On Bounding Problems of Quantitative Information Flow*

Hirotoshi Yasuoka and Tachio Terauchi

Tohoku University
yasuoka@kb.ecei.tohoku.ac.jp,
terauchi@ecei.tohoku.ac.jp

Abstract. Researchers have proposed formal definitions of quantitative information flow based on information theoretic notions such as the Shannon entropy, the min entropy, the guessing entropy, and channel capacity. This paper investigates the hardness of precisely checking the quantitative information flow of a program according to such definitions. More precisely, we study the "bounding problem" of quantitative information flow, defined as follows: Given a program M and a positive real number q, decide if the quantitative information flow of M is less than or equal to q. We prove that the bounding problem is not a k-safety property for any k (even when q is fixed, for the Shannon-entropy-based definition with the uniform distribution), and therefore is not amenable to the self-composition technique that has been successfully applied to checking non-interference. We also prove complexity theoretic hardness results for the case when the program is restricted to loop-free boolean programs. Specifically, we show that the problem is PP-hard for all the definitions, showing a gap with non-interference which is coNP-complete for the same class of programs. The paper also compares the results with the recently proved results on the comparison problems of quantitative information flow.

1 Introduction

We consider programs containing high security inputs and low security outputs. Informally, the quantitative information flow problem concerns the amount of information that an attacker can learn about the high security input by executing the program and observing the low security output. The problem is motivated by applications in information security. We refer to the classic by Denning [11] for an overview.

In essence, quantitative information flow measures *how* secure, or insecure, a program (or a part of a program –e.g., a variable–) is. Thus, unlike non-interference [9,12], that only tells whether a program is completely secure or not completely secure, a definition of quantitative information flow must be able to

* This work was supported by MEXT KAKENHI 20700019, 20240001, and 22300005, and Global COE Program "CERIES".

D. Gritzalis, B. Preneel, and M. Theoharidou (Eds.): ESORICS 2010, LNCS 6345, pp. 357–372, 2010.

distinguish two programs that are both interferent but have different degrees of "secureness."

For example, consider the programs $M_1 \equiv \texttt{if}\ H = g\ \texttt{then}\ O := 0\ \texttt{else}\ O := 1$ and $M_2 \equiv O := H$. In both programs, H is a high security input and O is a low security output. Viewing H as a password, M_1 is a prototypical login program that checks if the guess g matches the password.[1] By executing M_1, an attacker only learns whether H is equal to g, whereas she would be able to learn the entire content of H by executing M_2. Hence, a reasonable definition of quantitative information flow should assign a higher quantity to M_2 than to M_1, whereas non-interference would merely say that M_1 and M_2 are both interferent, assuming that there are more than one possible value of H.

Researchers have attempted to formalize the definition of quantitative information flow by appealing to information theory. This has resulted in definitions based on the Shannon entropy [11,6,15], the min entropy [24], the guessing entropy [14,1], and channel capacity [18,16,22]. All of these definitions map a program (or a part of a program) onto a non-negative real number, that is, they define a function $\mathcal{X}$ such that given a program M, $\mathcal{X}(M)$ is a non-negative real number. (Concretely, $\mathcal{X}$ is $SE[\mu]$ for the Shannon-entropy-based definition with the distribution μ, $ME[\mu]$ for the min-entropy-based definition with the distribution μ, $GE[\mu]$ for the guessing-entropy-based definition with the distribution μ, and CC for the channel-capacity-based definition.) Therefore, a natural verification problem for quantitative information flow is to decide, given M and a quantity $q \geq 0$, if $\mathcal{X}(M) \leq q$. The problem is well-studied for the case $q = 0$ as it is actually equivalent to checking non-interference (cf. Section 2.1). The problem is open for $q > 0$. We call this the *bounding problem* of quantitative information flow.

The problem has a practical relevance as a user is often interested in knowing if her program leaks information within some allowed bound. That is, the bounding problem is a form of quantitative information flow *checking* problem (as opposed to *inference*). Much of the previous research has focused on information theoretic properties of quantitative information flow and approximate (i.e., incomplete and/or unsound) algorithms for checking and inferring quantitative information flow. To fill the void, in a recent work [29], we have studied the hardness and possibilities of deciding the *comparison problem* of quantitative information flow, which is the problem of precisely checking if the information flow of one program is larger than that of the other, that is, the problem of deciding if $\mathcal{X}(M_1) \leq \mathcal{X}(M_2)$ given programs M_1 and M_2. The study has lead to some remarkable results, summarized in Section 3 and Section 4 of this paper to contrast with the new results on the bounding problem. However, the hardness results on the comparison problem do not imply hardness of the bounding problem.[2] Thus, this paper settles the open question.

[1] Here, for simplicity, we assume that g is a program constant. See Section 2 for modeling attacker/user (i.e., low security) inputs.

[2] But, they imply the hardness of the inference problem because we can compare $\mathcal{X}(M_1)$ and $\mathcal{X}(M_2)$ once we have computed them.

We summarize the main results of the paper below. Here, $\mathcal{X}$ is $SE[U]$, $ME[U]$, $GE[U]$ or CC, where U is the uniform distribution.

- Checking if $\mathcal{X}(M) \leq q$ is not a k-safety property [25,8] for any k.
- Restricted to loop-free boolean programs, checking if $\mathcal{X}(M) \leq q$ is PP-hard.

Roughly, a verification problem being k-safety means that it can be reduced to a standard safety problem, such as the unreachability problem, via self composition [3,10]. For instance, non-interference is a 2-safety property (technically, for the termination-insensitive case[3]), and this has enabled its precise checking via a reduction to a safety problem via self composition and applying automated safety verification techniques [25,21,27]. Also, our recent work [29] has shown that deciding the comparison problem of quantitative information flow for all distributions for the entropy-based definitions (i.e., checking if $\forall\mu.SE[\mu](M_1) \leq SE[\mu](M_2)$, $\forall\mu.ME[\mu](M_1) \leq ME[\mu](M_2)$, and $\forall\mu.GE[\mu](M_1) \leq GE[\mu](M_2)$) are 2-safety problems (and in fact, all equivalent).

We also prove a complexity theoretic gap with these related problems. We have shown in the previous paper [29] that, for loop-free boolean programs, both checking non-interference and the above comparison problem for entropy-based definitions with universally quantified distributions are coNP-complete. (PP is believed to be strictly harder than coNP. In particular, coNP = PP implies the collapse of the polynomial hierarchy to level 1.)

Therefore, the results suggest that the bounding problems of quantitative information flow are harder than the related problems of checking non-interference and the quantitative information flow comparison problems with universally quantified distributions, and may require different techniques to solve (i.e., not self composition).

The rest of the paper is organized as follows. Section 2 reviews the existing information-theoretic definitions of quantitative information flow and formally defines the bounding problems. Section 3 proves that the bounding problems are not k-safety problems. Section 4 proves that the bounding problems are PP-hard (even) when restricted to loop-free boolean programs. Section 5 discusses some implications of the hardness results. Section 6 discusses related work, and Section 7 concludes. All the proofs appear in the extended report [28].

2 Preliminaries

We introduce the information theoretic definitions of quantitative information flow that have been proposed in literature. First, we review the notion of the *Shannon entropy* [23], $\mathcal{H}[\mu](X)$, which is the average of the information content, and intuitively, denotes the uncertainty of the random variable X.

Definition 1 (Shannon Entropy). *Let X be a random variable with sample space $\mathbb{X}$ and μ be a probability distribution associated with X (we write μ explicitly for clarity). The Shannon entropy of X is defined as*

[3] We restrict to terminating programs in this paper. (The termination assumption is nonrestrictive because we assume safety verification as a blackbox routine.).

$$\mathcal{H}[\mu](X) = \sum_{x \in \mathbb{X}} \mu(X = x) \log \frac{1}{\mu(X = x)}$$

(The logarithm is in base 2.)

Next, we define *conditional entropy*. Informally, the conditional entropy of X given Y denotes the uncertainty of X after knowing Y.

Definition 2 (Conditional Entropy). *Let X and Y be random variables with sample spaces $\mathbb{X}$ and $\mathbb{Y}$, respectively, and μ be a probability distribution associated with X and Y. Then, the conditional entropy of X given Y, written $\mathcal{H}[\mu](X|Y)$ is defined as*

$$\mathcal{H}[\mu](X|Y) = \sum_{y \in \mathbb{Y}} \mu(Y = y) \mathcal{H}[\mu](X|Y = y)$$

where

$$\mathcal{H}[\mu](X|Y = y) = \textstyle\sum_{x \in \mathbb{X}} \mu(X = x|Y = y) \log \frac{1}{\mu(X=x|Y=y)}$$
$$\mu(X = x|Y = y) = \frac{\mu(X=x,Y=y)}{\mu(Y=y)}$$

Next, we define (conditional) mutual information. Intuitively, the conditional mutual information of X and Y given Z represents the mutual dependence of X and Y after knowing Z.

Definition 3 (Mutual Information). *Let X, Y and Z be random variables and μ be an associated probability distribution.*[4] *Then, the conditional mutual information of X and Y given Z is defined as*

$$\begin{aligned} \mathcal{I}[\mu](X;Y|Z) &= \mathcal{H}[\mu](X|Z) - \mathcal{H}[\mu](X|Y,Z) \\ &= \mathcal{H}[\mu](Y|Z) - \mathcal{H}[\mu](Y|X,Z) \end{aligned}$$

Let M be a program that takes a high security input H and a low security input L, and gives the low security output O. For simplicity, we restrict to programs with just one variable of each kind, but it is trivial to extend the formalism to multiple variables (e.g., by letting the variables range over tuples). Also, for the purpose of the paper, unobservable (i.e., high security) outputs are irrelevant, and so we assume that the only program output is the low security output. Let μ be a probability distribution over the values of H and L. Then, the semantics of M can be defined by the following probability equation. (We restrict to terminating deterministic programs in this paper.)

$$\mu(O = o) = \sum_{\substack{h,\, \ell \,\in\, \mathbb{H},\, \mathbb{L} \\ M(h,\, \ell) \,=\, o}} \mu(H = h, L = \ell)$$

Note that we write $M(h, \ell)$ to denote the low security output of the program M given inputs h and ℓ. Now, we are ready to introduce the Shannon-entropy based definition of quantitative information flow (QIF) [11,6,15].

[4] We abbreviate the sample spaces of random variables when they are clear from the context.

Definition 4 (Shannon-Entropy-based QIF). *Let M be a program with a high security input H, a low security input L, and a low security output O. Let μ be a distribution over H and L. Then, the Shannon-entropy-based quantitative information flow is defined*

$$\begin{aligned} SE[\mu](M) &= \mathcal{I}[\mu](O; H|L) \\ &= \mathcal{H}[\mu](H|L) - \mathcal{H}[\mu](H|O, L) \end{aligned}$$

Intuitively, $\mathcal{H}[\mu](H|L)$ denotes the initial uncertainty knowing the low security input and $\mathcal{H}[\mu](H|O, L)$ denotes the remaining uncertainty after knowing the low security output.

As an example, consider the programs M_1 and M_2 from Section 1. For concreteness, assume that g is the value 01 and H ranges over the space $\{00, 01, 10, 11\}$. Let U be the uniform distribution over $\{00, 01, 10, 11\}$, that is, $U(h) = 1/4$ for all $h \in \{00, 01, 10, 11\}$. Computing their Shannon-entropy based quantitative information flow, we have,

$$\begin{aligned} SE[U](M_1) &= \mathcal{H}[U](H) - \mathcal{H}[U](H|O) = \log 4 - \tfrac{3}{4} \log 3 \approx .81128 \\ SE[U](M_2) &= \mathcal{H}[U](H) - \mathcal{H}[U](H|O) = \log 4 - \log 1 = 2 \end{aligned}$$

Hence, if the user was to ask if $SE[U](M_1) \leq 1.0$, that is, "does M_1 leak more than one bit of information (according to $SE[U]$)?", then the answer would be no. But, for the same query, the answer would be yes for M_2.

Next, we introduce the *min entropy*, which Smith [24] recently suggested as an alternative measure for quantitative information flow.

Definition 5 (Min Entropy). *Let X and Y be random variables, and μ be an associated probability distribution. Then, the min entropy of X is defined*

$$\mathcal{H}_\infty[\mu](X) = \log \frac{1}{\mathcal{V}[\mu](X)}$$

and the conditional min entropy of X given Y is defined

$$\mathcal{H}_\infty[\mu](X|Y) = \log \frac{1}{\mathcal{V}[\mu](X|Y)}$$

where

$$\begin{aligned} \mathcal{V}[\mu](X) &= \max_{x \in \mathbb{X}} \mu(X = x) \\ \mathcal{V}[\mu](X|Y = y) &= \max_{x \in \mathbb{X}} \mu(X = x | Y = y) \\ \mathcal{V}[\mu](X|Y) &= \textstyle\sum_{y \in \mathbb{Y}} \mu(Y = y) \mathcal{V}[\mu](X|Y = y) \end{aligned}$$

Intuitively, $\mathcal{V}[\mu](X)$ represents the highest probability that an attacker guesses X in a single try. We now define the min-entropy-based definition of quantitative information flow.

Definition 6 (Min-Entropy-based QIF). *Let M be a program with a high security input H, a low security input L, and a low security output O. Let μ be a distribution over H and L. Then, the min-entropy-based quantitative information flow is defined*

$$ME[\mu](M) = \mathcal{H}_\infty[\mu](H|L) - \mathcal{H}_\infty[\mu](H|O, L)$$

Whereas Smith [24] focused on programs lacking low security inputs, we extend the definition to programs with low security inputs in the definition above. It is easy to see that our definition coincides with Smith's for programs without low security inputs. Also, the extension is arguably natural in the sense that we simply take the conditional entropy with respect to the distribution over the low security inputs.

Computing the min-entropy based quantitative information flow for our running example programs M_1 and M_2 from Section 1 with the uniform distribution, we obtain,

$$ME[U](M_1) = \mathcal{H}_\infty[U](H) - \mathcal{H}_\infty[U](H|O) = \log 4 - \log 2 = 1$$
$$ME[U](M_2) = \mathcal{H}_\infty[U](H) - \mathcal{H}_\infty[U](H|O) = \log 4 - \log 1 = 2$$

Hence, if a user is to check whether $ME[U]$ is bounded by q for $1 \leq q < 2$, then the answer would be yes for M_1, but no for M_2.

The third definition of quantitative information flow treated in this paper is the one based on the guessing entropy [17], that is also recently proposed in literature [14,1].

Definition 7 (Guessing Entropy). *Let X and Y be random variables, and μ be an associated probability distribution. Then, the guessing entropy of X is defined*

$$\mathcal{G}[\mu](X) = \sum_{1 \leq i \leq m} i \times \mu(X = x_i)$$

where $m = |\mathbb{X}|$ and $x_1, x_2, \ldots, x_m$ satisfies $\forall i, j. i \leq j \Rightarrow \mu(X = x_i) \geq \mu(X = x_j)$.

The conditional guessing entropy of X given Y is defined

$$\mathcal{G}[\mu](X|Y) = \sum_{y \in \mathbb{Y}} \mu(Y = y)\mathcal{G}[\mu](X|Y = y)$$

where

$$\mathcal{G}[\mu](X|Y = y) = \textstyle\sum_{1 \leq i \leq m} i \times \mu(X = x_i|Y = y)$$
$$m = |\mathbb{X}| \ \text{ and } \ \forall i, j. i \leq j \Rightarrow \mu(X = x_i|Y = y) \geq \mu(X = x_j|Y = y)$$

Intuitively, $\mathcal{G}[\mu](X)$ represents the average number of times required for the attacker to guess the value of X. We now define the guessing-entropy-based quantitative information flow.

Definition 8 (Guessing-Entropy-based QIF). *Let M be a program with a high security input H, a low security input L, and a low security output O. Let μ be a distribution over H and L. Then, the guessing-entropy-based quantitative information flow is defined*

$$GE[\mu](M) = \mathcal{G}[\mu](H|L) - \mathcal{G}[\mu](H|O, L)$$

Like with the min-entropy-based definition, the previous research on guessing-entropy-based quantitative information flow only considered programs without low security inputs [14,1]. But, it is easy to see that our definition with low security inputs coincides with the previous definitions for programs without low security inputs. Also, as with the extension for the min-entropy-based definition, it simply takes the conditional entropy over the low security inputs.

We test GE on the running example from Section 1 by calculating the quantities for the programs M_1 and M_2 with the uniform distribution.

$$GE[U](M_1) = \mathcal{G}[U](H) - \mathcal{G}[U](H|O) = \tfrac{5}{2} - \tfrac{7}{4} = 0.75$$
$$GE[U](M_2) = \mathcal{G}[U](H) - \mathcal{G}[U](H|O) = \tfrac{5}{2} - 1 = 1.5$$

Hence, if a user is to check whether $GE[U]$ is bounded by q for $0.75 \leq q < 1.5$, then the answer would be yes for M_1, but no for M_2.

The fourth and the final existing definition of quantitative information flow that we introduce in this paper is the one based on *channel capacity* [18,16,22], which is simply defined to be the maximum of the Shannon-entropy based quantitative information flow over the distribution.

Definition 9 (Channel-Capacity-based QIF). *Let M be a program with a high security input H, a low security input L, and a low security output O. Then, the channel-capacity-based quantitative information flow is defined*

$$CC(M) = \max_{\mu} \mathcal{I}[\mu](O; H|L)$$

Unlike the Shannon-entropy based, the min-entropy based, and the guessing-entropy based definitions, the channel-capacity based definition of quantitative information flow is not parameterized by the distribution over the inputs. As with the other definitions, let us test the definition on the running example from Section 1 by calculating the quantities for the programs M_1 and M_2:

$$CC(M_1) = \max_{\mu} \mathcal{I}[\mu](O; H) = 1$$
$$CC(M_2) = \max_{\mu} \mathcal{I}[\mu](O; H) = 2$$

Note that $CC(M_1)$ (resp. $CC(M_2)$) is equal to $ME[U](M_1)$ (resp. $ME[U](M_2)$). This is not a coincidence. In fact, it is known that $CC(M) = ME[U](M)$ for all programs M without low security inputs [24].

2.1 Non-interference

We recall the notion of non-interference [9,12].

Definition 10 (Non-intereference). *A program M is said to be non-interferent iff for any $h, h' \in \mathbb{H}$ and $\ell \in \mathbb{L}$, $M(h, \ell) = M(h', \ell)$.*

It can be shown that for the definitions of quantitative information flow $\mathcal{X}$ introduced above, $\mathcal{X}(M) \leq 0$ iff M is non-interferent.[5] That is, the bounding

[5] Technically, we need the non-zero-ness condition on the distribution for the entropy-based definitions. (See below.).

problem (which we only officially define for positive bounds –see Section 2.2–) degenerates to checking non-interference when 0 is given as the bound.

Theorem 1. *Let μ be a distribution such that $\forall h \in \mathbb{H}, \ell \in \mathbb{L}.\mu(h,\ell) > 0$. Then,*

- *M is non-interferent if and only if $SE[\mu](M) \leq 0$.*
- *M is non-interferent if and only if $ME[\mu](M) \leq 0$.*
- *M is non-interferent if and only if $GE[\mu](M) \leq 0$.*
- *M is non-interferent if and only if $CC(M) \leq 0$.*

The equivalence result on the Shannon-entropy-based definition is proven by Clark et al. [5]. The proofs for the other three definitions are given in the extended report [28].

2.2 Bounding Problem

We define the *bounding problem* of quantitative information flow for each definition of the quantitative information flow introduced above. The bounding problem for the Shannon-entropy based definition $B_{SE}[\mu]$ is defined as follows: Given a program M and a positive real number q, decide if $SE[\mu](M) \leq q$. Similarly, we define the bounding problems for the other three definitions $B_{ME}[\mu]$, $B_{GE}[\mu]$, and B_{CC} as follows.

$$\begin{aligned} B_{ME}[\mu] &= \{(M,q) \mid ME[\mu](M) \leq q\} \\ B_{GE}[\mu] &= \{(M,q) \mid GE[\mu](M) \leq q\} \\ B_{CC} &= \{(M,q) \mid CC(M) \leq q\} \end{aligned}$$

3 K-Safety Property

We show that none of the bounding problems are k-safety problems for any k. Informally, a program property is said to be a *k-safety* property [25,8] if it can be refuted by observing k number of (finite) execution traces. A k-safety problem is the problem of checking a k-safety property. Note that the standard safety property is a 1-safety property. An important property of a k-safety problem is that it can be reduced to a standard safety (i.e., 1-safety) problem, such as the unreachability problem, via a simple program transformation called *self composition* [3,10]. This allows one to verify k-safety problems by applying powerful automated safety verification techniques [2,13,20,4] that have made remarkable progress recently.

As stated earlier, we prove that no bounding problem is a k-safety property for any k. To put the result in perspective, we compare it to the results of the related problems, summarized below. Here, $\mathcal{X}$ is $SE[U]$, $ME[U]$, $GE[U]$, or CC, and $\mathcal{Y}$ is SE, ME, or GE. (Recall that U denotes the uniform distribution.)

(1) Checking non-interference is a 2-safety problem, but it is not 1-safety.
(2) Checking $\mathcal{X}(M_1) \leq \mathcal{X}(M_2)$ is not a k-safety problem for any k.
(3) Checking $\forall \mu.\mathcal{Y}[\mu](M_1) \leq \mathcal{Y}[\mu](M_2)$ is a 2-safety problem.

The result (1) on non-interference is classic (see, e.g., [19,3,10]). The results (2) and (3) on comparison problems are proven in our recent paper [29]. Therefore, this section's results imply that the bounding problems are harder to verify (at least, via the self-composition approach) than non-interference and the comparison problems for the entropy-based definitions of quantitative information flow with universally quantified distributions.

Formally, k-safety property is defined as follows.

Definition 11 (k-safety property). *We say that a property $P \subseteq Prog \times \mathbb{R}^+$ is a k-safety property iff $(M, q) \notin P$ implies that there exists $T \subseteq [\![M]\!]$ such that $|T| \leq k$ and $\forall M'.T \subseteq [\![M']\!] \Rightarrow (M', q) \notin P$.*

Here, *Prog* denotes the set of all programs, and $\mathbb{R}^+$ is the set of positive real numbers. $[\![M]\!]$ denotes the semantics (i.e., traces) of M, represented by the set of input/output pairs $\{((h, \ell), o) \mid h \in \mathbb{H}, \ell \in \mathbb{L}, o = M(h, \ell)\}$. Note that the original definition of k-safety property is only defined over programs [25,8]. However, because the bounding problems take the additional input q, we extend the notion to account for the extra parameter.

We now state the main results of this section which show that none of the bounding problems are k-safety problems for any k. Because we are interested in hardness, we focus on the case where the distribution is the uniform distribution. That is, the results we prove for the specific case applies to the general case.

Theorem 2. *Neither $B_{SE}[U]$, $B_{ME}[U]$, $B_{GE}[U]$, nor B_{CC} is a k-safety property for any k such that $k > 0$.*

We defer the details of the theorem to Section 3.1 (see also Section 5.2) as it can actually be obtained as a corollary of its results.

3.1 K-Safety under a Constant Bound

The result above appears to suggest that the bounding problems are equally difficult for all the definitions of quantitative information flow. However, holding the parameter q constant (rather than having it as an input) paints a different picture. We show that the problems become k-safety for different definitions for different k's under different conditions in this case.

First, for q fixed, we show that the bounding problem for the channel-capacity based definition of quantitative information flow is k-safety for $k = \lfloor 2^q \rfloor + 1$. (Also, this bound is tight.)

Theorem 3. *Let q be a constant. Then, B_{CC} is $\lfloor 2^q \rfloor + 1$-safety, but it is not k-safety for any $k \leq \lfloor 2^q \rfloor$.*

We briefly explain the intuition behind the above result. Recall that a problem being k-safety means the existence of a *counterexample* trace set of size at most k. That is, for $(M, q) \notin B_{CC}$, we have $T \subseteq [\![M]\!]$ such that $|T| \leq \lfloor 2^q \rfloor + 1$ such that any program that also contains T as its traces also does not belong to B_{CC} (with q), that is, its channel-capacity-based quantitative information flow

is greater than q. Then, the above result follows from the fact that the channel-capacity-based quantitative information flow coincides with the maximum over the low security inputs of the logarithm of the number of outputs [16], therefore, any T containing $\lfloor 2^q \rfloor + 1$ traces of the same low security input and disjoint outputs is a counterexample.

For concreteness, we show how to check B_{CC} via self composition. Suppose we are given a program M and a positive real q. We construct the self-composed program M' shown below.

$$\begin{array}{l} M'(H_1, H_2, \ldots, H_n, L) \equiv \\ \quad O_1 := M(H_1, L); O_2 := M(H_2, L); \ldots; O_n := M(H_n, L); \\ \quad \mathsf{assert}(\bigvee_{i,j \in \{1,\ldots,n\}} (O_i = O_j \wedge i \neq j)) \end{array}$$

where $n = \lfloor 2^q \rfloor + 1$. In general, a self composition involves making k copies the original program so that the resulting program would generate k traces of the original (having the desired property). By the result proven by Malacaria and Chen [16], it follows that M' does not cause an assertion failure iff $(M, q) \in B_{CC}$.

Next, we show that for programs without low security inputs, $B_{ME}[U]$ and $B_{GE}[U]$ are also both k-safety problems (but for different k's) when q is held constant.

Theorem 4. *Let q be a constant, and suppose $B_{ME}[U]$ only takes programs without low security inputs. Then, $B_{ME}[U]$ is $\lfloor 2^q \rfloor + 1$-safety, but it is not k-safety for any $k \leq \lfloor 2^q \rfloor$.*

Theorem 5. *Let q be a constant, and suppose $B_{GE}[U]$ only takes programs without low security inputs. If $q \geq \frac{1}{2}$, then, $B_{GE}[U]$ is $\lfloor \frac{(\lfloor q \rfloor + 1)^2}{\lfloor q \rfloor + 1 - q} \rfloor + 1$-safety, but it is not k-safety for any $k \leq \lfloor \frac{(\lfloor q \rfloor + 1)^2}{\lfloor q \rfloor + 1 - q} \rfloor$. Otherwise, $q < \frac{1}{2}$ and $B_{GE}[U]$ is 2-safety, but it is not 1-safety.*

The result for $ME[U]$ follows from the fact that for programs without low security inputs, the min-entropy based quantitative information flow with the uniform distribution is actually equivalent to the channel-capacity based quantitative information flow [24]. The result for $GE[U]$ may appear less intuitive, but, the key observation is that, like the channel-capacity based definition and the min-entropy based definition with the uniform distribution (for the case without low security inputs), for any set of traces $T = [\![M]\!]$, the information flow of a program containing T would be at least as large as that of M. Therefore, by holding q constant, we can always find a large enough counterexample T. The reason $B_{GE}[U]$ is 2-safety for $q < \frac{1}{2}$ is because, in the absence of low security inputs, the minimum non-zero quantity of $GE[U](M)$ is bounded (by 1/2), and so for such q, the problem $GE[U](M) \leq q$ is equivalent to checking non-interference.[6]

But, when low security inputs are allowed, neither $B_{ME}[U]$ nor $B_{GE}[U]$ are k-safety for any k, even when q is held constant.

[6] In fact, the minimum non-zero quantity property also exists for *ME*[U] without low security inputs and *CC*. There, the minimum non-zero quantity is 1, which agrees with the formulas given in the theorems.

Theorem 6. *Let q be a constant. (And let $B_{ME}[U]$ take programs with low security inputs.) Then, $B_{ME}[U]$ is not a k-safety property for any $k > 0$.*

Theorem 7. *Let q be a constant. (And let $B_{GE}[U]$ take programs with low security inputs.) Then, $B_{GE}[U]$ is not a k-safety property for any $k > 0$.*

Finally, we show that the Shannon-entropy based definition (with the uniform distribution) is the hardest of all the definitions and show that its bounding problem is not a k-safety property for any k, with or without low-security inputs, even when q is held constant.

Theorem 8. *Let q be a constant, and suppose $B_{SE}[U]$ only takes programs without low security inputs. Then, $B_{SE}[U]$ is not a k-safety property for any $k > 0$.*

Intuitively, Theorems 6, 7, and 8 follow from the fact that, for these definitions, given any potential counterexample $T \subseteq [\![M]\!]$ to show $(M, q) \notin B_{\mathcal{X}}$, it is possible to find M' containing T whose information flow is arbitrarily close to 0 (and so $(M', q) \in B_{\mathcal{X}}$). See Section 5.2 for further discussion.

Because k tends to grow large as q grows for all the definitions and it is impossible to bound k for all q, this section's results are unlikely to lead to a practical verification method. Nevertheless, the results reveal interesting disparities among the different proposals for the definition of quantitative information flow.

4 Complexities for Loop-Free Boolean Programs

In this section, we analyze the computational complexity of the bounding problems when the programs are restricted to loop-free boolean programs. The purpose of the section is to compare the complexity theoretic hardness of the bounding problems with those of the related problems for the same class of programs, as we have done with the k-safety property of the problems.

That is, we compare against the comparison problems of quantitative information flow and the problem of checking non-interference for loop-free boolean programs. The complexity results for these problems are summarized below. Here, $\mathcal{X}$ is $SE[U]$, $ME[U]$, $GE[U]$, or CC, and $\mathcal{Y}$ is SE, ME, or GE.

(1) Checking non-interference is coNP-complete
(2) Checking $\mathcal{X}(M_1) \leq \mathcal{X}(M_2)$ is PP-hard.
(3) Checking $\forall\mu.\mathcal{Y}[\mu](M_1) \leq \mathcal{Y}[\mu](M_2)$ is coNP-complete.

The results (1) and (3) are proven in our recent paper [29]. The result (2) tightens our (oracle relative) #P-hardness result from the same paper, which states that for each C such that C is the comparison problem for $SE[U]$, $ME[U]$, $GE[U]$, or CC, we have $\#\mathrm{P} \subseteq \mathrm{FP}^C$. (Recall that the notation FP^A means the complexity class of function problems solvable in polynomial time with an oracle for the problem A.) #P is the class of counting problems associated with NP. PP is the class of decision problems solvable in probabilistic polynomial time. PP is known

$$
\begin{array}{rll}
M ::= & x := \psi \mid M_0; M_1 & wp(x := \psi, \phi) = \phi[\psi/x] \\
& \mid \ \mathsf{if}\ \psi\ \mathsf{then}\ M_0\ \mathsf{else}\ M_1 & wp(\mathsf{if}\ \psi\ \mathsf{then}\ M_0\ \mathsf{else}\ M_1, \phi) \\
\phi, \psi ::= & \mathsf{true} \mid x \mid \phi \wedge \psi \mid \neg\phi & \quad = (\psi \Rightarrow wp(M_0, \phi)) \wedge (\neg\psi \Rightarrow wp(M_1, \phi)) \\
 & & wp(M_0; M_1, \phi) = wp(M_0, wp(M_1, \phi))
\end{array}
$$

Fig. 1. The syntax and semantics of loop-free boolean programs

to contain both coNP and NP, PH $\subseteq \mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$ [26], and PP is believed to be strictly larger than both coNP and NP. (In particular, PP = coNP would imply the collapse of the polynomial hierarchy (PH) to level 1.)

We show that, restricted to loop-free boolean programs, the bounding problems for the entropy-based definitions with the uniform distribution (i.e., $SE[U]$, $ME[U]$, and $GE[U]$) and the channel-capacity based definition (i.e., CC) are all PP-hard. The results strengthen the hypothesis that the bounding problems for these definitions are quite hard. Indeed, they show that they are complexity theoretically harder than non-interference and the comparison problems with the universally quantified distributions for loop-free boolean programs, assuming that coNP and PP are separate.

We define the syntax of loop-free boolean programs in Figure 1. We assume the usual derived formulas $\phi \Rightarrow \psi$, $\phi = \psi$, $\phi \vee \psi$, and false. We give the usual weakest precondition semantics in the figure.

To adapt the information flow framework to boolean programs, we make each information flow variable H, L, and O range over functions mapping boolean variables of its kind to boolean values. For example, if x and y are low security boolean variables and z is a high security boolean variable, then L ranges over the functions $\{x, y\} \to \{\mathsf{false}, \mathsf{true}\}$, and H and O range over $\{z\} \to \{\mathsf{false}, \mathsf{true}\}$.[7] (Every boolean variable is either a low security boolean variable or a high security boolean variable.) We write $M(h, \ell) = o$ for an input (h, ℓ) and an output o if $(h, \ell) \models wp(M, \phi)$ for a boolean formula ϕ such that $o \models \phi$ and $o' \not\models \phi$ for all output $o' \neq o$. Here, $\models$ is the usual logical satisfaction relation, using h, ℓ, o, etc. to look up the values of the boolean variables. (Note that this incurs two levels of lookup.)

As an example, consider the following program.

$$M \ \equiv \ z \ \texttt{:=}\ x; w \ \texttt{:=}\ y; \texttt{if}\ x \wedge y\ \texttt{then}\ z \ \texttt{:=}\ \neg z\ \texttt{else}\ w \ \texttt{:=}\ \neg w$$

Let x, y be high security variables and z, w be low security variables. Then,

$$
\begin{array}{ll}
SE[U](M) = 1.5 & GE[U](M) = 1.25 \\
ME[U](M) = \log 3 \approx 1.5849625 & CC(M) = \log 3 \approx 1.5849625
\end{array}
$$

[7] We do not distinguish input boolean variables from output boolean variables. But, a boolean variable can be made output-only by assigning a constant to the variable at the start of the program and made input-only by assigning a constant at the end.

We now state the main results of the section, which show that the bounding problems are PP-hard for all the definitions of quantitative information flow considered in this paper.

Theorem 9. $PP \subseteq B_{SE}[U]$

Theorem 10. $PP \subseteq B_{ME}[U]$

Theorem 11. $PP \subseteq B_{GE}[U]$

Theorem 12. $PP \subseteq B_{CC}$

We remind that the above results hold (even) when the bounding problems $B_{SE}[U]$, $B_{ME}[U]$, $B_{GE}[U]$, and B_{CC} are restricted to loop-free boolean programs. We also note that the results hold even when the programs are restricted to those without low security inputs. These results are proven by a reduction from MAJSAT, which is a PP-complete problem. MAJSAT is the problem of deciding, given a boolean formula ϕ over variables $\overrightarrow{x}$, if there are more than $2^{|\overrightarrow{x}|-1}$ satisfying assignments to ϕ (i.e., whether the majority of the assignments to ϕ are satisfying).

5 Discussion

5.1 Bounding the Domains

The notion of k-safety property, like the notion of safety property from where it extends, is defined over all programs regardless of their size. (For example, non-interference is a 2-safety property for all programs and unreachability is a safety property for all programs.) But, it is easy to show that the bounding problems would become "k-safety" properties if we constrained and bounded the input domains because then the size of the semantics (i.e., the input/output pairs) of such programs would be bounded by $|\mathbb{H}|\times|\mathbb{L}|$. In this case, the problems are at most $|\mathbb{H}|\times|\mathbb{L}|$-safety.[8] (And the complexity theoretic hardness degenerates to a constant.) But, like the k-safety bounds obtained by fixing q constant (cf. Section 3.1), these bounds are high for all but very small domains and are unlikely to lead to a practical verification method. Also, because a bound on the high security input domain puts a bound on the maximum information flow, the bounding problems become a tautology for $q \geq c$, where c is the maximum information flow for the respective definition.

5.2 Low Security Inputs

Recall the results from Section 3.1 that, under a constant bound, the bounding problems for both the min-entropy based definition and the guessing entropy-based definition with the uniform distribution are k-safety for programs without

[8] It is possible to get a tighter bound for the channel-capacity based definition by also bounding the size of the output domain.

low security inputs, but not for those with. The reason for the non-k-safety results is that the definitions of quantitative information flow ME and GE (and in fact, also SE) use the conditional entropy over the low security input distribution and are parameterized by the distribution. This means that the quantitative information flow of a program is averaged over the low security inputs according to the distribution. Therefore, by arbitrarily increasing the number of low security inputs, given any set of traces T, it becomes possible to find a program containing T whose information flow is arbitrarily close to 0 (at least under the uniform distribution). This appears to be a property intrinsic to any definition of quantitative information flow defined via conditional entropy over the low security inputs and is parameterized by the distribution of low security inputs. Note that the channel-capacity based definition does not share this property as it is defined to be the maximum over the distributions. The non-k-safety result for $B_{SE}[U]$ holds even in the absence of low security inputs because the Shannon entropy of a program is the average of the *surprisal* [7] of the individual observations, and so by increasing the number of high security inputs, given any set of traces T, it becomes possible to find a program containing T whose information flow is arbitrarily close to 0.

6 Related Work

This work continues our recent research [29] on investigating the hardness and possibilities of verifying quantitative information flow according to the formal definitions proposed in literature [11,6,15,24,14,1,18,16,22]. Much of the previous research has focused on information theoretic properties of the definitions and proposed approximate (i.e., incomplete and/or unsound) methods for checking and inferring quantitative information flow according to such definitions. In contrast, this paper (along with our recent paper [29]) investigates the hardness and possibilities of precisely checking and inferring quantitative information flow according to the definitions.

This paper has shown that the bounding problem, that is, the problem of checking $\mathcal{X}(M) \leq q$ given a program M and a positive real q, is quite hard (for various quantitative information flow definitions $\mathcal{X}$). This is in contrast to our previous paper that has investigated the hardness and possibilities of the comparison problem, that is, the problem of checking $\mathcal{X}(M_1) \leq \mathcal{X}(M_2)$ given programs M_1 and M_2. To the best of our knowledge, this paper is the first to investigate the hardness of the bounding problems. But, the hardness of quantitative information flow inference, a harder problem, follows from the results of our previous paper, and Backes et al. [1] have also proposed a precise inference method that utilizes self composition and counting algorithms.

While the focus of the work is on verification, in the light of the disparities among the different definitions (cf. Section 3.1 and Section 5), it may be interesting to judge the different proposals based on the hardness of verification. Researchers have also proposed definitions of quantitative information flow that are not considered in the paper. These include the definition based on the notion

of *belief* [7], and the ones that take the maximum over the low security inputs [15,14]. These are subjects of future study.

7 Conclusions and Future Work

In this paper, we have formalized and proved the hardness of the bounding problem of quantitative information flow, which is a form of (precise) checking problem of quantitative information flow. We have shown that no bounding problem is a k-safety property for any k, and therefore that it is not possible to reduce the problem to a safety problem via self composition, at least when the quantity to check against is unrestricted. The result is in contrast to non-interference and the comparison problem for the entropy-based quantitative information flow with universally quantified distribution, which are 2-safety properties. We have also shown a complexity theoretic gap with these problems, which are coNP-complete, by proving the PP-hardness of the bounding problems, when restricted to loop-free boolean programs.

We have also shown that the bounding problems for some quantitative information flow definitions become k-safety for different k's under certain conditions when the quantity to check against is restricted to be a constant, highlighting interesting disparities among the different definitions of quantitative information flow.

A possible future research direction is to investigate the entropy-based bounding problems with their distributions universally quantified, that is, the problem of deciding if $\forall\mu.\mathcal{Y}[\mu](M) \leq q$ where $\mathcal{Y}$ is instantiated with SE, ME, or GE. This is partly motivated by our recent work [29] that has obtained remarkable results by universally quantifying over the distributions in the entropy-based definitions in the comparison problems. (That is, checking $\forall\mu.SE[\mu](M_1) \leq SE[\mu](M_2)$, $\forall\mu.ME[\mu](M_1) \leq ME[\mu](M_2)$, and $\forall\mu.GE[\mu](M_1) \leq GE[\mu](M_2)$ are all equivalent and 2-safety, and so that they can all be checked simultaneously via self composition, and that they are coNP-complete when restricted to loop-free boolean programs –cf. Section 1–.) We actually already know the answer for the Shannon-entropy based definition. That is, $\forall\mu.SE[\mu](M) \leq q$, as this is simply is equivalent to $CC(M) \leq q$, the channel-capacity bounding problem. The problem is open for the other two entropy-based definitions of quantitative information flow.

References

1. Backes, M., Köpf, B., Rybalchenko, A.: Automatic discovery and quantification of information leaks. In: IEEE Symposium on Security and Privacy, pp. 141–153 (2009)
2. Ball, T., Rajamani, S.K.: The SLAM project: debugging system software via static analysis. In: POPL, pp. 1–3 (2002)
3. Barthe, G., D'Argenio, P.R., Rezk, T.: Secure information flow by self-composition. In: CSFW, pp. 100–114 (2004)
4. Beyer, D., Henzinger, T.A., Jhala, R., Majumdar, R.: The software model checker Blast. STTT 9(5-6), 505–525 (2007)
5. Clark, D., Hunt, S., Malacaria, P.: Quantified interference for a while language. Electr. Notes Theor. Comput. Sci. 112, 149–166 (2005)

6. Clark, D., Hunt, S., Malacaria, P.: A static analysis for quantifying information flow in a simple imperative language. Journal of Computer Security 15(3), 321–371 (2007)
7. Clarkson, M.R., Myers, A.C., Schneider, F.B.: Belief in information flow. In: CSFW, pp. 31–45 (2005)
8. Clarkson, M.R., Schneider, F.B.: Hyperproperties. In: CSF, pp. 51–65 (2008)
9. Cohen, E.S.: Information transmission in computational systems. In: SOSP, pp. 133–139 (1977)
10. Darvas, Á., Hähnle, R., Sands, D.: A theorem proving approach to analysis of secure information flow. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 193–209. Springer, Heidelberg (2005)
11. Denning, D.E.R.: Cryptography and data security. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (1982)
12. Goguen, J.A., Meseguer, J.: Security policies and security models. In: IEEE Symposium on Security and Privacy, pp. 11–20 (1982)
13. Henzinger, T.A., Jhala, R., Majumdar, R., Sutre, G.: Lazy abstraction. In: POPL, pp. 58–70 (2002)
14. Köpf, B., Basin, D.: An information-theoretic model for adaptive side-channel attacks. In: CCS, pp. 286–296 (2007)
15. Malacaria, P.: Assessing security threats of looping constructs. In: POPL, pp. 225–235 (2007)
16. Malacaria, P., Chen, H.: Lagrange multipliers and maximum information leakage in different observational models. In: PLAS, pp. 135–146 (2008)
17. Massey, J.L.: Guessing and entropy. In: ISIT, p. 204 (1994)
18. McCamant, S., Ernst, M.D.: Quantitative information flow as network flow capacity. In: PLDI, pp. 193–205 (2008)
19. McLean, J.: A general theory of composition for trace sets closed under selective interleaving functions. In: IEEE Security and Privacy, pp. 79–93 (1994)
20. McMillan, K.L.: Lazy abstraction with interpolants. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 123–136. Springer, Heidelberg (2006)
21. Naumann, D.A.: From coupling relations to mated invariants for checking information flow. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 279–296. Springer, Heidelberg (2006)
22. Newsome, J., McCamant, S., Song, D.: Measuring channel capacity to distinguish undue influence. In: PLAS, pp. 73–85 (2009)
23. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
24. Smith, G.: On the foundations of quantitative information flow. In: de Alfaro, L. (ed.) FOSSACS 2009. LNCS, vol. 5504, pp. 288–302. Springer, Heidelberg (2009)
25. Terauchi, T., Aiken, A.: Secure information flow as a safety problem. In: Hankin, C., Siveroni, I. (eds.) SAS 2005. LNCS, vol. 3672, pp. 352–367. Springer, Heidelberg (2005)
26. Toda, S.: PP is as hard as the polynomial-time hierarchy. SIAM J. Comput. 20(5), 865–877 (1991)
27. Unno, H., Kobayashi, N., Yonezawa, A.: Combining type-based analysis and model checking for finding counterexamples against non-interference. In: PLAS, pp. 17–26 (2006)
28. Yasuoka, H., Terauchi, T.: On bounding problems of quantitative information flow (2010), `http://www.kb.ecei.tohoku.ac.jp/~yasuoka`
29. Yasuoka, H., Terauchi, T.: Quantitative information flow - verification hardness and possibilities. In: CSF (2010)