repeat this step 10,000 times, and obtain $I_j; j = 1, 2\cdots, 10,000$.
3) We use the $100(1-\alpha)$th percentile of the sampling distribution of $I_1, \cdots, I_{10,000}$ as the cut-off to be used for rejection for the test with level α.
4) We now again simulate samples $\mathbf{Y}_0$ of length m from F_0 and $\mathbf{Y}_1$ of length n from F_1, and again combine them in one single sample which we denote by $\mathbf{Y}$.
5) Define a 0-1 valued vector $\mathbf{X}$, whose j-th element, x_j, is 0 or 1 according to whether the jth element of $\mathbf{Y}$, y_j, is from $\mathbf{Y}_0$ or $\mathbf{Y}_1$, i.e. $\mathbf{X}$ is a vector of length $N := (n+m)$ with n zeroes followed by m 1s. Compute the mutual information estimate between $\mathbf{Y}$ obtained in step 4 with the obtained $\mathbf{X}$. $\mathbb{H}_0$ is rejected if the obtained mutual information estimate is greater than the cut-off obtained in step 3.
6) Repeat steps 4 and 5 1000 times to estimate the power of the test, given by the proportion of rejections of $\mathbb{H}_0$.

Following the methods described in [35] and [6], the cut-offs of the CVM test and the BWS tests for 5% level are estimated by obtaining the sampling distribution of the said test statistics for relevant values of m and n based on samples from the standard normal distribution of sizes m and n. The sampling distribution is computed based on 10000 samples. We compare the percentage of rejections of $\mathbb{H}_0$ by the tests based on 1000 random pairs of samples of two sizes, 100 and 500. When the sample sizes grow larger, all the tests start performing well; we have also explored the power of the tests when the samples are of size 2500. As all the tests perform very well for that size, we omit the large size samples from our discussion for the sake of brevity.

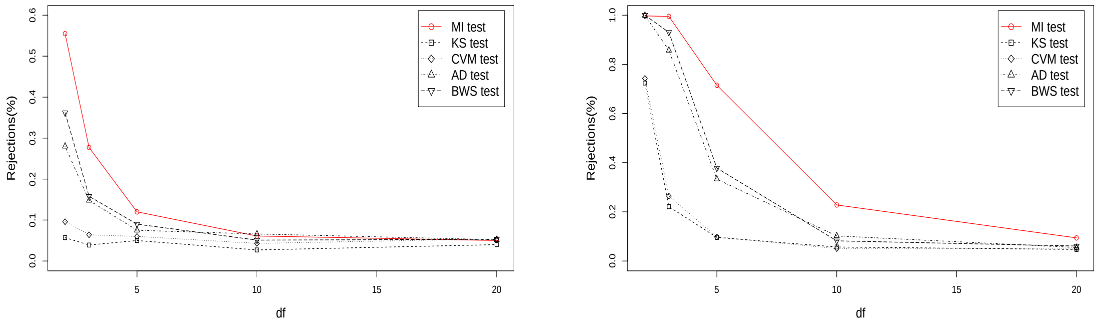
As examples of distributions where the support is finite, so that Theorem 1 holds and the mutual information based test statistic works well, we compare the uniform distribution on $(-1,1)$ with other symmetric uniform distributions on intervals $(-a,a)$. We also compare the uniform distribution on the unit interval with the uniform distributions on intervals $(0,a)$, for values of a gradually increasing from 1. The results are shown respectively in Figures 2 and 3. These graphs show the rate at which the test rejects the null hypothesis and decides the distributions are different. Going left to right the distributions become more different, so as expected the rejection rate rises.



(a) Sample size m=n=100

(b) Sample size m=n=500

Fig. 4. Estimated power of tests based on 1000 replications comparing $N(0,1)$ samples with $N(0,\sigma^2)$ samples.



(a) Sample size m=n=100

(b) Sample size m=n=500

Fig. 5. Estimated power of tests based on 1000 replications comparing $N(0,1)$ samples with samples from t distributions with various degrees of freedom.

Calculating the results to the 5% level means that all of the tests have a similar false positive rate, when the distributions are the same (computing the p-value reduces the risk of false positives). As soon as there is a difference in the distributions, our mutual information test (marked MI on the graphs) has the best rejection rate.

For the tests in Figure 2, the two distributions have the same mean but a different standard deviation. Our test fares significantly better than the others, especially with the smaller number of samples. The second set of test in Figures 2, have a different mean and standard deviation, in this case all of the tests do better, but our test is still the best at distinguishing the distributions.

A. Tests with infinite supports

We now look at how our test handles some cases of similar distributions with infinite support. While Theorem 1 does not apply in this case, our test is still useable, as are the other two sample tests, so it is interesting to compare their performance. We first compare samples from the standard normal distribution with samples from normal distributions with zero mean and variance increasing away from 1. The results are summarised in Figure 4. The mutual test clearly appears to be the most powerful, as can be seen in Figure 4.

We next observe the discriminatory power of the tests when a sequence of distributions is compared with their eventual limit: we compare samples from the standard normal distribution with t-distributions with gradually increasing degrees of freedom (the t-distribution is a lopsided normal distribution for low degrees of freedom and tends to the normal as the degrees of freedom increase). The results are presented in Figure 5. The mutual information test exhibits superior discriminatory power when comparing the standard normal distribution with t-distributions with different degrees of freedom. Whereas for the other tests power decreases to around 0.05 very qucikly, the power of the MI test decreases much more slowly; for example, when tested on the basis of 500 samples, the MI test is the only test with any discriminatory power between the standard normal and the t-distribution with 20 degrees of freedom.

For a further comparison, we look at two distributions with different shapes but equal means and variances. We compare the $N(0,1/12)$ distribution with the uniform distribution on the interval $(-0.5,0.5)$ for various sample sizes. The results are presented in Figure 6. In this situation, the mutual information test is the only test that can detect a difference with one hundred samples. The BWS test and the AD test started to reliably detect the difference between these distributions with six and seven hundred samples. The KS and the CVM tests do not perform well, this suggests that these tests are over reliant on the value of the mean and the standard deviation of the two samples when deciding if they are different.

There was only one case in which our test was not better than the other tests we discuss. This was when the two sets of samples were drawn from distributions that were identical in every aspect, except for a small different in their means. Figure 7 show the results of running the test with samples drawn from $N(0,1)$ and $N(\mu,1)$ for a range of μ. These graphs shows that when the difference in the mean equals 0.25 and 0.5 the other tests detect more of a difference. This would seem to be because our test is sensitive to a great deal of different factors, so when all of these factors are identical, apart from the mean, our tests find less of a difference than the other tests.

B. Tests with multiple inputs

In this subsection, we give some results from applying our test to systems with many possible secret values. While Theorem 1 only guarantees a good convergence rate in the two-sample case, our test is sound for any number of possible secret values, and it has demonstrated a good convergence rate for all of the systems we have looked at.

For our first example we look at tracing a passport when we do not know the nationality of the passport we are replaying the message to. Here we have 5 possible secret values depending on which passport we replay the message to. X=0:the UK passport we are trying to trace, x=1:a different UK passport x=2:a French passport, x=3:Greek, x=4:Irish and x=5:German. As expected, our test showed that the information leak remained, and the estimate of mutual information was, much higher: 2.31. This is because, as well as learning if the target passport is the one they are trying to trace, the attackers can also learn the nationality of the bearer.

Next, we tried the same test with data from the rewritten, fixed passport and 5 other countries. Our test showed that an information leak remained and the mutual information was 2.33. This was because each nationality of passport had a different response time. So while our fix of the information leak in e-passports makes a single passport untraceable, the nationality of the passport is still leaked. We note that it is possible to find the nationality of a passport by other means [32], [11], and that fixing this particular leak would require a level of international cooperation that is, most likely impossible to achieve.

As a second demonstration we looked at an example of Simple Power Analysis, taken from a book by Mangard, Oswald and Popp [22]. Simple Power Analysis involves observing the power usage of a smartcard, and using these observations to deduce what values the smart card is processing. So in this situation the secret values $\mathcal{X}$ are a secret (8-bit) byte stored in the chip and the observations $\mathcal{Y}$ are the measurements of the power used by the card. Mangard, Oswald and Popp construct a smart card, and show that 362ms into a particular operation the power used is:

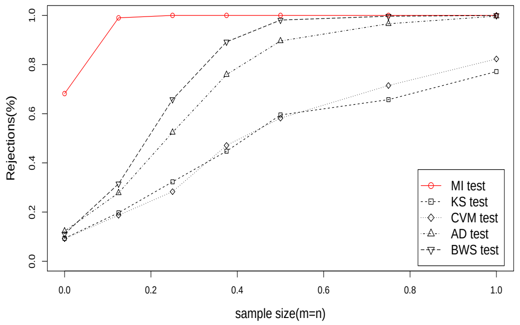


Fig. 6. Plot of the estimated power of various tests based on 1000 replications comparing $N(0,1/12)$ samples with samples from $U(-0.5,0.5)$ distribution for different sample sizes.

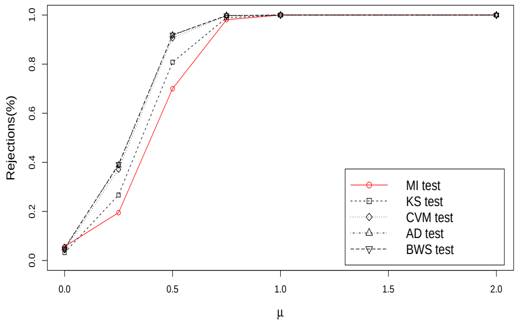


Fig. 7. Plot of the estimated power of various tests based on 1000 replications comparing $N(0,1)$ samples with samples from $N(\mu,1)$ distribution for 100 samples.

$$P = P_{data} + P_{el.noise} + P_{const} \tag{7}$$

where $P_{const} = 137.57mV$ is the constant power usage of the card, $P_{el.noise}$ $N(0,1.63)$ is noise in the system, and P_{data} depends on the hamming weight of the secret byte (i.e., the number of bits set to 1) as follows:

Hamming Weight	Power	Hamming Weight	Power
0	-22.67	5	4.96
1	-16.92	6	10.53
2	-11.35	7	16.68
3	-5.86	8	25.12
4	-0.49		

Mangard, Oswald and Popp analyse their smart card by taking 200 samples for each of the possible 256 secret values. They round their data to the nearest millivolt and treat it has discrete, and then calculate the leakage as the signal to noise ratio of the observations as their measurement of leakage.

We simulated data for the power usage of this smart card using Equation 7, and applied our test. We take only 10 samples for each secret value; this results in a p-value of 0, clearly showing that there is an information leak. The estimate of mutual information is 2.65, which tallies with Mangard et al.'s measure of the leakage. We are able to perform this analysis with an order of magnitude fewer samples because treating the measurements as continuous provides us with more information, and our shuffling method of constructing a test for zero leakage is extremely sensitive.

We note that this analysis does not recover the key; rather its aim is just to detect the existence of a leak. Gierlichs et al.'s [14] present a mutual information based framework that can be used to learn the key from observations of the smart card, which is already known to leak information. Our method is a general-purpose test to detect the presence of a leak in anything from a smart card or an e-passport to a multi-threaded program.

VII. Conclusion

We have shown that continuous mutual information can be an effective measure of information leakage for probabilistic systems. The discrete version of mutual information treats all the observations of a system as unrelated; when there is a natural, dense ordering on the data, discrete mutual information ignores the extra information this ordering provides. Continuous mutual information, on the other hand, does take account of this ordering and therefore provides a better measure.

We have used continuous mutual information to develop a statistical test for the presence of an information leak. When calculating an estimate of mutual information for a probabilistic system from sampled data, there will be some noise in the results. Given some sampled data, we create a number of artificial samples with the same observable public values and a randomly chosen secret value. The mutual information of this new data set will be zero, because now the secret values are unrelated to the observable output. This shows us what zero information leakage looks like for a particular system and we can then compare our original sampled data to this profile for zero leakage.

We show that our test is a practical way of looking for information leaks by analysing some possible fixes for an information leak in e-passports. We first experiment with adding a simple time delay to one of the messages; this massively reduces the information leak, but our test shows that it is still present and could still be exploited. We try out some other two-sample statistical tests from the literature and we find that in some cases all of the other tests would miss an information leak that we could detect. We next experiment with implementing the passport protocols and this does remove all traces of the leak, giving us, possibly, the only untraceable e-passport in the world.

Using a number of simulations, we found that our new test outperformed the existing two sample tests in almost all situations. We believe one of the reasons our test performs well is that it is based on the powerful notion of mutual information. We have benefited from a great deal of work carried out by the computer security community, which has shown that information theory to provide powerful measures of information leakage. Another possible reason our test outperforms the older Cramér-von Mises test [1] and Kolmogorov-Smirnov test [13], may be because these tests were designed to be carried out by hand. Therefore, these tests make a number of assumptions in order to simplify the calculations. The more recent two sample tests that we use as a comparison are the current state-of-the-art tests for deciding when two sets of sampled data come from the same distribution. Our mutual information test outperformed all of these tests.

Acknowledgments: We would like to thank Chris Novakovic, Erica Butler and the anonymous referees who made helpful comments about this paper.

References

[1] T. W. Anderson. On the distribution of the two-sample Cramér-von Mises Criterion. *Annals of Mathematical Statistics*, 1962.

[2] A. Antos and I. Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures and Algorithms*, 19, 2002.

[3] M. Backes and B. Köpf. Formally bounding the side-channel leakage in unknown-message attacks. In *ESORICS*, pages 517–532, 2008.

[4] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F. Stabaert, and N. Veyrat-Charvillon. Mutual information analysis: a comprehensive study. *Journal of Crytology*, 24:269–291, 2011.

[5] T. Batu, S. Dasgupta, R. Kumar, and R. Rubinfeld. The complexity of approximating entropy. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 678–687. ACM, 2002.

[6] W. Baumgartner, P. Weiß, and H. Schindler. A nonparametric test for the general two-sample problem. *Biometrics*, 54, 1998.

[7] K. Chatzikokolakis. *Probabilistic and Information-Theoretic Approaches to Anonymity*. PhD thesis, Ecole Polytechnique, 2007.

[8] K. Chatzikokolakis, T. Chothia, and A. Guha. Statistical measurement of information leakage. In *16th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 6015. LNCS, 2010.

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

[10] T. Chothia and A. Guha. A test for the two-sample problem using mutual information to fix an information leak in e-passports. Technical report, University of Birmingham, 2011.

[11] T. Chothia and V. Smirnov. A traceability attack against e-passports. In *FC10: Proceedings of the 14th International Conference on Financial Cryptography and Data Security 2010*. LNCS, 2010.

[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] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, New York., 1971.

[14] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis: A generic side-channel distinguisher. In *CHES '08: Proceeding sof the 10th international workshop on Cryptographic Hardware and Embedded Systems*, 2008.

[15] J. Heusser and P. Malacaria. Quantifying information leaks in software. In *Proceedings of 2010 Annual Computer Security Applications Conference (ACSAC'10)*, 2010.

[16] ICAO. Machine Readable Travel Documents. Doc 9303. Part 1. Technical report, International Civil Aviation Organization, 2006.

[17] P. C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, pages 104–113, 1996.

[18] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 2 (12):*, 1995.

[19] B. Köpf and M. Dürmuth. A provably secure and efficient countermeasure against timing attacks. In *Computer Security Foundations Symposium (CSF)*, 2009.

[20] B. Köpf and A. Rybalchenko. Approximation and randomization for quantitative information-flow analysis. In *Computer Security Foundations Symposium (CSF)*, 2010.

[21] B. Köpf and G. Smith. A provably secure and efficient countermeasure against timing attacks. In *Computer Security Foundations Symposium (CSF)*, 2010.

[22] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[23] S. McCamant and M. D. Ernst. Quantitative information flow as network flow capacity. In *PLDI '08: Proceedings of the 2008 ACM SIGPLAN conference on Programming language design and implementation*, pages 193–205. ACM, 2008.

[24] J. K. Millen. Covert channel capacity. In *IEEE Symposium on Security and Privacy*, pages 60–66, 1987.

[25] I. S. Moskowitz, R. E. Newman, and P. F. Syverson. Quasi-anonymous channels. In *IASTED CNIS*, pages 126–131, 2003.

[26] S. J. Murdoch. Hot or not: revealing hidden services by their clock skew. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 27–36. ACM, 2006.

[27] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of Tor. In *Proc. of the 2005 IEEE Symposium on Security and Privacy*. IEEE CS, 2005.

[28] J. Newsome, S. McCamant, and D. Song. Measuring channel capacity to distinguish undue influence. In *PLAS '09: Proceedings of the ACM SIGPLAN Fourth Workshop on Programming Languages and Analysis for Security*, pages 73–85. ACM, 2009.

[29] L. Paninski. Estimation of entropy and mutual information. *Neural Comp.*, 15(6):1191–1253, June 2003.

[30] A. N. Pettitt. A two-sample anderson-darling rank statistic. *Biometrika*, 1976.

[31] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Pacific Grove, California, 3rd edition, 2006.

[32] H. Richter, W. Mostowski, and Erik Poll. Fingerprinting Passports. In *NLUUG Spring Conference on Security*, 2008.

[33] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1986.

[34] G. Smith. On the foundations of quantitative information flow. In *FOSSACS '09: Proceedings of the 12th International Conference on Foundations of Software Science and Computational Structures*, pages 288–302, Berlin, Heidelberg, 2009. Springer-Verlag.

[35] Y. Xiao, Gordon A., and A. Yakovlev. A C++ program for the Cramér-von Mises two sample test. *Journal of Statistical Software*, 17, 2007.

[36] Y. Zhu and R. Bettati. Anonymity vs. information leakage in anonymity systems. In *Proc. of ICDCS*, pages 514–524. IEEE Computer Society, 2005.