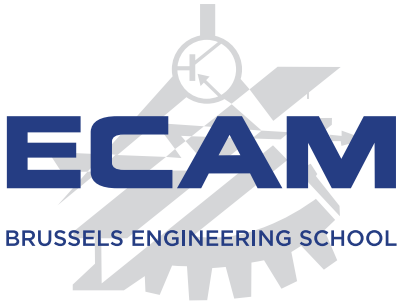


I5020 Computer Security

Session 2

Introduction to Cryptography and Symmetric Encryption

Sébastien Combéfis

Fall 2019

Objectives

- Introduction to cryptography
  - Definition of cryptosystems and basic cryptographic tools
  - Examples of simple cryptosystems and their cryptanalysis
- Symmetric encryption based on private keys

  Block and stream ciphers, DES and AES algorithms
- Applications of symmetric cryptography

  Exchanging messages, protecting files, banking, etc.


Cryptosystem

Alice and Bob (1)

- Alice and Bob exchange messages on communication channel

  Insecure channel, with Eve trying to intercept the exchanges

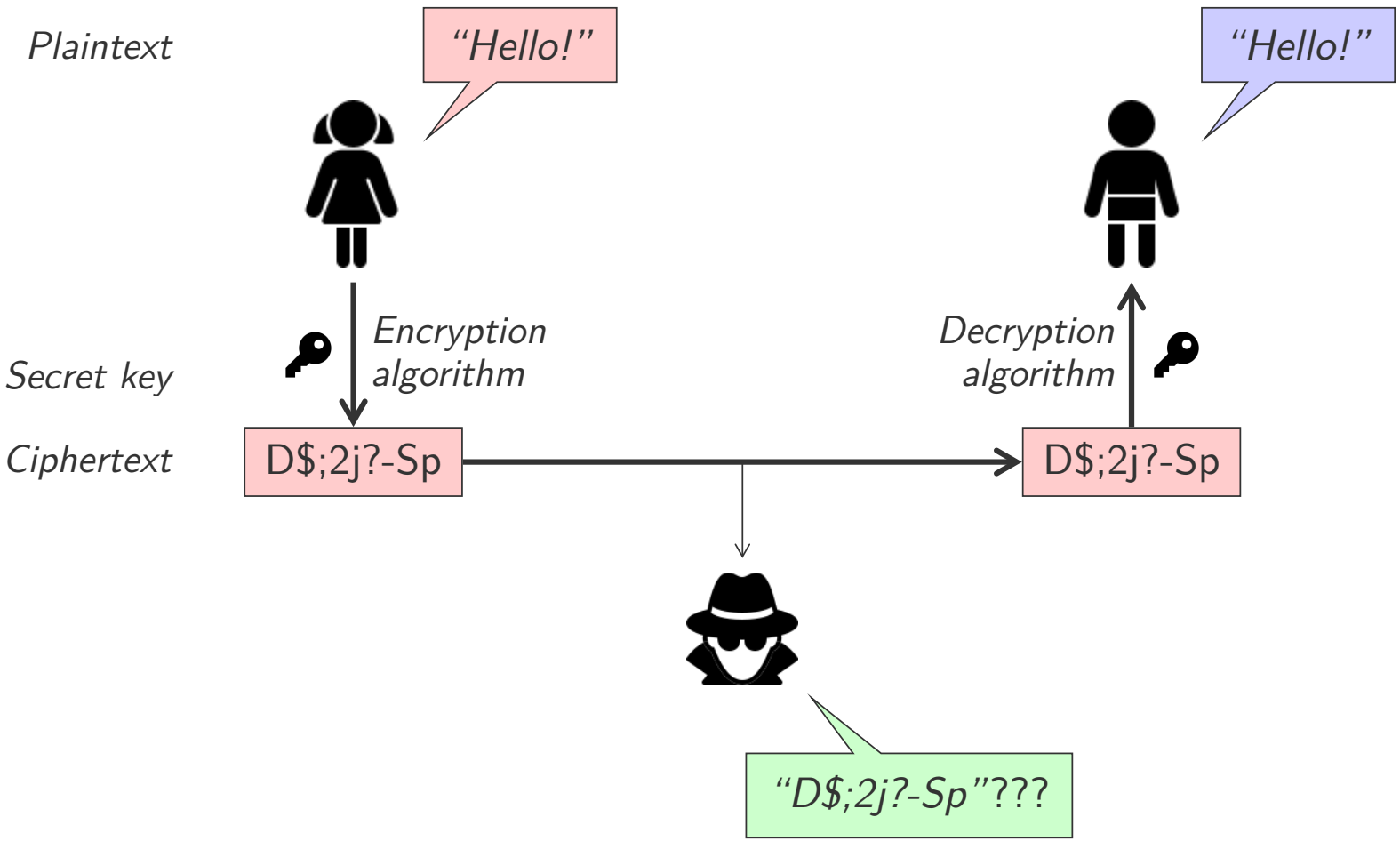
- Cryptography turns a clear text into a ciphered text

  Transmission of the ciphered text, Eve cannot understand it

- Only Alice and Bob can read the message thanks to a key

  This key needs to be shared between both stakeholders

Alice and Bob (2)



Cryptosystem

- Cryptosystem used for a secure communication

  Set of five elements used to exchange messages

- Representation by a five-tuple $\langle \mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D} \rangle$ such that:

  1. $\mathcal{P}$ finite set of plaintexts
  2. $\mathcal{C}$ finite set of ciphertexts
  3. $\mathcal{K}$ finite set of possible keys (*keyspace*)
  4. $\forall K \in \mathcal{K} : \exists (e_K : \mathcal{P} \to \mathcal{C}) \in \mathcal{E}$, (*encryption rule*)
     $(d_K : \mathcal{C} \to \mathcal{P}) \in \mathcal{D}$: (*decryption rule*)
     $\forall x \in \mathcal{P} : \; d_K(e_K(x)) = x$

Encryption Rule

- Encryption rule e_K is an injective function (one-to-one)

  That is, $e_K(x_1) \neq e_K(x_2)$ *when* $x_1 \neq x_2$

- Unambiguous decryption of a ciphertext with d_K

  That is, $d_K(y_1) = x_1 \implies \nexists x_2 : e_K(x_2) = y_1$

- Encryption function can perform permutation of plaintexts

  This is only possible when using the same alphabet $(\mathcal{P} = \mathcal{C})$

Cryptology

- Cryptography secures data

  People using this science are the "good" guys

- Cryptanalysis analyses and breaks secure communication

  People using this science are the "bad" guys

- The Kerchhoff's principle states that cryptosystem is known

  Need for security given that opponent knows the cryptosystem

Attack Model

- Four main attack models depending on known information
  - **Ciphertext-only**: opponent possesses $\mathbf{y}$
  - **Known plaintext**: opponent possesses a pair $(\mathbf{x}, \mathbf{y})$
  - **Chosen plaintext**: opponent can generate $\mathbf{y}$ given any $\mathbf{x}$
  - **Chosen ciphertext**: opponent can generate $\mathbf{x}$ given any $\mathbf{x}$
- Objective of the adversary is to find the secret key

  To be able to decrypt any ciphertext intercepted

History of Cryptography

Shift Cipher

- Shift cipher based on modular arithmetic

  $a \equiv b \pmod{m}$ *if* b *divides* $b - a$ *(a congruent to b modulo m)*

- Shifting letters from the alphabet to get the ciphertext

  Commonly referred to as the Caesar Cipher for $K = 3$

- Formal definition for English alphabet (26 letters)
  - $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$
  - $e_K(x) = (x + K) \mod 26$ $\qquad (x \in \mathbb{Z}_{26})$
  - $d_K(y) = (y - K) \mod 26$ $\qquad (y \in \mathbb{Z}_{26})$

Shift Cipher Example

- Encrypt ordinary English text thanks to a correspondence

  Alphabetic characters associated to residues mod 26 *(*$A \leftrightarrow 0$*...)*

- Formal definition of Caesar Cipher
  - $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$ and $K = 3$
  - $e_3(x) = (x + 3) \mod 26$
  - $d_3(y) = (y - 3) \mod 26$

- Encryption example
  - Plaintext "`hello`" corresponds to $\mathbf{x} = (8, 5, 12, 12, 15)$
  - Ciphertext $\mathbf{y} = (11, 8, 15, 15, 18)$ is "`khoor`"

Practical Cryptosystem

- Practical cryptosystem should satisfy certain properties
  1. Good time and space complexities for e_K and d_K
  2. An opponent with y should be unable to determine x or K
- Several properties should also be satisfied on keys

  The size of the keyspace must be as large as possible

Shift Cipher Cryptanalysis

- Finding parameters of a cryptosystem by cryptanalysis

  By analysing the public elements, such as ciphertexts

- Exhaustive key search is efficient for shift cipher
  - Cracked after trying $26/2 = 13$ decryption rules on average
  - The secret key K is also found with this technique!

Substitution Cipher

- Substituting letters from the alphabet by others

  Similar to "cryptogram puzzles" found in newspapers

- Formal definition for English alphabet (26 letters)
  - $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$
  - $\mathcal{K}$ is the set of all possible permutations π on 26 symbols 0, 1...
  - $e_\pi(x) = \pi(x)$
  - $d_\pi(y) = \pi^{-1}(y)$ (*where* π^{-1} *is the inverse permutation to* π)
- Shift Cipher is a particular case of the Substitution Cipher

  It only includes 26 *of the* 26! *possible permutations*

Substitution Cipher Example

- Formal definition
  - $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$
  -

z	1	2	3	4	5	6	7	8	9	10	11	12	13
$\pi_P(z)$	16	26	1	19	4	22	7	15	24	2	13	5	20

z	14	15	16	17	18	19	20	21	22	23	24	25	26
$\pi_P(z)$	10	23	3	18	8	14	25	21	11	6	9	17	12

  - $e_{\pi_P}(x) = \pi_P(x)$
  - $d_{\pi_P}(y) = \pi_P^{-1}(y)$
- Encryption example
  - Plaintext "`hello`" corresponds to $\mathbf{x} = (8, 5, 12, 12, 15)$
  - Ciphertext $\mathbf{y} = (15, 4, 5, 5, 23)$ is "`odeew`"

Substitution Cipher Cryptanalysis

- Exhaustive key search takes a lot of time since $|\mathcal{K}| = 26!$

  All the possible permutations of 26 *letters, more than* 4.0×10^{26}

- Can be cryptanalysed with the letter occurrences frequencies
  - Comparison with the frequencies of the used natural language
  - Analysing the frequencies of bigrams (pairs of letters)

Affine Cipher (1)

- Encryption functions are restricted to affines functions

  $e(x) = (ax + b) \mod m$, *with* $a, b \in \mathbb{Z}_m$

- Integers relatively prime if they have no common dividers

  Integers a *and* b *are relatively prime iff* $gcd(a, b) = 1$

- Number of $x \in \mathbb{Z}_m$ prime with m with Euler's totient function

  $$\phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1}) \qquad \text{where } m = \prod_{i=1}^{n} p_i^{e_i}$$

- Multiplicative inverse of $a \in \mathbb{Z}_m$, denoted $a^{-1} \in \mathbb{Z}_m$ such that

  $a \cdot a^{-1} \equiv a^{-1} \cdot a \equiv 1 \pmod{m}$ *(only exists if* $gcd(a, m) = 1$*)*

Affine Cipher (2)

- Formal definition for English alphabet (26 lettres)
  - $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$
  - $\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : gcd(a, 26) = 1\}$
  - $e_K(x) = (ax + b) \mod 26$ (*where* $K = (a, b)$)
  - $d_K(y) = a^{-1}(y - b) \mod 26$ (*where* $K = (a, b)$)
- There are $|\mathcal{K}| = 26\phi(26)$ possible keys

  Since 26 choices for b and then $\phi(26)$ choices for a
- Affine Cipher also a particular case of the Substitution Cipher

  It is therefore also sensitive to cryptanalysis by frequencies

Affine Cipher Example

- Formal definition
  - $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$
  - Let's choose the key $K = (5, 2)$ (*we have* $gcd(5, 26) = 1$)
  - $e_{(5,2)}(x) = (5x + 2) \mod 26$
  - $d_{(5,2)}(y) = 21(y - 2) \mod 26$ (*since* $5 \cdot 21 \equiv 1 \pmod{26}$)
- Encryption example
  - Plaintext "`hello`" corresponds to $\mathbf{x} = (8, 5, 12, 12, 15)$
  - Ciphertext $\mathbf{y} = (16, 1, 10, 10, 25)$ is "`pajjy`"

Vigenère Cipher

- A different substitution for each m letters of the plaintext

  As opposed to previous ciphers that are monoalphabetic ones

- Formal definition for English alphabet (26 letters)
  - $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$ (*where* $m \in \mathbb{N}_0$)
  - $e_K(x_1, ..., x_m) = (x_1 + k_1, ..., x_m + k_m)$ (*where* $K = (k_1, ..., k_m)$)
  - $d_K(y_1, ..., y_m) = (y_1 - k_1, ..., y_m - k_m)$ (*where* $K = (k_1, ..., k_m)$)

- One letter can be switched to m other distinct ones
  - The key is a string with length m called keyword
  - The process is referred to as a polyalphabetic cipher

Vigenère Cipher Example

- Formal definition
  - $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^5$
  - Let's choose $K = (8, 15, 21, 19, 5)$, that is, keyword "`house`"
  - $e_K(\mathbf{x}) = (x_1 + k_1, ..., x_5 + k_5)$
  - $d_K(\mathbf{y}) = (y_1 - k_1, ..., y_5 - k_5)$
- Encryption example
  - Plaintext "`hello`" corresponds to $\mathbf{x} = (8, 5, 12, 12, 15)$
  - Ciphertext $\mathbf{y} = (16, 20, 7, 5, 20)$ is "`ptget`"

Vigenère Cipher Cryptanalysis

- Exhaustive key search takes a lot of time since $|\mathcal{K}| = 26^m$

  Strings with m letters chosen from 26 with repetition

- Cryptanalysis first has to find the length of the keyword m

  Possible with Kasiski test or coincidence index

- Then it has to find the keyword $K = (k_1, ..., k_m)$

  From the mutual coincidence index of two strings

Permutation Cipher

- Altering the positions of the m letters of the plaintext

  The characters of the plaintext are therefore kept unchanged

- Formal definition for English alphabet (26 letters)
  - $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$
  - $\mathcal{K}$ is the set of all possible permutations π of $\{1, ..., m\}$
  - $e_\pi(x_1, ..., x_m) = (x_{\pi(1)}, ..., x_{\pi(m)})$
  - $d_\pi(y_1, ..., y_m) = (y_{\pi^{-1}(1)}, ..., y_{\pi^{-1}(m)})$

    (*where* π^{-1} *is the inverse permutation to* π)

Permutation Cipher Example

- Formal definition
  - $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^5$
  -

z	1	2	3	4	5
$\pi_S(z)$	2	4	1	5	3

  - $e_{\pi_S}(\mathbf{x}) = (x_2, x_4, x_1, x_5, x_3)$
  - $d_{\pi_S}(\mathbf{y}) = (y_3, y_1, y_5, y_2, y_4)$
- Encryption example
  - Plaintext "`hello`" corresponds to $\mathbf{x} = (8, 5, 12, 12, 15)$
  - Ciphertext $\mathbf{y} = (5, 12, 8, 15, 12)$ is "`elhol`"

Block and Stream Cipher

- Block Cipher encrypts successive plaintexts with same key K

  $\mathbf{y} = y_1 y_2 \ldots = e_K(x_1) e_K(x_2) \ldots$

- Stream Cipher uses a keystream $\mathbf{z} = z_1 z_2 \ldots$

  $\mathbf{y} = y_1 y_2 \ldots = e_{z_1}(x_1) e_{z_2}(x_2) \ldots$

- Two types of stream cipher schemes
  - Synchronous way derives a keystream from a single key
  - Periodic way with period d if $z_i = z_{i+d}$ for all $i \geq 1$

Synchronous Stream Cipher (1)

- Vigenère Cipher defined as a synchronous stream cipher
  - $z_i = \begin{cases} k_i & \text{if } 1 \leq i \leq m \\ z_{i-m} & \text{if } 1 \geq m+1 \end{cases}$ (*for a keyword with length* m)
  - Periodic keystream with period m: $k_1 k_2 \ldots k_m k_1 k_2 \ldots k_m k_1 k_2 \ldots$
- Stream ciphers often defined on binary alphabet $\mathcal{P} = \mathcal{C} = \mathbb{Z}_2$
  - Additions modulo 2 : $e_z(x) = (x + z) \mod 2$
    $d_z(y) = (y + z) \mod 2$
  - XOR operation implemented very efficiently in hardware

Synchronous Stream Cipher (2)

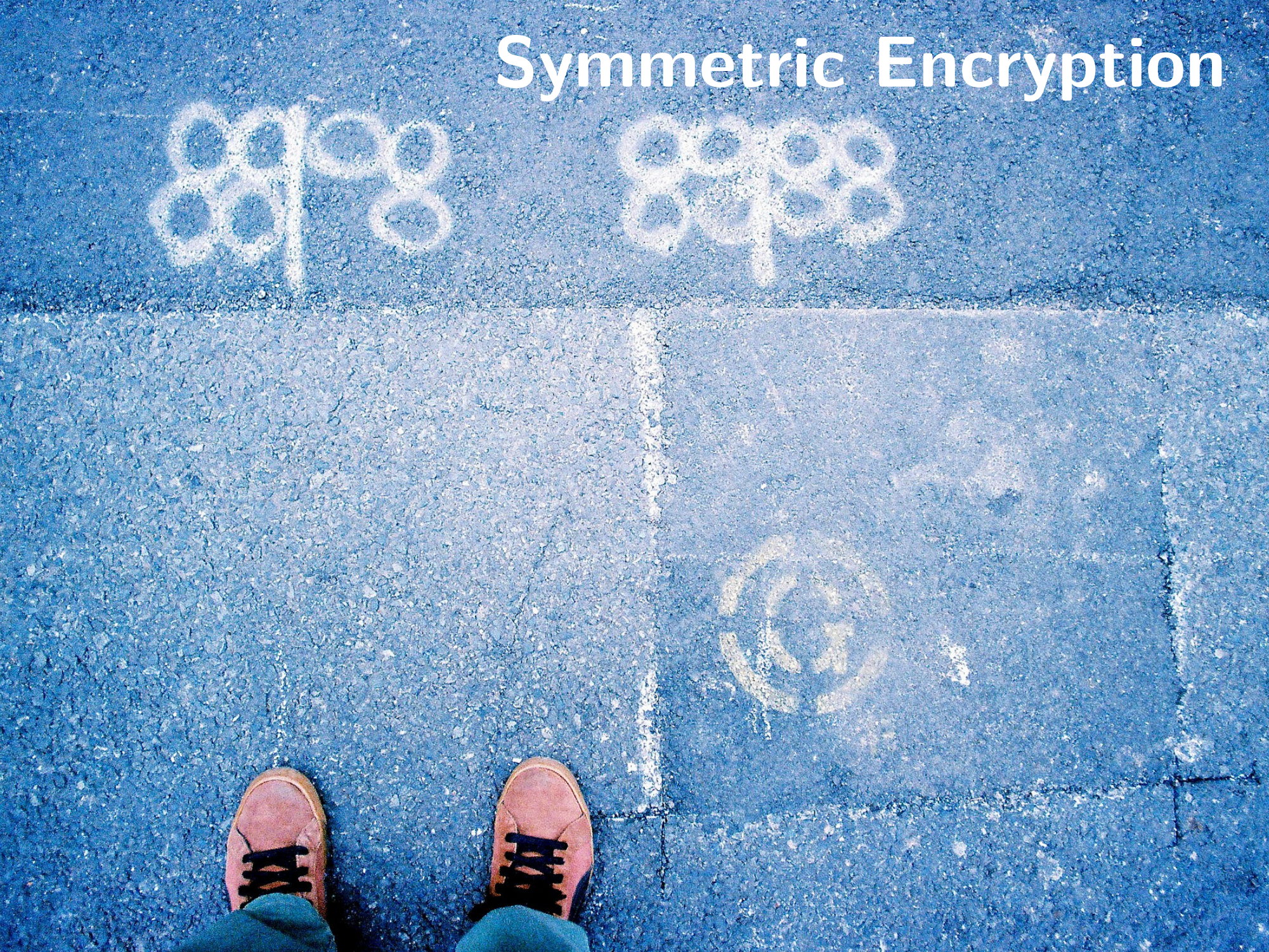
- Keystream generated by a linear recurrence of degree m
  - $z_{i+m} = \sum_{j=0}^{m-1} c_j z_{i+j} \mod 2$, with $c_j \in \mathbb{Z}_2$ specified constants
  - Key consists of the $2m$ values $k_1, ..., k_m, c_0, ..., c_{m-1}$
  - If c_j carefully chosen, smallest possible period will be $2^m - 1$
  - A "short" key can give rise to a keystream with a long period
- Keystream obtained efficiently with shift register hardware

  Linear Feedback Shift Register (LFSR) with m stages

Non-synchronous Stream Cipher

- Stream Cipher can also be non-synchronous
  - Depends on previous plaintext or ciphertext ($x_1 \ldots$ and/or $y_1 \ldots$)
  - And also depends on the key K
- Autokey Cipher developped by Vigenère
  - Using the plaintext to construct the keystream (aside of K)
  - Very insecure since there are only 26 possible keys
  - Keystream $z_1 = K$, $z_i = x_{i-1}$ and $e_z(x) = (x + z) \mod 26$, $d_z(y) = (y - z) \mod 26$
- Plaintext "`rendezvous`" with $K = 8$ gives "`zvrqhdujim`"

Symmetric Encryption



Product Cryptosystem

- Given two cryptosystems $\mathbf{S}_\star = \langle \mathcal{P}, \mathcal{C}, \mathcal{K}_\star, \mathcal{E}_\star, \mathcal{D}_\star \rangle$

  Both with the same plaintext and ciphertext spaces $\mathcal{P} = \mathcal{C}$

- Possible to defined the product $\mathbf{S}_1 \times \mathbf{S}_2$ cryptosystem
  - $\mathbf{S}_1 \times \mathbf{S}_2 = \langle \mathcal{P}, \mathcal{C}, \mathcal{K}_1 \times \mathcal{K}_2, \mathcal{E}, \mathcal{D} \rangle$
  - Keys are pairs $K = (K_1, K_2)$ with $K_1 \in \mathcal{K}_1$ and $K_2 \in \mathcal{K}_2$
  - Encryption function $e_{(K_1,K_2)}(x) = e_{K_2}(e_{K_1}(x))$
  - And decryption function is $d_{(K_1,K_2)}(y) = d_{K_1}(d_{K_2}(y))$

Iterated Cipher (1)

- Modern block cipher algorithm are product based

  Based on a sequence of permutation and substitution operations

- Iterated Cipher composed of two elements
  - **Round function** g for the $\mathcal{N}$ similar rounds
  - **Key schedule** algorithm to construct round keys $(K^1, ..., K^{\mathcal{N}})$

- Round function g takes two inputs and must be injective

  $g^{-1}(g(w, y), y) = w$

Iterated Cipher (2)

- Encryption operation of a plaintext applying the $\mathcal{N}$ rounds

$$
\begin{aligned}
&w^0 \leftarrow x \\
&w^1 \leftarrow g(w^0, K^1) \\
&\dots \\
&w^i \leftarrow g(w^{i-1}, K^i) \\
&\dots \\
&y \leftarrow w^{\mathcal{N}}
\end{aligned}
$$

- Decryption operation following the opposite process

$$
\begin{aligned}
&w^{\mathcal{N}} \leftarrow y \\
&\dots \\
&w^i \leftarrow g^{-1}(w^{i+1}, K^{i+1}) \\
&\dots \\
&w^0 \leftarrow g^{-1}(w^1, K^1) \\
&x \leftarrow w^0
\end{aligned}
$$

Substitution–Permutation Network (1)

- Substitution–Permutation Network (SPN)

  Special type of iterated cipher with a couple of small changes

- Characterised by two values $\ell, m \in \mathbb{N}_0$ with
  - Substitution π_S: $\{0,1\}^\ell \to \{0,1\}^\ell$ (*S-box*)
  - Permutation π_P: $\{1, ..., \ell m\} \to \{1, ..., \ell m\}$

- Plaintext/ciphertext are vectors with block length ℓm
  - Binary string with ℓm bits: $x = (x_1, ..., x_{\ell m})$
  - Or the concatenation of m substrings with ℓ bits: $x = x_{\langle 1 \rangle} \| ... \| x_{\langle m \rangle}$ with $x_{\langle i \rangle} = (x_{(i-1)\ell+1}, ..., x_{i\ell})$

Substitution–Permutation Network (2)

- SPN with $\mathcal{N}$ rounds, each except last performing

  m substitutions with π_S followed by a permutation with π_P

Algorithm 1: Substitution–Permutation Network

Function $SPN(x, \pi_S, \pi_P, (K^1, ..., K^{\mathcal{N}+1}))$

- $w^0 \leftarrow x$
- **for** $r \leftarrow 1$ **to** $\mathcal{N} - 1$ **do**
  - $u^r \leftarrow w^{r-1} \oplus K^r$
  - **for** $i \leftarrow 1$ **to** m **do**
    - $v^r_{\langle i \rangle} \leftarrow \pi_S(u^r_{\langle i \rangle})$
  - $w^r \leftarrow (v^r_{\pi_P(1)}, ..., v^r_{\pi_P(\ell m)})$
- $u^{\mathcal{N}} \leftarrow w^{\mathcal{N}-1} \oplus K^{\mathcal{N}}$
- **for** $i \leftarrow 1$ **to** m **do**
  - $v^{\mathcal{N}}_{\langle i \rangle} \leftarrow \pi_S(u^{\mathcal{N}}_{\langle i \rangle})$
- $y \leftarrow v^{\mathcal{N}} \oplus K^{\mathcal{N}+1}$
- **return** y

SPN Example (1)

- Complete example of a Substitution–Permutation Network
  - $\ell = 3$ and $m = 4$, that is, messages with $\ell m = 12$ bits
  - π_S defined as follows (3-bit binary words):

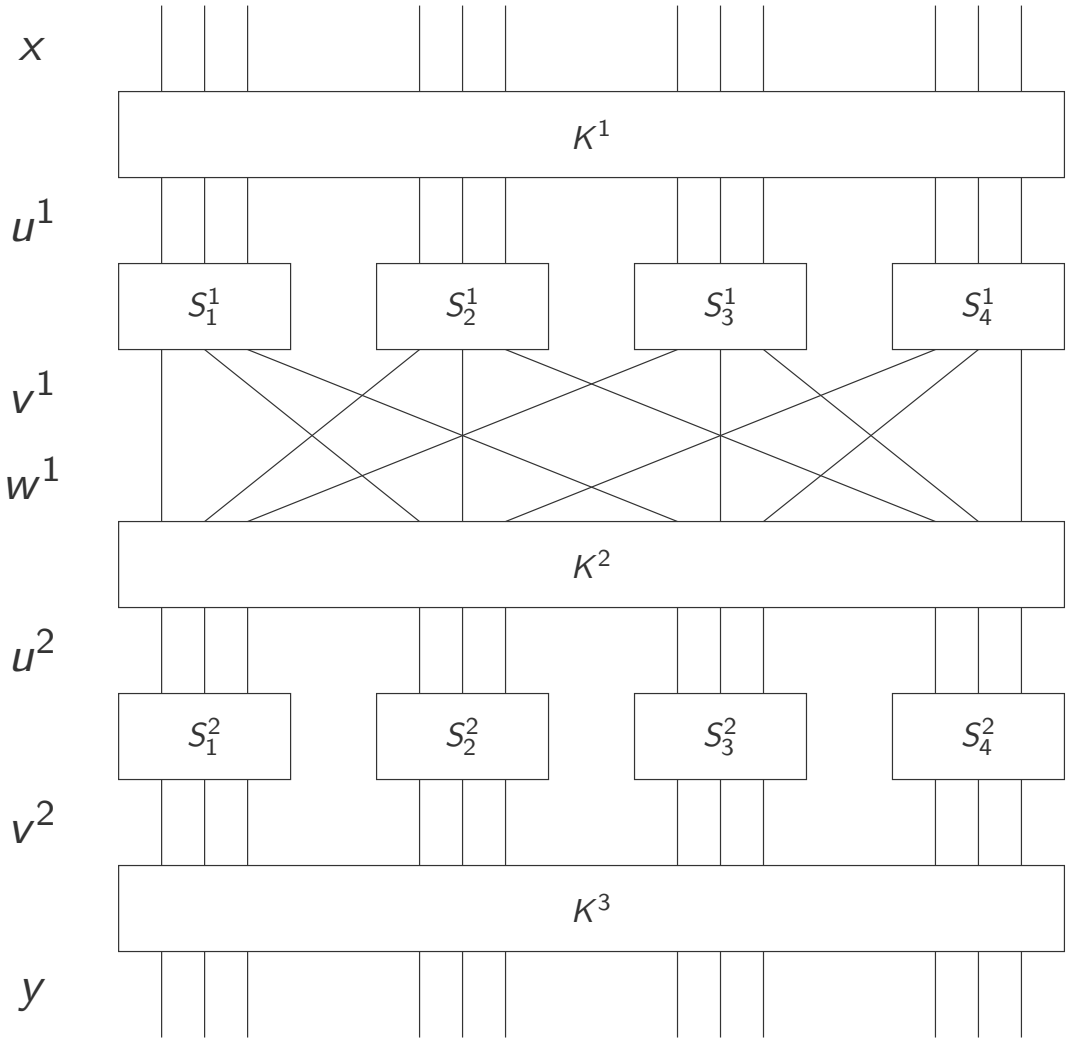
z	0	1	2	3	4	5	6	7
$\pi_S(z)$	4	2	5	6	0	3	7	1

  - π_P defined as follows (on bits of 12-bit binary word):

z	1	2	3	4	5	6	7	8	9	10	11	12
$\pi_P(z)$	1	4	7	2	5	10	2	8	11	6	9	12

  - Key $K = 001\ 110\ 101\ 001\ 010\ 011$
  - Key schedule with K^i the 12 bits starting at k_{3i-2}

SPN Example (2)



001 101 010 110

001 110 101 001

000 011 111 111

100 110 001 001

110 010 000 011

110 101 001 010

000 111 001 001

100 001 010 010

101 001 010 011

001 000 000 001

Data Encryption Standard (DES) (1)

- Special type of Iterated Cipher called Feistel Cipher

  Published on March 17, 1975 and adopted on January 15, 1977

- Basic form of a Feistel Cipher
  - Each state u^i divided into two halves of equal length L^i and R^i
  - Function f does not need to be injective
  - Round function g has the following form:

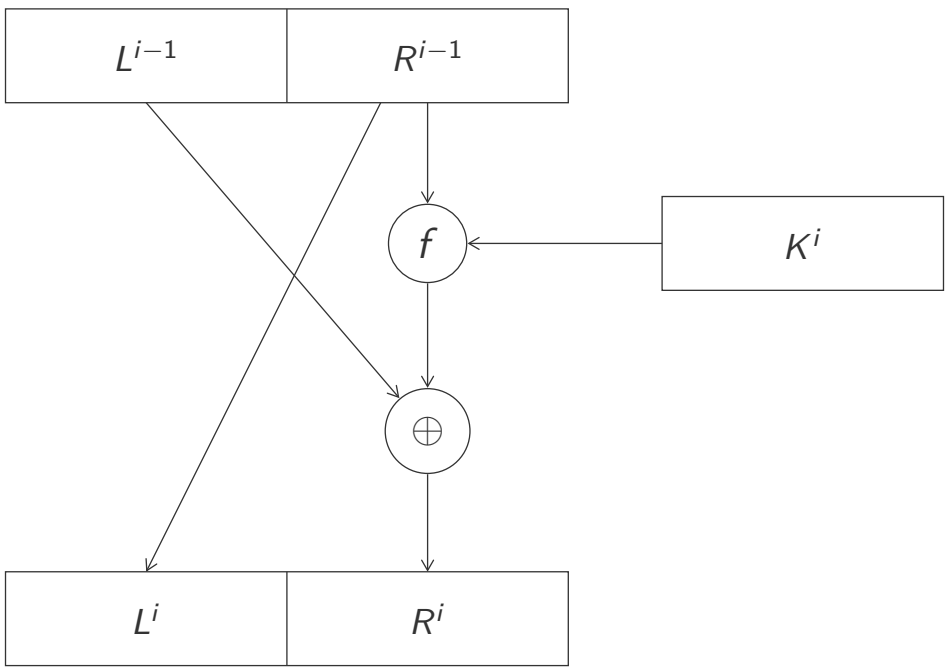
$$g(L^{i-1}, R^{i-1}, K^i) = (L^i, R^i)$$
$$= (R^{i-1}, L^{i-1} \oplus f(R^{i-1}, K^i))$$

DES Round

- One DES round inverts and combines left and right parts

Such a round is always invertible

$$\begin{aligned} L^{i-1} &= R^i \oplus f(L^i, K^i) \\ R^{i-1} &= L^i \end{aligned}$$



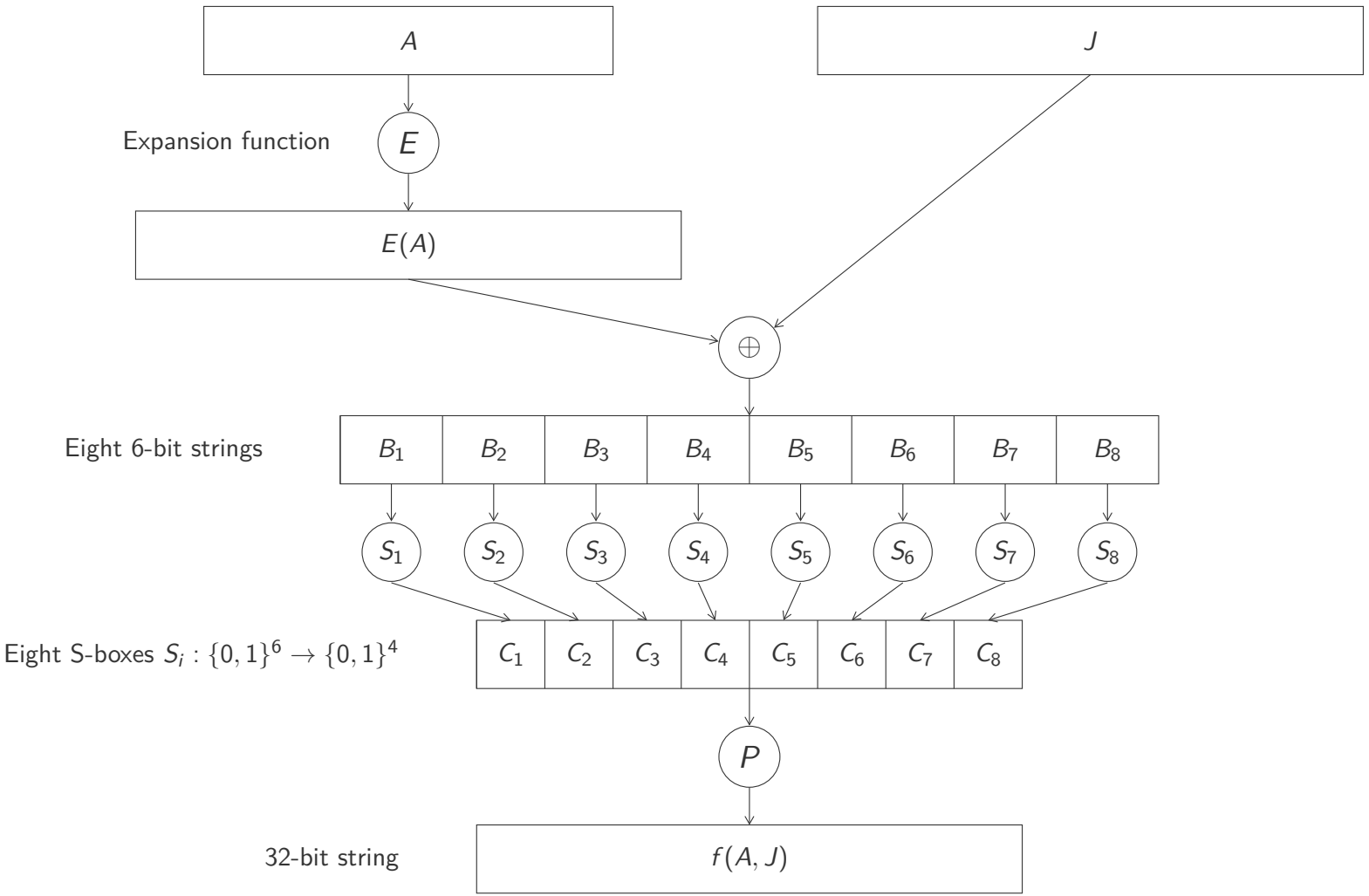
Data Encryption Standard (DES) (2)

- Data Encryption Standard (DES) is a Feistel Cipher

  With 16 *rounds*, 64-*bit block length and* 56-*bit key*

- Operations before and after the 16 rounds
  - Initial permutation **IP** on plaintext: $\mathbf{IP}(x) = L^0 R^0$
  - Final inverse permutation $\mathbf{IP}^{-1}$: $y = \mathbf{IP}^{-1}(R^{16} L^{16})$
- f function takes right part of current state and round key
  - $f : \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$
  - A substitution (with a S-box) followed by a (fixed) permutation

DES f Function



Advanced Encryption Standard (AES)

- NIST began process of replacing DES on January 2, 1997
  - Block length of 128 bits, support keys of 128, 192 and 256 bits
  - Rijndael submission selected, adopted on November 26, 2001
  - Open and international selection process
- Iterated Cipher with $\mathcal{N} = 10, 12, 14$ depending on key length

  Operations required by AES similar to those of SPN


Symmetric Cryptography
Application

Symmetric Cryptography

- Type of encryption where a single secret key is used

  Must be exchanged between communicating entities

- Two different types of symmetric cryptographic algorithms
  - Block algorithms encrypts data block by block in memory
  - Stream algorithms encrypts data as it streams

- Many different algorithms have been proposed and developed

  IDEA, DES, AES, Blowfish, RC5, RC6 (blocks) and RC4 (stream)

Advantage and Drawback

- Fast and very efficient to compute

  Better use of networks and CPU power

- Typically used for bulk encryption of large amount of data

  For example, database encryption with key stored in the engine

- Very important drawback is the key management

  Key exhaustion, attribution data, scalability...

Other Application

- Information validation when receiving a message

  If decrypted message correct, some guarantee about sender

- A fingerprint of a message can be computed as a hash

  Due to the highly complex plaintext/ciphertext relationship

- Basis for Random Number Generation (RNG)

  For not sensible applications

References

- Douglas R. Stinson, & Maura B. Paterson, *Cryptography: Theory and Practice* (Fourth Edition), CRC Press, 2017. (ISBN: 978-1-138-19701-5)
- steve, *Cryptography with Alice and Bob*, September 17, 2014. https://wordtothewise.com/2014/09/cryptography-alice-bob
- Ray Alderman, *Cryptology, cryptography, and cryptanalysis*, December 22, 2015. http://mil-embedded.com/guest-blogs/cryptology-cryptography-and-cryptanalysis
- parserite, *Cryptography for Absolute Beginners*, October 17, 2018. https://medium.com/@hashelse/cryptography-for-absolute-beginners-3e274f9d6d66
- Ahsan Barkati, *A complete description of Data Encryption Standard (DES)*, February 26, 2019. https://medium.com/@ahsanbarkati/the-des-data-encryption-standard-16466b45c30d
- zeroFruit, *What is AES? ? Step by Step*, February 13, 2019. https://medium.com/@14wnrkim/what-is-aes-step-by-step-fcb2ba41bb20
- Peter Smirnoff, & Dawn M. Turner, *Symmetric Key Encryption - why, where and how it?s used in banking*, January 18, 2019. https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking

Credits

- Icons from https://icons8.com/icons.
- Adam Foster, December 7, 2011, https://www.flickr.com/photos/twosevenoneonenineeightthreesevenatenzerosix/6655759625.
- Ruth Tate, March 7, 2008, https://www.flickr.com/photos/roneal/2322161594.
- Ishikawa Ken, January 19, 2013, https://www.flickr.com/photos/chidorian/8967757788.
- Blogtrepreneur, September 29, 2016, https://www.flickr.com/photos/143601516@N03/29972713206.