1 Week 1

1.1 Basics

1.2 Cryptographic primitives

A cryptographic primitive is a fundamental building block used in cryptographic protocols and algorithms. These primitives provide basic functionalities and are combined to implement more complex cryptographic operations. Important symmetric primitives include:

- block ciphers
- stream ciphers
- message authentication codes
- key derivation functions
- random number generators
- hash functions

Important asymmetric primitives include:

- public key cryptography
- digital signatures
- key exchange protocols
- identity-based cryptography

1.2.1 Computational feasibility

We will call a task computationally infeasible if its cost as measured by either the amount of memory used or the runtime is finite but impossibly large.

An encryption algorithm has a security level of n bits if the best known attack requires $\mathcal{O}(2^n)$ steps.

1.3 "Easy" symmmetric ciphers

1.3.1 Vernam (XOR) cipher

1.3.2 Caesar (modular addition) cipher

1.3.3 Rot (circular shift) cipher

1.4 Modular arithmetic

1.4.1 Modular multiplication

1.4.2 Galois cipher

1.5 Scenarios of Attacks and CPA-IND goal

1.6 Attack Scenarios

Ciphertext-only attack. The cryptanalyst has the ciphertext of several messages, all of which have been encrypted using the same algorithm. The cryptanalyst's job is to recover the plaintext

of as many messages as possible, or better yet deduce the keys used to encrypt the messages, in order to decrypt other messages encrypted with the same key.

Known-plaintext attack. The cryptanalyst also has access to the plaintext of several encrypted messages. His job is to deduce the key used to encrypt those messages.

Chosen-plaintext attack. The cryptanalyst can also choose how the plaintext gets encrypted. The goal is to deduce the encryption key.

Adaptive-chosen-plaitext attack. A special case of chosen-plaintext attack. The cryptanalyst can choose the plaintext to encrypt and modify his choice based on the results of the previous encryption.

Chosen-ciphertext attack. The cryptanalyst can choose different ciphertext to be decrypted and has access to decrypted plaintext. The goal is to deduce the key.

Chosen-key attack. The cryptanalyst has some knowledge about the relationship between different keys.

Rubber-hose analysis The cryptanalyst uses violence and/or bribery to get the key. Straightforward!

1.7 CPA-IND and Probabilistic Encryption

Ciphertext indistinguishability is a property of many encryption schemes. Intuitively, if a cryptosystem possesses the property of indistinguishability, then an adversary will be unable to distinguish pairs of ciphertexts based on the message they encrypt. The property of indistinguishability under chosen plaintext attack is considered a basic requirement for most provably secure public key cryptosystems.

A cryptosystem is considered secure in terms of indistinguishability if no adversary, given an encryption of a message randomly chosen from a two-element message space determined by the adversary, can identify the message choice with probability significantly better than that of random guessing.

CPA-IND experiment:

1. A key k is generated by running $Gen(1^n)$.
2. The adversary A is given input 1^n and oracle access to $Enc_k(\cdot)$.
3. A random bit $b \leftarrow \{0,1\}$ is chosen, and a ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to A. We call c the challenge ciphertext.
4. The adversary A continues to have oracle access to $Enc_k(\cdot)$, and outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

2 Week 2

2.1 Randomness

Randomness is found everywhere in cryptography: in the generation of secret keys, in encryption schemes, and even in the attacks on cryptosystems. Without randomness, cryptography would be impossible because all operations would become predictable, and therefore insecure.

2.1.1 TRNG

True random number generators (TRNGs) are characterized by the fact that their output cannot be reproduced. RNGs are based on physical processes that cannot be reproduced. In computer systems, modern CPUs are often equipped with hardware-based TRNGs or else there is a TPM (trusted platform module) on the motherboard which contains a TRNG. In computer systems without a hardware TRNG, random processes within the computer are used as entropy sources, e.g., fine-grained timing measurements of interrupts or other random data from device drivers.

2.1.2 PRNG

(Cryptographycally secure) Pseudorandom number generators (PRNGs) generate sequences which are computed from an initial seed value. Formally, a PRNG is a function $G : \mathbb{Z}_2^n \to \mathbb{Z}_2^{l(n)}, l(n) > n$, such that no adversary A can succeed the PRG-IND experiment with probability $> 1/2$.

PRG-IND experiment:

1. A uniform bit $b \in \{0,1\}$ is chosen. If $b = 0$, then choose a uniform $r \in \{0,1\}^{l(n)}$; if $b = 1$ then choose a uniform $s \in \{0,1\}^n$ and set $r = G(s)$.
2. The adversary A is given r, and outputs a bit b'.
3. The output of the experiment is defined to be 1 if $b' = b$ and 0 otherwise.

Usually, PRNGs are constructed procedurally from two algorithms:

- Init: takes as input a seed £s£ and an optional initialization vector, and outputs an initial state st_0
- GetBits: takes as input st_i, outputs a bit y and updates the state to st_{i+1}

2.2 Linear Congruential Generator

A linear congruential generator (LCG) is an algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piecewise linear equation.

The generator is defined by the recurrence relation:

$$X_{n+1} = (aX_n + c) \mod m$$

where m is the modulus, a is the multiplier, c is the increment, and X_0 is the seed, and are all integer constants. If $c = 0$, the generator is called Lehmer RNG.

2.2.1 Park Miller RNG

The Park-Miller random number generator (after Stephen K. Park and Keith W. Miller), is a type of linear congruential generator (LCG) that operates in multiplicative group of integers modulo n. The general formula is

$$X_{k+1} = a \cdot X_k \mod m$$

where m is a prime number or a power of a prime number, the multiplier a is an element of high multiplicative order modulo m (e.g., a primitive root modulo n), and the seed X_0 is coprime to m.

2.3 Stream Ciphers and Operation Modes

Stream ciphers encrypt bits individually. This is achieved by adding a bit from a key stream to a plaintext bit.

2.3.1 Synchronized

In synchronized stream ciphers the key stream depends only on the key.

2.3.2 Asynchronized

In asynchronized stream ciphers the key stream depends also in ciphertext.

2.4 Linear-Feedback Shift Register LFSR

An LFSR consists of clocked storage elements (flip-flops) and a feedback path. The number of storage elements gives us the degree of the LFSR. In other words, an LFSR with m flip-flops is said to be of degree m. The feedback network computes the input for the last flip-flop as the XOR-sum of certain flip-flops in the shift register.

2.4.1 Fibonacci LFSRs

The bit positions that affect the next state are called the taps. The rightmost bit of the LFSR is called the output bit, which is always also a tap. To obtain the next state, the tap bits are XOR-ed sequentially; then, all bits are shifted one place to the right, with the rightmost bit being discarded, and that result of XOR-ing the tap bits is fed back into the now-vacant leftmost bit. To obtain the pseudorandom output stream, read the rightmost bit after each state transition.

The arrangement of taps for feedback in an LFSR can be expressed in finite field arithmetic as a polynomial mod 2. This means that the coefficients of the polynomial must be 1s or 0s. This is called the feedback polynomial or reciprocal characteristic polynomial. For example, if the taps are at the 16th, 14th, 13th and 11th bits, the feedback polynomial is

$$x^{16} + x^{14} + x^{13} + x^{11} + 1$$

The "one" in the polynomial does not correspond to a tap, it corresponds to the input to the first bit.

2.4.2 Galois LFSRs

In the Galois configuration, when the system is clocked, bits that are not taps are shifted one position to the right unchanged. The taps, on the other hand, are XORed with the output bit before they are stored in the next position. The new output bit is the next input bit. The effect of this is that when the output bit is zero, all the bits in the register shift to the right unchanged, and the input bit becomes zero. When the output bit is one, the bits in the tap positions all flip (if they are 0, they become 1, and if they are 1, they become 0), and then the entire register is shifted to the right and the input bit becomes 1.

2.4.3 Correlation Attack

Correlation attacks are a class of cryptographic known-plaintext attacks for breaking stream ciphers whose keystreams are generated by combining the output of several linear-feedback shift registers (LFSRs) using a Boolean function.

2.5 Permutation

2.5.1 Keccak-Sponge Key Stream

2.5.2 Keccak-f Permutations

3 Week 3