Network Security and Cryptography
Symmetric-key cryptography

Lecture 0: Module practicalities

Mark Ryan

Network Security and Cryptography is divided into three parts:

1. Symmetric-key cryptography (lecturer: Mark Ryan)
2. Public-key cryptography (lecturer: Rishi Bhattacharyya)
3. Network Security (lecturer: Marius Muench)

Delivery methods:

- Live lectures
- Pre-recorded lectures (backup + more detail)
- Lecture-room Q+A, where lecturer asks the Qs
- Exercises, both assessed and non-assessed

For details, see Canvas page.

Network Security and Cryptography
Symmetric-key cryptography

Lecture 1: Intro to cryptography

Mark Ryan

What is cryptography?

Cryptography is the practice and study of techniques for secure communication and data storage in the presence of third parties (called adversaries, or attackers).

Here, "security" means *confidentiality*, *integrity*, and *authenticity*.

Cryptography is all around us:

- `https://` on the web
- WhatsApp end-to-end encryption
- Volume encryption on laptops, and phones
- Signature verification on software downloads
- . . .

Symmetric and asymmetric cryptography

Symmetric crypto

The same key is used to encrypt and to decrypt.
- Generate a key k
- Encrypt a "message" m with k:

$$c = \mathrm{Enc}_k(m)$$

- Decrypt the "ciphertext" c with the same key k:

$$m' = \mathrm{Dec}_k(c).$$

Asymmetric crypto

The encryption key and the decryption key are different.
- Generate a key sk, pk consisting of a secret part and a public part. Publish the public part.
- Someone will encrypt a "message" m with your pk:

$$c = \mathrm{Enc}_{pk}(m)$$

- You will decrypt the "ciphertext" c with the corresponding secret key sk:

$$m' = \mathrm{Dec}_{sk}(c).$$

Is this module for me?

From the canvas page:

Cryptography is a highly mathematical subject. It is assumed that you are comfortable with mathematical reasoning, including the ability to understand and evaluate proofs. You should be knowledgeable about modular arithmetic, and you have a good understanding of probability. You should also know how to read, write and change programs.

Can you solve these in 30-60 seconds each?

1. $53 \times 120 \mod 256$
2. $(x^2 + 3x + 1)(x^3 - 1)$
3. Ox3F $\oplus$ 0xFF
4. I toss a fair coin three times. What is the probability that I get *at least one head*?
5. I toss a fair coin n times. What is the probability that I get *exactly one head*?
6. gcd(21,35)
7. $x^{-k} \cdot (xy)^k \cdot y$
8. How may years is 2^{64} seconds?

Encryption

Encryption essential for security on the internet

Confidentiality, integrity, authenticity cannot be guaranteed otherwise

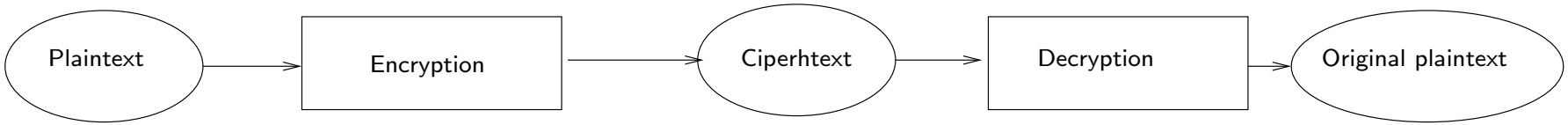
Works in principle as follows:

- Alice and Bob share a secret key.
- Alice uses secret key to scramble data: encryption
- Alice sends scrambled data to Bob
- Bob unscrambles data with secret key: decryption

Issue: how to achieve the premise that A and B share a key?

Terminology

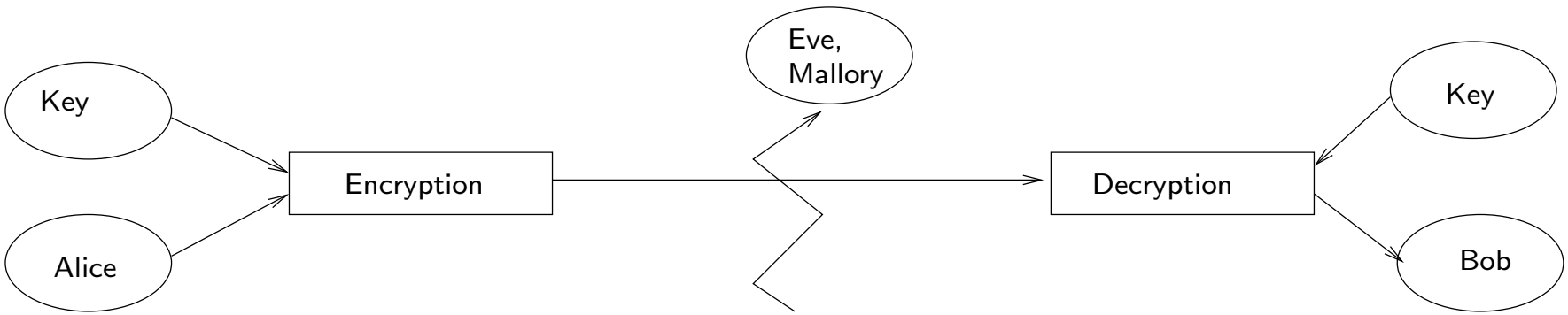
Plaintext	Message before encryption
Encryption	Process of scrambling a message
Ciphertext	An enciphered message
Decryption	Process of unscrambling a message



Players

Have the following main players:

- Alice: sender of an encrypted message
- Bob: intended receiver of encrypted message. Assumed to the key.
- Eve: (Passive) attacker intercepting messages and trying to identify plaintexts or keys
- Mallory: (Active) attacker intercepting and modifying messages to identify plaintexts or keys



Three simple ciphers, and a more complicated one

- Columnar transposition cipher
- Substitution cipher
- One-time pad

- DES (Data Encryption Standard, 1976)

Columnar transposition cipher

Used already since antiquity

- Key: Column size
- Encryption: Arrange message in columns of fixed size (the key). Add dummy text to fill the last column. Ciphertext consists of rows.
- Decryption: Calculate row size by dividing message length by the key. Arrange message in rows of this size. Plaintext consists of columns.

Is this cipher secure?

Security of columnar transposition cipher

Is this cipher secure?
Informal answer: No.
Given any ciphertext, attacker tries all possible values for the key.
For a message of size n there are at most n possibilities for the key, hence attacker will obtain plaintext.

Monoalphabetic substitution cipher

- Key: permutation of the alphabet
- Encryption: Apply the permutation
- Decryption: Apply the inverse permutation

Here is one way to choose the key:

- Choose keyword (or keyphrase)
- remove all duplicate letters from keyword
- start cipher-alphabet with letters from duplicate-free keyword
- and the end of the codeword continue with next unused letter of alphabet following last letter in codeword
- continue filling in letters in alphabetical order leaving out already used letters

How difficult is it to break this cipher? How many keys are there?

Security of the monoalphabetic substitution cipher

How difficult is it for the attacker to break this cipher?
Have $26! \approx 2^{86}$ possible keys
But: Have other tools available, e.g. frequency analysis
Frequency of letter occurrence varies dramatically amongst letters
In English text, 12.7% of all letters are "e", and 0.2% of all letters are "x".

One-time pad

The one-time pad consists of having a single-use key that is the same length as the plaintext being encrypted. Encryption is done by XOR'ing the plaintext and the key. Decryption is done by XOR'ing the ciphertext and the key.

- Key: Random bitstring $k_1, \ldots, k_n$, as long as message $m_1, \ldots, m_n$
- Encryption: $k_1 \oplus m_1, \ldots, k_n \oplus m_n$
- Decryption of ciphertext $c_1, \ldots, c_n$: $k_1 \oplus c_1, \ldots, k_n \oplus c_n$

How secure is this encryption system? (Hint: how many keys are there? Would frequency analysis work?)

How practical is this system?

Security of one-time pad

The one-time pad is *perfectly secure* (see later). The attacker cannot learn anything by looking at ciphertexts*.

* except the size of the ciphertext.

But it is not very practical, because:

- The key is long (same length as the plaintext)
- The key can be used only once.