Pseudo-random Functions

Debdeep Mukhopadhyay
IIT Kharagpur

PRG vs PRF

- We have seen the construction of PRG (pseudo-random generators) being constructed from any one-way functions.
- Now we shall consider a related concept:
  - Pseudo-random functions
  - instead of strings we consider functions
- It does not make much sense to call a fixed function pseudo-random.

Keyed Functions

- So, we have keyed functions.
- A keyed function $F:\{0,1\}^{*}x\{0,1\}^{*} \rightarrow \{0,1\}^{*}$
- The first input is called the key.
- The key is chosen randomly and then fixed, resulting in a single argument function, $F_k: \{0,1\}^{*} \rightarrow \{0,1\}^{*}$
- Assume that the functions are length preserving, meaning that the inputs, output and key are all of the same size.

Pseudo-random functions

- No **polynomial time** adversary should be able to distinguish whether it is **interacting** with:
  - F_k (for a randomly chosen k) or,
  - f (where f is chosen at random from the set of all functions mapping n bit strings to n bit strings).

Cardinality of all possible functions

- Set of all keyed functions:
  - The former is chosen from a distribution over at most 2^n distinct functions.
- Set of all possible random functions:
  - The later is from 2^{n2^n} functions.
- Despite this, the behavior of the functions must look the same to a PPT adversary.

Formally

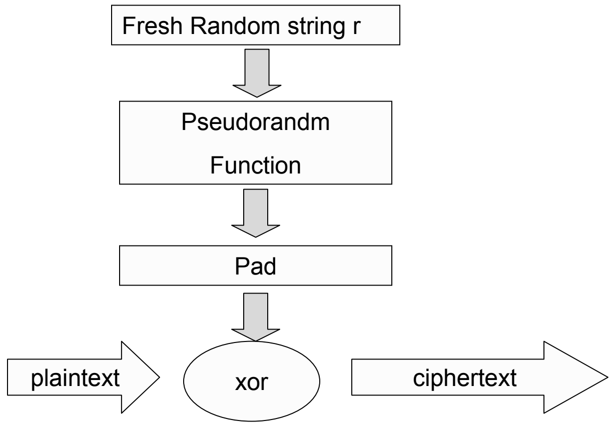
Let $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficient length preserving, keyed function.

F is said to be pseudo-random function if for all probabilistic polynomial time distinguisher D, there exists negligible function ε(n):

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

where k is chosen uniformly at random and f is chosen uniformly at random from the set of functions mapping n-bit strings to n-bit strings.

Encryption with a PRF



Some finer points

- If x and x' differ, outputs of $F_k(x)$ and $F_k(x')$ should not be correlated.
- Distinguisher D is not given the key:
  - it is meaningless to talk about pseudorandomness once the key is given.
  - one can compute $y' = F_k(0^n)$
  - then query the oracle at 0^n
  - if the oracle is for F_k, always $y = y'$
  - if the oracle is for random f, $y = y'$ with a probability of 2^{-n}. thus we have a distinguisher.

Security against CPA

- **Defn:** An adversary, A, should not be able to distinguish the encryptions of two arbitrary messages.

CPA Ind Exp

Experiment: $\text{Priv}^{\text{CPA}}_{\text{A},\Pi}(n)$

1. A key is generated by running Gen(n)
2. Adversary A is given n and oracle access to $\text{Enc}_{\text{k}}(.)$,
and outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $\text{b} \in \{0,1\}$ is chosen, and a ciphertext $\text{c=Enc}_{\text{k}}(m_b)$ is computed and given to A as a challenge. We call c the challenge ciphertext.
4. Adversary A continues to have oracle access to $\text{Enc}_{\text{k}}(.)$ and outputs a bit b'.
5. Output of the experiment is 1, if b'=b, and 0 otherwise.

A succeeds when $\text{Priv}^{CPA}_{\text{A},\Pi}(n) = 1$

Definition of Indistinguishable under CPA

Any encryption scheme Π=(Gen,Enc,Dec) has indistinguishable encryptions under CPA (called CPA-secure) is for all PPT adversary A, there exists a negligible ε(n) st.,

$$\Pr[\mathrm{Priv}_{\mathrm{A},\Pi}^{CPA}(n)=1] \le \frac{1}{2} + \varepsilon(\mathrm{n})$$

where the probabilities are taken over the random coins used by A, as well as the random coins used in the experiment.

CPA secured encryption

- the scheme has to be probabilistic:
  - consider a deterministic encryption: $ENC_k(m)=F_k(m)$
  - Given $c=ENC_k(m_b)$ it is possible to ask for $ENC_k(m_0)$ and $ENC_k(m_1)$ and see for a match. Accordingly b is discovered easily.
  - thus the scheme is not CPA secured.

A CPA secure encryption scheme from any PRF

Let F be a PRF. Define an encryption as follows:

1. Gen: on input n (security parameter), choose $k \leftarrow \{0,1\}^n$ uniformly at random as the key.
2. Enc: on input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ uniformly at random and output the ciphertext:

$$c = \langle r, F_k(r) \oplus m \rangle$$

3.Dec: On input a key k and a ciphertext <r,s>:

$$m = F_k(r) \oplus s$$

Theorem

If F is a pseudorandom function, then the above construction is a fixed length symmetric key scheme for messages of length n that has indistinguishable encryptions under a chosen plaintext attack.

Proof

- Follows a general principle.
- Prove that the system is secured when a truly random function is used.
- Next prove that if the system was insecure when the pseudorandom function was used, then we can make a distinguisher against the PRF.

Proof

Let $\widetilde{\Pi}=(\widetilde{\text{Gen}},\widetilde{Enc},\widetilde{Dec})$ be an encryption scheme that is exactly the same as Π=(Gen,Enc,Dec), except that a true random function f is used in place of F_k.

Thus $\widetilde{\text{Gen}}(n)$ chooses a random function f $\leftarrow \text{Func}_n$ and $\widetilde{Enc}$ just like Enc except that f is used instead of F_k.

Claim: For every adversary A that makes at most q(n) queries to its encryption oracle:

$$\Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1] \le \frac{1}{2} + \frac{q(n)}{2^n}$$

Proof: Each time a message m is encrypted a random $r \leftarrow \{0,1\}^n$ is chosen and the ciphertext is $\{r, m \oplus f(r)\}$

Let r_c be the random string used when generating the challenge ciphertext $c=<r_c, f(r_c) \oplus m>$.

Define, Repeat as the event that r_c is used by the encryption oracle to answer at least one of A's queries.

Clearly, $\Pr[\mathrm{Repeat}] \le \frac{q(n)}{2^n}$

Also, $\Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1 \mid \overline{\mathrm{Repeat}}] = \frac{1}{2}$.

$\therefore \Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1] = \Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1 \wedge \mathrm{Repeat}] + \Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1 \wedge \overline{\mathrm{Repeat}}]$

$\le \Pr[\mathrm{Repeat}] + \Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\tilde{\Pi}}(n)=1 \mid \overline{\mathrm{Repeat}}] = \frac{1}{2} + \frac{q(n)}{2^n}$

Construct a Distinguisher for the PRF

Let $\Pr[\mathrm{Priv}^{\mathrm{CPA}}_{\mathrm{A},\Pi}(n)=1] \ge \frac{1}{2} + \varepsilon(n)$

If ε is not negligible then the difference between this is also non-negigible. Such a gap will enable us to distinguish the PRF from a true random function.

Distinguisher D:

D is given input n and oracle $O:\{0,1\}^n \to \{0,1\}^n$.

D answers the queries made by A in the CPA IND EXP.

1. Run A(n). Whenever A queries its encryption oracle on a message m, answer this query in the following way:
   a) Choose $r \leftarrow \{0,1\}^n$ uniformly at random.
   b) Query O(r) and obtain response s'
   c) Return to A the ciphertext $<r, s' \oplus m>$
2. When A outputs $m_0, m_1 \in \{0,1\}^n$, choose a random bit $b \leftarrow \{0,1\}$.
   a) Choose $r \leftarrow \{0,1\}^n$ uniformly at random.
   b) Query O(r) and obtain response s'
   c) Return to A the ciphertext $<r, s' \oplus m_b>$
3. Continue answering A's queries as above. When A outputs a bit b', D outputs 1 if b=b' and 0 otherwise.

1. If D's oracle is a PRF, then the view of A when run as a sub-routine by D is distributed identically to the view of A in experiment $\mathrm{Priv}^{CPA}_{A,\Pi}(n)$.

Thus, $\Pr[D^{F_k}(n)=1] = \Pr[\mathrm{Priv}^{CPA}_{A,\Pi}(n)=1]$.

2.If D's oracle is a random function, then the view of A when run as a sub-routine by D is distributed identically to the view of A in experiment $\mathrm{Priv}^{CPA}_{A,\tilde{\Pi}}(n)$.

Thus, $\Pr[D^{f}(n)=1] = \Pr[\mathrm{Priv}^{CPA}_{A,\tilde{\Pi}}(n)=1]$.

Thus, $\Pr[D^{F_k}(n)=1] - \Pr[D^{f}(n)=1] \geq \varepsilon(n) - \frac{q(n)}{2^n}$,

which is non-negligible if ε(n) is so.

This violates the PRF property of the F_k.

A CPA-secured scheme for messages of arbitrary length

Consider, $m = m_1 m_2 ... m_l$, each m_i is an n-bit block.
The ciphertext is:

$$< r_1, F_k(r_1) \oplus m_1, r_2, F_k(r_2) \oplus m_2, ..., r_2, F_k(r_l) \oplus m_l >$$

Corollary:

If F is a pseudorandom function, then the scheme above is a private-key encryption scheme for arbitrary message that has indistinguishable encryptions under a chosen-plaintext attack.

Pseudo-random Permutations and Block Ciphers

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length preserving, keyed function.
It is called a keyed *permutation* if for every key, the function, F_k is one-one.
Since the function is length preserving, it is also a bijection, and hence an inverse permutation exists.
We call it F_k^{-1}.
The keyed permutation is efficient if given k and x, it is easy to compute both $F_k(x)$ and $F_k^{-1}(x)$.

Randomly chosen permutations and randomly chosen functions are not distinguishable by polynomial queries

If F is a pseudorandom permutation then it is also a pseudorandom function.

Pseudorandom Permutation

- It is also a permutation.
- Moreover there exists an efficient inverse, P_K^{-1}.
- A pseudorandom permutation is also a pseudorandom function.
- Strong pseudorandom permutation: No efficient algorithm A can distinguish well between $\langle P_K(.), P_K^{-1}(.)\rangle$ from $\langle \Pi(.), \Pi^{-1}(.)\rangle$ for a randomly chosen key and random permutation, Π.

$$A^{P_K(.),P_K^{-1}(.)} \text{ behaves like } A^{\Pi(.),\Pi^{-1}(.)}$$

Building Pseudorandom Permutations

- We can build pseudorandom permutations from pseudorandom functions, F
- Define $\boxed{D_F(x,y) = y, F(y) \oplus x}$
- Note that this is injective and that does not depend whether F is injective or not.
- Note that D_F and D_F^{-1} are efficiently computable.
- This construction was originally due to Horst Feistel.

Strong Pseudorandom Permutations

Let $F:\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, keyed permutation. We say that F is a strong pseudorandom permutation if for all probabilistic polynomial time distinguishers $D,$ there exists a negligible function $negl$ such that:

$$|\Pr[D^{F_k(.),F_k^{-1}(.)}(n)=1] - \Pr[D^{f(.),f^{-1}(.)}(\mathrm{n})=1] \le \varepsilon(\mathrm{n}),$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f is chosen uniformly from the set of permutations on $n-$bit strings.

Block Ciphers

- The analogue for strong pseudorandom permutations is block ciphers.
- **Note:** Block ciphers themselves are not secured encryption schemes.

  $c=F_k(m)$ is not CPA secured (Why?)

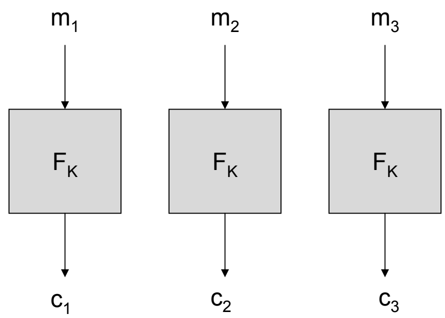
  So, block ciphers are building blocks for efficient encryption schemes and not encryption schemes by themselves.

Modes of Operations of block ciphers

- These are ways of encrypting arbitrary length messages using a block cipher.
- The difference between the ciphertext length and the message length is small in this case.
- It may be noted, that messages of arbitrary length can be padded so that they are multiples of the block length, n.
- Since this can be done without any ambiguity, we assume that the messages are made of l blocks, each of length n.

Modes of Encryption

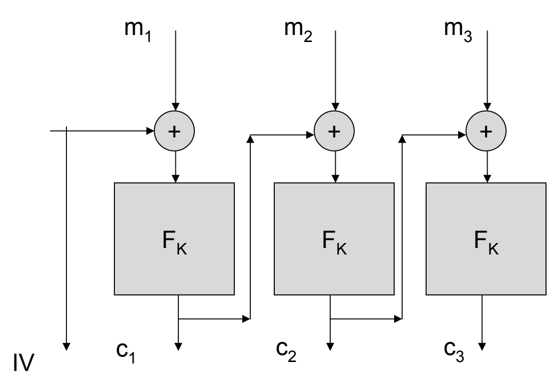
- Electronic Code Book (ECB)



Deterministic encryption and thus cannot be CPA-secure.

Not message indistinguishable either.

Cipher Block Chaining (CBC)

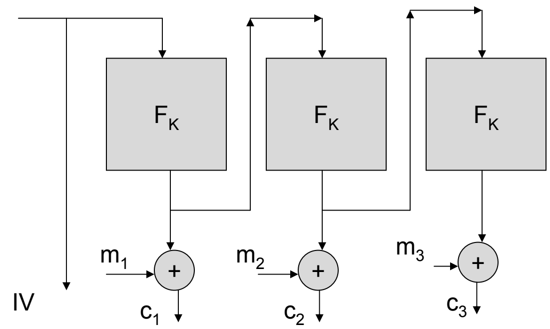


Parallelization not possible.

A random IV (initial vector) of size n bits is chosen. IV is sent in the clear for decryption.

Probabilistic and if F is a pseudo-random permutation then CBC is CPA-secure.

Output Feedback Mode (OFB)



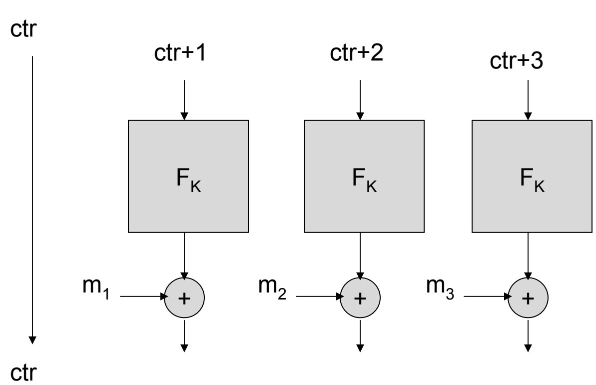
If F is a Pseudorandom function then this is secure against CPA.

Note that F need not be a permutation.

Parallelism not possible.

But pre-processing of the key stream can lead to extremely fast operations.

Counter Mode



Theorem

If F is a pseudo-random function, then randomized counter mode has indistinguishable encryptions under a chosen-plaintext attack (CPA).

Proof Idea

First consider that a truly random function, f, is used.
Let ctr* denote the initial value ctr, when the challenge ciphertext is generated in the experiment Priv^{cpa}.
For the i^{th} block of the message, thus ctr*+i was used to generate f(ctr*+i).
Now, if ctr*+i was never accessed before, then the key stream is random and like a one time pad. Thus the adversary has no advantage in deciding whether m_0 or m_1 was the corresponding plaintext for the challenge ciphertext.
So, we have to find what is the probability that ctr*+i was actually "matches" with one of the queries of the adversary A.

Proof Idea

The adversary A makes q(n) queries. The starting IV value for the ith query is denoted by ctr_i. Let each message be of block-length, q(n).

We divide the entire scenario into two mutually exclusive cases:

1. There do not exist any i, j, j' for which $ctr^*+j = ctr_i + j'$.

$Here: \Pr[\mathrm{Priv}_{A,\Pi}^{CPA}=1] = \frac{1}{2}.$

2. There exists i,j,j' for which $ctr^*+j = ctr_i + j'$.

In this case, A can easily determine $f(ctr^*+j)=f(ctr_i+j')$ and thus compute m_j. Thus he can predict whether m_0 or m_1 was encrypted.

Let $\mathrm{Overlap}_i$ denote the even that the sequence $ctr_i+1,\dots,ctr_i+q(n)$ overlaps the sequence $ctr^*+1,\dots,ctr^*+q(n)$.

Consider, $ctr^*+1,\dots,ctr^*+q(n)$

$ctr_i+1,\dots,ctr_i+q(n)$

$Overlap_i$ occurs when $ctr_i+1 \le ctr^*+q(n)$ and

when $ctr_i+q(n) \ge ctr^*+1$

This happens when: $ctr^*+1-q(n) \le ctr_i \le ctr^*+q(n)-1$

Proof

We define the event Overlap, as when $\mathrm{Overlap}_i$ occurs for any i,

that is: $\Pr[\mathrm{Overlap}] \le \sum_{i=1}^{q(n)} \Pr[\mathrm{Overlap}_i]$

Now, $\Pr[\mathrm{Overlap}_i] = \frac{2q(n)-1}{2^n} \Rightarrow \Pr[\mathrm{Overlap}] \le \frac{2q(n)^2}{2^n}.$

$$\Pr[\mathrm{Priv}_{A,\Pi}^{CPA}=1] \le \Pr[Overlap] + \Pr[\mathrm{Priv}_{A,\Pi}^{CPA}=1 \mid \overline{Overlap}]$$

$$= \frac{2q(n)^2}{2^n} + \frac{1}{2}$$

The next step is to reason that if the random function is replaced by the pseudo-random function, and the scheme is not CPA-secure, then we can frame a PPT algorithm D, which is able to distinguish the function F_k from a random function f. This proof is left as an exercise.

Block length and security

- Interestingly, we see that it is not only the key length but the block length also which decides the security.
- Consider a block length of 64 bits.
- The adversary's success probability in the CPA sense is thus around $\frac{1}{2} + q^2/2^{63}$. Thus if we have around 2^{30} guesses, then we have a practical attack! (only 1 GB queries and storage required).
- So, we need to increase the block length.