Form Methods Syst Des (2015) 47:265–286
DOI 10.1007/s10703-015-0232-5

Evaluation of anonymity and confidentiality protocols using theorem proving

Tarek Mhamdi[1] · **Osman Hasan**[1] · **Sofiène Tahar**[1]

Published online: 20 June 2015

Abstract Anonymity and confidentiality protocols constitute crucial parts in many network applications as they ensure anonymous communications between entities in a network or provide security in insecure communication channels. Evaluating the properties of these protocols is therefore of paramount importance, especially in the case of safety-critical applications. However, traditional analysis techniques, like simulation, cannot ascertain accurate analysis in this domain. We propose to overcome this limitation by conducting an information leakage analysis of anonymity and cryptographic protocols within the trusted kernel of a higher-order-logic theorem prover. For this purpose, we first introduce two novel measures of information leakage, namely the information leakage degree and the conditional information leakage degree and then present a higher-order-logic formalization of information measures and the underlying required theories of measure, probability and information. For illustration purposes, we use the proposed framework to evaluate the security properties of the one-time pad encryption system as well as the properties of an anonymity-based single MIX. We show how this formal analysis allowed us to find a counter-example for a theorem that was reported in the literature to describe the leakage properties of this single MIX.

Keywords Information leakage · Anonymity · Confidentiality · Theorem proving · Dependable systems and software

✉ Sofiène Tahar
tahar@ece.concordia.ca

Tarek Mhamdi
mhamdi@ece.concordia.ca

Osman Hasan
o_hasan@ece.concordia.ca

[1] Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada

1 Introduction

Anonymity networks such as Crowds [24] and Tor [9] have been proposed to provide anonymous communication between entities in a network. Analyzing the anonymity properties of these protocols consists in finding out how much information an attacker can learn about the senders and receivers in the network. One way to do so is through quantitative analysis of information flow [26,28] which allows to measure how much information about the high security inputs of a system can be leaked, accidentally or maliciously, by observing the systems outputs and possibly the low security inputs.

Quantitative analysis of information flow has also been proposed to analyze confidentiality protocols [28]. In fact, while these protocols aim to preserve sensitive and confidential data and prevent it from being leaked or tainted, a small leakage of information is sometimes necessary, as is the case for a voting protocol, where the tally of votes should be publicly revealed eventhough the individual votes should be kept secret. Password checking is another example of information leakage by design, where a rejected password reveals information about what the secret password is not.

Various measures of information flow have been proposed in the literature. For instance, Serjantov and Danezis [27] and Diaz et al. [8] independently proposed to use the entropy to define the quality of anonymity and to compare different anonymity systems. Malacaria [16] defined the leakage of confidential information in a program as the conditional mutual information between its outputs and secret inputs, given the knowledge of its low security inputs. Deng et al. [7] proposed relative entropy as a measure of the amount of information revealed to the attacker after observing the outcomes of the protocol, together with the a priori information. Chatzikokolakis et al. [2] modeled anonymity protocols as noisy channels and used the channel capacity as a measure of the loss of anonymity. Zhu and Bettati [29] proposed the anonymity degree as a measure of the anonymity properties in a MIX network.

In this paper, we propose two novel measures of information leakage, namely the *information leakage degree* and the *conditional information leakage degree*. We present the intuition behind these definitions and compare them to existing measures. The motive behind these new measures is that they not only quantify the leakage of information but they also describe the quality of leakage compared to the maximum leakage that the system allows under extreme situations, namely the perfect identification scenario and the perfect security scenario. We also compare the proposed information leakage degree to the anonymity degree introduced in [29] and show that our definition is related but more generic.

Traditionally, paper-and-pencil based analysis or computer simulations have been used for quantitative analysis of information flow. Paper-and-pencil analysis does not scale well to complex systems and is prone to human error. Computer simulation, on the other hand, lacks in accuracy due to the usage of computer arithmetics, such as floating or fixed point numbers, that leads to numerical approximations. These analysis inaccuracies may result in compromising national security and finances given the safety and security-critical nature of systems where information flow analysis is usually used.

As an alternative approach, we propose a machine-assisted analysis of information flow by conducting the analysis within the trusted kernel of a higher-order-logic theorem prover [11]. Theorem proving [13] is a field of computer science and mathematical logic that allows to conduct computer-assisted formal proofs of the correctness of systems and programs using mathematical reasoning. The implementation and specification of a system are both expressed in terms of logical formulas and the proof of correctness is derived from a very small set of axioms and inference rules. This deduction style ensures that only valid formulas

are provable. We propose to use higher-order logic [1] because its high expressiveness is required to formalize, or write in a formal language, all the mathematical theories needed to conduct the quantitative analysis on information flow. This includes the higher-order logic formalization of measure theory, Lebesgue integration, probability and information theory concepts.

We build upon the existing formalization of measure, integration and probability [18], and information theories [19] to provide a complete framework to formally reason about quantitative properties of information flow analysis within the sound core of the HOL4 theorem prover and thus guarantee accuracy of the analysis. In particular, this paper presents an extension of existing theories of measure, Lebesgue integration and probability [18] to cater for measures involving multiple random variables. Building upon this formalization, we present a higher-order-logic formalization of the Kullback–Leibler (KL) divergence [6] from which we can derive the formalization of most of the information leakage measures presented in the literature so far and the information leakage degrees that we propose in this paper.

We illustrate the usefulness of the framework for formal quantitative analysis of information flow by tackling two applications, an anonymity-based single MIX application [29] and the one-time pad (OTP) encryption system [21]. We provide a higher-order-logic formalization of the single MIX as well as the channel capacity which we use as a measure of information leakage within the MIX. We then formally verify that a sender using the MIX as a covert channel, can transmit information through the MIX at a rate equal to the maximum channel capacity without having to communicate with all the receivers. This result allowed us to identify a flaw in the paper-and-pencil based analysis of a similar problem [29] which clearly indicates the usefulness of the proposed technique. We also provide a higher-order-logic formalization of the different blocks of an OTP encryption system and use the formalization of information leakage measures to prove that this encryption type offers, indeed, a perfectly secure communication.

The rest of the paper is organized as follows: in Sect. 2, we present an overview of higher-order-logic theorem proving and the HOL4 theorem prover, followed by the formalization of information measures as well as the required underlying theories of measure and probability. In Sect. 3, we introduce two novel measures of information leakage and highlight their distinguishing characteristics. We show in Sect. 4 how we can use the proposed framework to evaluate the properties of an anonymity-based single MIX and the confidentiality properties of the OTP encryption. We discuss related work in Sect. 5 and conclude the paper in Sect. 6.

2 Formalization of information leakage in HOL

The formalization of measures of information leakage in higher-order logic requires the formalization of probability theory and main concepts on information theory including the Shannon entropy, mutual information and conditional mutual information. We start with a brief preview of the HOL4 theorem prover and then show how we formalize the probability and information theories in HOL.

2.1 HOL4 theorem prover

HOL4 is an interactive theorem prover which is capable of conducting proofs in higher-order logic. It utilizes the simple type theory of Church [3] along with Hindley–Milner polymorphism [22] to implement higher-order logic. HOL has been successfully used as a verification

Table 1 HOL symbols and functions

HOL symbols	Meaning
$\wedge$	Logical *and*
$\vee$	Logical *or*
$\neg$	Logical *negation*
::	Adds a new element to a list
++	Joins two lists together
(a, b)	A pair of two elements
fst	First component of a pair
snd	Second component of a pair
λx.t	Function that maps x to $t(x)$
{x|P(x)}	Set of all x such that $P(x)$

framework for both software and hardware as well as a platform for the formalization of pure mathematics.

In order to ensure secure theorem proving, the logic in the HOL system is represented in the strongly-typed functional programming language ML [23]. An ML abstract data type is used to represent higher-order-logic theorems and the only way to interact with the theorem prover is by executing ML procedures that operate on values of these data types. The HOL core consists of only five basic axioms and eight primitive inference rules, which are implemented as ML functions. Soundness is assured as every new theorem must be verified by applying these basic axioms and primitive inference rules or any other previously verified theorems/inference rules.

The HOL theorem prover includes many proof assistants and automatic proof procedures [12] to assist the user in directing the proof. The user interacts with the proof assistant through an interface and provides it with the necessary tactics, which are ML functions that break the proof goals into simpler subgoals that might need further simplification or could simply be solved using the various automatic proof procedures.

In order to facilitate reutilization of verified theorems, HOL allows its users to store a collection of valid HOL types, constants, axioms and theorems as a HOL theory file in computers. Once stored, HOL theories can be loaded in the HOL system and the corresponding definitions and theorems can be utilized right away. Thus, HOL theories allow us to build upon existing results in an efficient way without going through the tedious process of regenerating these results using the basic axioms and primitive inference rules. Various mathematical concepts have been formalized and saved as HOL theories by HOL users. Out of this useful library of HOL theories, we utilize the theories of Booleans, lists, sets, positive integers, real numbers, measure and probability in this paper. In fact, one of the primary motivations of selecting the HOL theorem prover for our work was to benefit from these built-in mathematical theories.

Table 1 provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories, in this paper.

2.2 Probability theory in HOL

Probability provides mathematical models for random phenomena and experiments. The classical approach to formalize probabilities defines the probability of an event A as $p(A) =$

$\frac{N_A}{N}$, where N_A is the number of outcomes favorable to the event A and N is the number of all possible outcomes of the experiment. The main limitation of this approach is the assumption that required all outcomes to be equally likely (equiprobable). Thus a concept of probability is used to define probability itself and hence this definition cannot be used as a basis for a mathematical theory. Besides that, for many random experiments the outcomes are not equally likely. Finally, the definition does not work for the cases when the number of possible outcomes is infinite.

Kolmogorov [14] introduced the axiomatic definition of probability which provides a mathematically consistent way for assigning and deducing probabilities of events. This approach consists in defining a set of all possible outcomes, Ω, called the sample space, a set F of events which are subsets of Ω and a probability measure p such that (Ω, F, p) is a measure space with $p(\Omega) = 1$.

Using measure theory to formalize probability has the advantage of providing a mathematically rigorous treatment of probabilities and a unified framework for discrete and continuous probability measures. In this context, a probability measure is a measure function, an event is a measurable set and a random variable is a measurable function. The expectation of a random variable is its integral with respect to the probability measure.

Basic definitions in the formalization of probability and measure theory in HOL are based on the work of [5]. Our contributions consist in going beyond these definitions to provide important theorems that will allow us to operate with the basic concepts such us random variables and their expected values. For instance, the formalization of [5] does not allow us to work with the sum of random variables as a random variable itself; we would have to add it as an assumption. Another important shortcoming is the lack of the formally verified properties of the expected value of a random variable such as the linearity and monotonicity.

In [20], we have presented a formalization of measure theory and Lebesgue integration based on the set of extended-real numbers, which is the set of real numbers augmented by the negative and positive infinity. This has allowed us to prove several important limiting and convergence theorems. On the other hand, it rendered the analysis more complex and at times tedious. In the current paper, however, we define the measure theory over standard real numbers to take into account the fact that probabilities are finitely valued. This has lead to important simplifications of the theorems in the library and made it easier to use the formalization without having to prove unnecessary and sometimes cumbersome assumptions in this context of probability. Furthermore, it allowed us to make use of the rich libraries of definitions and theorems related to limits and sequences which are already available in HOL. In the following, the symbol $\vdash$ denotes a theorem and $\models$ denotes a definition.

We define `subset_class` of a set Ω as a set of subsets of Ω. A sigma algebra over Ω is a special subset class that contains the empty set and is closed under countable unions and complementation relative to the space. Members of the sigma algebra over Ω are called the measurable sets of Ω. The higher-order logic formalization of a sigma algebra is as follows:

```
⊨ sigma_algebra (Ω, A) =
    subset_class Ω A ∧
    {} ∈ A ∧
    ∀s. s ∈ A ⇒ Ω\s ∈ A ∧
    ∀c. countable c ∧ c ⊆ A
     ⇒ ⋃c ∈ A
```

where $\Omega\backslash s$ denotes the complement of s within Ω and $\bigcup c$ the union of all elements of c. A set is countable if its elements can be counted one at a time, or in other words, if every element of the set can be associated with a natural number.

A measure function is a non-negative function $\mu: \mathcal{A} \to \mathbb{R}$ satisfying the countable-additivity condition, which states that the measure of a countable union of pairwise disjoint measurable sets is the sum of their respective measures. A triplet $(\Omega, \mathcal{A}, \mu)$ is a measure space *iff* $\mathcal{A}$ is a sigma algebra over the space Ω and $\mu: \mathcal{A} \to \mathbb{R}$ is a measure function. The formalization of a measure space is then the following

```
⊨ measure_space (Ω, A, mu) =
    sigma_algebra (Ω, A) ∧
    positive (Ω, A, mu) ∧
    countably_additive (Ω, A, mu)
```

A probability space (Ω, F, p) is defined as a measure space having a unit space measure.

```
⊨ prob_space (Ω, F, p) =
    measure_space (Ω, F, p) ∧
    (p Ω = 1)
```

A probability measure is a measure function and an event is defined as a measurable set.

```
⊨ ∀p. prob p = measure p
⊨ ∀p. events p = measurable_sets p
```

Two events A and B are independent iff $p(A \cap B) = p(A)p(B)$. Here $A \cap B$ is the intersection of A and B, which is the event that both events A and B occur.

```
⊨ ∀p a b. indep p a b =
   a ∈ events p ∧ b ∈ events p ∧
   prob p (a ∩ b) = prob p a * prob p b
```

$X: \Omega \to \mathbb{R}$ is a random variable iff X is $(F, \mathcal{B}(\mathbb{R}))$ measurable

```
⊨ random_variable X Ω F p Borel =
    prob_space (Ω, F, p) ∧
    X ∈ measurable (Ω, F) Borel
```

where F denotes, as previously, the set of events. A measurable function $X: \Omega \to \mathbb{R}$ is a function for which the inverse image of a measurable set is a measurable set, in other words, $\forall A \in \mathcal{B}(\mathbb{R}),\ X^{-1}(A) \in F$. The Borel sigma algebra over a space Ω is the sigma algebra generated by the open sets of Ω. In other words, it is the smallest sigma algebra containing the open sets of Ω. The higher-order-logic formalization of these concepts can be found in [19]. Here we focus on real-valued random variables but the definition can be adapted for random variables having values on any topological space thanks to our general definition of the Borel sigma algebra.

We also formally verified the properties of random variables in HOL. If X and Y are random variables and $c \in \mathbb{R}$ then the following functions are also random variables: cX, $|X|$, X^n, $X + Y$, XY, e^X and $max(X, Y)$.

The probability mass function $p_X: \mathcal{B}(\mathbb{R}) \to [0, 1]$ of a random variable X is defined as the function assigning to every $A \in \mathcal{B}(\mathbb{R})$, the probability of $X^{-1}(A)$, also notated $\{X \in A\}$.

$$p_X(A) = p(\{X \in A\}) = p\left(X^{-1}(A)\right),$$

```
⊨ pmf p X = (λA. prob p (PREIMAGE X A ∩ Ω))
```

where PREIMAGE denotes the HOL function for inverse image and the intersection with the sample space Ω is required because HOL functions are total and should be defined on all variables of the specific type instead of only on Ω.

To be able to define information measures involving multiple random variables, as is the case for the information leakage degrees defined above, we first need to formalize joint distributions as well as products of measure spaces. The joint distribution of two random variables defined on the same probability space is defined as

$$p_{XY}(A) = p(\{(X, Y) \in A\}),$$

```
⊨ joint_distribution p X Y =  (λA. prob p
      (PREIMAGE (λx. (X x, Y x)) A ∩ Ω))
```

The joint distribution of any number of variables can be defined in a similar way. We formally verified a number of joint distribution properties in HOL [17] and some of the useful ones are given below:

```
⊢ 0 ≤ p X Y A
⊢ p X Y = p Y X
⊢ p X Y (A × B)  ≤ p X A
⊢ p X Y (A × B)  ≤ p Y B
```

We also formally verified that the joint distribution is absolutely continuous with respect to the product of marginal distributions. A measure μ is said to be absolutely continuous with respect to ν if $\mu(A) = 0$ for every set A for which $\nu(A) = 0$. We also prove the following useful properties in HOL.

$$p_X(A) = \sum_{y \in Y(\Omega)} p_{XY}(A \times \{y\}),$$
$$p_Y(B) = \sum_{x \in X(\Omega)} p_{XY}(\{x\} \times B).$$

The joint distribution of two random variables is defined over the sigma-algebra corresponding to the product of measure spaces defined in the following. The product of two measure spaces $(X_1, \mathcal{S}_1, \mu_1)$ and $(X_2, \mathcal{S}_2, \mu_2)$ is defined as the measure space $(X_1 \times X_2, \mathcal{S}, \mu)$, where $\mathcal{S}$ is the sigma algebra on $X_1 \times X_2$ generated by subsets of the form $A_1 \times A_2$ where $A_1 \in \mathcal{S}_1$, and $A_2 \in \mathcal{S}_2$. The measure μ is defined for σ-finite measure spaces as

$$\mu(A) = \int_{X_1} \mu_2 (\{y \in X_2 | (x, y) \in A\})\, d\mu_1,$$

and $\mathcal{S}$ is defined using the sigma operator which returns the smallest sigma algebra containing a set of subsets, i.e., the product subsets in this case. The integral used in the definition of product measure is the Lebesgue integral which we formalized in [18].

Let $g(s_1)$ be the function $s_2 \to (s_1, s_2)$, then the product measure is formalized as

```
⊨ prod_measure m1 m2 =
    (λa. integral m1 (λs1.
        measure m2 (PREIMAGE g(s1) a)))
```

We verified in HOL that the product measure can be reduced to $\mu(a_1 \times a_2) = \mu_1(a_1) \times \mu_2(a_2)$ for finite measure spaces.

```
⊢ prod_measure m1 m2 (a1 × a2) =
      measure m1 a1 × measure m2 a2
```

We use the above definitions to define products of more than two measure spaces as follows. $X_1 \times X_2 \times X_3 = X_1 \times (X_2 \times X_3)$ and $\mu_1 \times \mu_2 \times \mu_3$ is defined as $\mu_1 \times (\mu_2 \times \mu_3)$. As stated above, we use the joint distribution and products of measure spaces as a basis to define several concepts of information theory that involve multiple random variables, as is the case for the information leakage degrees.

In this section we defined basic concepts of probability like the events, probability measures and random variables. We also used the formalization of Lebesgue integral [18] to formalize the main statistical properties of random variables, such as the expectation and the variance. Further details about this formalization can be found in [17].

2.3 Information theory in HOL

In this section, we first provide a formalization of the Radon–Nikodym derivative [10] which is then used to define the KL divergence. Based on the latter, we define most of the commonly used measures of information. We start by providing general definitions which are valid for both discrete and continuous cases and then prove the corresponding reduced expressions where the measures considered are absolutely continuous over finite spaces. We build on the foundations, presented in [19], to provide a more general formalization of information theory including the properties of measures of information.

2.3.1 Radon–Nikodym derivative

The Radon–Nikodym derivative of a measure ν with respect to the measure μ is defined as a non-negative measurable function f, satisfying the following formula, for any measurable set A [10].

$$\int_A f d\mu = \nu(A).$$

We formalize the Radon–Nikodym derivative in HOL as

```
⊨ RN_deriv m v =
   @f. f IN measurable (X, S) Borel ∧
   ∀x ∈ X, 0 ≤ f x ∧ ∀a ∈ S,
   integral m (λx. f x * I_a x) = v a
```

where @ denotes the Hilbert-choice operator. The existence of the Radon–Nikodym derivative is guaranteed for absolutely continuous measures by the Radon–Nikodym theorem stating that if ν is absolutely continuous with respect to μ, then there exists a non-negative measurable function f such that for any measurable set A,

$$\int_A f d\mu = \nu(A).$$

We proved the Radon–Nikodym theorem in HOL for finite measures which can be easily generalized to σ-finite measures [19].

```
⊢ ∀m v s st.
    measure_space (s, st, m) ∧
    measure_space (s, st, v) ∧
    abs_cont (s, st, m) (s, st, v) ⇒
```

```
∃f. f ∈ measurable (s, st) Borel ∧
∀x ∈ s, 0 ≤ f x < ∞ ∧
∀a ∈ st,
    integral m (λx. f x * I_a x) = v a
```

The formal reasoning about the above theorem is primarily based on the Lebesgue monotone convergence and the following lemma which, to the best of our knowledge, has not been referred to in mathematical texts before.

Lemma 1 *If P is a non-empty set of extended-real valued functions closed under the max operator, g is monotone over P and $g(P)$ is upper bounded, then there exists a monotonically increasing sequence $f(n)$ of functions, elements of P, such that*

$$\sup_{n\in\mathbb{N}} g(f(n)) = \sup_{f\in P} g(f).$$

Finally, we formally verified various properties of the Radon–Nikodym derivative. For instance, we prove that for absolutely continuous measures defined over a finite space, the derivative reduces to

```
⊢ ∀x ∈ s, u{x} ≠ 0 ⇒
        RN_deriv u v x = v{x}/u{x}
```

The following properties play a vital role in formally reasoning about the Radon–Nikodym derivative and have also been formally verified.

```
⊢ ∀x ∈ s, 0 ≤ RN_deriv m v x < ∞
⊢ RN_deriv ∈ measurable (s, st) Borel
⊢ ∀a ∈ st,  v a =
       integral m (λx. RN_deriv m v x * I_a x)
```

2.3.2 Kullback–Leibler divergence

The KL divergence [6], $D_{KL}(\mu||\nu)$ is a measure of the distance between two distributions μ and ν. It can be used to define most information-theoretic measures such as the mutual information and entropy and can, hence, be used to provide a unified framework to formalize most information leakage measures. It is because of this reason that we propose to formalize the KL divergence in this paper as it will facilitate formal reasoning about a wide variety of information flow related properties. The KL divergence is defined as

$$D_{KL}(\mu||\nu) = -\int_X log\frac{d\nu}{d\mu}d\mu,$$

where $\frac{d\nu}{d\mu}$ is the Radon–Nikodym derivative of ν with respect to μ. The KL divergence is formalized in HOL as

```
⊨ KL_divergence b m v =
       -integral m (λx. logr b((RN_deriv m v)x))
```

where b is the base of the logarithm. D_{KL} is measured in *bits* when $b = 2$. We formally verify various properties of the KL divergence. For instance, we prove that for absolutely continuous measures over a finite space, it reduces to

$$D_{KL}(\mu||\nu) = \sum_{x\in s} \mu\{x\} \log \frac{\mu\{x\}}{\nu\{x\}},$$

```
⊢ KL_divergence b u v =
        SIGMA (λx. u{x} logr b (u{x}/v{x})) s
```

We also prove the following properties

```
⊢ KL_divergence b u u = 0
⊢ 1 ≤ b  ⇒  0 ≤ KL_divergence b u v
```

The non-negativity of the KL divergence for absolutely continuous probability measures over finite spaces is extensively used to prove the properties of information theory measures like the mutual information and entropy. To prove this result, we use the Jensen's inequality and the concavity of the logarithm function.

We show in the subsequent sections how we use the KL divergence to formalize the mutual information, Shannon entropy, conditional entropy and the conditional mutual information, which are some of the most commonly used measures of information leakage.

2.3.3 Mutual information and entropy

The mutual information has been proposed as a measure of information leakage [29] from the secure inputs S of a program to its public outputs O as it represents the mutual dependence between the two random variables S and O. The mutual information is defined as the KL divergence between the joint distribution and the product of marginal distributions. The following is a formalization of the mutual information in HOL.

```
⊨ I(X;Y) = KL_divergence b (p X Y)
                prod_measure (p X) (p Y)
```

We prove various properties of the mutual information in HOL, such as the non-negativity, symmetry and reduced expression for finite spaces, using the result that the joint distribution is absolutely continuous w.r.t. the product of marginal distributions.

```
⊢ 0 ≤ I(X;Y)
⊢ I(X;Y) = I(Y;X)
⊢ I(X;Y) = 0 ⇔ X and Y independent
⊢ I(X;Y) = SIGMA (λ(x,y). p{(x,y)}
                  logr b (p{(x,y)}/p{x}p{y})) s
```

The Shannon entropy `H(X)` was one of the first measures to be proposed to analyze anonymity protocols and secure communications [8,27] as it intuitively measures the uncertainty of a random variable `X`. It can be defined as the expectation of p_X or simply as $I(X;X)$.

```
⊨ H(X) = I(X;X)
```

We prove that it can also be expressed in terms of the KL divergence between p_X and the uniform distribution p_X^u, where N is the size of the alphabet of X.

```
⊢ H(X) = log(N) - KL_divergence b (p X) (p_u X)
```

The cross entropy $H(X, Y)$ is the entropy of the random variable (X, Y) and hence there is no need for a separate formalization of the cross entropy.

The conditional entropy is defined in terms of the KL divergence as follows:

```
⊨ H(X|Y) = log(N) - KL_divergence b (p X Y)
                      prod_measure (p_u X) (p Y)
```

Some of the major entropy properties that we formally verified in HOL include:

```
⊢ 0 ≤ H(X) ≤ log(N)
⊢ max(H(X),H(Y)) ≤ H(X,Y) ≤ H(X) + H(Y)
⊢ H(X|Y) = H(X,Y) - H(Y)
⊢ 0 ≤ H(X|Y) ≤ H(X)
⊢ I(X;Y) = H(X) + H(Y) - H(X,Y)
⊢ I(X;Y) ≤ min(H(X),H(Y))
⊢ H(X) = -SIGMA (λx. p{x} logr b (p{x})) s
```

2.3.4 Conditional mutual information

The conditional mutual information $I(X;Y|Z)$ allows one to measure how much information about the secret inputs X is leaked to the attacker by observing the outputs Y of a program given knowledge of the low security inputs Z. This property was used by Malacaria [16] to introduce the conditional mutual information as a measure of information flow for a program with high security inputs and low security inputs and outputs. The conditional mutual information is defined as the KL divergence between the joint distribution p_{XYZ} and the product measure $p_{X|Z}p_{Y|Z}p_Z$. The HOL formalization is as follows.

```
⊨ I(X;Y|Z) = KL_divergence b (p X Y Z)
      (prod_measure (p X|Z) (p Y|Z) (p Y))
```

We formally verify the following reduced form of the conditional mutual information for finite spaces by first proving that p_{XYZ} is absolutely continuous w.r.t. $p_{X|Z}p_{Y|Z}p_Z$ and then apply the reduced form of the KL divergence.

$$I(X;Y|Z) = \sum_{(x,y,z)\in\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}} p(x,\ y,\ z)\log\frac{p(x,y,z)}{p(x|z)p(y|z)p(z)}.$$

When the two random variables X and Y are independent given Z, the conditional mutual information $I(X;Y|Z) = 0$. In fact, in this case, $\forall x,\ y,\ z.\ p(x,\ y,\ z) = p(x,\ y|z)p(z) = p(x|z)p(y|z)p(z)$.

```
⊢ indep_rv_cond p X Y Z ⇒ I(X;Y|Z) = 0
```

We also prove a few other important results regarding the conditional mutual information which will be useful later in our work.

```
⊢ 0 ≤ I(X;Y|Z)
⊢ I(X;Y|Z) = H(X|Z) - H(X|Y,Z)
⊢ I(X;Y|Z) = I(X;(Y,Z)) - I(X;Z)
⊢ I(X;Y|Z) ≤ H(X|Z)
```

So far, we have provided a higher-order-logic formalization of the KL divergence which we used to define various measures of information. Overall, the HOL definitions and proof scripts of the above formalization required around 15,000 lines of code [17].

This framework allows us to conduct many analyses of quantitative information flow using a theorem prover and hence guaranteeing the soundness of the analysis. We introduce, in the next section, two new measures of information as well as their formalization in higher-order logic. We also compare these measures to existing ones.

3 Degrees of information leakage

Information leakage is a measure of how much information about the high security inputs of a system is leaked, accidentally or maliciously, by observing the systems outputs and the low security inputs. Various measures of information leakage have been proposed in the literature, ranging from entropy to the channel capacity [2,7,8,16,27]. We introduce two new measures of information, namely the information leakage degree and the conditional information leakage degree, and compare them to the anonymity degree introduced in [29] to show that our definitions are more generic.

3.1 Information leakage degree

Consider a program having a set of secret inputs, represented by the random variable X and a set of public outputs, represented by Y. We define the information leakage degree of this program as

$$D = \frac{H(X|Y)}{H(X)},$$

where $H(X)$ and $H(X|Y)$ represent the Shannon entropy of X and the conditional entropy of X given Y, respectively.

```
⊨ D p X Y = H(X|Y) / H(X)
```

To better understand the intuition behind this definition, let us consider the two extreme cases of a completely secure program and a completely insecure program. Complete security, intuitively, happens when the knowledge of the public output Y of a program does not affect the uncertainty about the secret input X. This is equivalent to the requirement that X is independent of Y. In this case $H(X|Y) = H(X)$ and the information leakage degree is equal to 1. On the other hand, when the output of the program completely identifies its secret input, the entropy $H(X|Y)$ is equal to 0 and hence the information leakage degree is equal to 0 in this case of perfect identification. For situations between the two extremes, we prove that the information leakage degree lies within the interval $(0, 1)$. The result is derived in HOL as follows

```
⊢ 0 ≤ D p X Y ≤ 1
```

Using the properties of the mutual information, $I(X; Y)$, we prove that the information leakage degree is also equal to

```
⊢ D p X Y = 1 - I(X;Y) / H(X)
```

This result illustrates the significance of the information leakage degree definition since the mutual information measures how much information an adversary can learn about the input X after observing the output Y. This also allows to compare our definition to the anonymity degree proposed in [29] as

$$D' = 1 - \frac{I(X; Y)}{log N},$$

where N is the size of the alphabet of X. The perfect identification scenario, which is achieved when the anonymity is totally broken, should be represented by an anonymity degree that is always equal to zero, regardless of the system inputs. However, the definition proposed above is equal to zero only when the input random variable X is uniformly distributed. In fact, in the perfect identification scenario $I(X; Y)$ is equal to $H(X)$. When X is uniformly distributed,

$H(X) = log(N)$ and $D' = 0$. For all other distributions, $H(X) < log(N)$ and $D' > 0$. This is not a desirable property of the anonymity degree for the perfect identification case.

Our definition is more general. In fact, when X is uniformly distributed, the two measures coincide $D = D'$. However, in the general case, we believe that our definition is more accurate since, for instance, in the perfect identification scenario, D is always equal to 0 regardless of the input distribution. It is also always equal to 1 for the perfect anonymity case.

It should be noted that Zhu and Bettati [29] considered using the entropy as a normalization factor instead of $log(N)$ but opted for the latter arguing that the input distribution is already accounted for in the mutual information. We believe that this argument is not at all convincing. We also find that analyzing the perfect identification case in their paper, leads to a confusion on what normalization factor was finally used. This analysis is valid only when the entropy is used.

3.2 Conditional information leakage degree

We propose another variation of information leakage degree that is more general and can cover a wider range of scenarios. First, consider a program which has a set of high security inputs S, a set of low security inputs L and a set of public outputs O. The adversary wants to learn about the high security inputs S by observing the outputs O given the knowledge of the low security inputs L. To capture this added information for the adversary (low security inputs), we propose the following definition, which we call the conditional information leakage degree

$$D_c = \frac{H(S|(O,\ L))}{H(S|L)}.$$

This can be formalized in HOL as

```
⊨ D_c p S L O = H(S|(O,L)) / H(S|L)
```

Just like the previous case, consider the two extremes of perfect security and perfect identification. When the outputs and the secret inputs are independent, given L, the conditional entropy $H(S|(O,\ L))$ is equal to $H(S|L)$ which results in a conditional leakage degree equal to 1 for perfect security. However, if the public inputs and outputs completely identify the secret inputs, then $H(S|(O,\ L))$ is equal to 0 and so is the conditional leakage degree in the case of perfect identification. As in the case of leakage degree, we are also able to show that the conditional information leakage degree lies within the interval $(0, 1)$.

```
⊢ 0 ≤ D_c p X Y Z ≤ 1
```

We also prove that the conditional information leakage degree can be written in terms of the conditional mutual information and the conditional entropy.

```
⊢ D_c p S L O = 1 - I(S;O|L) / H(S|L)
```

This shows that this definition is clearly a generalization of the information leakage degree for the case of programs with additional low security inputs. We provide more intuition to interpret this definition by proving the data processing inequality (DPI) [6], which is an important result in information theory that is used, for instance, in statistics to define the notion of sufficient statistic. Random variables X, Y, Z are said to form a Markov chain in that order (denoted by $X \to Y \to Z$) if the conditional distribution of Z depends only on Y and is conditionally independent of X. Specifically, X, Y and Z form a Markov chain $X \to Y \to Z$ if the joint probability mass function can be written as $p(x,\ y,\ z) = p(x)p(y|x)p(z|y)$. We formalize this in HOL as follows

```
⊨ markov_chain p X Y Z =
   ∀ x y z. p X Y Z {(x, y, z)} =
   p X {x} * p Y|X {(y, x)} * p Z|Y {(z, y)}
```

We prove that $X \to Y \to Z$ is equivalent to the statement that X and Z are conditionally independent given Y. In fact, $p(x)p(y|x)p(z|y) = p(x, y)p(z|y) = p(x|y)p(z|y)p(y)$. This in turn is equivalent to $I(X; Z|Y) = 0$. This result allows us to prove the DPI theorem stating that, if $X \to Y \to Z$ then $I(X; Z) \leq I(X; Y)$.

We prove the DPI theorem using the properties of the mutual information. In fact, as shown previously, $I(X; (Y, Z)) = I(X; Z) + I(X; Y|Z)$. By symmetry of the mutual information, we also have $I(X; (Y, Z)) = I(X; Y) + I(X; Z|Y) = I(X; Y)$. The last equality results from the fact that $I(X; Z|Y) = 0$ for a Markov chain. Using the non-negativity of the conditional mutual information, it is straightforward to conclude that $I(X; Z) \leq I(X; Y)$.

We make use of the DPI to interpret the conditional information leakage degree. For a system with high security inputs S, low security inputs L and outputs O, if the outputs depend only on the low security inputs, i.e., $p(O|S, L) = p(O|L)$ then $S \to L \to O$ and S and O are conditionally independent given L. This is the perfect security scenario, for which $D_c = 1$. Using the DPI, we conclude that $I(S; O) \leq I(S; L)$. This means that when the conditional mutual information leakage is equal to 1, no clever manipulation of the low security inputs, by the attacker, deterministic or random, can increase the information that L contains about S, $(I(S; L))$.

The information leakage degrees that we have introduced in this section can be used to reason about information flow analysis of real-world protocols and programs. In the next section, we show two simple yet illustrative examples of how to use a theorem prover and our formalization of information measures to evaluate the properties of security protocols.

4 Applications

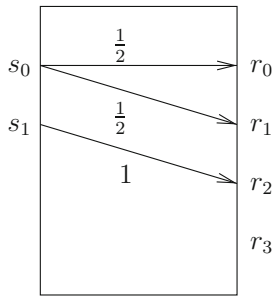
In order to illustrate the effectiveness and utilization of the formalization presented in the previous section, we use it to perform quantitative analysis of information flow and apply it to evaluate the anonymity properties of an anonymity-based single MIX as well as the properties of a classical encryption technique, namely the OTP.

4.1 Anonymity-based single MIX

In this section, we use our formalization to reason about an anonymity-based single MIX, designed to hide the communication links between a set of senders and a set of receivers. We model a single MIX as a communication node connecting m senders $(s_1, \ldots, s_m)$ to n receivers $(r_1, \ldots, r_n)$. The single MIX is determined by its inputs (senders), outputs (receivers) and the transition probabilities. We can also add clauses in the specification to capture additional information about the MIX like structural symmetry. The following is a formalization of the single MIX given in Fig. 1.

```
⊨ MIX_channel s m X Y =
   (X(s) = {0; 1}) ∧
   (Y(s) = {0; 1; 2; 3}) ∧
   (p Y|X {0} {0} = 1/2) ∧
   (p Y|X {1} {0} = 1/2) ∧
   (p Y|X {2} {1} = 1)
```

Fig. 1 Single MIX



Zhu and Bettati [29] used the single MIX to model an anonymity-based covert-channel where a sender is trying to covertly send messages through the MIX. They used the channel capacity as a measure of the maximum information that can be leaked through the MIX and can be used as a measure of the quality of anonymity of the network. A communication between a sender s_i and a receiver r_j is denoted by $[s_i, r_j]$. The term $p([s_u, r_v]_s|[s_i, r_j]_a)$ represents the probability that the communication $[s_u, r_v]$ is suspected given that $[s_i, r_j]$ is actually taking place. This model describes attacks on sender–receiver anonymity. The input symbols of the covert-channel are the actual sender–receiver pairs $[s, r]_a$ and the output symbols are the suspected pairs $[s, r]_s$. In this case, $p([s, r]_s|[s, r]_a)$ represents the result of the anonymity attack. We consider the case where an attacker can establish a covert-channel by having one sender s_1 communicate with any combination of j receivers. The same reasoning can be applied to multiple senders. The authors claim the following result [29]:

Lemma 2 *For a single sender s_1 on a single MIX, the maximum covert-channel capacity is achieved when s_1 can communicate to all receivers.*

We initially tried to formally verify this result, using the foundational results presented in the previous section, but we found a counter-example for an assumption upon which the paper-and-pencil proof of Lemma 2 is based [29]. The erroneous assumption states that the maximum of the mutual information is achieved when all input symbols have non-zero probabilities regardless of the transition probabilities (the results of the anonymity attack). We are able to prove in HOL that it is not necessary for the sender s_1 to communicate with all receivers to achieve capacity.

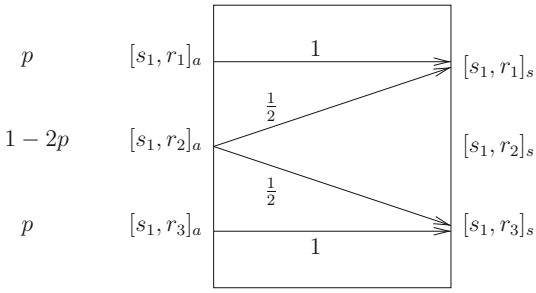
First, we provide a higher-logic-formalization of the channel capacity which is defined as the maximum, over all input distributions, of the mutual information between the input and the output of the channel. We formalize it in HOL using the Hilbert-choice operator; i.e., if it exists, the capacity is some c such that $c = I_m(X; Y)$ for some probability distribution m and for any input distribution p, $I_p(X; Y) \leq c$.

```
⊨ capacity s X Y = @c.
    ∃m. c = I_m(X;Y) ∧ ∀m. I_m(X;Y) ≤ c
```

Next, consider the covert-channel depicted in Fig. 2. To simplify the notation, let $x_i = [s_1, r_i]_a$ and $y_i = [s_1, r_i]_s$. This covert-channel is formalized in HOL as

```
⊨ MIX_channel_1 s m X Y =
    (X(s) = {0; 1; 2}) ∧
    (Y(s) = {0; 1; 2})  ∧
    (p X {0} = p X {2}) ∧
    (p Y|X {0} {0} = 1) ∧
```

Fig. 2 Single MIX example




```
(p Y|X {0} {1} = 1 / 2) ∧
(p Y|X {0} {2} = 0) ∧
(p Y|X {1} {0} = 0) ∧
(p Y|X {1} {1} = 0) ∧
(p Y|X {1} {2} = 0) ∧
(p Y|X {2} {0} = 0) ∧
(p Y|X {2} {1} = 1 / 2) ∧
(p Y|X {2} {2} = 1)
```

We prove that its mutual information is equal to $2p$.

```
⊢ ∀X Y s. MIX_channel_1 s m X Y ⇒
      I(X;Y) = 2 * p X {0}
```

We also prove that the capacity is equal to 1 and corresponds to $p = \frac{1}{2}$. This means that the input distribution that achieves the channel capacity is $[p\{x_0\} = \frac{1}{2},\ p\{x_1\} = 0,\ p\{x_2\} = \frac{1}{2}]$. Hence, we prove that the sender s_1 does not need to communicate with the receiver r_2 and still achieve maximum capacity, contradicting Lemma 2. Notice that with $p = \frac{1}{2},\ I(X;Y) = H(X) = 1$ which implies that the degree of information leakage $D = 0$. So for this covert-channel, the maximum capacity corresponds to perfect identification.

Unlike the paper-and-pencil based analysis, a machine-assisted analysis of quantitative information flow using theorem proving guarantees the accuracy of the results. In fact, the soundness of theorem proving inherently ensures that only valid formulas are provable. The requirement that every single step of the proof needs to be derived from axioms or previous theorems using inference rules, allows us to find missing assumptions and even sometimes wrong statements as was the case in the single MIX application. We have detected this problem while conducting the proof using the HOL4 theorem prover and more specifically when trying to prove the intermediate erroneous result.

4.2 One-time pad

The OTP is a simple yet solid encryption system that provides, if used correctly, an unbreakable security. The encryption is performed by modular addition of every character of the plaintext with a character from a secret random key of at least the same length as the original message. If the key is truly random and never reused in whole or in part, then it can be proven that the OTP encryption provides a perfect security. We formally prove this property using the HOL4 theorem prover using the higher-order-logic formalization we presented in Sect. 2.

The OTP encryption technique takes its name from the paper pads that have been historically used to distribute the keys, making it easy to simply pull the top sheet of the pad and

Fig. 3 A Russian one-time pad, captured by MI5 (*courtesy* Marcus J. Ranum)

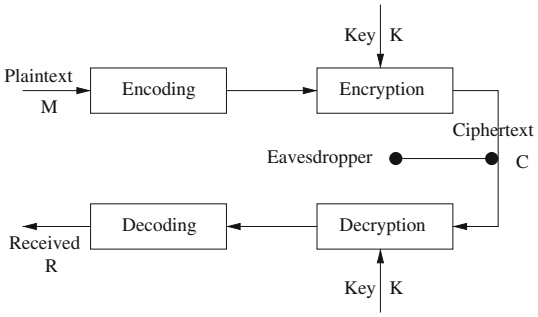


Fig. 4 One-time pad encryption

destroy it after use. An example of a Russian OTP that was captured by MI5 is depicted in Fig. 3.

The OTP has been extensively used to secure the communications of various international intelligence agencies and was used for instance in the Washington/Moscow hotline to provide perfectly secure communication between Washington and the Kremlin and without disclosing any other secret cryptographic technology.

The main challenges for this encryption technique are the generation of truly random keys and their distribution to both sender and receiver. This sometimes makes the technique impractical and limits the types of its applications to the cases where, for example, absolute security is a real must, regardless of the costs. Still, the OTP is available as a backup encryption option if other theoretically less secure but more practical encryption systems are unavailable for reasons of war or attacks. The OTP encryption is also very important in the situation where both sender and receiver need to do all the work by hand without the use of a computer, whether because one is not available or to avoid possible vulnerabilities of a standard computer.

The structure of a typical OTP encryption system is depicted in Fig. 4. The plaintext is first encoded into digits or bits then fed to the encryption block which performs a modular additional (modulo 10 or modulo 2) to produce a cipher text. The latter is transmitted to the receiver side which performs the inverse operations to recover the original message.

4.2.1 Encoding–decoding

A checkerboard [25] is a conversion scheme to convert alphabetic plaintext into digits to prepare it for encryption. Several types of checkerboards have been proposed. We use a *straddling checkerboard* in which the more frequent letters in a language are encoded with

Table 2 Straddling checkerboard example

	0	1	2	3	4	5	6	7	8	9
	A	T		O	N	E		S	I	R
2	B	C	D	F	G	H	J	K	L	M
6	P	Q	U	V	W	X	Y	Z	.	/

a lower number of digits, leading to a compressed output and, hence, shorter messages to be transmitted. Besides, a straddling checkerboard allows to achieve a simple form of information diffusion, or in other words, it reduces the redundancy in the statistics of the plaintext. An example checkerboard for the English language can be found in Table 2. We formalize the straddling checkerboard as the function `checkerboard` of the HOL type,

```
⊨ checkerboard: char -> num
```

We present the definition of `checkerboard` associated with Table 2 for the first-row letters as well as `P` and `/`.

```
⊨ (checkerboad #''A'' = 0)  ∧
   (checkerboad #''T'' = 1)  ∧
   (checkerboad #''O'' = 3)  ∧
   (checkerboad #''N'' = 4)  ∧
   (checkerboad #''E'' = 5)  ∧
   (checkerboad #''S'' = 7)  ∧
   (checkerboad #''I'' = 8)  ∧
   (checkerboad #''R'' = 9)  ∧
   (checkerboad #''P'' = 60) ∧
   (checkerboad #''/'' = 69)
```

Using the above definition of the straddling checkerboard, we formalize the encoding and decoding blocks as `encode` and `decode` functions, respectively. The encoder takes as input a string representing the alphabetic plaintext which it explodes into a list of characters, each of which is processed through the checkerboard, and returns a list of digits. The decoder performs the inverse operations to convert a list of digits back to a string. The functions `encode` and `decode` have the following HOL types:

```
⊨ encode: string → num list;
⊨ decode: num list → string
```

4.2.2 Encryption–decryption

The encryption and decryption blocks are formalized as two functions, `encrypt` and `decrypt`, taking as input a pair of same length lists of digits and returning a list of digits.

```
⊨ encrypt:(num list, num list) → num list
⊨ decrypt:(num list, num list) → num list
```

The encryption is performed by a *modulo* 10 addition, digit by digit, of the list representing the encoded message and the list of digits representing the OTP key. The result of this operation is a ciphertext which is also represented by a list of digits. On the receiver side, the

ciphertext is decrypted by subtracting, *modulo* 10, the key from ciphertext, resulting into a list of numbers that represent the original message. In the case where the plaintext is encoded into bits instead of digits, both encryption and decryption are performed by a simple XOR operation. We formalize `encrypt` in higher-order logic, recursively. h_1 and h_2 represent the first elements or heads of the lists and t_1 and t_2 their tails. The :: operator is the list constructor.

```
⊨  encrypt ([], []) = [] ∧
        ∀ t1 t2 h1 h2.
    encrypt (h1::t1, h2::t2) =
       (h1 + h2) MOD 10::encrypt (t1, t2)
```

Similarly, we formalize the decryption block as follows.

```
⊨  decrypt ([], []) = [] ∧
        ∀ t1 t2 h1 h2.
    decrypt (h1::t1, h2::t2) =
       (h1-h2) MOD 10::decrypt (t1, t2)
```

Finally, let m be the original message (plaintext), k be the OTP key and r be the received message after decryption and decoding. The OTP encryption is then formalized in HOL using the following `OTP` predicate.

```
⊢ ∀ m k r. OTP m k r ⇔
    r = decode(decrypt(encrypt(encode m, k), k))
```

As a reassuring property, we prove in HOL that the OTP as designed and formalized above, ensures that the received message is equal to the original message.

```
⊢ ∀ m k r. OTP m k r ⇒ (r = m)
```

4.2.3 Perfect security

We use our definition of information leakage degree and its formalization in higher-order logic, to prove in HOL that the OTP provides perfect security, i.e., $D = 1$.

We start by formalizing the notion of independence of random variables. Two random variables X and Y are independent iff $\forall A,\ B$, the events $\{X \in A\}$ and $\{Y \in B\}$ are independent.

```
⊨ indep_rv p X Y = ∀A B.
       A ∈ subsets Borel ∧
       B ∈ subsets Borel ⇒
     indep p (PREIMAGE X A ∩ Ω)
             (PREIMAGE Y B ∩ Ω)
```

Let M, C and K denote the random variables representing the plaintext, ciphertext and keys, respectively. Hence, K is uniformly distributed and is independent of M, which allows us to prove that

```
⊢  ∀ m ∈ 𝓜, c ∈ 𝓒.
       P(M=m|C=c) = P(M=m)
```

This follows from the following lemmas,

```
⊢ P(M=m|C=c) = P(M=m, C=c) / P(C=c)
⊢ P(M=m, C=c) = P(M=m, K=m ⊕ c)
⊢ P(M=m, K=m ⊕ c) = P(M=m) P(K=m ⊕ c)
⊢ P(K=m ⊕ c) = 2^-n
⊢ P(C=c) = 2^-n
```

Next, we prove that the conditional entropy of M given C is equal to the entropy of M and that the mutual information $I(M; C)$ is equal to zero.

```
⊢ H(M|C) = H(M)
⊢ I(M;C) = 0
```

Finally, it follows that the information leakage degree introduced in Sect. 3 is equal to 1, meaning that the OTP encryption is information-theoretically secure and there is no leakage of information about the secret input (plaintext) to a possible eavesdropper.

```
⊢ D(M, C) = 1
```

The assumption that k is never reused is actually not directly used in the mathematical proof of perfect secrecy of the OTP. In fact, key reuse would allow to break the encryption through heuristic cryptanalysis not through mathematical analysis. Instead, in a theoretically perfect setting, the one time use of the key is captured by the following assumptions: the ideal randomness of the keys generated and the uniform distribution of the keys. While ideal randomness is a theoretical unattainable concept, the one-time use is the practical realization of the ideal OTP protocol. Of course, the perfect security of the OTP is only ensured in this theoretically perfect setting.

5 Related work

Zhu and Bettati [29] proposed the notion of degree of anonymity which is close to our definition of information leakage degree but we showed that our definition is more general and the two are equal in the case of uniform distribution. Besides, we proposed the conditional information leakage degree, suitable for programs with low security inputs and proved the DPI to give more insight into the intuition behind this new definition. Moreover, our work is based on higher-order-logic theorem proving, which is arguably more sound than the paper-and-pencil based analysis of Zhu and Bettati. In fact, with our analysis we were able to detect the aforementioned problem with the analysis in [29] and provide a counter-example using theorem proving.

Coble [4] formalized some information theory in higher-order logic and used Malacaria's measure of information leakage, i.e., the conditional mutual information [16], to formally analyse the anonymity properties of the Dining Cryptographers protocol. Our formalization of information theory is an extended version of Coble's formalization, i.e., it supports Borel spaces and extended real numbers which allowed us to prove the Radon–Nikodym theorem. Coble's formalization of information theory does not offer these capabilities and thus cannot be used to formally verify the Radon–Nikodym theorem which is useful to verify the properties of various measures of information.

Chatzikokolakis et al. [2] modeled anonymity protocols as noisy channels and used the channel capacity as a measure of the loss of anonymity. In the case where some leakage is intended by design, like in an election protocol, they introduced the notion of conditional capacity which is related to the conditional mutual information. They used the PRISM model

checker [15] to assist in computing the transition probabilities and capacity of the Dining Cryptographers protocol. This analysis technique inherits the state-space explosion problem of model checking, limiting the number of state variables that can be used to represent the protocol. In [2], for instance, the anonymity property of the protocol have been proven for only three cryptographers. The same result can be derived using our framework for an arbitrary number N of cryptographers. In fact, probabilistic model checking is not designed to verify universally quantified generic mathematical relationships like we have been able to verify in the reported work.

The underlying theories over which we built this work are mainly from [18,19]. In [18], we provided a formalization of the measure theory and Lebesgue integration in HOL and proved some classical probability results like the Weak Law of Large Numbers. In [19], we formalized extended reals and based on them provided a more extensive formalization of measure and Lebesgue integration. We also formalized the Shannon entropy and relative entropy and proved the Asymptotic Equipartition Property. In the current paper, we enrich the underlying theories by adding, for instance, products of measure spaces and joint distributions. The main difference, however, is that in this paper we propose new measures of information leakage and formalize various other measures like mutual information and conditional mutual information based on a unified definition of the KL divergence. We use the framework in the evaluation of the anonymity properties of an anonymity-based single MIX and the confidentiality properties of the OTP encryption system.

6 Conclusions

In this paper, we proposed an alternative way to evaluate the anonymity and confidentiality properties of programs and protocols, by conducting a quantitative analysis on the information flow within these programs using a higher-order-logic theorem prover. The deduction style used in the theorem prover to derive proofs offers a high degree of trust in the accuracy of the analysis.

For this purpose, we provided a formalization of the KL divergence in the HOL4 theorem prover and used it to formalize various measures of information leakage that have been proposed in the literature such as the entropy, mutual information and conditional mutual information.

We have introduced two new measures of information, namely the information leakage degree and the conditional information leakage degree, and showed that these definitions are more generic to other comparable measures.

We also showed how formal analysis of information flow can be used in the evaluation of the properties of security protocols by proving the perfect security property of the OTP encryption system. We formalized the various blocks of the encryption system and proved the property as a general mathematical result, compared to computer simulation which can be used to prove such properties but less accurately due to numerical approximations and more importantly because it is not exhaustive.

We also showcased the benefit of using theorem proving to conduct such analysis when compared to paper-and-pencil analysis even for small applications. The benefit is even greater for larger systems or when dealing with parallel and distributed systems. In fact, we were able to come up with a counter-example to a result that appeared in a prominent paper [29] related to the anonymity-based single MIX for which we proved in HOL that the senders need not communicate with all the receivers to achieve channel capacity.

Our future plans include using this framework to study the properties of the Crowds [24] and Tor [9] protocols within the theorem prover.

References

1. Andrews PB (2002) An introduction to mathematical logic and type theory: to truth through proof. Springer, Heidelberg
2. Chatzikokolakis K, Palamidessi C, Panangaden P (2007) Anonymity protocols as noisy channels. In: Trustworthy global computing, LNCS, vol 4661. Springer-Verlag, Heidelberg, pp 281–300
3. Church A (1940) A formulation of the simple theory of types. J Symb Log 5:56–68
4. Coble AR (2008) Formalized information-theoretic proofs of privacy using the HOL4 theorem-prover. In: Privacy enhancing technologies, LNCS, vol 5134. Springer-Verlag, Heidelberg, pp 77–98
5. Coble AR (2010) Anonymity, information, and machine-assisted proof. PhD Thesis, University of Cambridge
6. Cover TM, Thomas JA (1991) Elements of information theory. Wiley-Interscience, New York
7. Deng Y, Pang J, Wu P (2007) Measuring anonymity with relative entropy. In: Formal aspects in security and trust, LNCS, vol 4691. Springer, pp 65–79
8. Diaz C, Seys S, Claessens J, Preneel B (2003) Towards measuring anonymity. In: Privacy enhancing technologies, LNCS, vol 2482. Springer, Heidelberg, pp 54–68
9. Dingledine R, Mathewson N, Syverson P (2004) Tor: the second-generation onion router. In: Proceedings of the 13th USENIX security symposium
10. Goldberg RR (1976) Methods of real analysis. Wiley, New York
11. Gordon MJC (1989) Mechanizing programming logics in higher-order logic. In: Current trends in hardware verification and automated theorem proving. Springer, New York, pp 387–439
12. Harrison J (1996) Formalized mathematics. Technical Report 36. Turku Centre for Computer Science, Finland
13. Harrison J (2009) Handbook of practical logic and automated reasoning. Cambridge University Press, Cambridge
14. Kolmogorov AN (1933) Grundbegriffe der Wahrscheinlichkeitsrechnung. Springer, Berlin. English translation (1950): foundations of the theory of probability. Chelsea, New York
15. Kwiatkowska M, Norman G, Parker D (2005) Quantitative analysis with the probabilistic model checker PRISM. Electron Notes Theor Comput Sci 153(2):5–31
16. Malacaria P (2007) Assessing security threats of looping constructs. SIGPLAN Notes 42(1):225–235
17. Mhamdi T (2013) Probability and information theories in HOL. Hardware Verification Group (HVG), Concordia University, Montreal, QC. https://github.com/mn200/hol/tree/master/src/probability
18. Mhamdi T, Hasan O, Tahar S (2010) On the formalization of the Lebesgue integration theory in HOL. In: Interactive theorem proving, LNCS, vol 6172. Springer, Heidelberg, pp 387–402
19. Mhamdi T, Hasan O, Tahar S (2011) Formalization of entropy measures in HOL. In: Interactive theorem proving, LNCS, vol 6898. Springer, Heidelberg, pp 233–248
20. Mhamdi T, Hasan O, Tahar S (2012) Quantitative analysis of information flow using theorem proving. In: International conference on formal engineering methods, LNCS, vol 7635. Springer-Verlag, Heidelberg, pp 119–134
21. Miller F (1882) Telegraphic code to insure privacy and secrecy in the transmission of telegrams. C.M. Cornwell, New York
22. Milner R (1977) A theory of type polymorphism in programming. J Comput Syst Sci 17:348–375
23. Paulson LC (1996) ML for the working programmer. Cambridge University Press, Cambridge
24. Reiter MK, Rubin AD (1998) Crowds: anonymity for web transactions. ACM Trans Inf Syst Secur 1(1):66–92
25. Rijmenants D (2004) One-time pad. Cipher Machines and Cryptology. http://users.telenet.be/d.rijmenants/en/table.htm
26. Sabelfeld A, Myers AC (2003) Language-based information-flow security. IEEE J Sel Areas Commun 21(1):5–19
27. Serjantov A, Danezis G (2003) Towards an information theoretic metric for anonymity. In: Privacy enhancing technologies, LNCS, vol 2482. Springer, Heidelberg , pp 259–263
28. Smith G (2009) On the foundations of quantitative information flow. In: Foundations of software science and computational structures, LNCS, vol 5504. Springer, York, pp 288–302
29. Zhu Y, Bettati R (2009) Information leakage as a model for quality of anonymity networks. IEEE Trans Parallel Distrib Syst 20(4):540–552