CSA E0 235: Cryptography 29-08-2019

Scribe for Lecture 5

Instructor: Arpita Patra *Submitted by: Akash A*

1 Computational Security:

1.1 Drawbacks of perfect security

We saw that even though perfect security is the holy grail of security notions, designing practical schemes based on it is very diffcult. Also, the underlying assumptions for perfect security are too stringent and therefore can be relaxed to match the real world environment. These relaxed assumptions lead to formation of Computational or Cryptographic Security.

1.2 Assumptions and Promises of Computational Security

Computational Security introduces the following relaxations to Perfect Security:

1. Security is only guaranteed against efficient adversaries that are polynomially bounded.
2. Adversaries can potentially succeed in breaking the scheme with some very small probability.

We have also seen that both of these modifications are essential for achieving practical encryption schemes. Computational security also promises to overcome the drawbacks of perfect security like:

- Shorter key for big messages
- Key reusability for multiple messages

Inorder to realize these promises, there is a need of additional concepts and tools. This lead to the concept of Pseudorandomness and Pseudorandom Generators.

2 Pseudorandomness and Pseudorandom Generators

2.1 Pseudorandomness

This is a property of probability distribution imposed on a set of binary strings of length l, say **S**. Consider two probability distributions on the set S:

- **G** : Some probability distribution on S
- **U** : Uniform probability distribution on S

Then, G is called pseudorandom if a string drawn according to G (called pseudorandom) is indistinguishable from a string drawn according to U(called random),to a PPT adversary. Pseudorandomness is a computational relaxation of true randomness.This means that it is enough that the string drawn appears random to the distinguisher but not need to truly random in reality.

2.2 Pseudorandom Generators

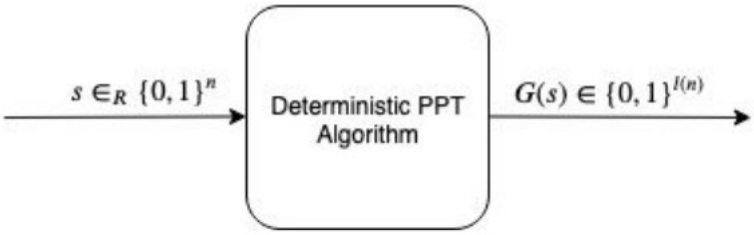

Figure 1: Representation of a PRG

A Pseudorandom Generator G is an efficient,deterministic algorithm that takes a short string s (known as seed) as input and generates a long string $l(s)$,where l is a polynomial defined over s.The requirements of G are as follows:

1. **Expansion :** $\forall\ l(n) > n$
2. **Pseudorandomness :**The output of G looks like a uniform string to any PPT observer

Consider the following PRG security game:

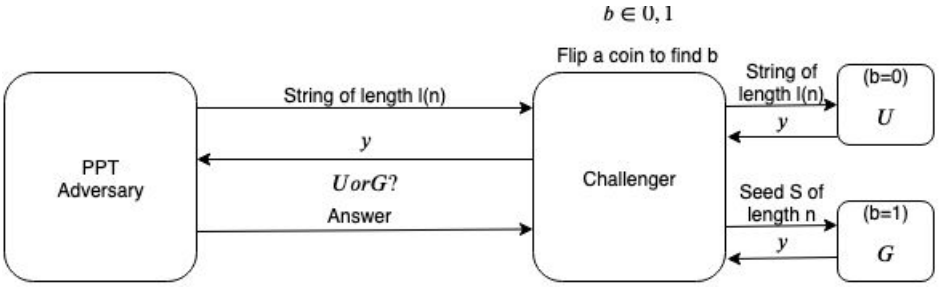

Figure 2: PRG security game

Step 1: The PPT Distinguisher A asks for a string of length $l(n)$ from the challenger C

Step 2: C flips a coin to find out $b \in \{0,1\}$.If $b = 0$,then C picks a random string y of length $l(n)$ from U(uniform distribution).If $b = 1$,then C sends a random seed s of length n to G and obtains string y of length $l(n)$ as output.Finally C forwards the string y to A and asks how y was chosen ,i.e, whether G or U?

Step 3: A responds his answer.
Let $A(U) = 1$ be the event that A answers U and it is correct
Let $A(G(s)) = 1$ be the event that A answers G and it is correct

Definition 1:(Pseudorandom Generator)
G is a Pseudorandom Generator if, $\forall$ PPT adversary A,$\exists$ a negligible function $negl$ such that

$$|Pr(A(U) = 1) - Pr(A(G(s) = 1)| \leq negl$$

2.3 Statistical Methods

Traditionally, statistical methods were used to find out whether G is a pseudo random generator or not. Typically, given a string S, compute a statistic using it and determine if it varies greatly from what a random string would produce. Some examples of such tests are:

Test 1: Count the fraction of 1's in S, let it be k.Ideally $k = 0.5$.S is pseudorandom if $|k - 0.5| < 0.01$

Test 2: Break S into $\frac{l}{2}$ consecutive pairs.
In each pair, count the fractional occurence of 00,01,11,00 strings and store it in K_{ij} where $i, j \in \{0, 1\}$.Ideally $k_{ij} = \frac{l}{2} * \frac{1}{4} = \frac{l}{8}$
S is pseudorandom if $(k_{00} - \frac{l}{8}^2) + (k_{01} - \frac{l}{8}^2) + (k_{10} - \frac{l}{8}^2) + (k_{11} - \frac{l}{8}^2) > 0.05$
This test is called as **chi-square test**

These statistical tests are not good enough because there are many $G's$ that pass these tests that are not PRGs.Also,these tests do not take the mathematical properties of G into consideration.An adversary can attempt to exploit these properties and break the PRG security.

2.4 Next Bit Test

Consider the following game:

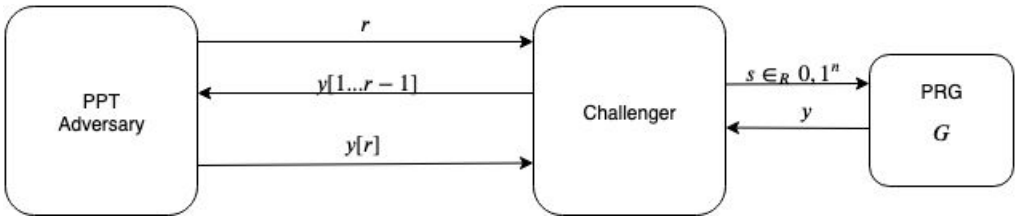

Figure 3: Next bit game

Step 1: The PPT Distinguisher A sends the index r to the challenger C

Step 2: C sends a random seed s of length n to G and obtains string y of length $l(n)$ as output.Then, C forwards the first $r - 1$ bits of string y to A and asks r^{th} bit of y

Step 3: A responds $y[r]$.
Let $A(y[1..r - 1]) = y[r]$ be the event that A guess the r^{th} bit correctly

G is a next bit secure PRG if $\forall$ PPT adversary A,$\exists$ a negligible function $negl$ such that

$$Pr(A(y[1..r - 1]) = y[r]) \leq \frac{1}{2} + negl$$

This means that given the first $r - 1$ bits of the pseudorandom string,any PPT adversary can guess the r^{th} bit with probability $negl$ more than $\frac{1}{2}$, which is almost equal to guessing. It can be shown that G is a secure PRG iff it is next bit secure.

2.5 Insecure PRG(example)

Consider the following PRG $G : \{0,1\}^n \to \{0,1\}^{n+1}$ defined by:

$$G(s) = s|s_{xor} \text{ ,where } s_{xor} = s[0] \oplus s[1] \oplus ... \oplus s[n]$$

To show this is not a secure PRG,we construct a PPT adversary as follows:

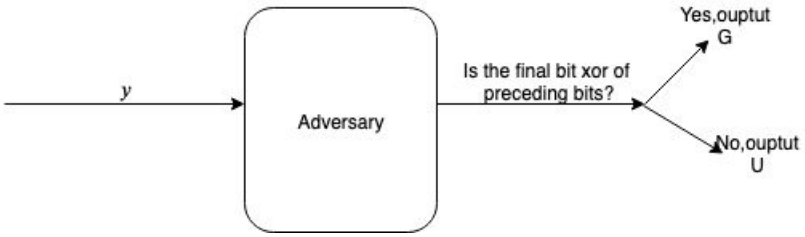

Figure 4: Insecure PRG

The Adversary A,on input of a string y, outputs G(implying y is pseudorandom)if the last bit of y = XOR of all the preceding bits of y.

1. This is true for all string which are output of G,therefore $Pr(A(G(s)) = 1) = 1$
2. If string y is randomly chosen from U,then the last bit will also be random.
   Therefore $Pr(A(U)) = 1) = \frac{1}{2}$

$$|Pr(A(U) = 1) - Pr(A(G(s) = 1)| = \frac{1}{2} \text{,which not negligible}$$

Hence,the given G is not a secure PRG

2.6 Unbounded Adversary

PRG can cracked by any unbounded adversary.This can be attributed to the fact the distribution of PRG is far from uniform.Consider a length doubling PRG:

$$G(s) : \{0,1\}^n \to \{0,1\}^{2n}$$

- Under U,the selection of any of the 2^{2n} strings is possible with probability $\frac{1}{2^{2n}}$
- However under G,the number of possible $2n$ length strings are atmost 2^n (as there are 2^n random seeds and each can produce $2n$ length string via G,and also overlaps possible)
  Therefore Pr[string s of length $2n \in G(s)] \leq \frac{2^n}{2^{2n}} = \frac{1}{2^n}$

This means that a vast majority of $2n$ length strings have zero probability of being an output of G.Hence,an attack can be constructed by an unbounded adversary
Given y by the challenger,A does the following

```
for each m in {0,1}^n
    compute k = G(m)
    if k == y output G
 if none of the k's == y output U
```

1. This is true for all string which are output of G,therefore $Pr(A(G(s)) = 1) = 1$

2. If string y is randomly chosen from U,then A outputs U with probablity utmost 2^{-n}(since,there are atleast 2^n strings that can only be truly random).
   Therefore $Pr(A(U)) = 1) \leq \frac{1}{2^n}$

$$|Pr(A(U) = 1) - Pr(A(G(s) = 1)| \geq 1 - \frac{1}{2^n}\text{,which not negligible}$$

Hence,PRG can be broken by an unbounded adversary.It can also be concluded that n chosen must be sufficiently large so that brute force is not feasible for a PPT distinguisher

2.7 Do PRG's exist?

There is no unconditional proof for existence of pseudorandom generators.They are very difficult to construct,however the crypto community has strong reasons to believe their existence. Firstly, they can be constructed under the rather weak assumption that *one-way functions* exist.Secondly, there are several practical constructions for PRG, called *stream-ciphers*. We will study about these primitives later in the course in detail.For know,we move forward with the assumption that:

PRG's exist

3 IND-secure SKE using PRG

3.1 Description

We construct the following scheme,under the assumption that PRG's exist.
Let the PRG be $G : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$

1. **Gen:**It takes the security parameter n as input and outputs a random string $k \in \{0,1\}^n$.This is the secret key shared between the communicating entities beforehand.

2. **Enc:**It takes the key k and message m as input and outputs the ciphertext $c = m \oplus G(k)$

3. **Dec:**It takes c and k as input and outputs the message $m = c \oplus G(k)$

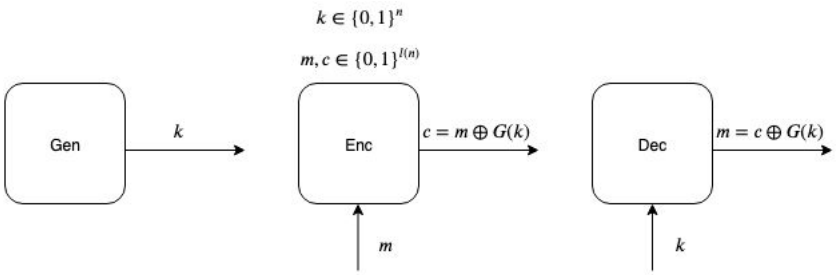

Figure 5: SKE scheme using PRG

Correctness: $Dec_k(Enc_k(m)) = m, \forall m \in M$

3.2 IND-secure game for SKE

Consider the following indistiguishability game $\text{PrivK}^{ind}_{A,\pi}(\text{n})$,where n is the security parameter and A is any PPT adversary and π is an encryption scheme (Gen, Enc, Dec) defined over (K, M, C)

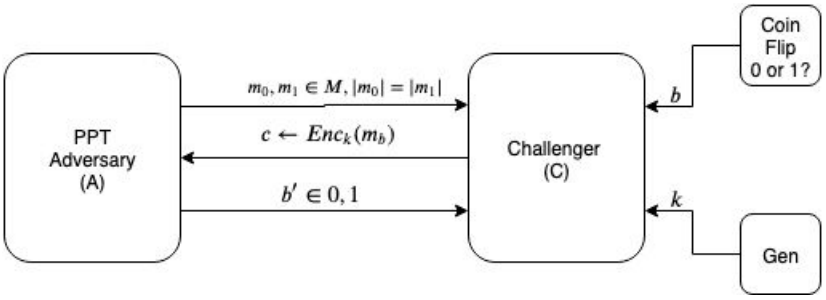

Figure 6: SKE scheme using PRG

Step 1: The PPT Adversary A sends to messages $m_0, m1$ of equal length to the challenger C

Step 2: C flips a coin to find out $b \in \{0, 1\}$ and encrypts m_b and sends it to A

Step 3: A responds his guess about which message is encrypted in b'.
If $b = b'$,then A won and C outputs 1
If $b \neq b'$,then A lost and C outputs 0
Let $\text{PrivK}^{ind}_{A,\pi}(\text{n}) = 1$ be an event in which A won.

π is ind-secure if $\forall$ PPT adversary A,$\exists$ a negligible function $negl$ such that

$$Pr(\text{PrivK}^{ind}_{A,\pi}(\text{n}) = 1)) \leq \frac{1}{2} + negl$$

3.3 Security of PRG based scheme

Theorem: If G is an PRG,then the scheme π defined using G is a ind-secure scheme

Proof : Let us suppose π is not an ind-secure SKE. This implies that there exist a PPT adversary A who can break this scheme with non-negligible probability.Formulating this we get:

$$\exists A, p(n) : Pr(\text{PrivK}^{ind}_{A,\pi}(\text{n}) = 1)) > \frac{1}{2} + \frac{1}{p(n)}\text{,for infinitely many } n$$

Now,lets construct a Distinguisher D for PRG-security game using A as follows:

Step 1: The distinguisher D receives the string y from Challenger C under PRG security game,meaning y can either be truly random or pseudorandom.Also,D receives two messages messages $m_0, m1$ of equal length from the adversary A under IND security game

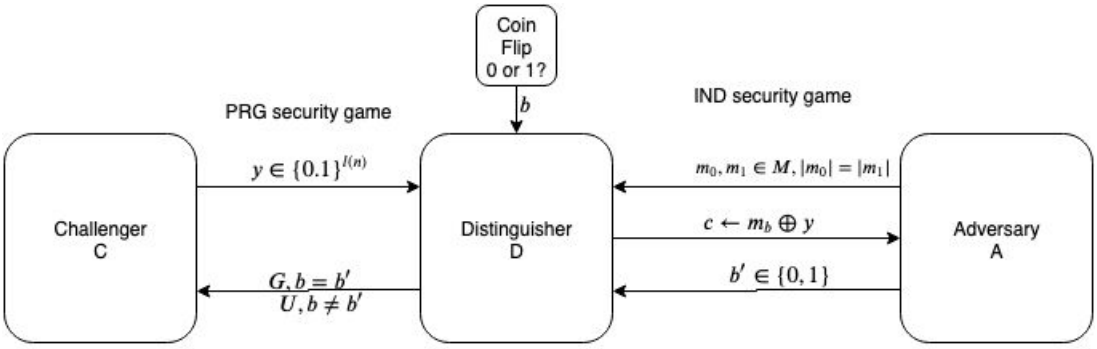

Figure 7: Construction of D using A

Step 2: C flips a coin to find out $b \in \{0,1\}$ and encrypts the message m_b using $c = m_b \oplus y$,treating as y as the encryption key.

Step 3: A responds his guess about which message is encrypted in b'.

Step 4: If $b = b'$,then D sends G to C,meaning y is pseudorandom
If $b \neq b'$,then D outputs U to C,meaning y is truly random

1. When y is pseudo-random,then the probability that D guesses it correctly is same as the probability that A wins the IND-secure game.Therefore,

$$Pr(A(G(s)) = 1) = Pr(\text{PrivK}^{ind}_{A,\pi}(\text{n}) = 1)) > \tfrac{1}{2} + \tfrac{1}{p(n)}, \text{for infinitely many } n$$

2. When y is truly random,the probability that D guess it correctly is no better than guessing.Therefore,

$$Pr(A(U) = 1) = \tfrac{1}{2}$$

$$\text{Thus,} |Pr(A(U) = 1) - Pr(A(G(s)) = 1| > \tfrac{1}{p(n)}, \text{for infinitely many } n$$

Hence the probability of distinguishing between a truly random generator and PRG G is not negligible. This implies that G is not a pseudorandom generator.
Thus π is not IND-secure $\implies$ G is not a PRG.
Since G is a secure PRG,then π is an IND-secure scheme.

Hence,Proved

4 Multi-message schemes

4.1 IND-game for multiple messages

Consider the following indistinguishability game for multi messages $\text{PrivK}^{multi}_{A,\pi}(\text{n})$,where n is the security parameter and A is any PPT adversary and π is an encryption scheme (Gen, Enc, Dec) defined over (K, M, C)

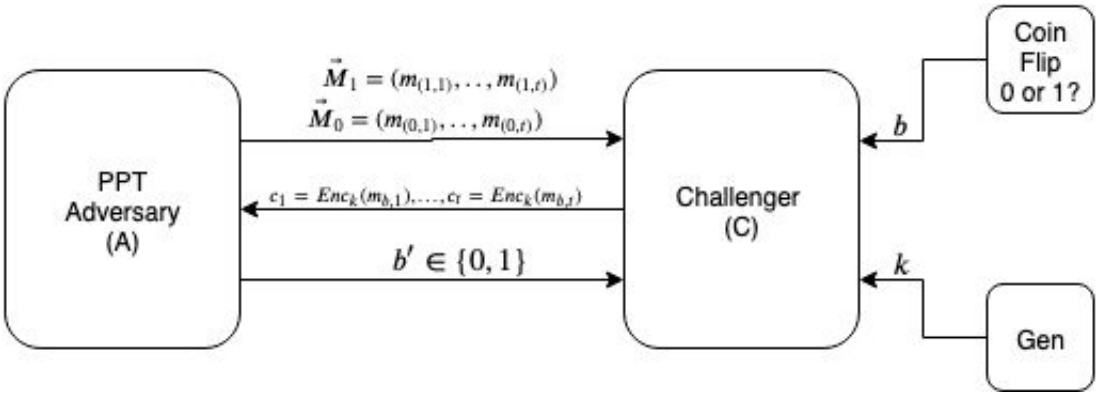

Figure 8: Multi Message Indistinguishable game

Step 1: The PPT Adversary A sends two message vectors $\vec{M}_0, \vec{M}_1$ of equal size to the challenger C

Step 2: C flips a coin to find out $b \in \{0,1\}$ and encrypts the vector $\vec{M}_b$ and sends it to A

Step 3: A responds his guess about which message vector is encrypted in b'.
If $b = b'$,then A won and C outputs 1
If $b \neq b'$,then A lost and C outputs 0
Let $\text{PrivK}^{ind}_{A,\pi}(\text{n}) = 1$ be an event in which A won.

π is multi-secure if $\forall$ PPT adversary A,$\exists$ a negligible function $negl$ such that

$$Pr(\text{PrivK}^{multi}_{A,\pi}(\text{n}) = 1)) \leq \frac{1}{2} + negl$$

4.2 Single message security vs Multiple messages security

Theorem: If π is a cipher whose Enc algorithm is a deterministic function of the key $k \in K$ and the plain-text $m \in M$,then π cannot have indistinguishable multiple encryption in the presence of an PPT adversary. Let us consider the following setup: Under this setup,A can always wins,therefore

$$Pr(\text{PrivK}^{multi}_{A,\pi}(\text{n}) = 1)) = 1$$

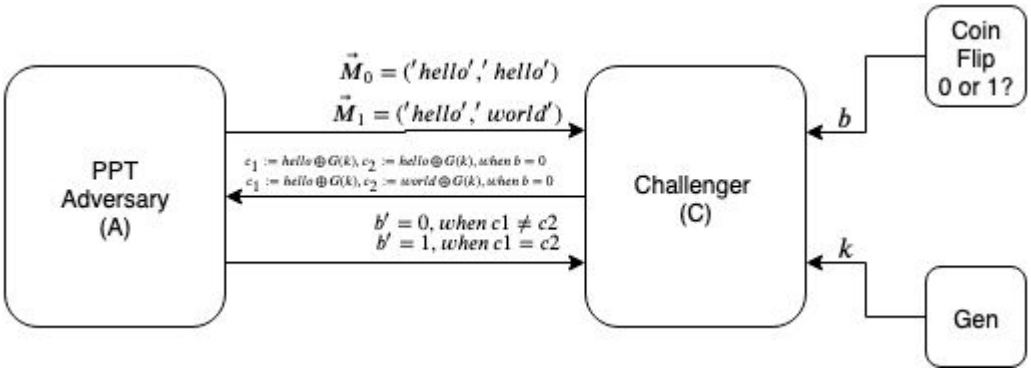

Figure 9: Attack using multi messages

This is attack is possible because of the following reasons:

- π is deterministic, encrypting same m results in same c
- The attack is possible for any deterministic Enc algorithm

Thus,we need to look for randomization during encryption.

5 Ciphertext Only Attack(COA)

Even though COA security notion is not a standard definition of SKE,there are interesting applications based on it:

- Roulette games
- Onion routing-(for sending anonymous messages): An onion is the data structure formed by wrapping up the message with successive layers of encryption.The number of layers can be as many as the intermediary routers in between the source and destination.In this setup,an intermediary router can only view the address of the next router to which it has to forward the message,by decrypting the topmost layer.As a result,not only the message remains hidden as it is transferred from one node to the next, but also no intermediary knows both the origin and final destination of the data, allowing the sender to remain anonymous.

6 References

[1] *Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, second edition. CRC Press, 2014.*

[2] *Arpita Patra. https://www.csa.iisc.ernet.in/ cris/e0_235.html Course Materials.*