CSA E0 235: Cryptography 3rd February 2016

Scribe for Lecture #7

Instructor: Arpita Patra *Submitted by: Aditi Rai*

1 Recall

In last lecture we discussed new definitions for secret key encryption (SKE), these definitions were: choosen plaintext attack (CPA), CPA security and CPA-mult-security. Also, we learned to use the notion of pseudorandomness in CPA-security and in this process we defined Pseudorandom functions, permutations (PRF,PRP) and strong PRP, also modelled PRF as an indistinguishability game.

In today's lecture we will construct a CPA-secure scheme from PRF and also give a proof of security of such a scheme.Later we will look at practical CPA-secure schemes from PRF/PRP/SPRP.

2 Encryption using PRFs

2.1 Pseudo Random Functions (PRFs)

Intutively a pseudorandom funtion (PRF) is a function wjose output begaviour looks like that of a TRF for an observer who is computationally bounded.This can be formally stated as:

$$\mid \mathsf{Pr}\left[D^{F^{(k)}}(1^n)=1\right]-\mathsf{Pr}\left[D^{f^{(\cdot)}}(1^n)=1\right]\mid\leq negl(n)$$

where the first probability is taken over uniform choice of $k \in \{0,1\}^n$ and the randomness of D, and the second probability is taken over uniform choice of $f \in Func_n$ and the randomness of D.

2.2 PRF based fixed length CPA-secure scheme

Let F be a pseudorandom function. We define a private-key encryption scheme $\Pi\,(Gen, Enc, Dec)$ for message length n as follows:

- Gen: generates uniform key $k \in \{0,1\}^n$ on 1^n as input
- Enc: takes a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$ as output, outputs ciphertext c by choosing $r \in \{0,1\}^n$ as

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

- Dec: takes a key $k \in \{0,1\}^n$ and a ciphertext $c = (c_0, c_1)$ as input and outputs the plaintext message as

$$m = (F_k(c_0) \oplus c_1(r))$$

Theorem 1 *If F_k is a PRF, then Π is a CPA-secure scheme.*

Proof We prove the theorem using proof by reduction. Assume Π is not a CPA secure scheme. Then there exists an adversary A_{CPA}, who can break the scheme Π with some non-negligible probability. Thus, there exists a polynomial *p(n)* such that

$$\mathsf{Pr}\left[\ \mathsf{PrivK}^{\mathsf{cpa}}_{\Pi,A_{CPA}=1}\right] < \tfrac{1}{2} + \tfrac{1}{p(n)}$$

Now we want to construct a good distinguisher who can distinguish F_k from a TRF with the help A_{CPA} such that

$$\mid \mathsf{Pr}\left[D^{F^{(k)}}(1^n) = 1\right] - \mathsf{Pr}\left[D^{f^{(.)}}(1^n) = 1\right] \mid > \tfrac{1}{q'(n)}$$

where q(n) is a polynomial and this will lead to a contradiction showing that the assumption was wrong. D represents the distinguisher who has given oracle access to some function and its goal now is to determine whether the function is PRF or TRF. The indistinguishability experiment emulated by D for A_{CPA} can be explained as:

- **Training Phase:** A_{CPA} submits its plaintext messages m to D. D picks a random string r $\in_R$ $(\{0,1\})^n$ and submits it to oracle 'O'. 'O' returns the value of function f say y. D then returns ciphertext, $c := (r, (m \oplus y))$ to A_{CPA}.
- **Challenge Phase:** A_{CPA} submits m_0 and m_1 to D . These are the challenge plaintexts and A_{CPA} can submit any message of its choice, the messages can be one of the messages A_{CPA} has already queried during training phase. D picks a random string r* $\in_R$ $(\{0,1\})^n$ and forwards it to D. Oracle returns the value of function f say y* to $A_{PRF}.A_{PRF}$ flips a coin and choose a bit b $\in_R$ $\{0,1\}$ and then returns ciphertext $c := (r*, (m_b \oplus y*))$ to A_{CPA}
- **Post Challenge Phase:** This phase is same as training phase.
- **Response Phase:** A_{CPA} finally submits a bit, *b'* as its guess regarding the encrypted challenge plaintext. If $b' = b$, then D outputs 1 else outputs 0.

The points to be noted here are:

- If D's oracle is a PRF, then the experiment emulated by D for A_{CPA} is identical to the experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\Pi,A_{CPA}}(n)$ as the key k is chosen at random and then every encryption is carried out by choosing random r, computing $y := F_k(r)$.This gives us,

  $$\mathsf{Pr}\left[D^{F^{(k)}}(1^n) = 1\right] = \ \mathsf{PrivK}^{\mathsf{cpa}}_{\Pi,A_{CPA}}(n) = 1,$$

  where key $k \in_R (\{0,1\})^n$

- If D's oracle is a truly random function, then an experiment $\widetilde{\Pi} = (\widetilde{Gen}, \widetilde{Enc}, \widetilde{Dec})$ can be looked at to understand the view of D in this case. The scheme $\widetilde{\Pi} = (\widetilde{Gen}, \widetilde{Enc}, \widetilde{Dec})$, the only difference between $\widetilde{\Pi}$ and Π is that a truly random function is used in encryption in place of F_k. Let Repeat denotes the event where the random string r* chosen to generate challenge ciphertext c, is queried somewhere in Challenge or Post-challenge phase.

    - Case 1: $r* = r$, chosen for some query in the training phase. In this case A obtains both $c := (r, (m \oplus f(r)))$,$c := (r*, (m_b \oplus f(r*)))$ as ciphertexts to m and m_b respectively. XORing these ciphertext will give the value $m + m_b$ and thus A_{CPA} can distinguish between m_0 and m_1 and can succeed in the expreiment with the probability 1.

        $$\text{Pr(Repeat)} \leq \tfrac{q(n)}{2^n}$$

        Since A_{CPA} can make at most q(n) queries and since r^* is chosen uniformly from $\{0, 1\}^n$.

    - Case 2: $r* \neq r$ for any query in the training phase.Here A learns nothing about the value of $f(r*)$ from its interaction with encryption oracle and A_{CPA} can succeed in the experiment only by guessing b, which happens with probability 1/2.

Now consider the probability that A_{CPA} succeeds in the experiment $\widetilde{\Pi}$

$$\begin{aligned}
Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1\right] &= Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1 \wedge Repeat\right] + \\
&\quad Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1 \wedge \overline{Repeat}\right] \\
&= Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1 \mid Repeat\right].Pr\left[Repeat\right] + \\
&\quad Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1 \mid \overline{Repeat}\right].Pr\left[\overline{Repeat}\right] \\
&\leq Pr\left[Repeat\right] + \left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n) = 1 \mid Repeat\right] \\
&\leq \frac{q(n)}{2^n} + \frac{1}{2}
\end{aligned}$$

Now calculating the probability with which D can win PRF inddinguishability experiment:

$$
\begin{aligned}
&\mid Pr\left[D^{F^{(k)}}(1^n)=1\right]-Pr\left[D^{f^{(.)}}(1^n)=1\right]\mid \\
&\qquad =\mid Pr\left[PrivK^{cpa}_{\Pi,A_{CPA}=1}\right]-Pr\left[PrivK^{cpa}_{A_{CPA},\widetilde{\Pi}}(n)=1\right]\mid \\
&\qquad =\Big(\frac{1}{2}+\frac{1}{p(n)}\Big)-\Big(\frac{q(n)}{2^n}+\frac{1}{2}\Big) \\
&\qquad =\frac{1}{p(n)}-\frac{q(n)}{2^n} \\
&\qquad >\frac{1}{q'(n)}
\end{aligned}
$$

But this contradicts our assumption that F_k is a pseudo-random function.Hence proved.

■

3 CPA-security for arbitrary length messages

Let $\Pi = (Gen, Enc, Dec)$ be a fixed-length CPA secure based on PRF/SPRP/PRF, and it encrypts messages of size n. Now to encrypt a message m of length $t.n$, a theoretical construction can be described as, we will first divide m into t n-length blocks and then encrypt each block using scheme Π. The ciphertext is obtained by appending the ciphertexts obtained from encryption of each block.

3.1 Block-cipher Modes of Operations

We are given a length-preserving block cipher F (may be a PRF/PRP/SPRP) with block length, n. It takes a key $k \in \{0,1\}^n$ and a value $x \in \{0,1\}^n$ as input and outputs $F_k(x) = F(k,x) \in \{0,1\}^n$ and our goal is to encrypt a message $m = m_1m_2....m_t$ (Without loss of generality $m_i \in \{0,1\}^n$ using F with ciphertext length as small as possible and with least randomness.The different modes that are explained in this section are:

- Electronic Code Block (ECB) Mode
- Cipher Block Chaining (CBC) Mode
- Output Feedback (OFB) Mode
- Counter (CTR) Mode

3.1.1 Electronic Code Book (ECB) Mode

Given a message $m = m_1m_2...m_t$ and a SPRP F_k, ECB encodes m as:

$$c_i = F_k(m_i)$$

And the decryption is carried out as:

$$m_i = F_k^{-1}(c_i)$$

For decryption to be possible, we need F_k^{-1} to be efficiently computable and thus we need F_k to be SPRP.The scheme is deterministic and is not CPA-secure.For example, lets take two messages $m_0 = aa$ and $m_1 = ab$, the corresponding cipher-text will be $c_1 = c_a c_a$ and $c_2 = c_a c_b$. Just by looking at the ciphertext returned by *Challenger*, the *Attacker* can guess the bit b with probability equals to 1.

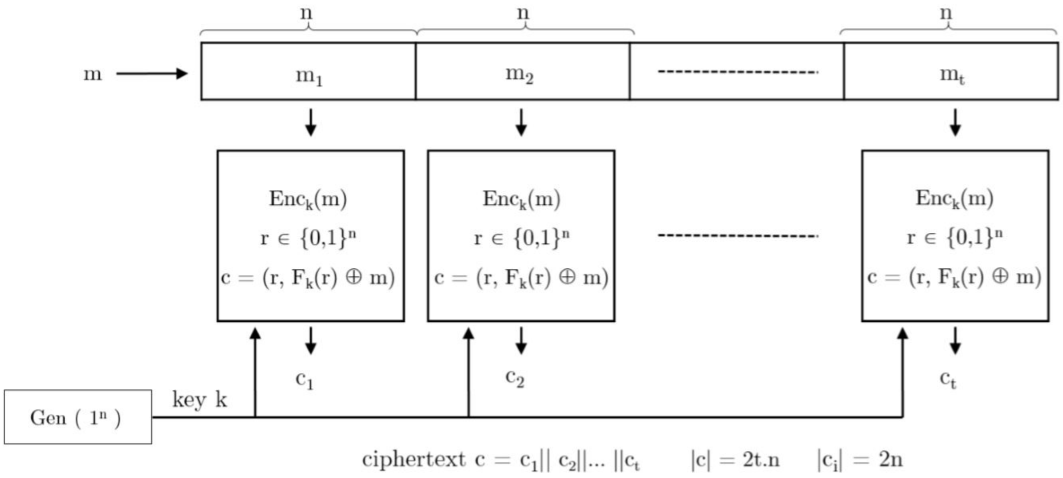

Fig 3a. Electronic Code Book (ECB) Mode

3.1.2 Cipher Block Chaining (CBC) Mode

The scheme used in this mode is non-deterministic as an *Initializtion Vector*, IV, of block length, n is chosen uniformly at random from $\{0,1\}^n$, and the encryption for message $m = m_1 m_2 .... m_t$ is carried out as:

$$c_0 = IV$$
$$c_i = F_k(m_i \oplus c_{i-1}) \ \ for \ \ 1 \leq i \leq t$$

The resulting ciphertext will be $c = c_0 c_1 ... c_t$ which will be of length (t+1)n. The decryption of ith block requires c_i and c_{i-1} as:

$$m_i = c_{i-1} \oplus F_k(c_i)$$

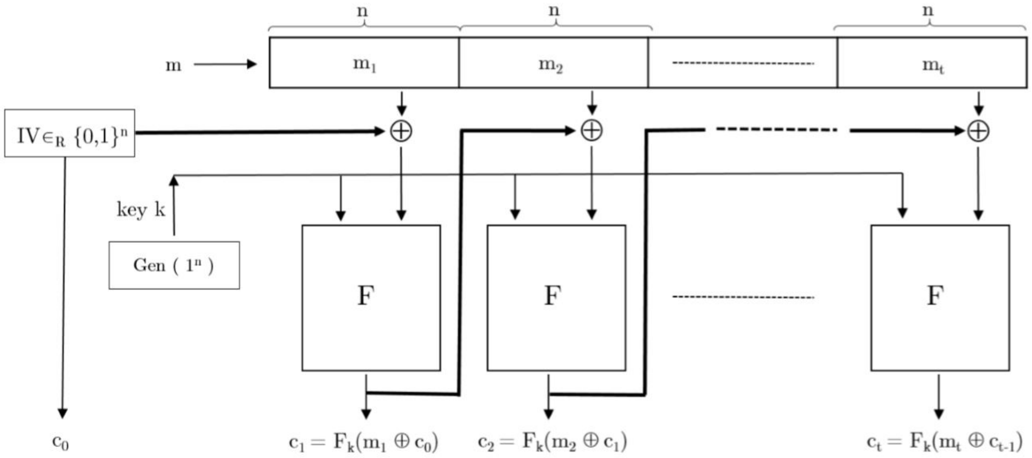

Fig 3b. Cipher Block Chaining(CBC) Mode

The CBC scheme explained above is IND-CPA secure but cannot carry out parallel encryption as each block has to wait for encryption of previous block. There exists a variant of CBC mode known as Chained CBC mode.

Chained CBC Mode

In chained CBC mode the last block of previous iphetext is used as the IV for the encrptio of the next message. This helps in reduction of bandwith as we are not sending IV each time. Say the first message $m = m_1m_2...m_t$is encrypted using some random IV, and then the next message $m' = m'_1m'_2...m'_t$ is encrypted using c_t as IV. The flaws in this scheme is that it is vulnerable to chosen-plaintext attack as the adversary knows in advance the IV for next message.

3.1.3 Output Feedback (OFB) Mode

In this mode, an *Initialization Vector*, IV, of block length n is chosen uniformly at random from $\{0, 1\}^n$ and a pseudorandom stream of y_i is generated using PRF's. The encryption of the message $m = m_1m_2...m_t$ is carried out as

$$\begin{aligned} y_0 &= IV \\ y_i &= F_k(y_{i-1}) \\ c_i &= y_i \oplus m_i \end{aligned}$$

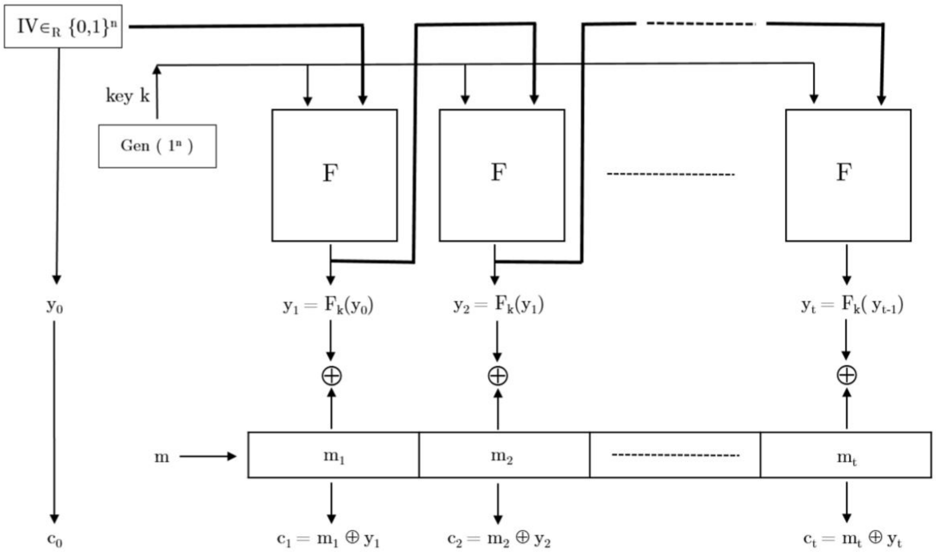

Fig 3c. Output Feedback (OFB) Mode

The psedorandom stream of pad is independent of *m*. So the scheme may not be parallalizable (since each block is dependent on the last) but, is pre-computable as the y_i are independent of *m*. This scheme is CPA secure.

3.1.4 Counter CTR Mode

This scheme uses a counter, CTR which is basically an *Initialization Vector* chosen uniformly at random from $\{0,1\}^n$. Then, a stream $y_i = F_k(CTR + i)$ is generated which is used in the encryption of $m = m_1 m_2 ... m_t$ as follows:

$$c_0 = IV(or\ CTR)$$
$$c_i = m_i \oplus y_i \ \ for\ \ 1 \leq i \leq t$$

the ciphertext in this scheme will be given as $c = c_0 c_1 .... c_t$ hich will be of length *(t+1)n*.

The decryption does not require F to be invertible, or even a permutation.But if F is PRF, then CTR mode can be CPA-secure.

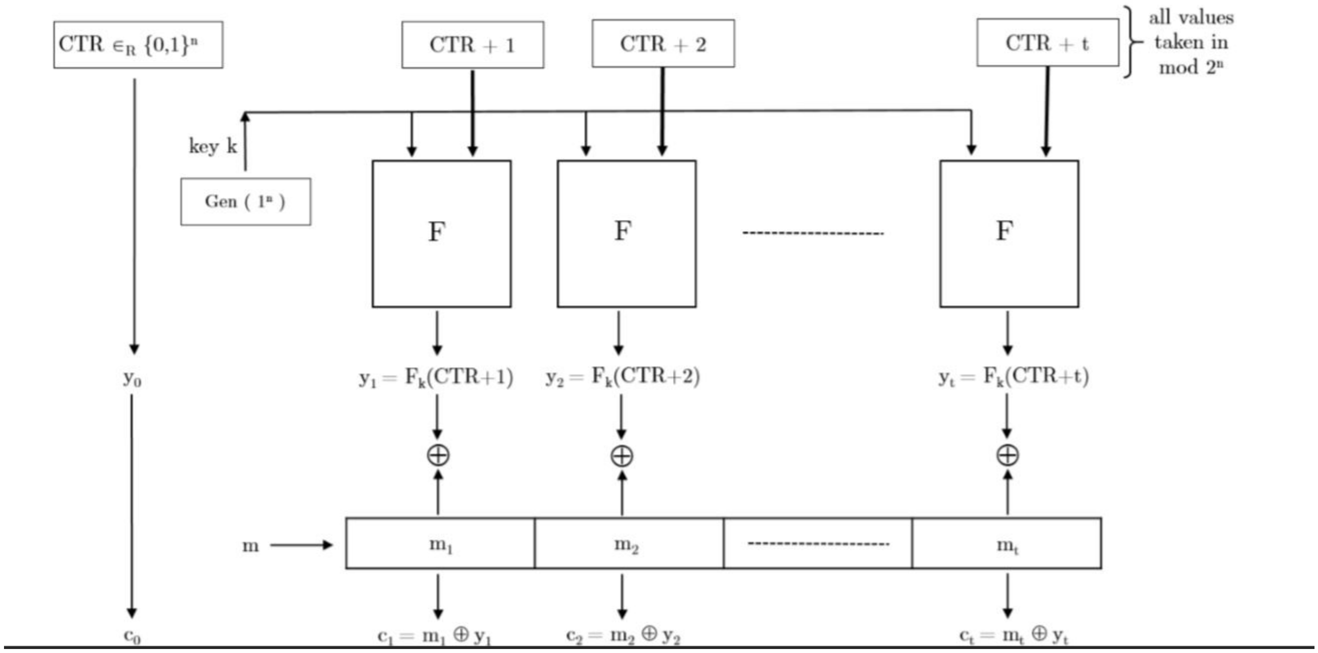

Fig 3d. Counter (CTR) Mode

Here we can see that the scheme is both pre-compuatble and parallelizable as the block of pseudorandom stream can be computed independently.

3.1.5 Comparison of the modes of operation

.	Theorectical construction	ECB	CBC	OFB	CTR
Randomness Usage	n/Block = t.n	No randomness	n	n	n
Ciphertext Expansion	2n/Block = 2t.n	t.n	(t+1).n	(t+1).n	(t+1).n
Ciphertext Computation Parallelizable	YES	YES	NO	NO(but pre-computable)	YES
Randomness Reusability	NO	–	–	YES	YES
Minimal Assumption	PRF	SPRP	SPRP	PRF	PRF
CPA Security	YES	NO	YES	YES	YES

4 References

1. Jonathan Katz,Yehuda Lindell : Introduction to Modern Cryptography, Second Edition.
2. Arpita Patra : Lecture Notes, Lecture 7.