CSA E0 235: Cryptography January 31st 2016

Scribe for Lecture 6

Instructor: Arpita Patra *Submitted by: O.S.L.Bhavana*

1 Recall of Lecture 5

- Introduced Pseudorandomness and Pseudo random generators and gave an indistinguishability based definition. Discussed attack by an unbounded adversary and existence of PRG's.
- Gave indistinguishability based definition for COA attack and constructed a COA-secure SKE under the assumptions of existence of PRG.
- To deal with the key reuse limitation introduced multiple message COA security and found that any scheme Π that is COA-mult secure is COA-secure and the converse is not necessarily true.
- Defined One-Way functions and the inverting experiment.

2 CPA Security

2.1 CPA indistinguishability experiment $PrivK^{CPA}_{A,\Pi}(n)$

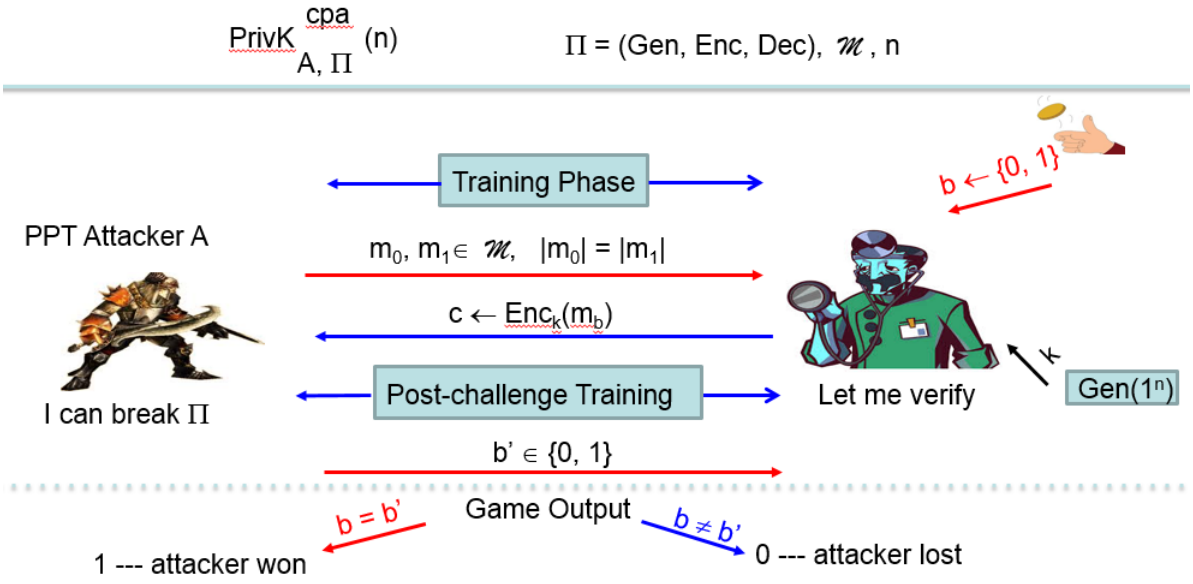

Figure 1 Indistinguishability experiment for CPA-security

- The challenger or verifier generates a key k using $Gen(1^n)$

- A is given black-box access to encryption oracle that is hard wired with a key k.
- A is allowed to query the encryption oracle polynomial number of times adaptively for any message $m \in M$of his choice and in return he would get $Enc_k(m)$. This process is called Pre- training phase.
- Then A chooses any two messages(even the messages for which he queried earlier)m_0, m_1 and sends to the challenger.
- Challenger Chooses a bit b uniformly at random and encrypts message m_b and sends the challenge cipher text to A
- A can again opt for training after he has seen the challenge cipher text this phase is call post-Training phase.
- A would then reply challenger with a bit b' and wins if he guesses message corresponding to the cipher text correctly i.e $b = b'$, loses otherwise.
- $PrivK^{CPA}_{A,\Pi}(n) = 1$if A wins, 0 otherwise.

2.2 Definition

A scheme Π is CPA-secure if for all PPT adversaries A there exists a negligible function $negl(.)$ such that

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

where probability is over the random tosses of A and random coins used in the experiment. Intuitively given knowledge of two messages cannot distinguish which of the two messages the cipher text corresponds to even when given black box access to encryption oracle.

2.3 Is CPA attack realistic?

One can feel that why would any legitimate party like to train adversary with corresponding cipher texts of messages he chose. And may feel this scenario is unrealistic. But CPA-attacks have made a significant place in history.

During world war II, US Navy cryptanalysts had discovered that Japan was planning an attack on Midway island in the Central Pacific. They had learned this by intercepting a communication message containing the ciphertext fragment "AF" that they believed corresponded to the plaintext "Midway island". Unfortunately, their attempts to convince Washington planners that this was indeed the case were futile; the general belief was that Midway could not possibly be the target. The Navy cryptanalysts then devised the following plan. They instructed the US forces at Midway to send a plaintext message that their freshwater supplies were low. The Japanese intercepted this message and immediately reported to their superiors that "AF" was low on water. The Navy cryptanalysts now had their proof that "AF" was indeed Midway, and the US forces dispatched three aircraft carriers to the location. The result is that Midway was saved, and the Japanese incurred great

losses. It is even said that this battle was the turning point in the war by the US against Japan in the Pacific.Although here Japan didn't intend to give corresponding message ciphertext pairs CPA attack happened.

2.4 CPA vs COA multiple message security

Just to brush up memory in COA mult experiment the attacker would choose and send 2 message vectors of some fixed length. He would in return get a cipher text vector corresponding to one of the message vectors.To win the Attacker is required to tell which message vector does the cipher text corresponds to. For more details refer to Lecture 5.

Consider the following COA multiple message attack the attacker sends two message vectors of length $P_1(n) + 1 + P_2(n)$. The message vectors are such that the first $P_1(n)$ and last $P_2(n)$ are same where $P_1(n), P_2(n)$ are some polynomials.

$$M_0 = (m_1, m_2, ..., m_{P_1(n)}, M'_0, M_1, ..., M_{P_2(n)})$$

$$M_1 = (m_1, m_2, ..., m_{P_1(n)}, M'_1, M_1, ..., M_{P_2(n)})$$

The challenger generates a key k and chooses a bit b
Now attacker would be challenged by the challenger with the following cipher text vector c
$C = (Enc_k(m_1), .., Enc_k(m_{P_1(n)}), Enc_k(M_b), Enc_k(M_1), ..., Enc_k(M_{P_2(n)}))$

Since Attacker knows the corresponding message , cipher text pairs for the first $P_1(n)$ messages and last $P_2(n)$ messages. Are the message cipher text pairs equivalent to Pre training and post training phases of CPA attack? Are CPA and COA-mult equivalent?
The answer is a big NO. Because in CPA attack the adversary can adaptively make queiries to the encryption oracle whereas in COA-mult he has to choose the messages non adaptively.

2.5 CPA multiple message security

2.5.1 CPA Multi Message indistinguishability experiment $PrivK^{CPA-Mult}_{A,\Pi}(n)$

- The challenger generates a key K using $Gen(1^n)$ and gives adversary black box access to encryption oracle $Enc_k(.)$

- A queries the encryption oracle with message vectors of his choice to get corresponding cipher text vector polynomial number of times during training phases.
- A chooses and sends two message vectors to challenger.
- Challenger chooses a message vector by choosing a uniformly random bit b and sends the cipher text vector corresponding to this message vector to the attacker
- A can again opt for Post training similar to pre-training phase.

- A outputs a bit b'. He wins if $b = b'$, loses otherwise.
- The output of the experiment is 1 if A wins and 0 otherwise.

.

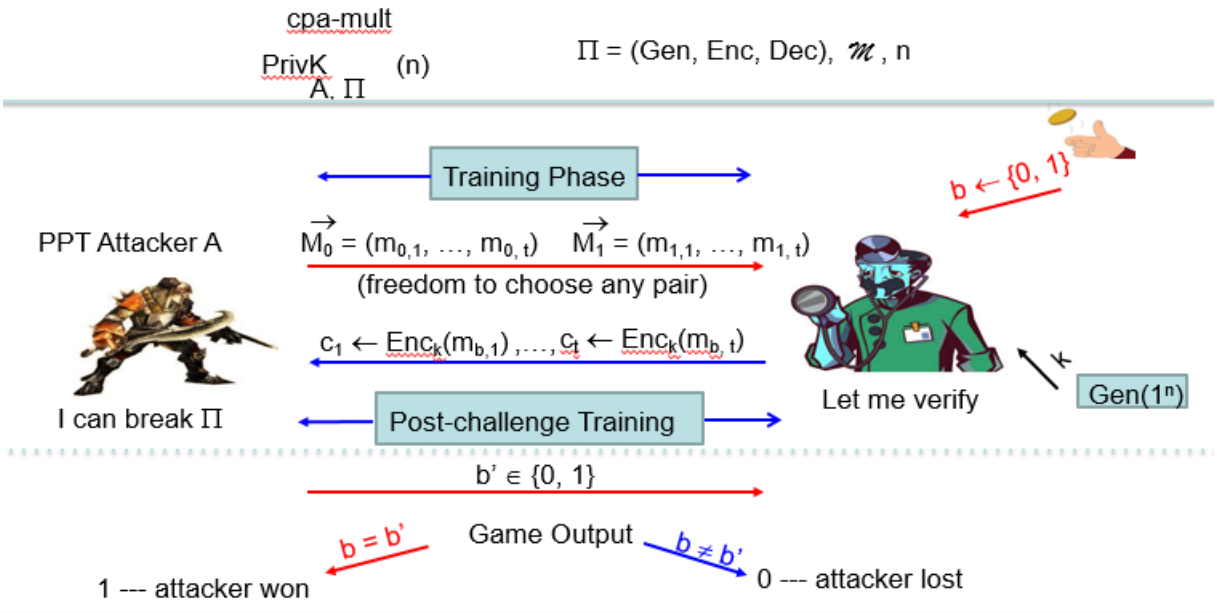

Figure 2 Indistinguishability experiment for CPA-multiple message security

2.5.2 Definition

A scheme Π is CPA-Multiple message secure if for all PPT adversaries A there exists a negligible function $negl(.)$ such that

$$Pr[PrivK_{A,\Pi}^{CPA-mult}(n) = 1] \leq \frac{1}{2} + negl(.)$$

where probability is over the random tosses of A and random coins used in the experiment.

2.5.3 CPA single message vs CPA multiple message security

If the length of message vectors in $PrivK_{A,\Pi}^{CPA-Mult}(n)$ is set to 1 then $PrivK_{A,\Pi}^{CPA-Mult}(n)$ reduces to $PrivK_{A,\Pi}^{CPA}(n)$
So any Π that is CPA-Multiple secure is CPA- secure also.
That converse is also true unlike COA security vs COA-multiple message security.
Any scheme Π that is CPA-secure is also CPA-multiple message secure. This is because of the randomization of encryption algorithm.

2.6 CPA security guarentee in practice

Ensures security against CPA even if multiple messages are encrypted using a single key and communicated.
CPA- security is the least that should be expected from a Secret Key Encryption(SKE)

Using a CPA-secure scheme we have overcome both drawbacks of Perfectly secure scheme i.e Key length as long as message and key re-usage.

2.7 Assumptions for a CPA-secure scheme

2.7.1 Can the encryption algorithm be deterministic for CPA-security?

For a deterministic algorithm for any given the output of the algorithm is always the same. If the encryption algorithm is deterministic adversary A would query the encryption oracle with the messages of his choice and gets the corresponding ciphertexts. And sends two such messages to the challenger and always wins by guessing the message corresponding to challenge cipher text correctly as the algorithm is deterministic and the cipher texts are known to him in Pre-training phase. Therefore, **For any CPA-secure scheme the encryption algorithm must be randomized.**

2.7.2 Need for randomness

So in order to have different cipher texts each time for same message, key pair we need fresh randomness in reach run of $Enc(.,.)$ algorithms.

Consider a truly random function $f : \{0,1\}^n \to \{0,1\}^n$ i.e for each $x \in \{0,1\}^n$ $f(x) = y \in_R \{0,1\}^n$
Each time the sender of Enc(.,.) algorithm needs fresh randomness an x is chosen such that $x \in_R \{0,1\}^n$ by flipping n coins then

$$Enc_f(m) = (x, m \oplus y)$$

Let $c = m \oplus y$

$$Dec_f(x, c) = c \oplus f(x)$$

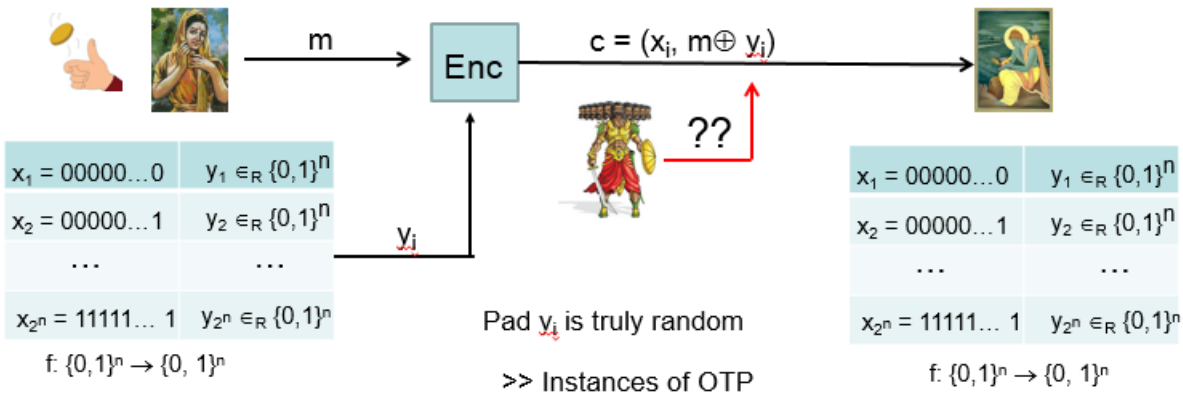

Figure 3 Truly random function

In the above scheme the function f acts as key. The key size i.e the function description size is $n.2^n$ which is a huge problem.

As randomness is a scarce resource and any true randomness has to come for physical process we cannot afford to have truly fresh randomness each time for enc.

2.8 Pseudorandom functions

In *Ö. Goldreich, S. Goldwasser and S. Micali. How to Construct Random Functions. JACM, 33(4), 792-807, 1986¨* designed Pseudo random functions that **looks like** a truly random function.

2.8.1 Truly Random functions

Let $Func_n = \{f : \{0,1\}^n \rightarrow \{0,1\}^n\}$
$|Func_n| = 2^{n.2^n}$
U is a uniform distribution over $Func_n$
A function chosen from $Func_n$ according to U is a Truly Random Function $f : \{0,1\}^n \rightarrow \{0,1\}^n$
The function description size of such truly random function is $n.2^n$.

2.8.2 Pseudo random function

A *Keyed function F* is a two input function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n\}$ where the first input is key k and second input is x. If the key k is fixed we have a function $\{f_k : \{0,1\}^n \rightarrow \{0,1\}^n\}$
A keyed function F is efficient if $F_k(x)$ is efficiently computable given k ,x by some deterministic polynomial time algorithm. From here on we are only interested in efficient keyed functions
Let $KFunc_n = \{F_k(.) : k \in \{0,1\}^n\}$ and $|KFunc_n| = 2^n$
A function sampled uniformly from $KFunc_n$ that behaves or looks like a Truly random function is called a **Pseudorandom function(PRF)**
Function description size of such a PRF is n bits because if the key k is given anyone who knows x can compute $F_k(x)$ efficiently.

2.8.3 Formulating Definition

Does the definition similar on the lines of PRG i.e The PPT attacker would be given a function and is asked to distinguish whether it is a PRF or TRF work? No because the function size is exponential PPT attacker can never even completely look at the function. Thus this kind of definition would be highly unjustified.

2.8.4 PRF Indistinguishability Game

- The challenger would choose a random bit b if b=1 $F_k \in_R KFunc_n$ else if b=0 $f \in_R Func_n$
- PPT attacker D would be given oracle access to f. D can query the oracle polynomial number of times adaptively.
- D outputs bit b'
- D wins if b=b', loses otherwise

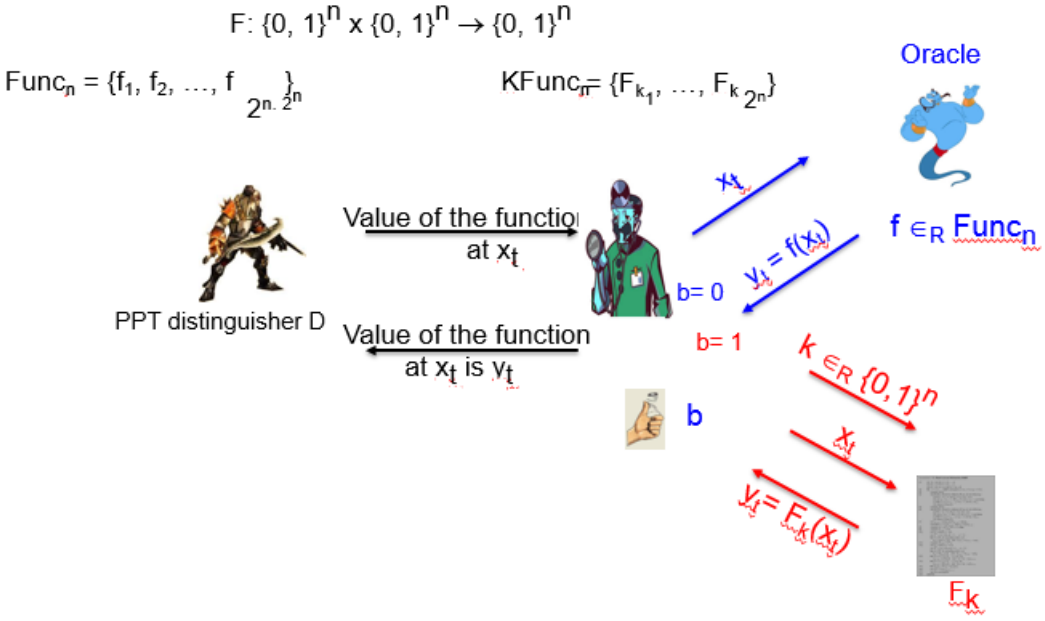

Figure 4 Indistinguishability experiment for PRF

2.8.5 Definition

Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n\}$ be an efficient, length preserving, keyed function. F is a PRF if for all PPT distinguishers D, there exists a negligible function negl(.) such that

$$|Pr[D^{F_k(.)}(1^n) - D^{f(.)}(1^n)]| \leq negl(n)$$

where $k \in_R \{0,1\}^n$ and $f \in_R Func_n$
Note: D is not given the key k as it is trivial to distinguish oracle for $F_k$from oracle for f. For example query the oracle at 0^n to obtain answer y. Compare y with $y' = F_k(0^n)$. $Pr[y = y'] = 2^{-n}$ for a random function oracle.

2.8.6 Lets try to construct a PRF

Consider the following length preserving keyed function $F(k, x) = k \oplus x$
For any x F(k,x) is uniformly random if k is uniformly random.

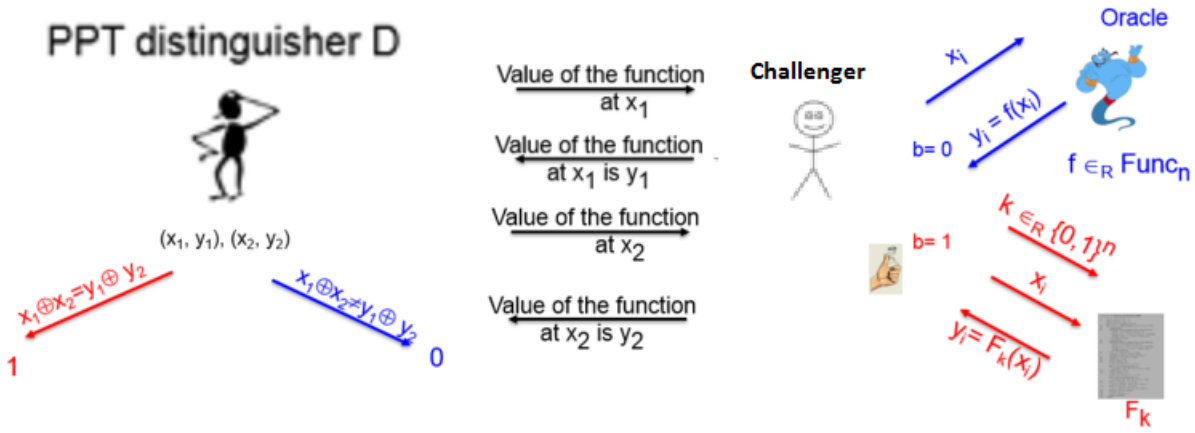

Figure 5 A faulty PRF construction

- The challenger chooses a PRF or TRF based on $b \in_R 0, 1$.
- D queries the oracle the oracle at any 2 points x_1, x_2 and receives y_1, y_2
- D outputs 1 if $x_1 \oplus y_1 = x_2 \oplus y_2$, 0 otherwise
- If it is an oracle oracle corresponding to PRF $Pr[D^{F_k(.)}(1^n)] = 1$
- If it a random function oracle $Pr[D^{f(.)}(1^n)] = 2^{-n}$
- $|Pr[D^{F_k(.)}(1^n) - D^{f(.)}(1^n)]| = 1 - 2^n$. Therefore F is not a PRF

2.8.7 Existence of PRF

No proof exists for existence of PRF's. But there exists efficient primitives called block ciphers that are belived to be PRF since no good distinguisher exists till now. PRF's exist only if PRGs exist. So PRF's can be constructed from any of the hard problems from which PRGs can be constructed. Now we have one more assumption to our assumption list that PRFs exist.

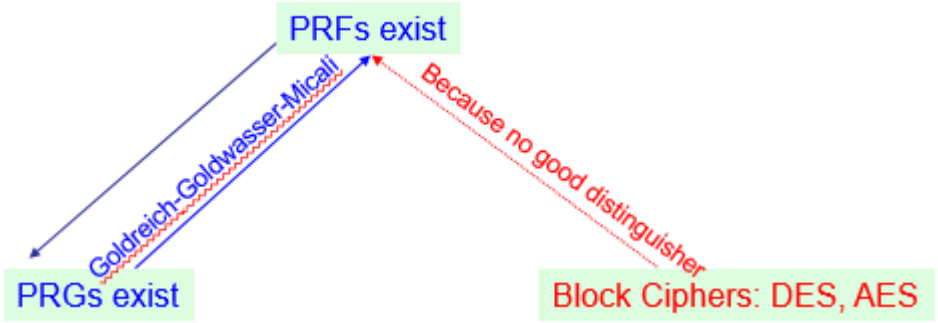

Figure 6 Implication between PRF,PRG,Block ciphers

2.9 Pseudo random permutations

Any function $f : \{0,1\}^n \to \{0,1\}^n$ is a permutation if it is bijective.
Just analogous to TRF and PRF
Let $Perm_n = \{$set of all permutation functions $f : \{0,1\}^n \to \{0,1\}^n\}and|Perm_n = 2^n!|$
$Kperm_n = \{$ set of keyed permutations $\}$and$|Kperm_n| = 2^n$
A keyed function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n\}$ is said to be keyed permutation if each $F_k : \{0,1\}^n \to \{0,1\}^n\}$ is a one-to-one(bijective if length preserving).
A keyed permutation F is efficient if $F_k(x)$ are efficiently computable by a deterministic polynomial time algorithms given x,k. An efiicient keyed permutation $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n\}$ is a Pseudo random permutation if for all PPT adversaries A there exists a negligible function negl(.) such that

$$|Pr[D^{F_k(.)}(1^n) - D^{f(.)}(1^n)]| < negl(.)$$

where $k \in_R \{0,1\}^n$ and $f \in_R Perm_n$

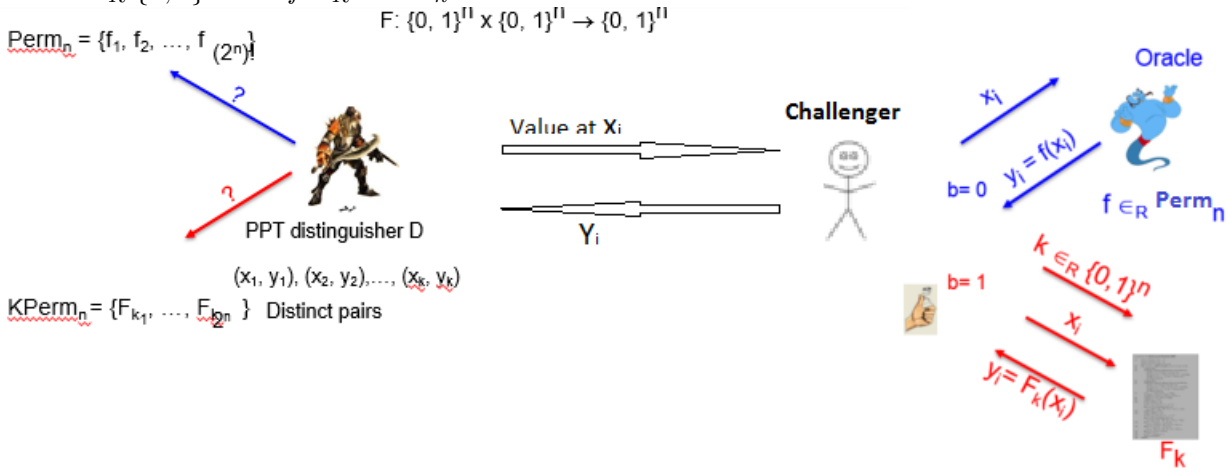

Figure 7 Indistinguishability experiment for PRP

- The challenger chooses a PRP or TRP based on $b \in_R 0, 1$.
- D queries the oracle the oracle at polynomial number of points $x_1..., x_t$ and receives $y_1, ..., y_t$
- D based on the above function values tries to distinguish PRP from TRP.

2.10 Strong Pseudo random permutations

A PRP F is efficient if $F_k^{-1}(x)$ are efficiently computable by a deterministic polynomial time algorithm given x,k. An efficient PRP $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n\}$ is a Strong Pseudo random permutation if for all PPT adversaries A there exists a negligible function negl(.) such that

$$|Pr[D^{F_k(.),F_k^{-1}(.)}(1^n) - D^{f(.),f^{-1}(.)}(1^n)]| < negl(.)$$

where $k \in_R \{0,1\}^n$ and $f \in_R Perm_n$

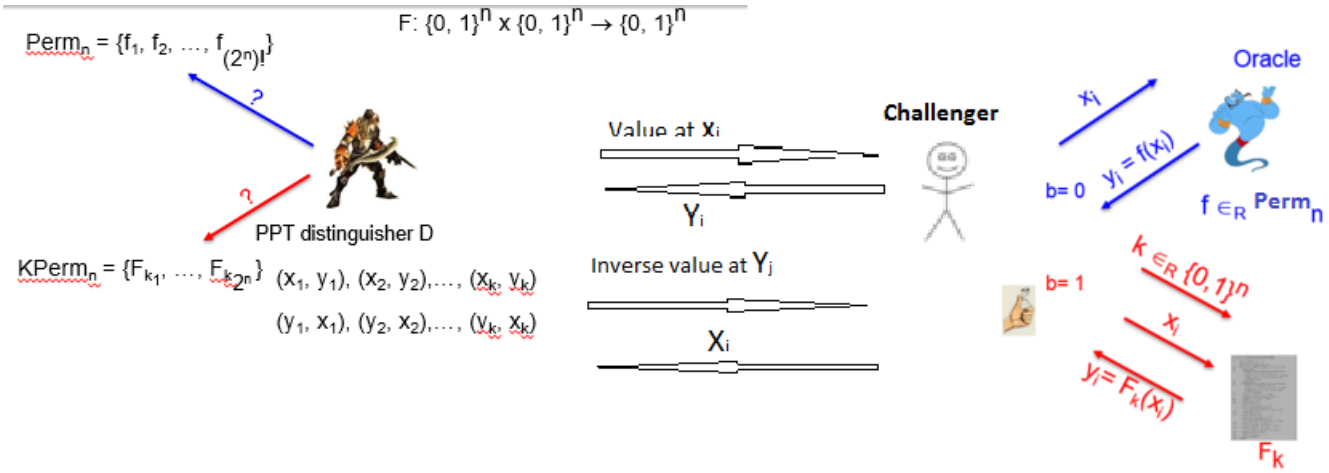

Figure 8 Indistinguishability experiment for SPRP

- The challenger chooses a SPRP or TRP based on $b \in_R 0, 1$.
- D queries the oracle the oracle at polynomial number of points $x_1..., x_t$ and receives $y_1, ..., y_t$, $y_1, ..., y_l$ to receive $x_1..., x_l$
- D based on the above function values tries to distinguish SPRP from TRP.

  Any SPRP is by default a PRP but the converse is not always true.

2.10.1 Existence of SPRP and PRP

There is no concrete throritical proof that PRP's and SPRP's exist. But they are strongly believed to exist.Block ciphers are believed to be PRP and SPRP as no strong distinguisher exists. In later lectures we will see constructions of PRP, SPRP based on PRF's.

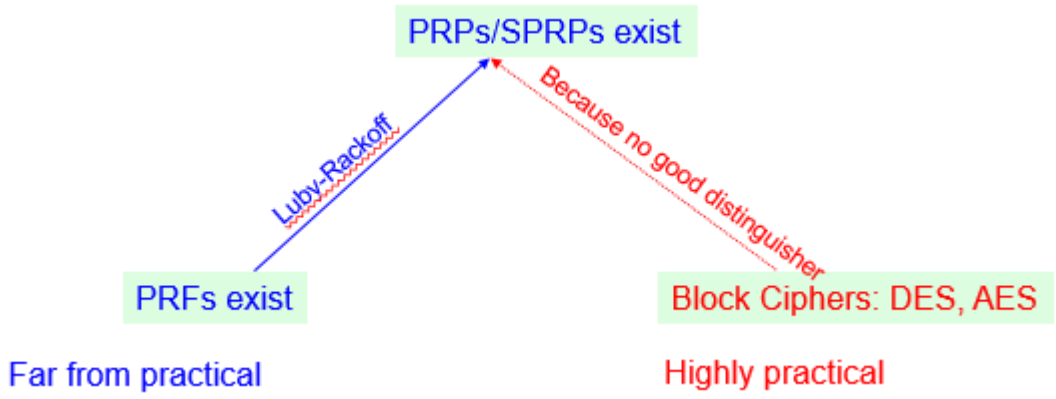

Figure 9 Implications between PRF, PRP/SPRP, Block ciphers

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.