CSA E0 235: Cryptography (January 25, 2016)

Scribe for Lecture 4

Instructor: Arpita Patra *Submitted by: (Seba Ann Kuruvilla)*

In the previous lectures we have seen various definitions of perfect security and their equivalences and have seen the Shannons theorem(useful tool for proving or disproving perfect secrecy) and its proof.We have also seen the inherent drawbacks of the perfect security in the OTP scheme,that the

1. The key space should be as large as the message space
2. The key cannot be reused

Due to these drawbacks, perfect security cannot be afforded, so we relax perfect security and introduce the notion of computational security. The important relaxations we make here to solve the above mentioned drawbacks are the following:

1. Make the adversary from having unbounded computational power to bounded (efficient/polynomial) computational power.
2. From a **'0'** break scheme to a scheme where break is allowed with negligible probability(ie.. from impossible to infeasible with high probability).

We will see theese relaxations are necessary and see them more precisely, and further will define secret key encryption in the world of computational security using two different paradigms

1. **Paradigm I**:Sematic Security for Secret Key Encryption which is the computational analogue of Shannon's perfect security
2. **Paradigm II**:Indistinguishability based security defintion for Secret Key Encryption which is the computational analogue of game based definition for Perfect Security.

We need to note that unlike the perfect security scheme, now we have assumptions in the computational security , which we will consider while creating schemes.

1 Need for a relaxed Threat Model

Let us Assume Secret Key Encryption scheme where many messages are encrypted using a single short key (ie .. we reuse the key as well the key length is shorter than message length).The adversary we consider in this situation is *unbounded* powerful instead of bounded.Let the adversary also have some number n of (cipher text,message) pairs with him. Eg:$(c_1, m_1)...(c_n, m_n)$ where $c_i = Enc_k(m_i)$.The adversary can bruteforce over all the keys in $O(|K|)$ time and find the corresponding key which gives the correct decryption for all the ciphers.But such an attack is possible only when the adversary has unbounded computing power.There for the relaxation is needed,so that such a brute force attack should not happen.Normally when the key is of length l bits the size of keyspace $|K|$ will be $O(2^l)$

2 Need for a relaxed Break Model

Again let us assume a Secure Key Encryption scheme where where many messages are encrypted using a single short key.The adversary we consider in this situation is *polynomial bounded* powerful instead of unbounded.Let the adversary also have some number n of (cipher text,message) pairs with him.Eg:$(c_1, m_1)...(c_n, m_n)$ where $c_i = Enc_k(m_i)$.Assume the system to have 0 break.One way to get over the scheme is to make a guess in constant ie..$O(1)$ time.If the adversary is lucky(which can can happen with $O(\frac{1}{|K|})$probability) he will get the correct key.Therefore the case of 0 break is not valid now.It means we need to relax the same.

3 Precisely Defining the Relaxations Using Asymptotic Approach

What is the Asymptotic approach , why it is used?When there are some technical and theoretical difficulties in explicitly bounding the maximum success probabilities of any adversary running for some specified time,ie..giving a concrete bound,we use an approach which uses complexity theory and introduce an integer valued security parameter **n**,that will parametrise the cryptographic scheme of the involved parties.Now let us precisely define the above mentioned factors(running time and success probabilities) in asymptotic approach ,using the security parameter **n**, in the world of computational security.

3.1 Polynomially Bounded $\rightarrow$ Probabilistic Polynomial Time

A probabilistic Poly time (PPT)(Feasible/Efficient) algorithm means and algorithm having running time which is polynomial in the input size.A *PPT* adversary means nothing but a PPT algorithm itself,ie.. the running time of the adversary is a polynomial in the input size.

A polynomial is nothing but a function $f(n) : Z^+ \rightarrow Z^+$ where there exist a finite number of c_i such that $f(n) < \sum_i c_i n^i \forall n$ *Eg:*$n^3 < c_1 n^1 + c_2 n^2 + c_3 n^3$

What is the limitation of a PPT adversary? The PPT adversary by definition cannot bruteforce over the keyspace as it takes $O(2^n)$ time for the same,if the key size is n.

3.2 Not Allowing 0 break $\rightarrow$ Negligible functions

Small or negligible functions are those functions which grows slower than any inverse polynomial.Eg:$\frac{1}{2^n}, \frac{1}{2^{\frac{n}{2}}}$.

Very small or neglible function $f(n)$'s are those functions for which every polynomial in n $p(n)$,there exist some positive integer N such that $f(n) < \frac{1}{p(n)} \forall n > N$. **Note:** $\frac{1}{n^{10}}$ is not a negligible function as $\frac{1}{n^{10}} \not< \frac{1}{n^{20}}$ but n^{20} is a polynomial.

Here in the world of computational security ,break is allowed with negligible probabilities.

3.3 Closure Properties of Polynomial and Neglible Functions

The polynomial and Neglible functions exhibits closure properties mentioned below

3.3.1 Polynomial functions(PPT)

Letp_1 and p_2 be polynomials in n then

1. $p_1 + p_2$ is a polynomial in n
2. $p_1 * p_2$ is a polynomial in n

3.3.2 Negligible functions

Let$negl_1$ and $negl_2$ be negligible functions in n then

1. $negl_1 + negl_2$ is a negligible function in n
2. $p(n) * negl_1$ is a negligible function in n for any polynomial $p(n)$

3.4 Security Parameter

Let us use asymptotic approach to say an adversary running for n^3 (PPT) time breaks the system with a probability $\frac{1}{2^n}$ $(negligible)$.As you increase the value of **n** here the life of the adversary becomes harder,it will take more time and have less chances of success.This **n** we refer here is a tunable parameter which tunes how difficult it is to break a cryptosystem.This will be the publicly known part of the scheme and will be an input to all associated algorithms.

Gen(Key Generation),Enc(Encryption),Dec(Decryption),Adversary will be polynomials in **n**..ie(Running time of the users and running time of the attackers will be polynomial in **n**) and success probabilities will be negligible in **n**.Usually the secret key size is same set to **n** and have values 128,256 etc...

Choosing n carefully is very essential... why??

Consider a scheme where designer claims that an adversary running for n^3 minutes can break his scheme with a probability of $2^{40}2^{-n}$.This is a negligible value,so the scheme is said to be secure/good.

Let us choose different values for **n**

1. If we choose $\mathbf{n} \leq 40$,we can see that the adversary running for40^3minutes(6 weeks)can break the scheme with a probability of 1.Now the scheme seems useless but its not the case we made a foolish choice of **n**
2. Now if we choose **n**=50 then the adversary running for 50^3 minutes will break the system with a probability of $\frac{1}{1000}$,this may or may not be acceptable to different schemes.
3. But if we choose **n**=500 an adversary working for only 200 years can break this scheme with a probability of 2^{-460} which is definitely acceptable.

There fore we can conclude that a wise choice of n is important.

Here **n** ,the security parameter, can be considered analogous to a knob. We need to set it in an optimal way, as an increase in **n** causes an increase in the running time of the user as well as the adversary,of which the former we want to reduce and latter we want to increase.

4 Concrete Approach

There is one more approach for defining the security of the system, which is the concrete approach ,where we set the value of **n** and run the scheme on different machines.Hence we can give concrete bounds or values to the running time and success probabilities of the schemes for Eg: *No adversary running for 5 years on a 4Giga Hertz machine can break the scheme with probability better than* 2^{-60}.

Although a concrete bound is what ultimately we want,we always stick on with the asymptotic approach as stating an asymptotic guarantee can give different concrete guarantees as ,the security parameter,the machine capacity etc..changes.One more reason is that a concrete approach guarantee is difficult to provide as there can be future advancements in computing power,change in the algorithm/software etc.. which cannot be taken care of when what we have is ,just a fixed concrete bound for a security scheme.

5 Revisiting the syntax of a Secret Key Encryption Scheme

The Secret Key Encryption scheme(SKE) mainly consists of three PPT algorithms which are as follows:

1. **Gen(1^n)**: Randomised key Generation Algorithm which takes **n** (security parameter) as input and ouputs a key **k** of length **n** according to some probability distribution.
2. $Enc_k(m)$:Encryption algorithm which takes a message of length **l(n)** and the key k from **Gen(1^n)** and outputs a cipher text **c**.It can be deterministic($c = Enc_k(m)$) or

randomised ($c \leftarrow Enc_k(m)$).This is a fixed length encryption scheme as the messages are all of same fixed length **l(n)**, which is a polynomial in **n**.

3. $Dec_k(m)$:Decryption algorithm ,which takes the cipher text **c** and key **k** and outputs message **m** which was encrypted in the cipher.This is usually a deterministic algorithm($m = Dec_k(c)$)

6 The two Paradigms of Security

6.1 Paradigm I $\rightarrow$ Semantic Security

This was proposed by S. Goldwasser and S. Micali(Probabilistic Encryption. Journal of Computer and System Sciences, 28(2): 270-299, 1984).
Here the **threat** is modelled as a Randomised,COA(cipher text only attack),PPT(Probabilistic Polynomial time) adversary and **break** is that we captured in Shannon's Perfect Security *ie..* given prior information about message, the cipher text leaks no additional information about the message.In perfect world we said it was **impossible** to break but here we say that **it is infeasible to break with high probability**.

Introducing two functions h(m) and f(m) They are functions of the message **m** where **h(m)** is called the history function and it gives the external information about the message ,if any, and **f(m)** is function of the message which the adversary aims to compute.

World of Semantic Security Here we introduce two worlds,in both they have access to history function,but in one world adversary is given the cipher text and and in another world we give the message length only.And the adversaries in both the worlds, let them be *A* and *A'* ,are asked to compute **f(m)**.
If in both the worlds the probabilities of the PPT adversaries,namely *A*and *A'*, to correctly compute **f(m)** is just negligibly apart ,we say our scheme is semantically secure.Thii idea is captured by the following equation.

$$\left|Pr[A(1^n, c, h(m))] = f(m)] - Pr[A'(1^n, |m|, h(m))] = f(m)]\right| \leq negl(n)$$

First probability in the above equation is taken over uniform distribution of key **k**,message sampler Samp(1^n),randomness of Adversary *A* and of the Encryption algorithm whereas the second probability is taken over message sampler Samp(1^n)(polynomial in **n**,randomness of Adversary *A'*.

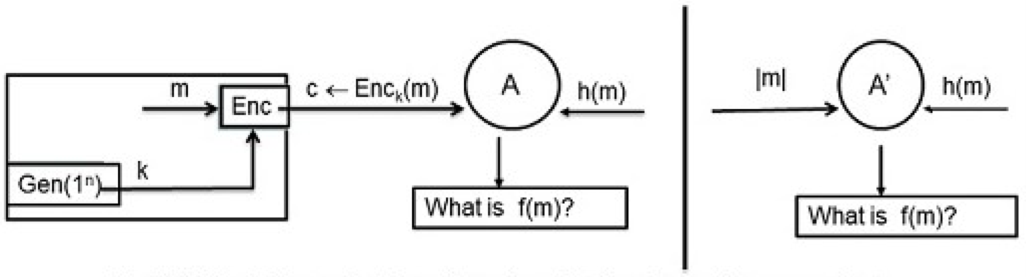

Fig: World of Semantic Security where both adversaries guess about f(m)

We can clearly see that Semantic security is the computational equivalent of the idea of perfect security as this equation captures the fact that the probability of an adversary to compute a function **f(m)** is almost the same ,regardless of the fact that it has access to the cipher text or not.(Access to the cipher text gives him only a negligible advantage to know about the original message).Here note that message length is not hidden,we aim in hiding the message content and not the length.

6.2 Paradigm II $\rightarrow$ Indistinguishability based Security

Here we explain the next paradigm ,which is an equivalent and easier definition of the same semantic security we have seen.It captures the idea with a difference in the break, that no PPT adversary should be able to say the difference between any two encrypted messages. Consider a scheme $\Pi = (Gen, Enc, Dec)$ and we use the following game between the PPT adversary A and a verifier.
The Game is as follows:

1. Adversary 'A' selects two messages m_0 and m_1 of equal length and gives it to the verifier.
2. Verifier flips a coin and chooses a bit $b \rightarrow 0, 1$
3. Verifier encrypts the corresponding message m_b using a key from **Gen** and produces a cipher text c
4. The cipher text is given to the adversary and he is asked to guess the bit b
5. if the guessed bit *b'* by the adversary is same as bit b the adversary wins and outputs a **1**(*b*==*b'*) else it output a **0**(*b*≠*b'*).

We denote this experiment using $PrivK^{ind}_{A,\Pi}$ where $\Pi = (Gen, Enc, Dec), m$.

Definition 1: A scheme is **ind-secure** if for every PPT attacker A there is negligible function $negl_n$ such that

$$Pr[PrivK^{ind}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + negl_n$$

Here the probability is taken over by the randomness used by A and the challenger.By the definition what we mean is that ,the scheme is **ind-secure** if the probability of the adversary to correctly guess the bit,is only negligibly greater than $\frac{1}{2}$,which is the probability that the adversary may ouput a random bit *ie..* It should not be able to distinguish between the encryptions of m_0 and m_1 with much probability than $\frac{1}{2}$

Note: An equivalent definition for indistuinguishability based security.If a scheme is **ind-secure** then for any PPT 'A' and any index 'i'

$$Pr[A(1^n, c) = m^i] \leq \frac{1}{2} + negl_n$$

*ie..*the probability of the adversary to predict correctly the ith bit of the message,seeing the cipher text, is only negligibly greater than $\frac{1}{2}$

Renaming the Indistuinguishability based definition In our scheme we use the Cipher Text only attack (COA) where the adversary can do nothing but just read the cipher text given to him.So we rename our indistuinguishability based experiment using the COA attack as $PrivK^{COA}_{A,\Pi}$

Definition 2: A Symmetric Key Encryption scheme is said to be COA secure(it having indistinguishable encryptions under COA attack) if for every adversary A there is a negligible function $negl_n$ such that

$$Pr[PrivK^{COA}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + negl_n$$

Here the intuition is that in the COA (Cipher text only attack) the adversary should not be able to distinguish between the messages ,m_0 or m_1, seeing just the cipher text.The further mentions will be based on this definition of security.

Note: An equivalent definition to the above COA secure scheme.A scheme $\Pi = (Gen, Enc, Dec)$ is **coa-secure** if for every PPT adversary A, there is a negligible function $negl_n$, such that

$$\left| Pr[Output(PrivK^{COA}_{A,\Pi}(n,0)) = 1] - Pr[Output(PrivK^{COA}_{A,\Pi}(n,1)) = 1] \right| \leq negl_n$$

Which means that an adversary A will give an output in the same way regardless whether m_0 or m_1 was encrypted. In other words,A is unable to distinguish between the two encryptions.

6.3 Sematic Vs Indistuinguishability Based Security

Summarising both the paradigms, **Semantic Security** tells that, given prior information about message, the ciphertext leaks no additional information about the message and **Indistui-nguishability based security** tells that, given the knowledge of two messages, it cannot be distinguished that ,the ciphertext corresponds to the first or second message.Both are equivalent definitions and can be shown that one implies the other and vice-versa.

7 Assumption for COA secure Secret Key Encryptions

The world of computational security overcame the limitations of perfect security world, and allows

1. Shorter key for bigger message
2. Reuse of the key

Now let us consider the OTP scheme,when we use the above properties ,the pad used can't just be the key.We need to expand the key length,for that we need functions which create the pad/mask using the key as input.For perfect security we need the pad to be truly random,but in the computational world of security we need the pads to just **look random** and not really **be random**.So we introduce here Pseudo Random Generators**(PRG)** which are tools used to cheat the PPT adversaries,about which we will study in the upcoming lectures.

7.1 Pseudorandomness

Before seeing the PRG(Pseudo Random Generators) its important to understand the concept of **Pseudo-Randomness** which is the property of a probability distribution.

Let **G** be a probability distribution and **U** be a uniform probability distribution,over the set of all binary strings of length l.We say distribution G is pseudo-random if a string drawn according to G is indistinguishable from a string drawn according to U ,to any PPT distinguisher .

A string drawn according to **G** is **pseudorandom** and a string drawn according to **U** is **random**.