Scribes: Sam Hopkins, Jed Liu Lecturer: Andrew Myers

1 Introduction

Quantitative information flow was first proposed by Denning [5] and Gray [6]. Its goal is to measure the amount of information leaked about a secret by observing the running of a program. Quantitative definitions of information leakage are often borrowed from information theory. Indeed, early models of quantitative information flow directly used the Shannon mutual information $I(X;Y)$, which measures the (expected) reduction in uncertainty about the value of random variable X given by learning the value of Y (see §2).

Doing information flow quantitatively immediately raises the following issues, among others:

- It is not clear how to analyze code to quantitatively measure its information leakage. Malacaria has done some work on this [7, 3], and there is a body of work on sampling executions of randomized programs and programs with randomized input data.
- Geoffrey Smith [8] claims that Shannon entropy is the wrong measure, because it fails to accurately capture what he calls the *vulnerability* of a program: the probability that the attacker's first guess at the high data is correct. This quantity is related to *min-entropy*. The work is further discussed in §3.
- Dealing with nondeterminism in the program may pose a problem to a theory of quantitative information flow. Orthogonally, using Shannon mutual information directly does not appear to result in a theory that allows compositional reasoning about program executions: if k bits are leaked in the first execution of a program and k' in the second, we would like our theory to say that together the executions leak $k + k'$ bits, but that need not be true if we work directly with mutual information. Clarkson et al. [4], discussed in §4, addresses both of these problems simultaneously.
- Naïvely, $I(X;Y)$ seems to value all secret data equally. However, different pieces of high-security data might be worth more than others—not all leaked bits are equal. This problem is beginning to be addressed [1], but is still at least a partially open question.

2 Uncertainty and Shannon entropy

Consider a program P that takes a high input `H` and produces a low output `L`. An adversary that observes `L` might be able to deduce something about `H`. One approach to quantifying the information leaked by P is to use *uncertainty*: we quantify the amount of information in `H` (the adversary's *initial uncertainty* before observing `L`) and the amount of unleaked information about `H` (the adversary's *remaining uncertainty* after observing `L`). From these, intuitively, we can derive the amount of information leaked to `L`:

$$\text{information leaked} = \text{initial uncertainty} - \text{remaining uncertainty}.$$

Early quantitative measures of uncertainty were based on *Shannon entropy* $H(X)$, which measures the expected number of bits optimally required to transmit the value of the (discrete) random variable X.

$$H(X) = -\sum_{x \in \mathcal{X}} \Pr[X = x] \lg \Pr[X = x],$$

where $\mathcal{X}$ is the support of X and lg is $\log_2$. For X drawn from a uniform distribution over the integers in the interval $[0, n)$, the Shannon entropy gives $H(X) = \lg n$ bits, as intuitively expected. An alternative characterization is $H(X)$ measures the uncertainty (or "surprisal") in the value of X: the average amount that one should be surprised by seeing the value.

We write $H(X, Y)$ to denote the Shannon entropy of a system with two random variables X and Y. When X and Y are dependent, it is often useful to disentagle them. The *conditional entropy* $H(X \mid Y)$

defines the information in X, given knowledge of Y:

$$\begin{aligned} H(X \mid Y) &= \sum_{y \in \mathcal{Y}} \Pr[Y = y] H(X \mid Y = y) \\ &= H(X, Y) - H(Y). \end{aligned}$$

Mutual information $I(X\,;\,Y)$ measures the mutual dependence of X and Y. It gives the amount of information shared between the variables: the entropy reduction in one variable by learning the other. Mutual information is symmetric and non-negative.

$$\begin{aligned} I(X\,;\,Y) &= I(Y\,;\,X) \\ &= H(X) + H(Y) - H(X, Y) \\ &= H(X) - H(X \mid Y) \\ &= H(X, Y) - H(X \mid Y) - H(Y \mid X) \geq 0. \end{aligned}$$

In the context of quantitative information flow, we have initial uncertainty $H(\mathtt{H})$ and remaining uncertainty $H(\mathtt{H} \mid \mathtt{L})$, so the information leaked by the program P is $H(\mathtt{H}) - H(\mathtt{H} \mid \mathtt{L}) = I(\mathtt{H}; \mathtt{L})$.

3 Vulnerability and min-entropy

Geoffrey Smith [8] observed that this Shannon-based definition is not a good characterization of security, because it does not offer good guarantees about the probability of guessing H. For example, consider the program

```
if H mod 8 == 0 then L := H
else L := 1
```

where H is an $8k$-bit integer. As measured by Shannon entropy, the information leakage is $I(\mathtt{H}; \mathtt{L}) = H(\mathtt{L}) \approx k + 0.169$ bits, so the remaining uncertainty $H(\mathtt{H} \mid \mathtt{L})$ is about $7k - 0.169$ bits. But, since the `then` branch is taken $1/8$ of the time, the adversary can guess H at least $1/8$ of the time.

On the other hand, here is a program that leaks the last $k + 1$ bits of H into L, leaving $H(\mathtt{H} \mid \mathtt{L}) = 7k - 1$ bits of uncertainty remaining:

$$\mathtt{L := H \;\&\; 0}^{7k-1}\mathtt{1}^{k+1}$$

Intuitively, this program is more secure: the adversary can guess H with probability $1/2^{7k-1}$. Shannon entropy, however, says that this program leaks more information (about 0.831 bits' worth).

To obtain a measure that gives better security guarantees about the probability of obtaining H in a single guess, Smith defines the *vulnerability* $V(X)$ of a random variable X to guessing:

$$V(X) = \max_{x \in \mathcal{X}} \Pr[X = x].$$

This is related to $H_\infty(X)$, the *min-entropy* of X:

$$H_\infty(X) = -\lg V(X). \tag{1}$$

Shannon entropy and min-entropy are instances of *Rényi entropy*. The Rényi entropy of order $\alpha \geq 0$ is

$$H_\alpha(X) = \frac{1}{1 - \alpha} \log \sum_{x \in \mathcal{X}} \Pr[X = x]^\alpha.$$

Shannon entropy is obtained in the limit as $\alpha \to 1$; min-entropy is obtained as $\alpha \to \infty$.

Smith defines *conditional vulnerability* analogously to conditional entropy, and *conditional min-entropy* analogously to (1) above:

$$\begin{aligned} V(X \mid Y) &= \sum_{y \in \mathcal{Y}} \Pr[Y = y] V(X \mid Y = y) \\ H_\infty(X \mid Y) &= -\lg V(X \mid Y). \end{aligned}$$

Smith then proposes defining initial uncertainty as $H_\infty(\mathtt{H})$ and remaining uncertainty as $H_\infty(\mathtt{H} \mid \mathtt{L})$, so the information leaked is the difference: $H_\infty(\mathtt{H}) - H_\infty(\mathtt{H} \mid \mathtt{L})$.

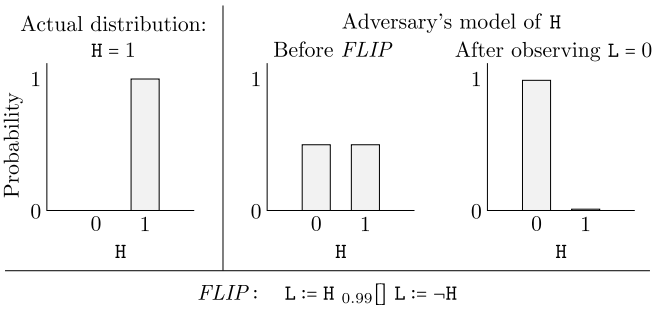

Figure 1: The adversary's accuracy can decrease.

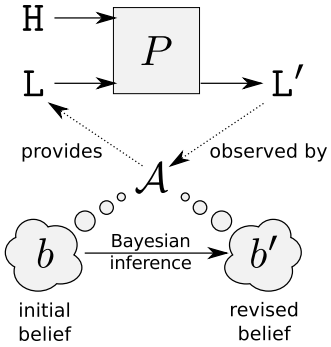

Figure 2: System model of [4]. The attacker uses Bayesian inference to revise his belief, based on his initial belief, the program input (which he provides), the program output, and the program text.

4 Beliefs

Michael Clarkson et al. [4] contend that uncertainty is not the right thing to measure. While the attacker might be absolutely certain about the value secret, he might also be absolutely wrong. Indeed, with information-theoretic measures of uncertainty, leakage can only monotonically increase with successive program runs, whereas an adversary might become more wrong about the secret.

For example, consider the program *FLIP* shown at the bottom of Figure 1. Most of the time, it sets L to H. But with probability 0.01, *FLIP* sets L to be ¬H. Figure 1 shows that when this happens, the attacker can become more wrong about H. Suppose H is 1, and the attacker believes H is equally likely to be 1 or 0. If the attacker observes L = 0 after running *FLIP*, then he will believe H = 1 with probability 0.99, resulting in a probability distribution that is further from the actual distribution than what he started with.

Because the attacker's distribution is therefore subjective, it should be treated as a *belief*. Rather than measuring changes in the uncertainty of the attacker, Clarkson et al. measures changes in the *accuracy* of the attacker's belief.

The system model used is shown in Figure 2. The attacker $\mathcal{A}$ starts with an *initial belief* b, which is a probability distribution over the set of states $\mathcal{S}$ (i.e., the support of H). He supplies low input L, runs the program P, and observes low output L′. The attacker obtains a *revised belief* b' via Bayesian inference, using all that is available to him (including the program text): b, L, L′, and P. Successive runs can be made, resulting in further revised beliefs.

The attacker's accuracy is the *distance* from his belief b to the true distribution $\dot{\mathtt{H}}$, as measured by *relative entropy*[1] (also known as the Kullback-Leibler divergence):

$$D(b \twoheadrightarrow \dot{\mathtt{H}}) = \sum_{\sigma \in \mathcal{S}} \dot{\mathtt{H}}(\sigma) \log \frac{\dot{\mathtt{H}}(\sigma)}{b(\sigma)}.$$

(Here, $\dot{\mathtt{H}}$ is the probability distribution that maps the true value of H to 1.) Information flow is quantified by the change in the attacker's accuracy:

$$\mathcal{Q}_{\dot{\mathtt{H}}}(b \twoheadrightarrow b') = D(b' \twoheadrightarrow \dot{\mathtt{H}}) - D(b \twoheadrightarrow \dot{\mathtt{H}}).$$

One nice property of this definition is its compositionality. If a series of program runs results in a series of belief revisions $b_0, b_1, \ldots, b_n$, the leakage of the entire series of runs is the sum of the leakages from the individual runs:

$$\mathcal{Q}_{\dot{\mathtt{H}}}(b_0 \twoheadrightarrow b_n) = \sum_{i=1}^{n} \mathcal{Q}_{\dot{\mathtt{H}}}(b_{i-1} \twoheadrightarrow b_i).$$

[1] While relative entropy does not satisfy the definition of a formal metric, this does not seem to matter.

5 Discussion

The discussion split into three questions.

- There seem to be lots of proposed models of quantitative information flow, all of which have nontrivial differences. Is there a taxonomy of such models? Is there a list of desiderata for them, or a list of theorems that should be provable about a model of quantitative information flow in order for it to qualify as such? The answer seems to be no, but that would be pretty nice. Maybe some theorem that relates the leakage measured by the model to some notion formulated in the language of differential privacy?
- Is there a connection between these leakage measures and differential privacy? Given that much of the privacy community seems to have settled on differential privacy as the right notion, this seems like an important question to address. One approach to this is [2].
- There are often natural coding interpretations of information-theoretic quantities which are enlightening to think about when using an entropic measure in the wild. How can we interpret the use of relative entropy in [4]? Since relative entropy measures efficiency losses when coding a message according to the wrong distribution, one proposal is to interpret it as follows: suppose we have an inside attacker who learns the high-security output but believes it to be distributed differently than it really is, and suppose that this attacker must exfiltrate the data on some bounded communication channel. Then the relative entropy between the actual distribution of the high data and the attacker's guess at the distribution should somehow measure how successfully the attacker can exfiltrate the data via the bounded channel.

References

[1] Mário S. Alvim, Andre Scedrov, and Fred B. Schneider. Not all bits are equal: Incorporating worth into information-flow measures. Technical report, Cornell University, Ithaca, NY, USA, April 2013. Available from `http://hdl.handle.net/1813/33124`.

[2] Gilles Barthe and Boris Kopf. Information-theoretic bounds for differentially private mechanisms. In *Proc. 24th IEEE Computer Security Foundations Symposium (CSF 2011)*, pages 191–204, Domaine de l'Abbaye des Vaux de Cernay, France, June 2011.

[3] David Clark, Sebastian Hunt, and Pasquale Malacaria. A static analysis for quantifying information flow in a simple imperative language. *Journal of Computer Security*, 15(3):321–371, August 2007. ISSN 0926-227X.

[4] Michael R. Clarkson, Andrew C. Myers, and Fred B Schneider. Quantifying information flow with beliefs. *Journal of Computer Security*, 17(5):655–701, October 2009. ISSN 0926-227X. Available from `http://www.cs.cornell.edu/andru/papers/jbelief.pdf`.

[5] Dorothy E. Denning. *Cryptography and Data Security*. Addison-Wesley, Reading, MA, USA, June 1982. ISBN 978-0201101508.

[6] James W. Gray III. Toward a mathematical foundation for information flow security. In *Proc. IEEE 1991 Symposium on Security and Privacy*, pages 21–35, Oakland, CA, USA, May 1991.

[7] Pasquale Malacaria. Assessing security threats of looping constructs. In *Proc. 34th ACM Symposium on Principles of Programming Languages (POPL)*, pages 225–235, Nice, France, January 2007. Available from `http://www.eecs.qmul.ac.uk/~pm/Papers/boundsWhile.pdf`.

[8] Geoffrey Smith. On the foundations of quantitative information flow. In *Proc. FOSSACS 2009: 12th International Conference on Foundations of Software Science and Computation Structures*, volume 5504 of *LNCS*, pages 288–302, York, UK, March 2009. Available from `http://www.cs.fiu.edu/~smithg/papers/fossacs09.pdf`.