CS 7880 Graduate Cryptography September 18, 2017

Lecture 4: Computationally secure cryptography

Lecturer: Daniel Wichs *Scribe: Lucianna Kiffer*

1 Topic Covered

- ε-security
- Computationally secure cryptography (asymptotic security)
- Computational indistinguishability
- Interactive model of security

2 ε-security

Recall that the statistical distance between two distributions is defined as follows:

$$\begin{aligned}
\mathrm{SD}(X,Y) &= \frac{1}{2}\sum_{z} |\Pr[X = Z] = \Pr[Y = Z]| \\
&= \max_{D \subseteq Z} |\Pr[x \in D] - \Pr[y \in D]| \\
&= \max_{D: Z \to \{0,1\}} |\Pr[D(X) = 1] - \Pr[D(Y) = 1]|
\end{aligned}$$

The intuition here is that if two distributions have tiny statistical distance then no test D can distinguish them from each other. Therefore, they are "essentially identical". This gives us a way to relax the security requirements for encryption. Recall that perfect security required that the distributions $\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m1)$ are indentical. We now relax this by only insisting that they are "statistically close".

Definition 1 An encryption scheme has ε-security if $\forall m_0, m_1 \in \mathcal{M}$:

$$\mathsf{SD}(\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m1)) \leq \varepsilon$$

where K is a uniformly random key in the set $\mathcal{K}$. $\diamondsuit$

Theorem 1 *In any ε-secure encryption scheme we have that* $\varepsilon \geq 1 - \frac{|\mathcal{K}|}{|\mathcal{M}|}$

Proof: let $\varepsilon = \max_{m_0,m_1} \mathsf{SD}(\mathsf{Enc}(K, m_0), \mathsf{Enc}(K, m_1))$

$$\geq \max_{m_0,m_1} \Pr[D_{m_0}(\mathsf{Enc}(K, m_0)) = 1] - \Pr[D_{m_0}(\mathsf{Enc}(K, m_1)) = 1]$$

where D_{m_0} is a distinguisher which is defined as $D_{m_0}(c) = 1$ if $\exists k \in \mathcal{K}$ such that $\mathsf{Dec}(k, c) = m_0$.
Taking the expectation over random choices for m_0, m_1, we get that the above is

$$\geq \Pr[D_{M_0}(\mathsf{Enc}(K, M_0)) = 1] - \Pr[D_{M_0}(\mathsf{Enc}(K, M_1)) = 1] \geq 1 - \frac{|\mathcal{K}|}{|\mathcal{M}|}$$

where M_0 and M_1 are the uniform distributions over $\mathcal{M}$. It's easy to see that $\Pr[D_{M_0}(\mathsf{Enc}(K, M_0)) = 1]$. Moreover $\Pr[D_{M_0}(\mathsf{Enc}(K, M_1)) = 1] \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}$ since for any ciphertext c in the support of $\mathsf{Enc}(K, M_1)$, there are at most $|\mathcal{K}|$ values of m_0 for which $D_{m_0}(c)$ would output 1 and the probability that the random variable M_0 takes on some such value is therefore bounded by $\frac{|\mathcal{K}|}{|\mathcal{M}|}$. □

Thus we are still bound by Shannon's impossibility result, saying that if the key is even 1-bit shorter than the message, $\varepsilon \geq \frac{1}{2}$. BUT, have not shown that the distinguisher is efficient. Thus, even though we can't get ε-security, maybe we can get somewhere if we restrict the distinguisher to be efficient.

3 Computationally Secure Cryptography

The goal is to define encryption schemes that a computationally constrained adversary cannot distinguish with some bounded probability (i.e. probability less than the chance of an asteroid hitting the earth).

One idea is to define (t, ε)-security where for any adversary whose run time is bounded by t, we would require that the success probability $\leq \varepsilon$. We won't use this definition because it gets quite cumbersome. The exact run-time depends on the particular model of computation (Turing Machine vs. JAVA programs) and also it's not clear what choices of t, ε are the right ones.

3.1 Asymptotic Security

Instead we define asymptotic security as follows:

- All schemes are parameterized by a "security parameter" n. As n gets bigger the scheme should asymptotically become more secure. For example, the scheme can set the size of the secret key to depend on n.
- Adversaries are PPT (probabilistic polynomial time), i.e. run in time polynomial in n and their other inputs. Adversaries have access to randomness (uniformly random bits). For a PPT algorithm A we write $A(x)$ to denote the random variable for A's output and $A(x; r)$ is the execution of a randomized algorithm for a particular input x and a particular randomness r.
- the adversary's "success probability" is negligible:

$$\varepsilon = \mathsf{negl}(n) \text{ if } \varepsilon(n) = \frac{1}{n^{\omega(1)}} \iff \forall c > 0,\ \varepsilon(n) = \frac{1}{\Omega(n^c)} \iff \forall c > 0\ \exists n_0\ s.t.\ \forall n > n_0,\ \varepsilon(n) \leq \frac{1}{n^c}$$

Examples of non-negligible functions include $\frac{1}{2}$, $\frac{1}{\log n}$ and $\frac{1}{n^2}$. We are looking for a negligible function like $\frac{1}{2^n}$.

Note that

$$\varepsilon(n) = \begin{cases} \frac{1}{2} \; n \text{ is odd} \\ \frac{1}{2^n} \; n \text{ is even} \end{cases}$$

is not negligible.

A useful property of negligible functions is that multiplying a polynomial function times a negligible function results in a negligible function.

Asymptotic security gives a sharp threshold for security without messy parameters. Almost all of the results we have can also be nicely translated to (t, ε)-security if desired.

We make a brief comment on uniform v.s. non-uniform models of computation: uniform computation adversaries have one algorithm that gets n as an input while non-uniform models of adversaries allow for different algorithms for each different n. For this class we will usually think of the adversary as uniform but might mention some cases where this distinction makes a difference.

4 Computational Indistinguishability

Consider the sequence of variables $X = \{X_n\}_n \in \mathbb{N}$ and $Y = \{Y_n\}_n \in \mathbb{N}$ (one for each security parameter). We define computational indistinguishability between the sequences as follows:

Definition 2 X, Y are computationally indistinguishable if $\forall$PPT distinguishers D, $\exists \varepsilon(n) = \mathsf{negl}(n)$ such that

$$|\Pr[D(X_n) = 1] - \Pr[D(Y_n) = 1]| \le \varepsilon(n)$$

denoted by $X \approx Y$. ♢

Theorem 2 *If $X \approx Y$ and $f : \{0,1\}^* \to \{0,1\}^*$ is a* PPT *function, then $f(X) \approx f(Y)$.*

Informal proof: If we can distinguish between $f(X)$ and $f(Y)$, then we could run f on X and Y and distinguish the outputs and it use it to distinguish between X and Y.

Proof: (reduction) Suppose there exists a PPT distinguisher D such that

$$|\Pr[D(f(X_n)) = 1] - \Pr[D(f(Y_n)) = 1]| \neq \mathsf{negl}(n)$$

We can construction a new PPT distinguisher $D' = D \circ f$ and get that

$$\Pr[D'(X_n) = 1] - \Pr[D'(Y_n) = 1]| \neq \mathsf{negl}(n)$$

since a polynomial function times a negligible function is still negligible. □

Theorem 3 *Hybrid Argument. If $X \approx Y$ and $Y \approx Z$, then $X \approx Z$.*

Proof: For any distinguisher D let

$$\begin{aligned}
\varepsilon(n) &= |\Pr[D(X_n)=1] - \Pr[D(Z_n)=1]| \\
&= |\Pr[D(X_n)=1] - \Pr[D(Y_n)=1] + \Pr[D(Y_n)=1] - \Pr[D(Z_n)=1]| \\
&\leq \underbrace{|\Pr[D(X_n)=1] - \Pr[D(Y_n)=1]|}_{\varepsilon_1(n)} + \underbrace{|\Pr[D(Y_n)=1] - \Pr[D(Z_n)=1]|}_{\varepsilon_2(n)}
\end{aligned}$$

Since $X \approx Y$ we know that ε_1 is negligible and since $Y \approx Z$ we know that ε_2 is negligible. Therefore $\varepsilon = \varepsilon_1 + \varepsilon_2$ is negligible which concludes the proof. □

5 Security Game

We usually define security via *games* which are interactive protocols between an *adversary* A trying to break the system and the world running the system (which we call the challenger). For some such Game and an adversary A we define $\mathsf{Game}_A(1^n)$ to denote output of the game; usually this will be the output of the adversary A at the end of the game.

Often we define security via two games $\mathsf{Game}^0, \mathsf{Game}^1$ which represent two possible options for what the world might be doing (e.g., encryptions of two different messages) and we require that the adversary cannot tell them apart. We define this as follows. DEFINITION 3 We say that two games $\mathsf{Game}^0, \mathsf{Game}^1$ are computationally indistinguishable, denoted by $\mathsf{Game}^0 \approx \mathsf{Game}^1$, if

$$\forall \mathsf{PPT} A\ \exists \mathsf{negl}\ \varepsilon() \text{ s.t. } |\Pr[\mathsf{Game}^0_A(1^n)=1] - \Pr[\mathsf{Game}^1_A(1^n)=1]| = \varepsilon(n)$$

♢

Computational indisitnguishability of games is analogous to computational indistinguishability of random variables. In particular, the same hybrid argument works for games.

Theorem 4 *If* $\mathsf{Game}^0 \approx \mathsf{Game}^1$ *and* $\mathsf{Game}^1 \approx \mathsf{Game}^2$ *then* $\mathsf{Game}^0 \approx \mathsf{Game}^2$

5.1 Computationally Secure Encryption

We consider an encryption scheme with key space $\mathcal{K}_n = \{0,1\}^n$ and message space $\mathcal{M}_n = \{0,1\}^{\ell(n)}$ where ℓ is some polynomial. The ciphertext space is $\mathcal{C}_n$. The scheme consists of algorithms:

$$\mathsf{Enc} : \mathcal{K}_n \times \mathcal{M}_n \to \mathcal{C}_n$$

$$\mathsf{Dec} : \mathcal{K}_n \times \mathcal{C}_n \to \mathcal{M}_n$$

we define the following game for proving computational security of encryption schemes: DEFINITION 4 One-time security game: OneSec^b where $b \in \{0,1\}$

- Adversary chooses $m_0, m_1 \in \mathcal{M}_n$ and sends it to the Challenger
- The Challenger samples a uniformly random key from the key space ($k \leftarrow \mathcal{K}_n$), and then sends the Adversary an encryption $c = \mathsf{Enc}(k, m_b)$ of the message m_b.

- The Adversary outputs some value b'.

$\diamondsuit$

DEFINITION 5 An encryption scheme is one-time computationally secure if

$$\mathsf{OneSec}^0 \approx \mathsf{OneSec}^1.$$

$\diamondsuit$