CSA E0 235: Cryptography (04.02.16)

Scribe for Lecture 7

Instructor: Arpita Patra *Submitted by: (Sudeep Chatterjee)*

1 Recall

- New definitions for SKE
  - CPA, CPA-security, CPA-mult-security

In our last lecture we have learned two main definition of cpa security. One is the usual cpa security and the other one is the multiple version of it, aka cpa-mult-security, which incorporates multiple message. They both capture chose plain text attack. The interesting thing to notice here that both of this scheme is equivalent. This means that if any scheme is secure according to single message cpa then it is also secure according the the mult-cpa version. Its quite obvious that simple cpa-security is just special version of multi-cpa-security with only one message.

So, we just need to focus and design for the cpa security for single message and it will automatically imply the mult-cpa version, as we have said earlier. Once we got the definition of the security in place, we started to look for appropriate assumptions on which we can construct schemes which are secure against the cpa notion. We also learned about the notion of PRF, PRP, SPRP.

2 Overview

In this lecture we will build a cpa secure scheme from PRF. We will use reduction based approach to prove this. But unfortunately we will see later that the scheme that we have constructed from PRF is not practically efficient.

3 Minicrypt

Minicrypt is the set of all those things that we can get by just relying on OWF assumption. This contains all those things that we have seen so far in symmetric key setting. In order to construct to all that we need in minicrypt it is enough to assume that OWF exists(necessary and sufficient). We can show the following

$OWF\ exists \implies PRG\ exists \implies PRF exists$
$PRG\ exists \implies PRF exists$
$PRF\ exists \implies cpa\ secure\ scheme\ exists$
$PRF\ exists \implies MAC\ exists$
$PRF\ exists \implies authenticated\ secret\ key\ encryption$

4 Motivation : Why we need PRF?

In the scenario, both the sender and reciever agree on some function, which is picked uniformly at ranom from the space of all functions.(wrt some domain and codomain). So, since in this case we are using TRF, both the parties agree on the truth table. The sender chooses a x_i which is from the domain of the function, and computes the $y_i = F_k(x_i)$. Once computed the sender sends the pair, $(x_i, m\ xor\ y_i)$. The receiver recieves the message as two parts of the cipher texts, (c_0, c_1). So, before trying to go further, let us see the flaw in this design. The exponential nature of the size of the truth table makes it not efficient to store it as a table in memory. Now we will replace TRF with PRF. Functionally a PRF is almost as same as a TRF to PPT distinguisher. Once, we used PRF we don't need the truth table anymore, the only thing we need its the key, as PRF are keyed functions. So both sender and reciever need to agree on the key which is picked U.A.R. Since the length of the key is n, so we have 2^n possibilities of keys. So, once the sender chooses an $r\varepsilon 0, 1^n$, then the sender computes $y = F_k(x_r)$, and send as usual. The reciever receives two components as before. Decrypting is pretty straightforward as follows.

1. Gen: k is chosen u.a.r. from K.
2. Encrypt: pick $r\varepsilon\{0,1\}^n$ and compute $c = (r, m\ xor\ F_k(r))$.
3. Decrypt: $c =< c_0, c_1 >$, $m = c_1\ xor\ F_k(c_0)$.

Theorem

If F_k is a PRF, then Π is a CPA-secure scheme

Proof

Assume that Π is NOT cpa-secure. This let us conclude that there exists some adversary A such that

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] > 1/2 + 1/p(n)$$

Now our goal is to construct a good distinguisher using this adversary, to distinguish between PRF and TRF. So, we want to construct a distinguisher D s.t.

$$|Pr[D^{F_k(.)}(n) = 1] - Pr[D^{f(.)}(n) = 1]| > 1/q(n)$$

We will use adversary A to construct D for the pseudorandom function F. The distinguisher D is given oracle access to a function which can be either a random function or a pseudo random function. To do this, D does the CPA game with A, and notices the response of the adversary A. If A is successful then D guesses that the function must be pseudorandom, otherwise, random. The experiment is done in the following way.

1 Run $A(1^n)$. The adversary makes poly-number of queries which in total runs in poly time.

   – Adversary queries its encryption roacle on message m.

- D chooses r u.a.r. from $0, 1$ and returns the ciphertext $< r, y'\ xor\ m >$ to adversary.

- Now A outputs messages $m_0, m_1 \varepsilon\ \{0,1\}^n (|m_0| = |m_1|)$ and choose a random bit b from $0, 1$ and selects again r^* u.a.r. from $0, 1^n$.(this is the challenge phase).

- D recieves y^* from its oracle function and returns the adversary $(r^*, m_b\ xor\ y^*)$.

  - Finally receving the cipher text, A will give the answer, by outputing a bit b'.

- Now having A's answer D has to formulate his answer. If the oracle function of D is a PRF then the entire game is exactly like the cpa game. So, D knows the probability that A will correctly guess is high, i.e.

$$Pr[A = 1] > 1/2 + 1/p(n)$$

- If TRF is used and if r^* of the post training phase is same as any of the r in the training phase then the probability that A will break the scheme with probability 1. Because when the oracle returns the ciphter text $< r, y >$ in response to a request to encrypt message m, the adversary learns the value of $f(r)$.

- If the r^* is never occurred in the training phase then adversary A learns nothing about the value of $f(r^*)$(since it is truly random function).So, adversary guess the correct b' with probability $1/2$.

Let **Repeat** denotes the event that r^* is repeated in the training phase.
Now, we claim that the probability of adversary A winning the game when the pads are coming from a TRF is $1/2$ given that **Repeat** doesn't happen.

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1/Repeat'] = 1/2$$

We want to find out the following, i.e. probability that A will win.

$$\begin{gathered} Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] \\ = Pr[PrivK^{cpa}_{A,\Pi}(n) = 1 \wedge Repeat'] + Pr[PrivK^{cpa}_{A,\Pi}(n) = 1 \wedge Repeat] \\ \leq Pr[PrivK^{cpa}_{A,\Pi}(n) = 1 | Repeat'] + P[Repeat] \end{gathered}$$

We have proved before that the probability of A's winning the game when Repeat event is not occured is exactly $1/2$. So substituting that in the above we get.

$$= 1/2 + P[Repeat]$$

Now we want to find the value of P[Repeat]. Let A makes $q(n)$ number of queries so, we denote Repeat as $R_1 \wedge R_2 \wedge ... \wedge R_n$. So,

$$\begin{gathered} P[Repeat] = P[R_1 \wedge R_2 \wedge ... \wedge R_n] \\ \leq \sum_{i=1}^{q(n)} Pr[R_i] = q(n)/2^n \end{gathered}$$

Probability that A will win is

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] \leq 1/2 + q(n)/2^n$$

From this calculations D's strategy will be clear. If the function is TRF then,

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] \leq 1/2 + q(n)/2^n$$

If the function is PRF then,

$$Pr[PrivK^{cpa}_{A,\Pi}(n) = 1] \geq 1/2 + 1/p(n)/2^n$$

Now,

$$\begin{aligned} &|Pr[D^{F_k(.)}(n) = 1] - Pr[D^{f(.)}(n) = 1] \\ &\geq 1/p(n) - q(n)/2^n \end{aligned}$$

This proves that D can distinguish a PRF from TRF with non-negligible probability.

5 CPA Security for arbitrarily length messages(Theoritical construction)

Till now we have built a construction where the length of the message is fixed n. What happens if the length of the message is longer than n. One approach is to break the whole messge into some blocks each of length n.(padding done to make equal size blocks).

$$\begin{aligned} m = m_1 m_2 ... m_k \text{ Each } m_i \text{ is encrypted to } c_i. \\ c = c_1 c_2 ... c_k \end{aligned}$$

But how good is this approach? Let us assume the number of blocks is l and size of each block is n. So, $|m| = nl$.

- Random bits are very expensive, very hard to dealt with, so we need to restrict the use of randomness.
- **Ciphertext expansion** The more each ciphertext is expanded the total overhead will increase ny factor l. So, its better to send using no cipher text expansion.
- If the computation of encrypt or decrypt is not parallelizable then it will take a lot of time to encrypt or decrypt a message.
- We should try to build on minimum assumptions. PRF is better.
- **CPA security** Yes

Electronic Code Block Mode

Given message, $m = m_1 m_2 ... m_l$, each of the block is encrypted using direct application of pseudorandom permutation. SPRP is used so decryption can be done using inverse function. Disdavantage is that no randomness is used, hence not even coa-secure for multiple messages. But this is parallelizable. Length of the cipher text and the plaintext are same.

Cipher Block Chaining

The message is divided in blocks and intially a random vector is chosen, and it is xored with the first message block and then encrypted with the key. The output cipher text works as the input "random" vector for the next block. Clearly this is not parallelizable. It assumes SPRP and give cpa-security. Only blockwise parallelization possible.

Encryption:

$$c_i = F_k(m_i \oplus c_{i-1}) \text{ for } i = 1...l$$

Decryption

$$m_i = F_k^{-1}(c_i) \oplus c_{i-1} \text{ for } i = 1...l$$

This scheme uses n bits per message for randomization. Choosing distinct random vector can save randomness but it is not cpa secure assignment.
Another variation that is done on these scheme, is that the final cipher text of the final blcok is fed as the input random vector of the next message. This reduces randomization. SSL 3.0 implements this method. However this is not cpa secure.

Output Feedback Method

Another variation is instead of serially inputting the random vector thought previous block, the function value generated by the random/pseudorandom function is fed to the next block instead of the cipher text. This is a cpa-secure scheme. But this is not parallelizable, but precomputable. In this case we can also incorporate the chaining method with this method and still get cpa-security.

Counter Method

This mode of operation is less comon that CBC mode. In this mode, it can be viewed as a way of generating a pseudorandom stream of block cipher. The stream is fed to each block parallely. Stream $r_i = F_k(ctr + i)$ is the way it is computed. Clearly this is parellelizable, CPA-secure.

6 Reference

[1] Katz, Jonathan, and Yehuda Lindell. Introduction to modern cryptography. CRC Press, 2014. [2] Arpita Patra. http://drona.csa.iisc.ernet.in/arpita/Cryptography16.html. Course Materials.