CS 127/CSCI E-127: Introduction to Cryptography

Prof. Salil Vadhan Fall 2013

Lecture Notes 5:

Computational Security

Reading.

- Katz-Lindell 3.0–3.2

1 Introduction

- Motivation: Recall *statistical security*: for every $m_0, m_1 \in \mathcal{M}$ and set T of ciphertexts,

$$|\Pr[\mathsf{Enc}_K(m_0) \in T] - \Pr[\mathsf{Enc}_K(m_1) \in T]| \leq \varepsilon.$$

  That is, there is no test T that distinguishes the encryptions of any pair of messages with probability better than ε.

  - Still requires $|\mathcal{K}| \geq (1-\varepsilon) \cdot |\mathcal{M}|$.

- *(Computational) indistinguishability*: only consider tests T defined by "feasible" algorithms $\mathcal{A}$, i.e. replace the event "$\mathsf{Enc}_K(m) \in T$" with "$\mathcal{A}(\mathsf{Enc}_K(m)) = 1$".

- First Goal: Construct computationally secure encryption schemes that go beyond the Shannon barrier (i.e. have $|\mathcal{K}| \ll |\mathcal{M}|$.

  - Still restricted to "one use" and passive adversary.

- Later: Model and achieve security for multiple messages and active adversaries.

2 Concrete formalization

- feasible adversary = time $\leq t$ on specific computational model (e.g. $t = 2^{100}$ cycles on a Pentium D) using a program of size $\leq t$.

- $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ should all run in time $\ll t$.

Definition 1 (indistinguishable encryptions (concrete version)) *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme over* $\mathcal{M}$ *where all messages in* $\mathcal{M}$ *have the same length.* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* (t, ε)-indistinguishable encryptions (against an eavesdropper) *if for every probabilistic algorithm* $\mathcal{A}$ *running in time* t *and for all* $m_0, m_1 \in \mathcal{M}$,

$$|\Pr[\mathcal{A}(\mathsf{Enc}_K(m_0)) = 1] - \Pr[\mathcal{A}(\mathsf{Enc}_K(m_1)) = 1]| \leq \varepsilon.$$

where the probabilities above are taken over $K \xleftarrow{R} \mathsf{Gen}$*, the coin tosses of* Enc_K*, and the coin tosses of* $\mathcal{A}$*.*

- Gen doesn't take any input.
- **Q:** if we want $t = 2^{100}$ and $\varepsilon = 2^{-100}$, what is the shortest key length n (among 64, 128, 256, 512, 1024) for which we might *hope* to achieve (t, ε)-indistinguishability (when the message length is $\gg n$)?

3 Asymptotic formalization

- Need a *security parameter* 1^n: n is chosen by the sender and receiver in advance depending on the level of security they want. Often, n corresponds to the key length of the scheme.
- A "feasible" adversary is any probabilistic poly(n)-time ("PPT") adversary $\mathcal{A}$. Our definition of PPT refers to a *uniform* algorithm, with a fixed program size independent of n. (However, in many treatments of cryptography, it is common to model adversaries as *nonuniform* algorithms, where there can be a different program for each value of n, and the program can be of size poly(n). The nonuniform model simplifies some definitions and proofs, but we will use a uniform treatment for consistency with Katz–Lindell.)
- Require that $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ all run in polynomial time (i.e. poly(n)). Gen now takes n as input (in unary).
- Main point: $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ run in some fixed polynomial time (e.g. time n^2) but security must hold against adversaries with even larger running time. Thus, as we set n larger and larger (e.g. as technology improves), the scheme takes much less time to use than it does to break.
- The message space can change with the security parameter: $\mathcal{M} = \bigcup_n \mathcal{M}_n$. For example, $\mathcal{M}_n$ can be $\{0,1\}, \{0,1\}^{\ell(n)}, \{0,1\}^*$.
- What should ε be? A function $\varepsilon : \mathbb{N} \to [0,1]$ is *negligible* if for every c, there exists n_0 s.t. $\varepsilon(n) < 1/n^c$ for all $n > n_0$.
- We have the adversary choose the messages m_0, m_1 (rather than requiring indistinguishability for all pairs $m_0, m_1 \in \mathcal{M}_n$). This issue does not come up in the concrete formalization (or a nonuniform model of security), because we allow the adversary to have a program size that is as large as its running time, so it can have any pair of messages hardwired into its code.
- Similarly to PS2, instead of having the adversary distinguish an encryption of m_0 from an encryption of m_1, it is equivalent to give the adversary an encryption of m_b for a randomly chosen $b \xleftarrow{R} \{0,1\}$, and require that the adversary cannot guess b with probability nonnegligibly better than $1/2$.

Definition 2 (indistinguishable encryptions (asymptotic version)) *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme over* $\mathcal{M} = \bigcup_n \mathcal{M}_n$ *where all messages in* $\mathcal{M}_n$ *have the same length.* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* (computationally) indistinguishable encryptions (against an eavesdropper) *if for every PPT* $\mathcal{A}$*, there is a negligible function* ε *such that the probability that* $\mathcal{A}(1^n)$ *succeeds in the adversarial indistinguishability game on security parameter* n *is at most* $(1 + \varepsilon(n))/2$.

- To handle varying message lengths (e.g. $\mathcal{M}_n = \{0,1\}^*$): restrict the adversary to choose a pair (m_0, m_1) with $|m_0| = |m_1|$.

- A hybrid between asymptotic and concrete security is to have the adversary's time bound and success probability parameterized by the security parameter n, i.e. consider $(t(n), \varepsilon(n))$-indistinguishable encryptions, but be more precise about the bounds. For example, we could seek $(2^{n/5}, 2^{-n/5})$-indistinguishable encryptions, which is a lot stronger than just requiring security against PPT algorithms and negligible success probability. The asymptotic framework above amounts to requiring (n^c, n^{-c})-indistinguishable encryptions for every constant c and all sufficiently large n. For cryptographic algorithms based on the hardness of factoring n-bit numbers, at best we could hope for something like $(2^{n^{1/3}}, 2^{-n^{1/3}})$-indistinguishable encryptions, due to the state of the art in factorization algorithms.

4 Examples of Insecure Schemes

- Shift cipher: $\mathsf{Gen}(1^n)$ outputs a uniformly random $k \xleftarrow{R} \{0, \ldots, 2^n - 1\}$, $\mathcal{M}_n = \{0, \ldots, 2^n - 1\}^*$, $\mathsf{Enc}_k(m_1, \ldots, m_t) = (m_1 + k \bmod 2^n, \ldots, m_t + k \bmod 2^n)$.

- Biased one-time pad: $\mathsf{Gen}(1^n)$: for $i = \{1, \ldots, n\}$, set $k_i = \{1$ with pr. $.49; 0$ with pr. $.51\}$. Output $k = k_1 \ldots k_n$. $\mathcal{M} = \{0,1\}^n$, $\mathsf{Enc}_k(m) = m \oplus k$.

5 Semantic Security

Definition 3 *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme over* $\mathcal{M} = \bigcup_n \mathcal{M}_n$. *An encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies* partial semantic security *if for every PPT* $\mathcal{A}$, *there is a PPT* $\mathcal{A}'$ *such that for every (*poly(n)*-time samplable*[1]*) distribution* M_n *on* $\mathcal{M}_n$ *and every* poly(n)*-time computable function* $f : \mathcal{M}_n \to \{0,1\}^*$, *we have*

$$\begin{aligned} \Pr\left[\mathcal{A}(1^n, \mathsf{Enc}_K(M_n)) = f(M_n)\right] &\leq \Pr\left[\mathcal{A}'(1^n, |M_n|) = f(M_n)\right] + \mathrm{neg}(n) \\ &\leq \max_v \{\Pr\left[f(M_n) = v\right]\} + \mathrm{neg}(n), \end{aligned}$$

where the probabilities are taken over M_n, $K \xleftarrow{R} \mathsf{Gen}(1^n)$, *and the coin tosses of* Enc *and* $\mathcal{A}$. *(As usual, the two occurrences of* M_n *in the probabilities refer to the same instantiation of this random variable.)*

- The function f captures the information about the message that the adversary is trying to compute.
- Examples:
  - $f(m) = m$: recovering entire plaintext.

[1] A distribution M_n is poly(n)-time samplable if there is a probabilistic polynomial-time algorithm S such that for every n, the output distribution of $S(1^n)$ is precisely M_n.

- $f(m) = m_1$: recovering first bit.

- Semantic security says that the best an adversary can compute f after seeing the ciphertext is essentially the same as if it were only given the length of the ciphertext.

- We call it "partial" semantic security, because the full definition of semantic security allows for giving the adversary partial information $h(M_n)$ about the message and allows for the adversary to choose the distribution M_n and the functions f and h (the definition of Katz–Lindell allows the former but not the latter). The full definition of semantic security is in fact equivalent to the encryption scheme having indistinguishable encryptions.

- The algorithm $\mathcal{A}'$ is often referred to as a *simulator* — it simulates what $\mathcal{A}$ learns about the message, but while being given less information (just the length of the message).

Theorem 4 *If an encryption scheme has indistinguishable encryptions, then it satisfies partial semantic security.*

Hence if we assume (or prove) indistinguishability (i.e. distinguishing encryptions is hard), then we can deduce semantic security (i.e. computing information about the message is hard).

Proof: We'll only prove that indistinguishable encryptions implies semantic security.

■

Note *reducibility argument*: we show how to convert a PPT adversary $\mathcal{A}$ violating partial semantic security into a PPT adversary $\mathcal{B}$ violating indistinguishability. Similar in spirit to the reductions done in **NP**-completeness (but more delicate to analyze, due to the average-case nature of security definitions).