CSA E0 235: Cryptography January 20, 2016

Scribe for Lecture 5

Instructor: Arpita Patra *Submitted by: Sruthi Sekar*

1 Brief Recap Of last lecture

We ventured into the world of Computational security in order to overcome the drawbacks accompanied by perfect security setting. The following was formalized:

- In the computational setting, we now assume an efficient adversary (threat model relaxed) and we allow the adversary to break the scheme with a negligible probability(break model relaxed). Hence we gave precise definitions for efficient (probabilistic polynomial time) algorithms and Negligible functions.
- We then gave two equivalent security definitions of a Secret Key Encryption scheme (in the computational setting), the Semantic security definition and the Indistinguishability based security definition (4).
- As the indistinguishabilty based definition is easier to work with and both definitions are equivalent, we work with this.
- With this, we saw that there is a need for additional assumptions to construct a coa-secure SKE. This led to the concept of Pseudorandomness and Pseudorandom Generators.

2 Psedurandomness and Pseduorandom Generators

2.1 Pseudorandomness

This is a property of a probability distribution. We consider the set of all binary strings of length l, say **S**. Consider two distributions:
G : some probability distribution on **S**
U : uniform probability distribution on **S**
Definition: G is pseudorandom if a string drawn from G is indistinguishable from a string drawn from U, to a probabilistic polynomial time distinguisher.
Although, this is a property of probability distribution:
a) A string drawn from G is called pseudorandom.
b) A string drawn from U is called random.
Remark : *Just like Ind security is a computational relaxation of perfect secrecy, pseudorandomness is a computational relaxation of true randomness.*

2.2 Pseudorandom Generators

A pseudorandom generator G is an efficient, deterministic algorithm for transforming a short, uniform string called the seed into a longer, "uniform looking"(or "pseudorandom") output string. The need for pseudorandom generators in cryptographic applications motivated a cryptographic approach to defining pseudorandom generators in the 1980s. The basic realization was that a good pseudorandom generator should pass all (efficient) statistical tests, i.e., for any efficient statistical test (or distinguisher) $\mathcal{D}$, the probability that $\mathcal{D}$ returns 1 when given the output of the pseudorandom generator should be close to the probability that $\mathcal{D}$ returns 1 when given a uniform string of the same length. Informally, then, the output of a pseudorandom generator should "look like" a uniform string to any efficient observer.

Formal Definition: : Let l be a polynomial and let G be a deterministic polynomial-time algorithm such that for any n and any input $s \in_R \{0,1\}^n$, the result $G(s)$ is a string of length $l(n)$. We say that G is a pseudorandom generator if the following conditions hold:

- **Expansion**: For every n, $l(n) > n$.
- **Pseudorandomness**: For any probabilistic polynomial time algorithm $\mathcal{D}$, $\exists$ negligible function negl such that

$$|Pr[\mathcal{D}(r) = 1] - Pr[\mathcal{D}(G(s)) = 1]| \leq \mathsf{negl}(n),$$

  where the first probability is taken over the uniform choice of $r \in \{0,1\}^{l(n)}$ and the randomness of $\mathcal{D}$, and the second one is over the uniform choice of $s \in \{0,1\}^n$ and the randomness of $\mathcal{D}$.(See Fig 1)

l is called the expansion factor of G.

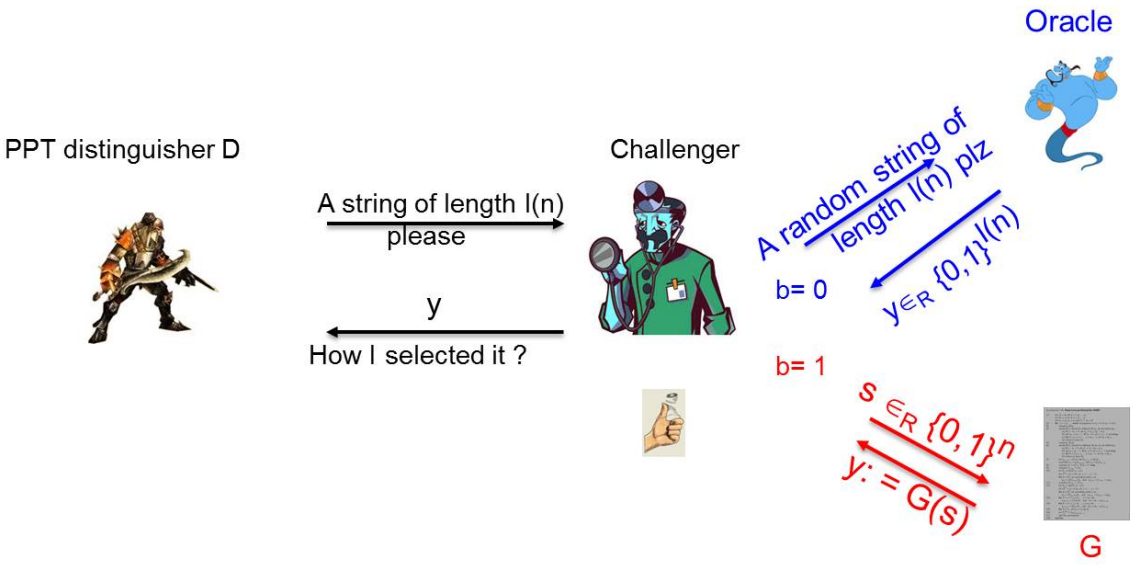

Figure 1: PRG Security

An example of a insecure PRG:

Define $G : \{0,1\}^n \to \{0,1\}^{n+1}$ by:

$$G(s) = ss' \text{ where } s' = \oplus_{i=1}^{n} s_i$$

To show that this is not a PRG, we define the following probabilistic polynomial time distinguisher $\mathcal{D}$:

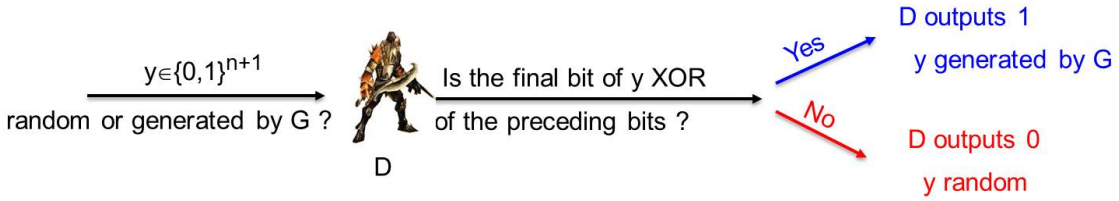

Figure 2: Insecure PRG

$\mathcal{D}$: on input $w \in \{0,1\}^{n+1}$, $\mathcal{D}$ outputs 1 if and only if $w_{n+1} = \oplus_{i=1}^{n} w_i$. (see Fig 2)
Since this holds for all strings output by G, we have $Pr[\mathcal{D}(G(s)) = 1] = 1$. If $w \in_R \{0,1\}^{n+1}$, the last bit w_{n+1} is uniform, and hence $Pr[\mathcal{D}(w) = 1] = \frac{1}{2}$.
$\implies |Pr[\mathcal{D}(w) = 1] - Pr[\mathcal{D}(G(s)) = 1]| = \frac{1}{2}$, which is not negligible. Hence, G is not a PRG.

PRG can be cracked by an unbounded adversary:

The distribution on the output of a pseudorandom generator G is far from uniform. Consider a length doubling PRG $G : \{0,1\}^n \to \{0,1\}^{2n}$, where $l(n) = 2n$.

- Under the uniform distribution on $\{0,1\}^{2n}$, each of the 2^{2n} strings is chosen with probability exactly 2^{-2n}.
- In contrast, consider the distribution of the output of G, when G is run on a uniform seed. When G receives an input of length n, the number of different strings in the range of G is atmost 2^n.
  $\implies$ the fraction of strings of length $2n$ in the range of G is atmost $2^n/2^{2n} = 2^{-n}$

Thus, a vast majority of the strings of length $2n$ do not occur as a output of G. Hence, given *an unlimited amount of time*, we can construct a distinguisher for the above PRG:

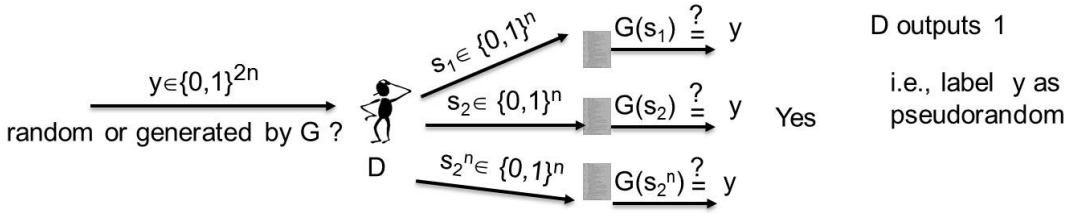

Figure 3: Unbounded distinguisher for a length doubling PRG

Exponential time distinguisher $\mathcal{D}$: $\mathcal{D}(w)$ outputs 1 if and only if $\exists s \in \{0,1\}^n$ such that $G(s) = w$. (Fig 3)

Remark : $\mathcal{D}$ takes exponential time to perform an exhaustive search, computing $G(s)$, for all $s \in \{0,1\}^n$. Moreover, by Kerckhoffs' principle, the specification of G is known to $\mathcal{D}$.

- If w is uniformly distributed in $\{0,1\}^{2n}$, then $Pr[\exists\ s\ s.t.\ G(s) = w] \leq 2^{-n}$
- If w is output of G, then $Pr[D(w) = 1] = 1$.

Hence, $|Pr[\mathcal{D}(w) = 1] - Pr[\mathcal{D}(G(s)) = 1]| \geq 1 - 2^{-n}$, which is non-negligible.
Remark : This brute force attack does not contradict the pseudorandomness of G, since the attack is not efficient.

Do PRGs exist?:
Although we do not know how to unconditionally prove the existence of pseudorandom generators, we have strong reasons to believe they exist.
First, they can be constructed under the rather weak assumption that *one-way functions* (See Section 3) exist, although these constructions are far from being practical.
Second, there exist several practical constructions of candidate pseudorandom generators called **stream ciphers** for which no efficient distinguishers are known.
Assumption: The very first assumption we make is that PRGs exist.

3 One Way Functions

This section aims at giving a brief overview of what a one way function means.
One-way functions are functions, which are *easy to compute*, but *hard to invert*.

The first condition can be formalized as: f is computable in polynomial time.

The second one can be formalized as: it is infeasible for any probabilistic polynomial time algorithm to invert f, that is, to find a preimage of a given value y, except with negligible probability.
Formal Defintion: Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function. Consider the inverting experiment $\mathbf{Invert_{A,f}(n)}$: (for a probabilistic polynomial time adversary $\mathcal{A}$ and any value n for security parameter)

- Choose uniformly, $x \longleftarrow \{0,1\}^n$. Compute $y := f(x)$.
- $x' \longleftarrow \mathcal{A}(1^n, y)$
- Output: 1 if f(x') = y; 0 otherwise.

We say f is a one way function if:

- **Easy to compute:** $\exists$ a polynomial time algorithm M_f computinf f, i.e., $M_f(x) = f(x)\ \forall x$
- **Hard to invert:** For every probabilistic polynomial time $\mathcal{A}$, there exists a negligible function negl such that:

$$|Pr[Invert_{A,f}(n) = 1]| \leq \mathsf{negl}(n)$$

4 COA Secure Scheme from PRG

Before defining a coa-secure encryption scheme, we recall the indistinguishability based security definition of a secret key encryption scheme for an adversary using a cipher-text only attack. Consider the following **Indistinguishability experiment** $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}(n)$ for a private key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any probabilistic polynomial time adversary $\mathcal{A}$ and any value n for security parameter:

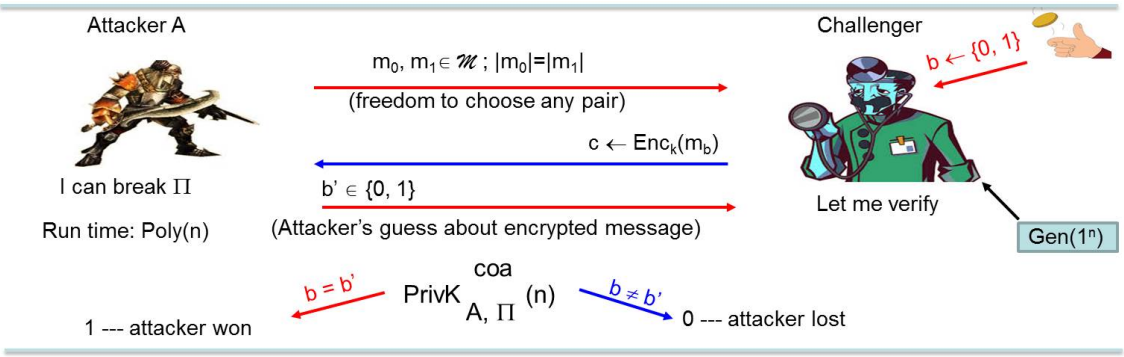

Figure 4: Indistinguishability experiment

Definition : A private key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is coa-secure if $\forall$ probabilistic polynomial time adversary $\mathcal{A}, \exists$ a negligible function negl s.t.

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

where the probability is taken over the randomness used by $\mathcal{A}$ and the randomness used in the experiment.
Remark : We use the indistinguishability based definition for security as we know that Semantic Security $\approx$ Ind Security.
The two main goals of the construction are:

- To construct a coa-secure encryption scheme that uses a key shorter than the message.
- To construct a scheme which allows reuse of key for multiple messages.

4.1 Proof By Reduction

Before defining an explicit construction of a scheme based on certain assumption, we see what we mean by "proof by reduction". It is a tool used in giving a mathematical proof of security of the encryption scheme.

To prove that a given construction is computationally secure, we must rely on unproven assumptions. We assume that some mathematical problem is hard, or that some low-level cryptographic primitive is secure, and then to prove that a given construction based on this problem/primitive is secure under this assumption.
There are four cases which may arise in a proof by reduction:
Case 1: If a scheme Π is secure then Π' is secure.

Case 2: If assumption A1 holds then A2 holds.
Case 3: If assumption A holds then Π is secure.
Case 4: If Π is secure then assumption A holds.
We consider the first case, explaining the procedure of proof by reduction.

The proof that a cryptographic construction is secure as long as some underlying problem is hard generally proceeds by presenting an explicit reduction showing how to transform any efficient adversary $\mathcal{A}'$ that succeeds in breaking the construction Π', into an efficient algorithm $\mathcal{A}$ that solves the problem Π that was assumed to be hard. Here are the steps:
To Prove: Some cryptographic construction Π' is secure, given that Π is secure.

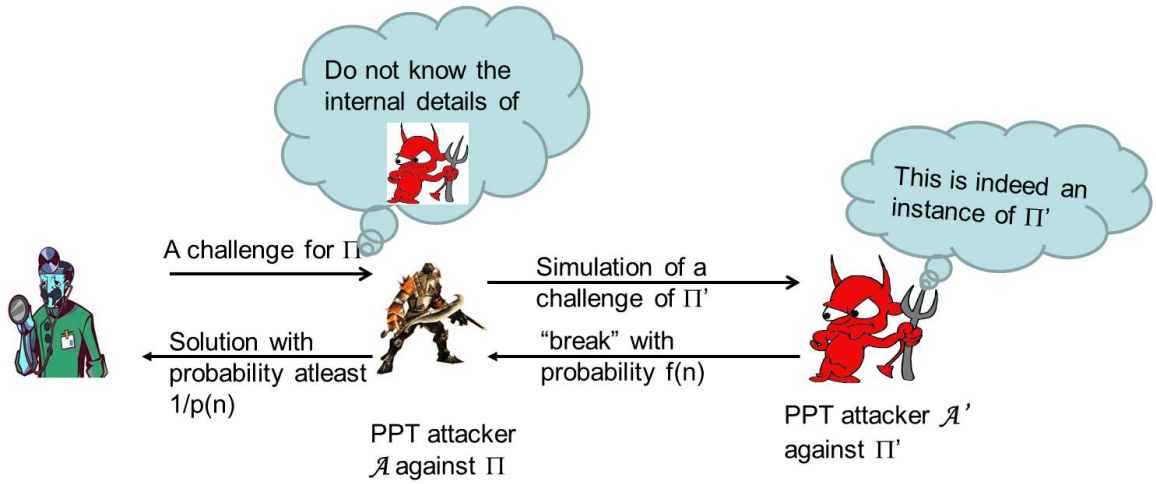

Figure 5: Proof by Reduction

We prove this by Contradiction:

- Assume to contrary that Π' is not secure. Then $\exists$ a probabilistic polynomial time Adversary $\mathcal{A}'$ that can break Π' with probability $f(n)$
- Construct an efficient (ppt) algorithm $\mathcal{A}$, called the **"reduction"**, that attempts to break Π, using $\mathcal{A}'$ as subroutine. (**Remark** : It is important to see here that, $\mathcal{A}$ knows nothing about how $\mathcal{A}'$ works. In other words, it is given a "black-box access" to $\mathcal{A}'$) So, given a challenge for $\mathcal{A}$ from Π, it simulates an instance of Π' for $\mathcal{A}'$, which should be identical to a view that $\mathcal{A}'$ would have received on direct interaction with Π'.
- If $\mathcal{A}'$ succeeds in breaking the instance of Π' that is being simulated by $\mathcal{A}$, then this should allow $\mathcal{A}$ to break the challenge that it was given, at least with inverse polynomial probability $\frac{1}{p(n)}$
- Since the randomness used in $\mathcal{A}$ is independent of that used in $\mathcal{A}'$, the probabilities are independent and we see that $\mathcal{A}$ breaks the challenge with probability at least $\frac{f(n)}{p(n)}$. Moreover, since $\mathcal{A}'$ is efficient, so is $\mathcal{A}$.

- If $f(n)$ is not negligible (by contradictory assumption), then by property of negligible functions $\frac{f(n)}{p(n)}$ is not negligible $\implies$ Π is not secure. This is a contradiction.

Hence we outlined the steps involved in proof by reduction, which will be used in proving the security of the coa-secure encryption scheme, which is constructed based on the assumption that PRGs exist.

4.2 A COA-Secure Fixed length Secret Key Encryption [i]

By the construction of One Time Pad, we XORed the message with a truely random string, which gave a perfectly secure encryption scheme. To use a shorter key, we can use a pseudorandom pad instead! The advantage of this is that now, the key, which will be a seed for the PRG, will be shorter than the message. So, the assumption that we make here is the existence of PRGs. Then, intuitively, the scheme must be secure, as a pseudorandom pad "looks random" and hence any probabilistic polynomial time adversary cannot distinguish between a message, encrypted using the one-time pad or a message, encrypted using this "pseudo-"one-time pad encryption scheme.

a) **Formal Definition of the scheme:** Fix some message length l and let G be a pseudorandom generator with expansion factor $l(n)$ (i.e., $|G(s)| = l(|s|)$. The scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is defined as below: **Correctness**: $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$

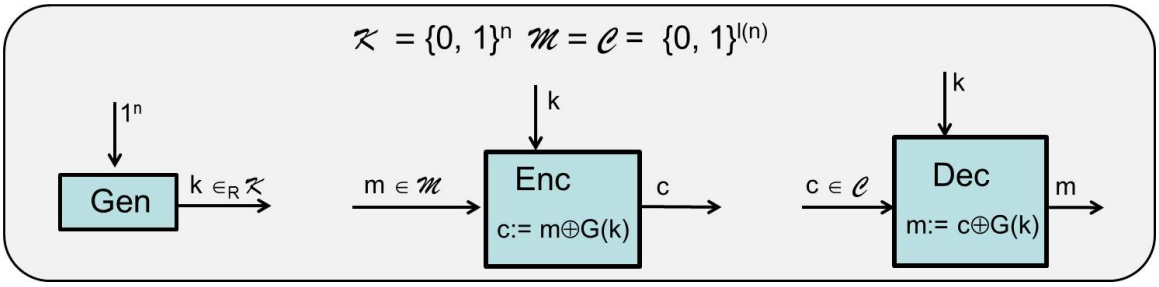

Figure 6: COA-Secure Encryption Scheme using PRG

b) **Precise Assumption:** We assume that Pseudo-random generators exist.

c) **Proof Of Security:** Now, we give a mathematical proof of the security of the scheme in Fig 6:

[i] to enable use of a key, whose length is shorter than the message length

Theorem 1: If G is a Pseudorandom Generator, then Π (in Fig 6) is a fixed length coa-secure Secret Key Encryption scheme.

Proof : We prove this by reduction. We wish to show that Π satisfies the Indistinguishability based security definition, i.e., ∀ probabilistic polynomial time adversary $\mathcal{A}$, ∃ a negligible function negl s.t.

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

$$\implies \sum_{\text{All ppt adversaries}} Pr[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A}_i,\Pi}(n) = 1] \leq \frac{1}{2} + \frac{1}{p_i(n)} \; \forall n \geq N_i$$

Assume to the contrary that Π is not coa-secure. This means, ∃ a probabilistic polynomial time distinguisher $\mathcal{A}$ and a polynomial $p(n)$ such that

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

We now construct a distinguisher $\mathcal{D}$ for breaking the PRG security, hence bringing a contradiction. $\mathcal{D}$ uses $\mathcal{A}$ as a subroutine. Clearly $\mathcal{D}$ runs in polynomial time (assuming $\mathcal{A}$ does).

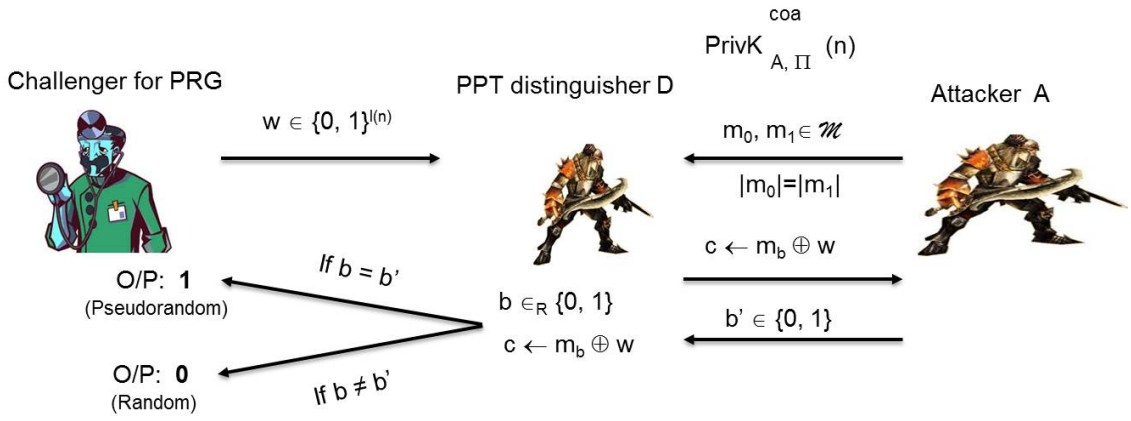

Figure 7: Reduction proof for security of Π

We define a modified encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$

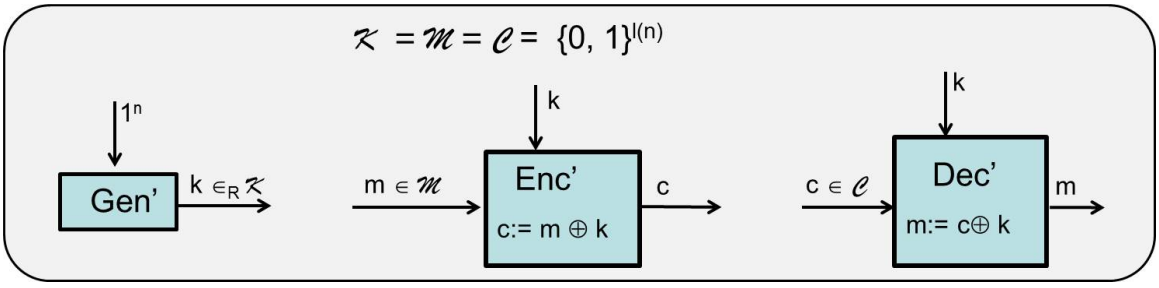

Figure 8: Modified encryption scheme Π'

Perfect Security of OTP $\implies Pr[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi'}(n) = 1] = \frac{1}{2}$

- If $w \in_R \{0,1\}^{l(n)}$, then the view of $\mathcal{A}$ when run as a subroutine by $\mathcal{D}$ is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi'}(n)$. This is because when $\mathcal{A}$ is run as a subroutine by $D(w)$ in this case, $\mathcal{A}$ is given a ciphertext $c = w \oplus m_b$, where $w \in \{0,1\}^{l(n)}$ is uniform. Since $\mathcal{D}$ outputs 1 exactly when $\mathcal{A}$ suceeds, therefore:

$$Pr[\mathcal{D}(w) = 1] = Pr[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi'}(n) = 1] = \frac{1}{2}$$

- If w is instead generated by choosing uniform $k \in \{0,1\}^n$ and then setting $w := G(k)$, then the view of $\mathcal{A}$ when run as a subroutine by $\mathcal{D}$ is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}(n)$.This is because when $\mathcal{A}$ is run as a subroutine by $D(w)$ in this case, $\mathcal{A}$ is given a ciphertext $c = w \oplus m_b$ where $w = G(k)$ for a uniform $k \in \{0,1\}^n$.Thus

$$Pr[\mathcal{D}(G(k)) = 1] = Pr[\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}(n) = 1]$$

Therefore, we have:

$$|Pr[\mathcal{D}(w) = 1] - Pr[\mathcal{D}(G(k)) = 1]| > \frac{1}{2} + \frac{1}{p(n)} - \frac{1}{2} = \frac{1}{p(n)} \text{ for infinitely many n}$$

$\implies$ $\mathcal{D}$ is a probabilistic polynomial time distinguisher for the PRG G. This is a contradiction to the security of the PRG and hence we have completed the proof by reduction.

4.3 Multi message COA Security [ii]

Definition 4 of security deals with the case where the communicating parties transmit a single ciphertext that is observed by an eavesdropper. It would be convenient, however, if the communicating parties could send multiple ciphertexts to each other, all generated using the same key, even if an eavesdropper might observe all of them. For such applications we need an encryption scheme secure for the encryption of multiple messages.
Consider the following **multiple message eavesdropping experiment** $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n)$ for a private key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any probabilistic polynomial time adversary $\mathcal{A}$ and any value n for security parameter:

[ii] formalizing key reusability

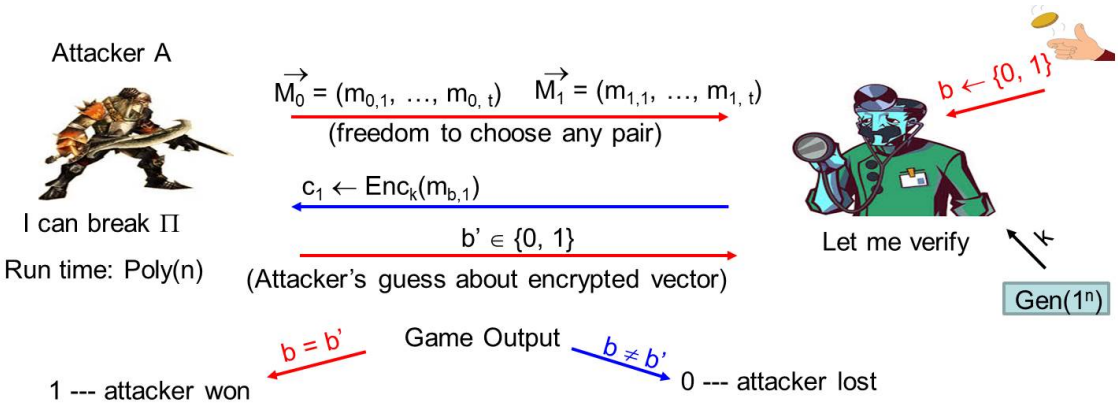

Figure 9: Multiple message COA-Security

Definition : A private key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is coa-mult-secure if $\forall$ probabilistic polynomial time adversary $\mathcal{A}, \exists$ a negligible function negl s.t.

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

where the probability is taken over the randomness used by $\mathcal{A}$ and the randomness used in the experiment.

Hence, any scheme that has indistinguishable multiple encryptions in the presence of an eavesdropper clearly also satisfies the indistinguishability based security Def 4. This is because $\mathsf{PrivK}^{\mathsf{coa}}_{\mathcal{A},\Pi}(n)$ is a special case of $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n)$, where the adversary outputs two lists containing only a single message each.

$\Longrightarrow$ Any cipher that is coa-mult-secure is coa-secure.

Remark : Converse of above statement is not true. This is stated in following theorem.

Theorem 2: There exists a private key encryption scheme that is coa-secure but not coa-mult-secure.

Proof: The one-time pad is perfectly secure, and hence is coa-secure. We show that it is not coa-mult-secure. Consider the experiment $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},OTP}(n)$:

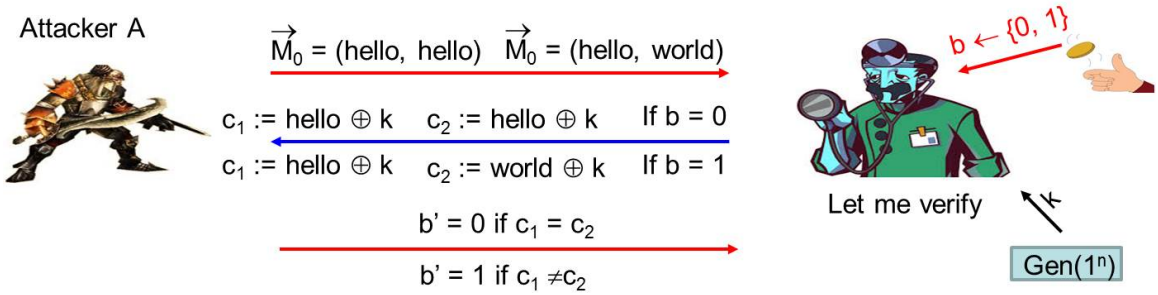

Figure 10: coa-secure but not coa-mult-secure

The first message contains the same plaintext twice, while the second contains two different messages. We analyze the probability that $b = b'$. Since the One Time pad is *deterministic*, encrypting the same message twice, using the same key yields the same ciphertext.Thus, if $b = 0$ then we must have $c_1 = c_2$ and $\mathcal{A}$ outputs 0 in this case. On the other hand, if $b = 1$, then a different message is encrypted each time, hence $c_1 \neq c_2$ and $\mathcal{A}$ outputs 1. Hence

$$Pr[\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},OTP}(n) = 1] = 1$$

Hence, the above encryption scheme is not coa-mult-secure.

Infact, the above attack can be mounted on any encryption scheme whose
Enc algorithm is deterministic. In this case, encrypting the same message multiple times with same key always gives the same cipher text and hence, above attack works. Stating this formally:
Theorem 3: If Π is an encryption scheme in which Enc is deterministic function of the key and the message, then Π cannot have indistinguishable multiple encryptions in the presence of an eavesdropper.
Proof: The proof of this uses the same attack as in Theorem 2. This follows from the discussion preceding the theorem.

How to overome this?
To construct a scheme, secure for encrypting multiple messages, we must design a scheme in which encryption is *randomized* so that when the same message is encrypted multiple times, different cipher texts can be produced.

Having achieved the two goals, i.e., ensuring use of a shorter key and allowing key re-usability, we now move on to strengthening the attack model. We now consider a more powerful adversary and build a scheme that satisfies this stronger security definition. This would imply that it satisfies coa security and coa-mult security.

5 Relation between One way functions, PRGs and COA-secure SKE

In the concluding section, we just state the relations between the two primitives, One way functions and Pseudorandom generators and a coa-secure SKE. We made two assumptions:

- PRGs exist
- COA-secure Secret key encryption scheme exists.

The following figure 11 depicts the relation:

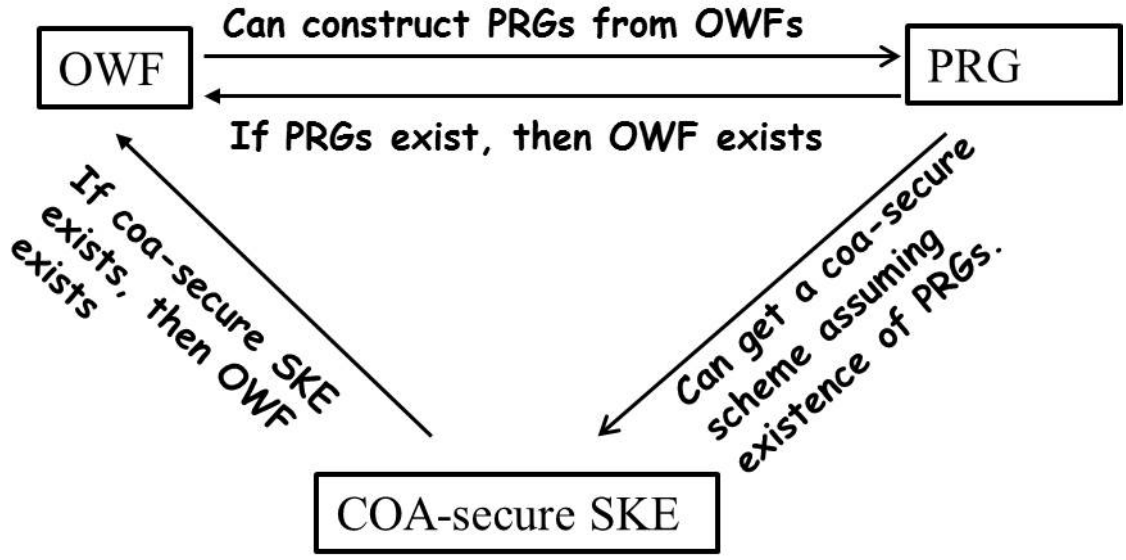

Figure 11: OWF, PRG and coa-secure SKE

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.