CSA E0 235: Cryptography	January 25, 2018
Scribe for Lecture 5	
Instructor: Arpita Patra	*Scribe: Pranshu Gaba*

1 Introduction

Last time, we looked at computational security, and made the definitions of probabilistic polynomial-time (PPT)/negligible function precise in terms of security parameter n. We looked at the semantic and the indistinguishability based definitions of security and showed that they are equivalent. We also looked at pseudorandomness and pseudorandom generators (PRGs).

Now we will continue the discussion on PRGs. We will make use of reduction-based proofs to prove that if PRGs exist, then a construction is secure according to ind-secure definition. We will find a construction for COA-secure symmetric key encryption, and finally, discuss the short comings of the current construction/definition, and try to come up with a better 1definition.

2 Pseudorandom Generators (PRGs) [1] [2]

Let $l(n)$ be a polynomial such that $l(n) > n$ for all n. Let U be the uniform distribution over $\{0,1\}^{l(n)}$. Let $G\colon \{0,1\}^n \to \{0,1\}^{l(n)}$ be a length-expanding function with probability distribution over $\{G(s) : s \in_R \{0,1\}^n\}$.

To determine whether G is a PRG, a game is played. A PPT distinguisher D asks the challenger for a string of length $l(n)$. The challenger flips a coin b. If $b = 0$, then the challenger picks a string from U. If $b = 1$, then the challenger picks a string from G. The distinguisher must now tell if the string was chosen from U or G. If the distinguisher guesses correctly, then we say $D(y) = 1$, else $D(y) = 0$.

We say that G is a PRG if for every PPT distinguisher D, there is a negligible function negl such that

$$\Big|\Pr_{r \in_R \{0,1\}^{l(n)}}[D(r) = 1] - \Pr_{s \in_R \{0,1\}^n}[D(G(s)) = 1]\Big| \leq \mathsf{negl}(n)$$

That is, the difference in probabilities of the adversary winning the game when the number is chosen from U, and when the number is chosen from G, should be negligible.

2.1 Attempt to construct a PRG

We will now try to construct a PRG. We have $s \in_R \{0,1\}^n$. Consider the function $G\colon \{0,1\}^n \to \{0,1\}^{n+1}$, which is defined to be $G(s) = ss'$, where $s' = \bigoplus_{i=1}^n s_i = s_1 \oplus s_2 \oplus \cdots \oplus s_n$. Note that output of G is one bit longer than s, so expansion factor $l(n)$ is $n + 1$.

Given a string from $y \in \{0,1\}^{n+1}$, the distinguisher must determine if it is generated from U or G, i.e. the distinguisher tries to guess the value of b.

The strategy adopted by a efficient distinguisher D is to output 1 if and only if the last bit of the message is equal to the XOR of all the preceding bits. There are two possible cases:

- If the last bit does not equal the XOR of all the preceding bits, the distinguisher D knows for sure that the message was from U. He guesses b correctly. $\Pr[D(r) = 1] = 1$
- Else if the last bit equals the XOR of all preceding bits, then there are two possible cases again:
  - If y is from G, then D is always correct. $\Pr[D(G(s)) = 1] = 1$, $s \in_R \{0,1\}^n$.
  - Else if y is from U, then D is correct with half probability. $\Pr[D(r)] = 1] = \frac{1}{2}$, $r \in_R \{0,1\}^{n+1}$.

The difference in the probabilities is

$$\big|\Pr[D(r) = 1] - \Pr[D(G(s))] = 1]\big| = 1 - \frac{1}{2}$$

This difference is constant, and not negligible, thus G that we chose is not a PRG. As it turns out, constructing a PRG is a very difficult task.

2.2 PRG can be cracked by an unbounded adversary

Consider the length-doubling PRG, where G takes in a seed of length n, and outputs a string of length $2n$. There are 2^n possible strings of length n, and 2^{2n} strings of length $2n$. If an adversary picks a string from $\{0,1\}^{2n}$ uniformly at random, then the probability that he picks a strings that came from a seed is $\frac{2^n}{2^{2n}} = \frac{1}{2^n}$, which is negligible. Most of the $2n$ length strings do not appear as the output of G. There is a negligible probability that he picks a string from the range of G.

If a PPT adversary is given a string y, then there is negligible probability that he will be able to distinguish correctly whether it is from U or G. This is because he cannot compare y with all the 2^n outputs of $G(s)$ in polynomial time.

However, if we give the adversary unbounded power, then he can list out all those 2^n outputs of $G(s)$. Then given any string y, he can check if it belongs in the range of G. The adversary's strategy would be:

- If y is in not the range of G, the distinguisher outputs 0 always because it was definitely not generated by G.
- Else if y is in the range of G, then the optimal strategy of D is to output 1 always.
  - If y is from G, then D is always correct. $\Pr[D(G(s)) = 1] = 1$, $s \in_R \{0,1\}^n$.
  - Else if y is from U, then D is correct with probability $\frac{2^n}{2^{2n}}$. $\Pr[D(r) = 1] = 2^{-n}$, $r \in_R \{0,1\}^{n+1}$.

When we test the security of this scheme, we see that the distinguisher is almost always able to distinguish correctly.

$$\left|\Pr[D(r) = 1] - \Pr[D(G(s)) = 1]\right| \geq \underbrace{1 - 2^{-n}}_{\text{not negligible}}$$

The difference in probabilities is $1 - 2^{-n}$, which is not negligible, so this scheme can be cracked by an unbounded adversary.

Cryptography is an applied science, and in reality, the adversary does not have unbounded power. It is reasonable to construct a scheme that can be broken by a PPT adversary only with at most negligible probability. However, while implementing this algorithm, it must be made sure that n is large enough so it cannot be broken using brute-force by a PPT adversary in a reasonable time.

In this entire section, we have assumed that PRGs exist. Their existence has neither been proven nor disproven, although it is strongly believed that they exist. However, it has been shown that if one-way functions exist (which are a lower-level primitives than PRGs), then PRGs exist [3]. We also believe that stream ciphers are PRGs, because no good distinguishers for them are known.

2.3 One-way functions

One-way functions are functions that are easy to compute, but almost always difficult to invert. We can play a game $\mathsf{Invert}_{\mathcal{A},f}(n)$ to check if f is a one-way function.

2.3.1 Inverting game

The challenger chooses number a number x from $\{0,1\}^n$ uniformly at random, and computes $y \coloneqq f(x)$. A PPT adversary $\mathcal{A}$ is given y. The adversary now tries to guess x', a pre-image of y.

If $f(x') = y$, then the adversary has won, and the output of the game is 1. Else if $f(x') \neq y$, then the adversary has lost, and the output of the game is 0. Note that $\mathcal{A}$ does not have to find the original x to win the game. It is sufficient for him to guess any pre-image to win the game.

We say that f is one-way if $\mathcal{A}$ wins the game with negligible probability.

2.3.2 Mathematical formulation

Function $f\colon \{0,1\}^* \to \{0,1\}^*$ is a one-way function if the following two conditions hold:

- Easy to compute: for every $x \in \{0,1\}^*$, $f(x)$ can be computed in $\mathsf{poly}(n)$ time.
- Hard to invert: For every PPT algorithm A, there is a negligible function $\mathsf{negl}(n)$ such that

$$\Pr[\mathsf{Invert}_{\mathcal{A},f}(n) = 1] \leq \mathsf{negl}(n) \approx \Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(f(x), 1^n) \in f^{-1}(f(x))] \leq \mathsf{negl}(n)$$

3 Proofs by Reduction

Proofs by reduction are very useful. We can use them to prove statements like

1. If an encryption scheme Π is secure, then another scheme Π' is also secure.
2. If an assumption A holds, then Π is secure.
3. If A_1 holds, then A_2 also holds.
4. If Π is secure, then A holds.

Proof by reduction involves proof by contradiction/contrapositive. We will see how to prove statement 1. The other statements can be proved in a similar fashion.

We want to prove that if Π is secure, then Π' is also secure.

Proof Assume that Π' is not secure, so we have a PPT attacker against Π' that can break it with non-negligible probability $f(n)$. The challenger sends a challenge for Π to PPT attacker against Π. Based on this challenge, the PPT attacker against Π sends a simulated challenge to PPT attacker against Π'. This PPT can break Π' with non-negligible probability $f(n)$.

Based on this break, the PPT against Π comes up with a solution for Π which can break Π with probability $1/p(n)$. The overall probability that the PPT attacker against Π breaks the scheme is at least $f(n)/p(n)$, which is not negligible. However, our statement included the assumption that Π is secure, so this should not be possible. This is a contradiction! Therefore Π' is secure, and no PPT attacker against Π' exists that can break with probability $f(n)$.

■

4 COA-secure SKE

We now construct a scheme Π that is COA-secure using the ind-secure definition. That is, for every PPT adversary $\mathcal{A}$, there is a negligible function negl, such that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n)$$

Let $\mathcal{K} = \{0,1\}^n$, and $\mathcal{M} = \mathcal{C} = \{0,1\}^{l(n)}$.

The Gen algorithm picks a key $k \in_R \mathcal{K}$ uniformly at random.

The Enc algorithm receives a key $k \in \mathcal{K}$, and a message $m \in \mathcal{M}$ and outputs the ciphertext $c := G(k) \oplus m$.

The Dec algorithm receives a key $k \in \mathcal{K}$ and a ciphertext $c \in \mathcal{C}$ and outputs the message $m := G(k) \oplus c$.

Look at $\mathsf{Dec}_k(\mathsf{Enc}_k(m))$. It is equal to m for all $m \in \mathcal{M}$, since $G(k) \oplus G(k) \oplus m = m$. This proves the correctness of this scheme. Using proof by reduction, we will show that if G is a PRG, then the above scheme is COA-secure.

Proof We prove by contradiction. If Π is not secure, then we show that it implies G is not a PRG, which is a contradiction, so Π must be secure.

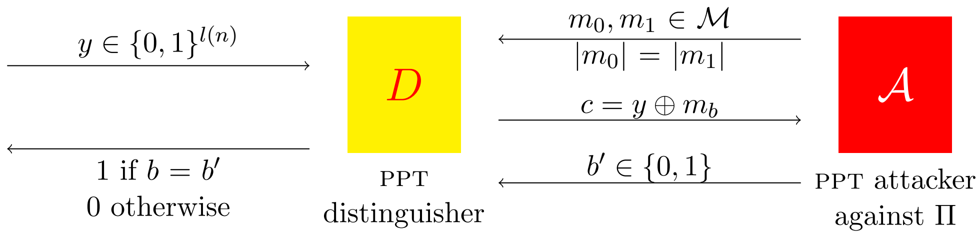
If Π is COA-secure, then for every PPT adversary $\mathcal{A}_i$, there is a $\mathsf{negl}_i(n)$ such that

$$\begin{aligned}\Pr\left[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A}_i,\Pi}(n) = 1\right] &\leq \frac{1}{2} + \mathsf{negl}_i(n) \\ &< \frac{1}{2} + \frac{1}{\mathsf{poly}(n)} \text{ for all } n > N \text{ for every } \mathsf{poly}(n)\end{aligned}$$

So Π is not COA-secure if there exists a PPT adversary $\mathcal{A}_i$ and a polynomial $\mathsf{poly}_i(n)$ such that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A}_i,\Pi}(n) = 1\right] > \frac{1}{2} + \frac{1}{\mathsf{poly}_i(n)} \text{ for infinitely many } n\text{'s}$$

Assume that Π is not secure. Then there exists a PPT adversary $\mathcal{A}$ such that the above inequality holds true for infinitely many n's.



The distinguisher D gets a string of length $l(n)$ from a challenger. He uses adversary $\mathcal{A}$ to help him determine whether y is a random string, or a pseudorandom string. So, D plays $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}$ with $\mathcal{A}$ with D as the challenger.

$\mathcal{A}$ sends two messages, m_0, m_1, of his choice to D. D must use y in the challenge to get more information about y from $\mathcal{A}$. D can do this by flipping a coin b. He sends the ciphertext $c = y \oplus m_b$ to $\mathcal{A}$. The adversary return his guess b' to D.

If y is a random string, then this case is similar to a perfect security. Even if the adversary has unbounded power, he cannot guess the bit with probability more than half. $\Pr[D(y) = 1] = \frac{1}{2}$

If y is a pseudorandom string, then since $\mathcal{A}$ is a very good adversary, he can break it with non-negligible probability. $\Pr[D(G(s)) = 1] > \frac{1}{2} + \frac{1}{\mathsf{poly}(n)}$

The difference in probabilities is not negligible.

$$\left|\Pr[D(y) = 1] - \Pr[D(G(s)) = 1]\right| > \underbrace{\frac{1}{\mathsf{poly}(n)}}_{\text{not negligible}}$$

This means that G is not a PRG. However, since we believe G is a PRG, therefore we also believe that Π is a COA-secure scheme. ■

5 Multiple-message COA Security

We have the overcome one of the limitations of perfect security: the key size can now be smaller than the message size, and the scheme will still be COA-secure. However, we still can't reuse the key. We can show that if a PPT adversary is allowed to play $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}$ twice with the same key, then he can break the scheme with probability 1.

Let us define a new game,

$$\mathsf{PrivK}^{\mathsf{coa-mult}}_{\mathcal{A}_i,\Pi}(n), \qquad \Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}), \mathcal{M}.$$

This is very similar to $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}$, the only difference being that the adversary sends two vectors of his choice, $\vec{M}_0 = (m_{0,1}, m_{0,2}, \ldots, m_{0,t})$ and $\vec{M}_1 = (m_{1,1}, m_{1,2}, \ldots, m_{1,t})$ instead of messages m_0 and m_1. Each vector has polynomially many messages.

A scheme Π is COA-mult secure if for every PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the above game is at most negligibly better than $\frac{1}{2}$.

$$\Pr\left[\mathsf{PrivK}^{\mathsf{coa-mult}}_{\mathcal{A}_i,\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n)$$

The game $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A}_i,\Pi}$ is special case of $\mathsf{PrivK}^{\mathsf{coa-mult}}_{\mathcal{A}_i,\Pi}$, with $|M_0| = |M_1| = 1$. Thus if a scheme Π is COA-mult-secure, then it is also COA-secure. However the converse is not true. We can show that multiple-message security is stronger than single-message security i.e. there exists a scheme which is COA-secure, but not COA-mult-secure.

The adversary can send the following two vectors $M_0 = (\texttt{hello}, \texttt{hello})$ and $M_1 = (\texttt{hello}, \texttt{world})$. The challenger gets a key k, and flips a coin b.

- If $b = 0$, he returns $c_1 \coloneqq \texttt{hello} \oplus k, c_2 \coloneqq \texttt{hello} \oplus k$.
- If $b = 1$, he returns $c_1 \coloneqq \texttt{hello} \oplus k, c_2 \coloneqq \texttt{world} \oplus k$.

It is now very easy, even for a PPT adversary to find b. He returns $b' = 0$ if $c_1 = c_2$ and $b' = 1$ if $c_1 = c_2$. He wins the game with probability 1.

$$\Pr\left[\mathsf{PrivK}^{\mathsf{coa-mult}}_{\mathcal{A}_i,\Pi}(n) = 1\right] = 1$$

We have found a scheme which is COA-secure, but not COA-mult-secure. This happens because the Enc algorithm used was deterministic. Encrypting the same message m twice with the same key twice gives the same cipher text.

Theorem. *If Π is a cipher whose Enc algorithm is a deterministic function of the key and plaintext, then Π cannot have indistinguishable multiple encryptions in the presence of an eavesdropper.*

To overcome this, we must make use of a randomized encryption scheme. This solve the problem of key reuse. The same key can then be used multiple times and the scheme will still be secure.

Next time, we will give more power to the adversary. (chosen plaintext attacks (CPA), which is stronger than both COA and COA-mult). We will construct a scheme which is CPA-secure.

References

[1] Jonathan Katz, Yehuda Lindell, *Introduction to Modern Cryptography*, CRC Press, 2nd edition, 2015.

[2] Arpita Patra, *Cryptography*, lecture notes, Cryptography E0 235, Indian Institute of Science, 2018.

[3] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC 89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, New York, NY, USA, 1989. ACM Press.