CS276 Scribe Notes

Hakim Weatherspon <hweather@cs.berkeley.edu>

February 11, 2004

1 Semantic Security

A different way to think about security for an encryption scheme is in terms of knowledge; that is, what kinds of information can an attacker deduce. A reasonable way to define security is that an attacker cannot learn anything from the ciphertext; that is, giving the attacker the ciphertext does not give the attacker any information. One way to formalize this notion of security is that whatever the attacker knew before having the ciphertext she knows no more after obtaining the ciphertext. That is, whatever the attacker could compute from the ciphertext the attacker could compute without ciphertext, then attacker learns nothing new by seeing cipertext.

To formally define semantic security, we first define a simulator S that states whatever an attacker A can compute on the ciphertext the simulator S can also compute given nothing. The idea is that any value the attacker could have computed with the ciphertext, the simulator can come up with another way to compute the same value without the ciphertext.

Definition. Semantic security

$\forall$ attackers A (given ciphertext)
$\forall$ distributions $\mathcal{X}$ of equal length messages from message space $\mathcal{M}$
$\forall$ interesting information $f : \mathcal{M} \to \{0,1\}^{\phi}$
$\forall$ partial information $h : \mathcal{M} \to \{0,1\}^{\phi}$

$\exists$ a simulator S (not given ciphertext) where

$$\Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(m), h(m))] - \Pr_{m \leftarrow \mathcal{X}}[S^{E_k}(h(m)) = f(m)] \leq \epsilon$$

The addition of an oracle E_k allows the above definition of semantic security to handle chosen plaintext attacks.

In summary, the attacker is an algorithm. The attacker also specifies the distribution on messages, interesting information, and partial information it would like to predict that would make its job easier. We allow the adversary every leeway. If the adversary does ϵ better than the simulator at predicting the interesting information, then the attacker has broken the encryption. Semantic security seems like a pretty strong statement about encryption

Resource bounds A and S:

- Asymptotic setting insists that A be polynomially bounded and that for all A there exists a polynomially bounded simulator S (may be a different polynomial bound) such that the difference is at most $\epsilon = \frac{1}{O(P)}$ for any polynomial time algorithm P.

- Concrete setting is more difficult to describe. Talk about t, t', ϵ-security where A is restricted to run in time at most t and S is restricted to run in time at most t'. Think about A running in something feasible and S is feasible but may take more time.

1.1 Prove semantic security is as secure as weak-find-then-guess security

Theorem. Semantic security $\iff$ weak-find-then-guess security

$\Rightarrow$ is the easy direction since semantic security seems to be such a strong notion of security.

Proof. ϵ-semantic security $\Rightarrow$ 2ϵ-weak-find-then-guess security

Given A, m_0, m_1
Let $\mathcal{X}$ be uniform on $\{m_0, m_1\}$
interesting information $f(m_b) = b$ (b indicates which msg we are looking at)
side information $h(m_b) = \epsilon$. (No side or partial information)

How good of job can a simulator do at predicting which of the two messages were drawn given no side information? That is, we draw a msg m_0 or m_1 from the space, run the simulator given no information about msg m_b and ask it to predict which msg was drawn. The simulator has no hope because it has no information about m so the best the simulator can do is probability $\frac{1}{2}$. That is, the best that a simulator can do is just flipping a coin.

Formally,

$$\begin{aligned}
\Pr_{m_0 \leftarrow \mathcal{X}}[A^{E_k}(E_k(m_0)) = 1] - \Pr_{m_1 \leftarrow \mathcal{X}}[A^{E_k}(E_k(m_1)) = 1] \quad &\leq 2(\Pr_{b \leftarrow \{0,1\}}[A^{E_k}(E_k(m_b)) = b] - \tfrac{1}{2}) \\
&\quad (b \text{ is } A\text{'s guess for which world it is in}) \\
&= 2(\Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(m)) = f(m)] - \\
&\qquad \Pr_{m \leftarrow \mathcal{X}}[S^{E_k}(E_k(m)) = f(m)]) \qquad \forall S \\
&\leq 2\epsilon
\end{aligned}$$

□

$\Leftarrow$ is the harder direction to prove.

Proof. ϵ-weak-find-then-guess security $\Rightarrow$ ϵ-semantic security

Suppose we have an algorithm that is ϵ-weak-find-then-guess secure, we need to show that it is semantically secure. To show that the algorithm is semantically secure means that we need to create a simulator S for all A attackers where S does just as good of a job as A.

Key idea is to construct a simulator S that uses algorithm A. That is, semantic security says that their output should be equivalent. Figure 1 gives an illustration.

Formally, if $A, \mathcal{X}, h, f$ are any semantic security attacker, define $S^{E_k}(p) = (A^{E_k}(0), p)$ (p is the partial information and 0 is the all 0's $\in \mathcal{M}$ but not necessarily contained in distribution $\mathcal{X}$).

Claim/intuition: attacker will not be able to distinguish 0 from another message drawn from distribution $\mathcal{X}$. If this claim is proved, then that would mean that that simulator S learned what attacker A learned. As a result, we need to prove the claim. The claim is going to follow from weak-find-then-guess security. Weak-find-then-guess security states that the difference between encryption of 0's and encryption of any other information cannot be distinguished.

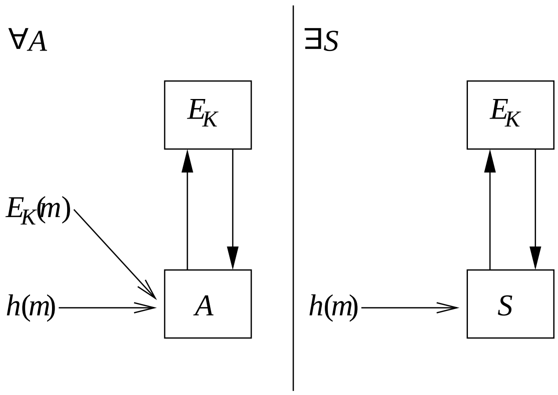

Figure 1: Illustration that simulator S performs as well as all attackers A.

Formally,

$$\begin{aligned}
&= \Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(m), h(m))] - \Pr_{m \leftarrow \mathcal{X}}[S^{E_k}(h(m)) = f(m)] \\
&= \Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(m), h(m))] - \Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(0), h(m)) = f(m)]
\end{aligned}$$

(If it can distinguish 0 from random, can distinguish 0 from concrete.)

$$= \sum_{i \in \mathcal{X}} p_x(i) \cdot \{ \Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(m), h(m))] - \Pr_{m \leftarrow \mathcal{X}}[A^{E_k}(E_k(0), h(m)) = f(m)] \}$$

$p_x(i)$ is a weighted argument that corresponds to the probability of choosing outcome i, the sum of which is less than or equal to ϵ. So, it is enough to show $\{\cdots\} \leq \epsilon, \forall i$. Relationship follows from security from find-then-guess attack.

Define $B^{E_k}(c) = \begin{cases} 1 & \text{if } A^{E_k}(c, h(i)) = f(i) \\ 0 & \text{otherwise} \end{cases}$

Note, choice of B depends on i. Assume fixed i.

By find-then-guess security

$$\begin{aligned}
\{\cdots\} &= \Pr_{m \leftarrow \mathcal{X}}[B^{E_k}(E_k(i)) = 1] - \Pr_{m \leftarrow \mathcal{X}}[B^{E_k}(E_k(0)) = 1] \\
&\leq \epsilon
\end{aligned}$$

□

1.2 Current State of the world

Figure 2 illustrates the current state of the world. Notice that it is enough to show an encryption is secure by proving that it is real-or-random or left-or-right secure. The sad state of affairs is that all proofs rely on unproven assumptions (unless the scheme relies on information theoretic results. e.g. we assume AES is secure).

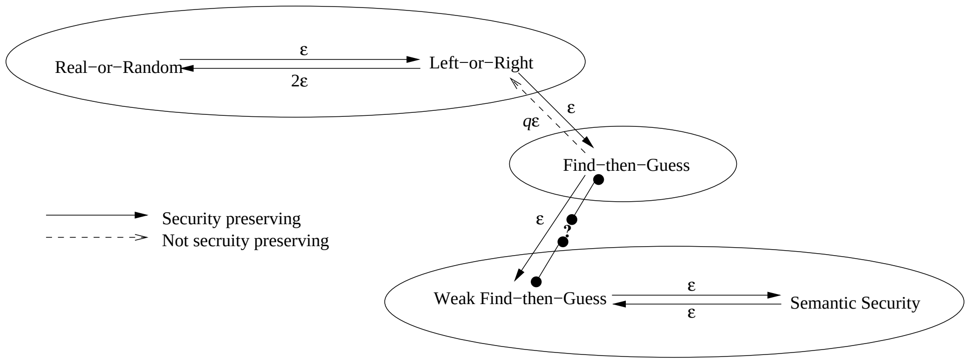

Figure 2: Current state of the world

2 Closing Note: How to reason about system security

Reason about a system

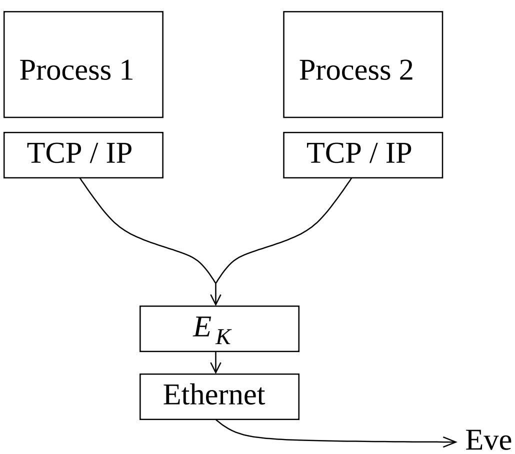

Figure 3: Example of system security analysis

Steps to verify system security

1. Verify E_k is real-or-random secure (attacker knows msg length, number of msgs, time when msg sent).
2. Process 2 + TCP/IP is secure with "angelic" transport.
3. Process 1 is Eve's oracle. Objective: path from Process 1 to Eve does not reveal side information.