

## 1 Lecture Plan

- See examples of stream ciphers used in practice.
- Define CPA-security

## 2 Stream Ciphers

- Stream ciphers are practical systems which behave like pseudorandom generators. However, there are no proofs available that a particular stream cipher is in fact a pseudorandom generator.
- Stream ciphers can be designed for either efficient hardware implementation or efficient software implementation.
- Hardware-oriented stream ciphers are based on feedback shift registers (FSRs).
- Linear feedback shift registers (LFSRs) are FSRs where the feedback function is linear.
- Example: Consider a four-bit shift register where the feedback is the XOR of all the four bits. If we initialize the state to 1100, then we get a cycle of period 5. The states are 1100, 1000, 0001, 0011, 0110.
- The output depends on the state of the LFSR. Once a state repeats, the output repeats. If an LFSR has  $n$  bits, then the period of the output sequence can be at most  $2^n - 1$ .
- Each LFSR can be associated with a feedback polynomial. If the feedback polynomial is primitive, then the period is maximal. A polynomial of degree  $n$  over  $\text{GF}(2)$  is primitive if it is irreducible and the smallest value of  $m$  for which the polynomial divides  $X^m + 1$  is  $m = 2^n - 1$ . Example:  $1 + X^3 + X^4$ .

### 2.1 A5/1

- Used to provide voice encryption in the GSM cellular system.
- Developed in 1987. Reverse engineered in 1999.
- Uses three LFSRs of lengths 19, 22, and 23.
- More details at <https://en.wikipedia.org/wiki/A5/1>.

## 2.2 RC4

- A software-oriented cipher designed by Ron Rivest of RSA Security in 1987. Reverse engineered and leaked in 1994.
- Has an internal state of 256 bytes initialized to  $S[i] = i$  for  $i = 0, 1, \dots, 255$ .
- More details on pages 92–93 in Chapter 5 of Serious Cryptography.
- It took 20 years for cryptanalysts to find flaws. Used in WEP (the first generation Wi-fi security protocol) and TLS (the protocol underlying HTTPS).

## 3 Chosen-Plaintext Attacks and CPA-Security

- Consider a scenario where the honest parties share a key  $k$  and the attacker can influence these parties to encrypt messages  $m_1, m_2, \dots$  using  $k$ . At some later point, the attacker observes the encryption of a message  $m$  (using the same key  $k$ ). He even knows  $m$  is one of the messages  $m_1, m_2, \dots$ . Security against chosen-plaintext attacks means that the attacker cannot tell which message was encrypted with probability significantly better than random guessing.
- Real-world chosen-plaintext attacks: WWII British mine locations, Battle of Midway
- Formally, chosen-plaintext attacks are modeled by giving the adversary  $\mathcal{A}$  access to an *encryption oracle*. It can be considered a black box which encrypts messages of  $\mathcal{A}$ 's choosing using a key  $k$  which is unknown to  $\mathcal{A}$ .
- Consider the following experiment  $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$ :
  1. A key  $k$  is generated by running  $\text{Gen}(1^n)$ .
  2. The adversary  $\mathcal{A}$  is given  $1^n$  and oracle access to  $\text{Enc}_k(\cdot)$ , and outputs a pair of messages  $m_0, m_1 \in \mathcal{M}$  with  $|m_0| = |m_1|$ .
  3. A uniform bit  $b \in \{0, 1\}$  is chosen. Ciphertext  $c \leftarrow \text{Enc}_k(m_b)$  is computed and given to  $\mathcal{A}$ .
  4. The adversary  $\mathcal{A}$  continues to have oracle access to  $\text{Enc}_k(\cdot)$ , and outputs a bit  $b'$ .
  5. The output of the experiment is defined to be 1 if  $b' = b$ , and 0 otherwise. If output is 1, we say that  $\mathcal{A}$  succeeds.

**Definition.** A private-key encryption scheme  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  has *indistinguishable encryptions under a plaintext attack*, or is **CPA-secure**, if for all probabilistic polynomial-time adversaries  $\mathcal{A}$  there is a negligible function  $\text{negl}$  such that, for all  $n$ ,

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

- Note that no deterministic encryption scheme can be CPA-secure.

## 4 Pseudorandom Functions

- Pseudorandom functions are “random-looking” functions.
- In this case, pseudorandomness will be a property of a distribution over functions.
- Given a security parameter  $n$ , a keyed function  $F : \{0, 1\}^{l_{key}(n)} \times \{0, 1\}^{l_{in}(n)} \rightarrow \{0, 1\}^{l_{out}(n)}$  is a two-input function, where the first input is called the key and is denoted by  $k$ . The functions  $l_{key}, l_{in}, l_{out}$  specify the lengths of the key, second input, and output respectively.
- We will only consider *efficient* keyed functions, i.e. there is a polynomial-time algorithm that computes  $F(k, x)$  given  $k$  and  $x$ .
- If the key  $k$  is fixed, we get a single-input function  $F_k : \{0, 1\}^{l_{in}(n)} \rightarrow \{0, 1\}^{l_{out}(n)}$  defined by  $F_k(x) = F(k, x)$ .
- $F$  is said to be *length-preserving* when  $l_{key}(n) = l_{in}(n) = l_{out}(n) = n$ .
- For simplicity, let us assume that  $F$  is length-preserving.
- Let  $\mathbf{Func}_n$  be the set of all functions with domain and range equal to  $\{0, 1\}^n$ .
- Informally, a keyed function  $F$  is said to be *pseudorandom* if the function  $F_k$  (for a uniform key  $k$ ) is indistinguishable from a function chosen uniformly from  $\mathbf{Func}_n$ . No efficient adversary should be able to distinguish (with a success probability non-negligibly better than  $\frac{1}{2}$ ) whether it is interacting with  $F_k$  (for uniform  $k$ ) or  $f$  (where  $f$  is uniformly chosen from  $\mathbf{Func}_n$ ).
- Note that  $|\mathbf{Func}_n| = 2^{n \cdot 2^n}$ . Visualize a lookup table having  $2^n$  rows with each row containing an  $n$ -bit string. Each row corresponds to an input  $x \in \{0, 1\}^n$  and the contents correspond to the output  $f(x)$ .
- Choosing a function  $f$  uniformly from  $\mathbf{Func}_n$  corresponds to choosing each row in the lookup table uniformly and independently of the other rows.
- For a given length-preserving keyed function  $F_k$ , choosing  $k$  uniformly from  $\{0, 1\}^n$  induces a distribution over at most  $2^n$  functions with domain and range equal  $\{0, 1\}^n$ .
- The definition of a pseudorandom function will be given with respect to an efficient (polynomial-time) distinguisher  $D$  which is given access to an *oracle*  $\mathcal{O}$  which is either equal to  $F_k$  (for uniform  $k$ ) or  $f$  (for uniform  $f$  from  $\mathbf{Func}_n$ ).  $D$  can query the oracle  $\mathcal{O}$  at any point  $x \in \{0, 1\}^n$  and the oracle returns  $\mathcal{O}(x)$ .  $D$  can adaptively query the oracle but can ask only polynomially many queries.

**Definition.** Let  $F$  be an efficient, length-preserving, keyed function.  $F$  is a **pseudorandom function** if for all PPT distinguishers  $D$ , there is a negligible function  $\text{negl}$  such that:

$$\left| \Pr \left[ D^{F_k(\cdot)}(1^n) = 1 \right] - \Pr \left[ D^{f(\cdot)}(1^n) = 1 \right] \right| \leq \text{negl}(n),$$

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  $f \in \mathbf{Func}_n$  and the randomness of  $D$ .

- $D$  is *not* given access to the key  $k$ . If  $k$  is known, it is easy to construct a distinguisher which succeeds with non-negligible probability (how?).
- Example of a non-pseudorandom, length-preserving, keyed function:  $F(k, x) = k \oplus x$ .

## 5 CPA-Secure Encryption from Pseudorandom Functions

- Let  $F$  be a pseudorandom function. Define a private-key encryption scheme for messages of length  $n$  as follows:

- **Gen**: On input  $1^n$ , choose  $k$  uniformly from  $\{0, 1\}^n$ .
- **Enc**: Given  $k \in \{0, 1\}^n$  and message  $m \in \{0, 1\}^n$ , choose uniform  $r \in \{0, 1\}^n$  and output the ciphertext

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

- **Dec**: Given  $k \in \{0, 1\}^n$  and ciphertext  $c = \langle r, s \rangle$ , output the plaintext message

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

**Theorem.** *If  $F$  is a pseudorandom function, then the above construction is a CPA-secure private-key encryption scheme for messages of length  $n$ .*

*Proof.* Done in class. □

## 6 References and Additional Reading

- Chapter 5 of *Serious Cryptography* by J.-P. Aumasson.
- Section 3.4, 3.5 from Katz/Lindell