

CSE 508 – Network Security

Date : 09/04/2009

Topics: Pseudo-Random Generators, Stream Ciphers, Computational Indistinguishability

1. Problem with One Time Pad

In One Time Pad (Fig 1) the key K has to be as big as the message M i.e. to send 1 GB of message we need to have a key of length 1 GB, which is annoying

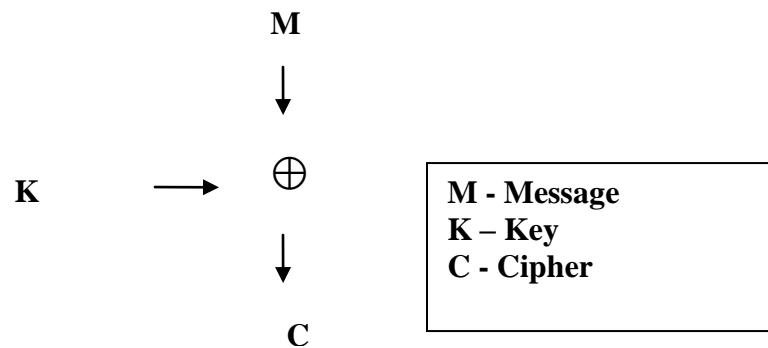


Fig 1: One Time Pad

2. Stream Cipher

Stream Cipher overcomes the above problem by using an *Expansion function* upon a key of much smaller length. The output of the expansion function is a continuous stream (Fig 2).

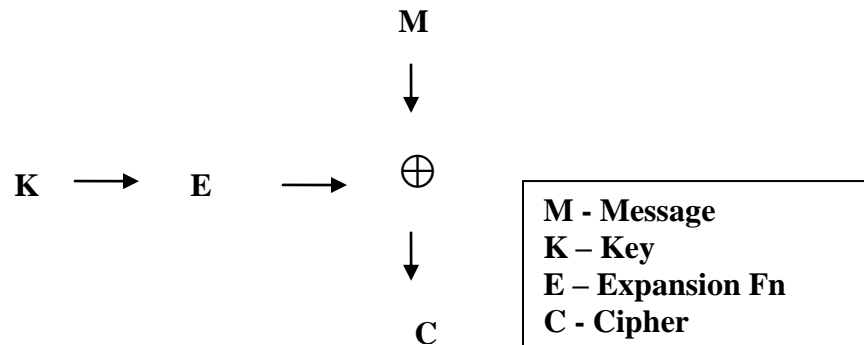


Fig 2: Stream Cipher

2.1. Desired properties of Expansion function

- It should generate truly random stream of bits
- It should be able to generate long stream of bits
- Given an output, an adversary should not be able to infer the seed of random generators used in the expansion function

- The stream shouldn't repeat itself
- The function should be efficient
- The function should be deterministic
- An adversary should not be able to predict unknown portion of the output from some known portion of the output (known as *Next-bit Test*)

We don't know whether an expansion function with all such desired properties actually exists.

2.2. Examples of Expansion functions

Example 1: Linear Congruential Generator (LCG)

The expansion function E in a LCG looks like:

```

unsigned int state;
int E(void)
{
    state = 322349 * state + 45656749; // large prime numbers
    return state % 2; //state machine change
}

```

Example 2: Linear Feedback Shift Register (LFSR)

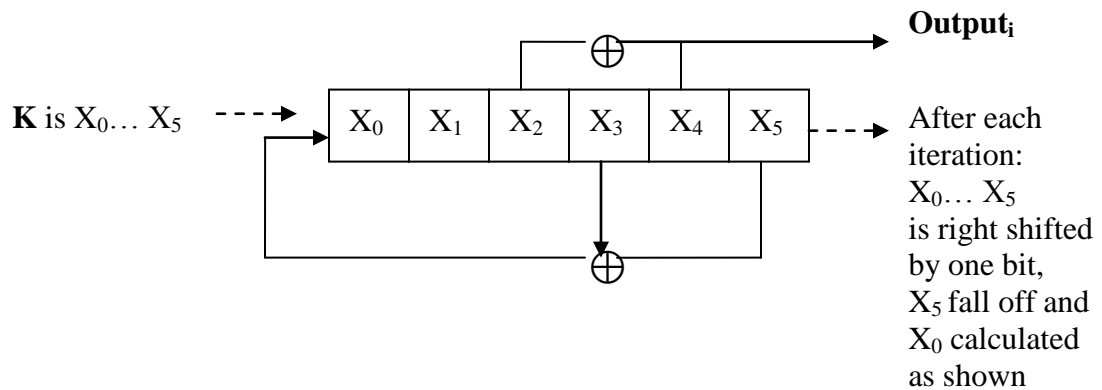


Fig 3: LFSR

If one chooses the taps carefully all possible $2^6 - 1$ values are generated (all except 0)

LCG and LFSR are not secure.

Example 3: Blum Blum Shub (BBS)

Pick two large primes P, Q. $N = P * Q$

Pick $X_0 \in \{2, 3, \dots, N-1\}$

Here Key = (X_0, N)

Let $X_i = X_{i-1}^2 \pmod N$

Now $\text{Output}_i = X_i \pmod 2$

Breaking BBS (predicting Output_i) is equivalent to factoring N

3. Probability Distribution

Definition: A probability distribution is a function that assigns a probability to each possible value in some set.

Example 1

$$\begin{aligned} S &= \{0,1\} \\ D(0) &= 0.3 \\ D(1) &= 0.7 \end{aligned}$$

Here D is the probability distribution function and it is a non-uniform distribution. $\sum D(S_i)$ should be equal to 1

The uniform distribution is a distribution function U in which $U_s(s) = 1/|S|$

$S \rightarrow$ Set

$s \rightarrow$ an element in the set

Notations: U_1 stands for the uniform distribution of 1-bit strings
 $X \leftarrow D$ means 'Drawing an X according the distribution D '

Example 2

$$D = U_{\{0,1,2,3,4,5\}}$$

$$X \leftarrow D$$

X	0	1	2	3	4	5
D	1/6	1/6	1/6	1/6	1/6	1/6

Example 3

If the distribution is $X^2 \pmod 6$

X	0	1	2	3	4	5
$X^2 \pmod 6$	0	1	4	3	4	1

Then D is

X	0	1	2	3	4	5
D	1/6	1/3	0	1/6	1/3	0

4. Statistical Indistinguishability

This is an approach to compare a stream cipher key and a truly random key

Definition: Two distributions D & D' are ϵ -statistically indistinguishable if for all algorithm A ,

$$\text{Adv } A = \left| \Pr[A(X) = 1 \mid X \leftarrow D] - \Pr[A(X) = 1 \mid X \leftarrow D'] \right| \leq \epsilon$$

A is called an adversary algorithm and is defined as follows

$$A(X) = 1, \text{ if } A \text{ guesses that } X \text{ was drawn from } D \\ = 0 \text{ otherwise}$$

In this case A is assumed to have infinite time and infinite knowledge.

5. Computational Indistinguishability

Definition: Distribution D & D' are t, ϵ -computationally indistinguishable if for all algorithm A running in time $\leq t$,

$$\text{Adv } A \leq \epsilon$$

Representation: $D \stackrel[t]{\sim} D'$

Statistical indistinguishability can be represented as $D \stackrel[\epsilon]{\sim} D'$

Theorem 1

$$\text{If } D_1 \stackrel[t]{\sim}_{\epsilon_1} D_2 \text{ \& } D_2 \stackrel[t]{\sim}_{\epsilon_2} D_3 \text{ then } D_1 \stackrel[t]{\sim}_{\epsilon_1 + \epsilon_2} D_3$$

Proof:

Let A be any algorithm running in time t

Then

$$\begin{aligned} & \left| \Pr[A(X) = 1 \mid X \leftarrow D_1] - \Pr[A(X) = 1 \mid X \leftarrow D_3] \right| \\ &= \left| \Pr[A(X) = 1 \mid X \leftarrow D_1] - \Pr[A(X) = 1 \mid X \leftarrow D_2] + \Pr[A(X) = 1 \mid X \leftarrow D_2] - \Pr[A(X) = 1 \mid X \leftarrow D_3] \right| \\ &\leq \left| \Pr[A(X) = 1 \mid X \leftarrow D_1] - \Pr[A(X) = 1 \mid X \leftarrow D_2] \right| + \\ & \quad \left| \Pr[A(X) = 1 \mid X \leftarrow D_2] - \Pr[A(X) = 1 \mid X \leftarrow D_3] \right| \\ &\leq \epsilon_1 + \epsilon_2 \end{aligned}$$

Theorem 2 (Data Processing Inequality)

If $D \stackrel{t}{\sim} D'$ and f is any function computable in time t' , then $f(D) \stackrel{t-t'}{\sim}_{\epsilon} f(D')$

Notation: $f(D) \implies X \leftarrow D; f(x)$

Proof: (Proof by contrapositive)

Let A' be a function which can distinguish D & D' as depicted in the Fig 4

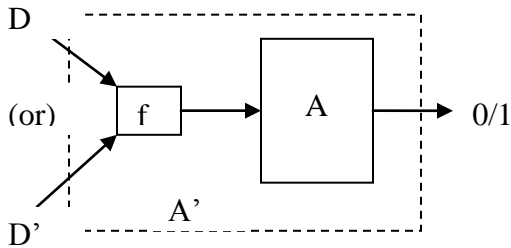


Fig 4: Data Processing Inequality

Suppose $\exists A$ running in time $t-t'$ such that $\text{Adv}_{f(D), f(D')} A > \epsilon$ (Negation of *then* proposition in the theorem)

$$\begin{aligned} \Pr [A'(X)=1 \mid X \leftarrow D] &= \Pr [A(f(X))=1 \mid X \leftarrow D] \\ &= \Pr [A(X)=1 \mid X \leftarrow f(D)] \end{aligned} \quad \text{---->(1)}$$

$$\begin{aligned} \Pr [A'(X)=1 \mid X \leftarrow D'] &= \Pr [A(f(X))=1 \mid X \leftarrow D'] \\ &= \Pr [A(X)=1 \mid X \leftarrow f(D')] \end{aligned} \quad \text{---->(2)}$$

So advantage in distinguishing D and D' is

$$\begin{aligned} \text{Adv}_{D, D'} A' &= \left| \Pr [A'(X)=1 \mid X \leftarrow D] - \Pr [A'(X)=1 \mid X \leftarrow D'] \right| \\ &= \left| \Pr [A(X)=1 \mid X \leftarrow f(D)] - \Pr [A(X)=1 \mid X \leftarrow f(D')] \right| \quad \text{---->from (1) \& (2)} \\ &= \text{Adv}_A \\ &> \epsilon \end{aligned}$$

So $\text{Adv}_{D, D'} A' > \epsilon$ (Negation of *if* proposition in the theorem)

Hence the theorem is proved by contrapositive