CSA E0 235: Cryptography 27-01-2016

Scribe for Lecture 5

Instructor: Arpita Patra *Submitted by: Indumathy.J*

1 PSEUDORANDOMNESS:

Why Pseudorandomness /Pseudorandom Generator???

We adopted the world of *Computational Security* to overcome the 2 inherent limitations of *Perfect security.*That is in Computational Security,

- We make the 'threat' Computationally Bounded - PPT COA adversary.(Polynomial effcient & Probablistic adversary)
- We allow 'Break' with a small negligible probability.
- Small keyspace which is not as big as message space is used.
- And Key can be reused.

To encrypt PT - Plain Text with a key we need to have 'random' key whose length is equal to that of the PT's. But to generate truly random bits of large size is an expensive, difficult and a slow process. Hence its enough in computational security if we go for *Pseudorandomness* which 'looks' random to a PPT adversary , but truly its not. Thus, Computational Security is practical.

What is Pseudorandomness /Pseudorandom Generator???

Pseudorandomness is a property of probability distribution on strings. Lets say,

G : Some probability distribution over set of binary strings of length 'n'.
U : Uniform Probability Distribution (i.e. sampling any string from this set has same probability) over the same set of strings.

We call a string sampled from the set of string according to G as **pseudorandom string** or **psedorandomly generated** if its indisinguishable from a string sampled according to U (which we call truly random string) from the same set of strings to a PPT Distinguisher.

2 PSEUDORANDOM GENERATOR:

We call a determnistic Polynomial time algorithm G as psedorandom generator if it takes an uniform input string s $\varepsilon_R \{0,1\}^n$, where s : seed and n : security parameter, The output G(s) is a string whose length is polynomial in n say l(n) where l is a polynomial if it satisfies the following condition :

- **Expansion :** For every n , $l(n) > n$ must hold. l is called *expansion factor* of G.
- **Pseudorandomness :** For any PPT Distinguisher D , the following probability must hold for any negligible function *negl*:

$$| Pr[D(G(s) = 1] - Pr[D(r) = 1] | \leqslant negl(n)$$

which means the probability for a PPT distinguisher D to say a string is sampled according to G or if its a truly random string is almost same. We'll consider the following Experiment / Game to understand this better.

3 THE PRG SECURITY / GAME / EXPERIMENT :

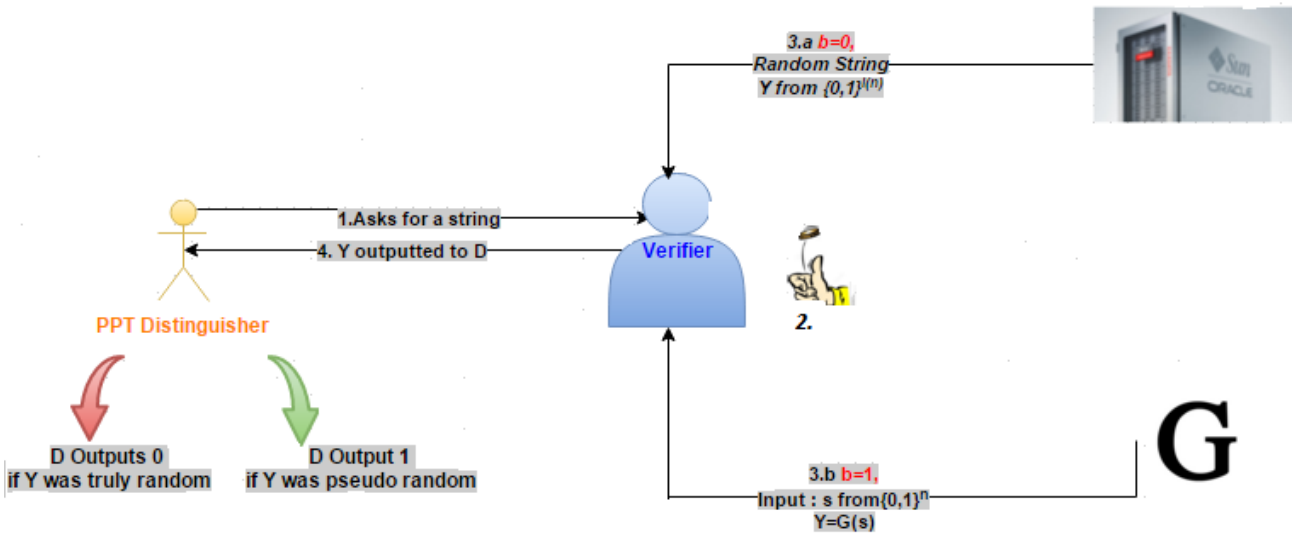
The aim of the game is to check if a PPT Distinguisher who can break a PRG system exists or not.The game is played between a challenger and a PPT Distinguisher D. Consider :

- U : Uniform Distribution over $\{0,1\}^{l(n)}$
- G : PseudoRandom Generator whose seed is a uniform string from $\{0,1\}^n$ and output G(s) is string of length l(n) . n - security parameter , l(n) - polynomial over n.

The Distinguisher asks for a string of length l(n) to the challenger .The challenger then chooses a random bit b ε_R {0,1}. If b = 0 , the the challenger outputs Y $\varepsilon_R \{0,1\}^{l(n)}$, which is a truly random string of length *l(n)* to D. Else he invokes G with a uniform seed chosen from $\{0,1\}^n$ as input and outputs $Y = G(s)$ to D.
Now D outputs a bit say b'.We call G as PRG if D is not able to distinguish wether y was sampled according to U or by G . This is given by the probability notion where *negl* is a negligible function :

$$|Pr[D(G(s) = 1] - Pr[D(r) = 1]| \leqslant negl(n)$$

i.e Probability of D outputting 1 if he sees Y=G(s) or Y = truly randm string is almost same. We need to note the randomness involved here is when choosing r (truly random string of length l(n)) or s(seed) and that of D who is a PPT Distinguisher.



3.1 An Example of construction of Pseudo Random generator :

Lets construct a PRG G with input seed s ε_R $\{0,1\}^n$ and expansion factor = n+1 .The Pseudorandom generator G works as follows : It ouputs the string Y=G(s) as ss' where the last bit s' is exor of 1st n bits.

Is the PRG Construction Secure???

We can easily have a PPT Distinguisher D who can guess that the string is a psudorandom string or truly random string in the PRG Security Game with a probability which is not $\leqslant$ a negligible value.

D can check if the last bit of Y is exor of the previous n bits and if so and if Y was output of G then Pr[D(G(s)=1] = 1

If suppose Y was a truly random string of length n+1 outputed in the game , then Pr[D(r) = 1] = 1/2 as Y ss' is a truly random string and no matter how s' was generated the probability that it would be equal to the ex-or of 1st 'n' bits is 1/2. Hence ,

$$|Pr[D(G(s) = 1] - Pr[D(r) = 1]| \leqslant 1/2 \text{ which is (non-negligible)}$$

Thus G is not a PRG.

Construction of PRG is a hard task!!!

3.2 OBSERVATION :

It can be shown that distribution on the output of a PRG G is far from Uniform distribution. This can be shown by considering the following scenario :

- Let G be a length doubling PRG , i.e $l(n) = 2n$, where n is the length of the input seed. Under uniform distribution on $\{0,1\}^{2n}$, the probability to choose a string is 2^{-2n}.

- In contrast, consider the distribution of the output of G . G's input is of length n and the number of different strings on the range of G is thus atmost 2^n. Thus the fraction of string of length 2n that are in the range of G is atmost $2^n/2^{2n} = 2^{-n}$ which dont occur as output of G.

Thus we can make an important observation that given an unbounded D (or unbounded time) PRG can be cracked.

3.3 Brute Force on seed space by an unbounded Distinguisher D / or given an unlimited amount of time, can actually crack PRG :

In the example above where G is a length doubling PRG , consider an unbounded Distinguisher D or consider he has been given exponential amount of time.D outputs 1 iff s ε_R $\{0,1\}^n$ such that $Y = G(s)$ by exhaustively computing G(s) for every s ε_R $\{0,1\}^n$ as by Kerchkhoff's law the entire specification of the Scheme (here PRG G) must be public except the secret key used in encryption. Thus , Pr[D(G(s)=1] =1 if $Y = G(s)$.

In contrast if Y was actually sampled from $0,1^{2n}$ uniformly , then the probability that there exists an s with G(s) = Y is atmost $1/2^n$ and so D outlputs 1 with probability atmost $1/2^n$ Thus , Pr[D(r) = 1] = 2^{-n} . Hence,

$$| Pr[D(G(s) = 1] - Pr[D(r) = 1 | > 1 - 1/2^n \text{ which is (non-negligible)}$$

Hence we can see an unbounded attacker can perform an brute force attack and crack PRG. Thus we must make sure that we select n sufficiently large so that an efficient Distinguisher D cant perform brute force attack on the seed space.

4 EXISTENCE OF PSEUDORANDOM GENERATORS:

We certainly understood that its hard to construct PRG , but there is no concrete proof for the existence PRG.But in the world of cryptography its strongly believed that PRG exists.(like the unproven assumption $P \neq NP$) This assumption may be constructed under 2 basis :

- First under the weak assumption that $one - way functions$ exist.Not practical and is due to the works of Goldreich-Levin, Yao .(Explained in section.7)
- Second based on a highly practical candidate for PRG called *StreamCipher*. **Stream Cipher** can be understood as a determnistic algorithm (just like G) which outputs pseudorandom bits gradually and as requested . Thus it offers greater efficiency in terms of only required number of bits can be requested and flexibility as there is no upper bound on the number of bits that can be requested.

Now we'll prove the security of a COA- SKE that uses PRG :

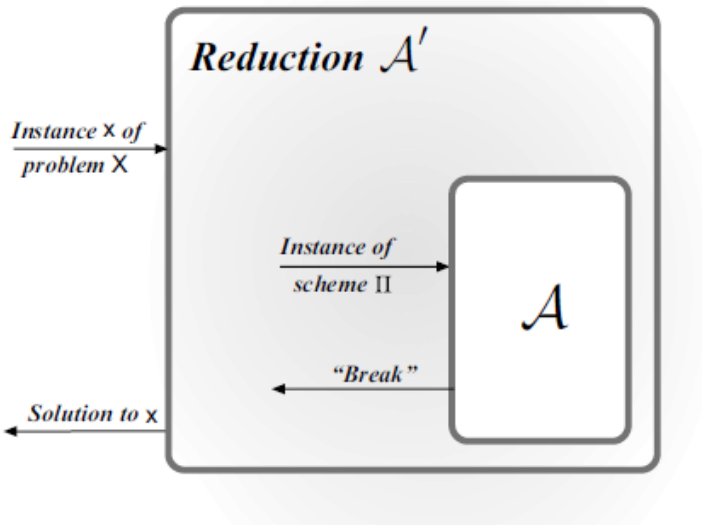
5 PROOF OF SECURITY OF THE PRG -BASED SKE:

This proof is done by the method of **REDUCTION.** Informally, to prove an Encryption construcion is secure by proof by *Reduction* is to start with an assumption that some mathematical problem is Hard which is believed to be true, or some known secure cryptographic pimitive and prove the security of the given construction using this problem or the primitive under the assumptions made.

This is done by contrapositive rule or proof by contradiction . Lets see the following simple scenario to show how we can use proof by reduction to prove a Cryptographic scheme π is secure.

5.1 Understanding proof by reduction:

- **INITIAL ASSUMPTION:** We start with the assumption that some problem **X** is hard, i.e it cannot be solved by any Polynomial time algorithm except with some negligible probability.
- Our goal is to prove that π is secure.Lets do that using proof by contradiction.
  - Lets fix some PPT Adversary A who can break π with good success probability $\xi(n)$.
  - Now we'll construct an efficient adversary say A' (called the reduction) who attempts to solve **X** using adversary A as a black box. i.e A' doesnt know how A works except the fact that he is good at attacking π.
  - So given an instance x of **X** A' has to simulate it into an instance of π in some manner and ask adversary A to break π for the simulated x. $\rightarrow$ **REDUCTION STEP.**
  - If A breaks the instance of π simulated by A' it must allow A' to solve the instance x atleast with an inverse probability $1/p(n)$.
  - It implies that A' solves **X** with probability $\xi(n)/p(n)$ as A solving π is independent of solving **X**.
  - If $\xi(n)$ is a non-negligible probability , Then it means B can break **X** with non-negligible probability which contradicts with our initial assumption. $\rightarrow$ **CONTRADICTION.**
- Thus given our assumption that **X** is hard, we can conclude that there is no efficient PPT sdversary A who can break π with non- negligible probability.



Now lets see the Construction of the PRG Based COA-Secure SKE Scheme and later prove the security of thus constructed scheme.

5.2 The PRG BASED COA- Secure SKE Scheme:

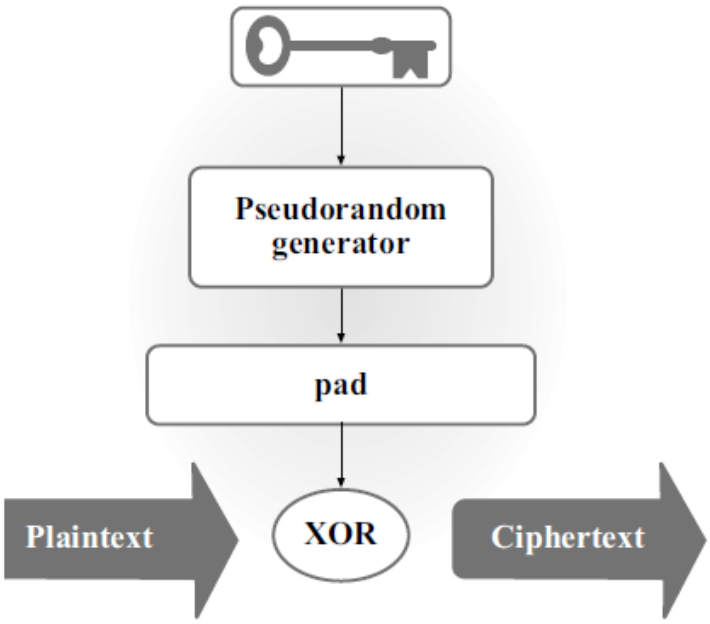
Construction of the scheme π :

Let G be a PRG with expansion factor *l(n)* , where n - security parameter and input seed length. Lets define a private-key encryption scheme for messages of length *l(n)* as follows:

- GEN : outputs a uniform key $k \, \varepsilon \{0,1\}^n$
- ENC : takes k and message $m \, \varepsilon \, \{0,1\}^{l(n)}$ as input and outputs cipher text c:=m⊕G(k).
- DEC : takes k and Ciphertext $c \, \varepsilon \, \{0,1\}^{l(n)}$ as input and outputs message m:=c⊕G(k).

$G(k)$- Pseudorandom key.

CORRECTNESS OF THE SCHEME:
Its easy to check that $DECK_k(ENC_k(\text{m})) = \text{m}$.
PROOF:
From the definition of ENC algorithm we get , $ENC_k(\text{m}) := \text{m} \oplus \text{G(k)}$.
From the definition of DEC algorithm we get , $DEC_k(ENC_k(\text{m})) := DECK_k($ m ⊕ G(k)) := m⊕G(k)⊕G(k) := m



5.3 THEOREM : If G is a PRG , then the above fixed-length private-key encryption scheme π is secure in the presence of an eavesdropper. i.e , COA-Secure.

PROOF:

- *INITIAL ASSUMPTION :* PRG Exists. and G is PRG
- Now we prove the security of π by reducing it to Pseudorandomness of the PRG G. We prove this by the proof by contradiction and lets assume the scheme is not COA-secure and there exists a PPT adversary who can break the scheme with non-negligible probability. Hence ,

$$\forall \text{ A Pr[A wins in } PrivK^{coa}_{A,\pi}(n)] \geqslant 1/2 + negl(n)$$

- For the negligible function negl(n) we can find a polynomial function P(n) and a N such that n$\geqslant$ N and negl(n) $<$ 1/P(n) :

$$\forall \text{ A Pr[A wins in } PrivK^{coa}_{A,\pi}(n)] > 1/2 + 1/P(n) \text{ ,for infinitely many n}$$

- Now we use the PPT adversary A who could break π to emulate a PPT Distinguisher D who can distinguish the output of G from a truly random string. This is done as follows there is a PRG verifier who chooses a bit b ε_R {0,1} and If b = 0 , the challenger outputs Y ε_R $\{0,1\}^{l(n)}$ to D. Else he inputs a uniform seed chosen from $\{0,1\}^n$ to G and outputs $Y = G(s)$ to D.
- If we can somehow show that D can break G with non-negligible probability we arrive at a contradiction to our inital assumption and we can thus prove that if

PRG exists then π is COA-Secure.

- **TO SHOW THAT:**

$$|Pr[D(G(s)) = 1] - Pr[D(r) = 1]| \geqslant 1/P(n) \text{ which is (non-negligible)}$$

- Now the π's adversary A sends two equal length (= l(n)) messages m_0 and m_1 to D.
- The D chooses another bit b1 ε_R {0,1} and encrypts the message m_{b1} with Y , C $\leftarrow m_{b1} \oplus$ Y and sends back to A. (Here D acts as $PrivK^{coa}_{A,\pi}(n)$ verifier.)
- The above step represents how D uses instance of PRG problem Y to simulate an instance C for the π. $\rightarrow$ REDUCTION STEP.
- We must note that D is completly unaware of how A works , he just knows A breaks π with non-negligible probability.
- NOTE: If suppose Y was actually a pseudorandom string then this game is simiar to $PrivK^{coa}_{A,\pi}(n)$.
  If not it would be similar to an OTP instance problem where C $\leftarrow m_{b1} \oplus$ Y is encrypted with a truly random key.
- Now A outputs a bit b' to D.
- D outputs 1 i.e the string Y is pseudorandomly generated by G if b=b'. As he knows that A is good at breaking π which uses Pseudorandom Key and the chances that Y was pseudorandom is high due to this fact.
- Else he outputs 0 i.e String Y must be a truly random string.
- Thus Pr[D(G(s))=1] $> 1/2 + 1/P(n) \rightarrow$ Probability of A breaking π
- PR[D(r) = 1] = 1/2 as Y is truly random string and no matter how A is powerful his probability of guessing b' would be 1/2 as for any message m_0 or m_1 we can find a random key Y with 1/2 probability.

$$\begin{aligned} | \, Pr[D(G(s)) = 1] - Pr[D(r) = 1] \, | &> (1/2 + 1/\mathrm{P(n)}) - 1/2) \\ &> 1/P(n) \text{ (non-negligible)} \end{aligned}$$

- Thus we see that G is not a PseudoRandom Generator which contradicts our initial assumption. Thus our assumption that π is not COA-SECURE is incorrect.

CONCLUSION: IF G IS PRG , THEN π Constructed using psedorandom key is COA-SECURE.

Now we have seen that using PRG , we have overcome one of the drawbacks of information theoritic perfect security i.e smaller keyspace than that of the message space can be used. To overcome the next drawback i.e Key reusability lets analyse MULTI -MESSAGE COA Secure Scheme.

6 MULTI MESSAGE COA - SECURITY :

Unlike Single-Message COA Security where the adversary over the channel can capture only one Cipher Text , But in reality he can observe the channel for longer time and can capture more than 1 cipher texts.This notion of multi-message COA- Secure SKE Scheme is defined by the following experiment / game :

The game $PrivK_{A,\pi}^{coa-mult}(n)$ is played between a challenger and a PPT adversary. The adversary picks two sets of Message vectors : $M_0 := < m_{0,1}$, $m_{0,2}$, ... , $m_{0,t} >$ and $M_1 = < m_{1,1}$, $m_{1,2}$, ..., $m_{1,t} >$ and sends them to the challenger. The challenger then chooses a uniform bit b ε_R $\{0, 1\}$ and sends the encrypted cipher text C := Enc(M_b) := $< Enc_k(m_{b,1})$, $Enc_k(m_{b,2})$, ... , $Enc_k(m_{b,t} >$. The adversary then guesses b'. If b = b', the adversary wins and otherwise the adversary loses.

6.1 DEFINITION:

A symmetric key encryption scheme π is said to be ciphertext-only-attack multiple message secure if for all PPT Adversaries A, there exists a negligible function negl(n) such that

$$\Pr[\text{A wins in } PrivK_{A,\pi}^{coa-mult}(n)] \leqslant 1/2 + negl(n)$$

6.2 RELATIONSHIP BETWEEN COA-SECURE AND COA-MULTI SECURE SCHEMES :

We need to note the relationship between the two notions of security i.e single message COA- Secure scheme and Multi message COA secure scheme. It is easy to observe that the single message security is a *special case* of multi message security when the 2 message vectors are of unit length.Hence any scheme that is COA-Multi secure is also COA-Secure. But The converse is not true.

6.3 SECURITY OF COA-MULTI SECURE SCHEME :

By allowing access to multiple messages, an adversary can figure out some information about the combination of the multiple messages. For Example :In the $PrivK_{A,\pi}^{coa-mult}(n)$ game let adversary A chooses $M_0 = <$ A,A $>$ and $M_1 = <$ A,B $>$ such that A and B are 2 different strings and ENC(A) $\neq$ ENC(B).The challenger outputs C = $<$ c1 , c2 $>$. Its easy for A to check if c1 = c2 , if so he knows that the 1st message vector has been chosen for encryption otherwise he the other message vector has been encrypted. Hence the adversary's output is always right by the choice of M_0 and M_1 .

The above break is due to the fact that encryption of same message always yields the

same cipher texts. In particular if any Encryption algorithm is determnistic one can exploit the above fact and break the system. Hence wee need to go for **RANDOMIZED ENCRYPTION ALGORITHM**.

7 ONE-WAY FUNCTION:

We just made 2 Assumptions that PRG Exists and this implies COA-Secure Schemes that use Pseudorandom key exists. Inorder to prove these assumptions we need to understand what is One-Way function.A short description of One-Way Function is outlined in this section.Informally One-way functions are those that are easy to compute but "difficult" to invert (almost always).

7.1 DEFINITION :

A Function f : $\{0,1\}^* \rightarrow \{0,1\}^*$ is said to be *One-Way Function* if for a x ε Domain of f and for a y ε Range of f it satisfies the below 2 properties :

- **EASY TO COMPUTE :** y = f(x) is easy computed easily in polynomial time.
- **HARD TO INVERT :** But the pre-image of y is difficult to compute.

7.2 THE INVERTING EXPERIMENT :

- The Experiment $Invert_{A,f}(n)$ is played by a PPT Adversary A(1^n) and a Verifier.
- A challenges the verifier that he can find the invert image of the one-way function f .
- The verifier picks any x ε_R $\{0,1\}^n$ from the Domain of f compute y=f(x) and gives y to the adversary.
- A outputs 1 if he find any one of the pre-image of Y say x' such that x'=f^{-1} (y) and he wins. x' may not be same as x
- If x' $\neq$ f^{-1} (y) A outputs 0 and he looses.
- Hence f is **ONE-WAY FUNCTION** if there exists a negligible function negl(n) such that :

$$\Pr[\text{A wins in } Invert_{A,f}(n)\ ] \leqslant negl(n)$$

7.3 CONSEQUENCE OF ONE WAY FUNCTION :

One can prove that

- If PRG exists then One-Way Function exists and hence,
- If coa-secure SKE exists, then One-Way Function exists.

References

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

[2] Arpita Patra. http://drona.csa.iisc.ernet.in/ arpita/Cryptography16.html. Course Materials.