

# Symmetric Cryptography

## Cryptographic Primitives

Design and Verification of Security Protocols and Security  
Ceremonies

Programa de Pós-Graduação em Ciências da Computação  
Dr. Jean Everson Martina

August-November 2016



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA

# Cryptography Aims

- To provide confidentiality;

# Cryptography Aims

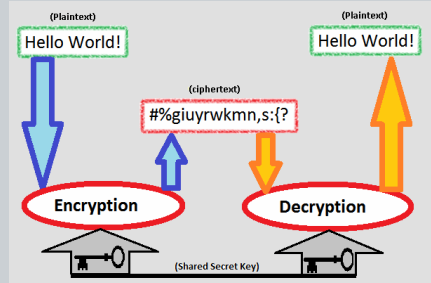
- To provide confidentiality;
- Securely exchange information in an insecure environment;

# Cryptography Aims

- To provide confidentiality;
- Securely exchange information in an insecure environment;
- Only valid users know how to perform decryption.

# Cryptography Aims

- To provide confidentiality;
- Securely exchange information in an insecure environment;
- Only valid users know how to perform decryption.



# Symmetric Cryptography Setting

- Encryption and Decryption are specified algorithms;

# Symmetric Cryptography Setting

- Encryption and Decryption are specified algorithms;
- Two approaches:

# Symmetric Cryptography Setting

- Encryption and Decryption are specified algorithms;
- Two approaches:
  - Keep encryption and decryption algorithms secret:  
*"security through obscurity"*;



# Symmetric Cryptography Setting

- Encryption and Decryption are specified algorithms;
- Two approaches:
  - Keep encryption and decryption algorithms secret:  
*"security through obscurity"*;
  - Add another variable, called a *key* that must be known to encrypt and decrypt a message;

# Symmetric Cryptography Setting

- Encryption and Decryption are specified algorithms;
- Two approaches:
  - Keep encryption and decryption algorithms secret:  
*"security through obscurity"*;
  - Add another variable, called a *key* that must be known to encrypt and decrypt a message;
- Symmetric-key cryptography uses same key used for both encryption and decryption.

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;
- We can not send key in a separate message because:

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;
- We can not send key in a separate message because:
  - If unencrypted an adversary could intercept it and learn the key;

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;
- We can not send key in a separate message because:
  - If unencrypted an adversary could intercept it and learn the key;
  - If encrypted it would require a key that had to be distributed somehow;

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;
- We can not send key in a separate message because:
  - If unencrypted an adversary could intercept it and learn the key;
  - If encrypted it would require a key that had to be distributed somehow;
- We usually assume the key is known;

# Symmetric Cryptography Problem

- Symmetric-key cryptography requires that the recipient of an encrypted message knows the key;
- We can not send key in a separate message because:
  - If unencrypted an adversary could intercept it and learn the key;
  - If encrypted it would require a key that had to be distributed somehow;
- We usually assume the key is known;
- We will talk about key distribution protocols in the next lectures.



# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;

# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;
- Substitution cipher: replace individual characters with different characters;

# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;
- Substitution cipher: replace individual characters with different characters;
- Permutation cipher: change the order of the characters using permutations;

# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;
- Substitution cipher: replace individual characters with different characters;
- Permutation cipher: change the order of the characters using permutations;
- Good modern symmetric ciphers use substitution and permutation;

# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;
- Substitution cipher: replace individual characters with different characters;
- Permutation cipher: change the order of the characters using permutations;
- Good modern symmetric ciphers use substitution and permutation;
- Its been the study of military schools for a long time;

# Classical Cryptography

- As remote as Julius Ceasar in ancient Rome or Sparta in ancient Greece;
- Substitution cipher: replace individual characters with different characters;
- Permutation cipher: change the order of the characters using permutations;
- Good modern symmetric ciphers use substitution and permutation;
- Its been the study of military schools for a long time;
- Developed the area of Cryptanalysis.

# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;

# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;
- Problems are that:



# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;
- Problems are that:
  - Characters and words do not enough entropy;

# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;
- Problems are that:
  - Characters and words do not enough entropy;
  - Context defines how characters are used and when they are used;

# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;
- Problems are that:
  - Characters and words do not enough entropy;
  - Context defines how characters are used and when they are used;
- You attack such ciphers by measuring the distribution on cipher-text and comparing it with a target language;

# Mono-alphabetic Ciphers

- Same encryption algorithm used for each character;
- Problems are that:
  - Characters and words do not enough entropy;
  - Context defines how characters are used and when they are used;
- You attack such ciphers by measuring the distribution on cipher-text and comparing it with a target language;
- It can be made more efficient by using context of doubles and triples.

# Poly-alphabetic Ciphers

- Change the substitution pattern on a character-by-character basis;

# Poly-alphabetic Ciphers

- Change the substitution pattern on a character-by-character basis;
- Uses a key to configure the selection of various mono-alphabetic ciphers;

# Poly-alphabetic Ciphers

- Change the substitution pattern on a character-by-character basis;
- Uses a key to configure the selection of various mono-alphabetic ciphers;
- We can not apply frequency analysis directly because the translations scheme changes every new character we input;

# Poly-alphabetic Ciphers

- Change the substitution pattern on a character-by-character basis;
- Uses a key to configure the selection of various mono-alphabetic ciphers;
- We can not apply frequency analysis directly because the translations scheme changes every new character we input;
- You attack if the user reuses the key throughout the encryption process;



# Poly-alphabetic Ciphers

- Change the substitution pattern on a character-by-character basis;
- Uses a key to configure the selection of various mono-alphabetic ciphers;
- We can not apply frequency analysis directly because the translations scheme changes every new character we input;
- You attack if the user reuses the key throughout the encryption process;
- You find the period of the key and then use frequency analysis to crack all the mono alphabetic cipher independently.

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems the led to Enigma being cracked are:

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems that led to Enigma being cracked are:
  - Characters never mapped to themselves;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems that led to Enigma being cracked are:
  - Characters never mapped to themselves;
  - $E() = D()$ ;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems that led to Enigma being cracked are:
  - Characters never mapped to themselves;
  - $E() = D()$ ;
  - Operator errors;



# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems that led to Enigma being cracked are:
  - Characters never mapped to themselves;
  - $E() = D()$ ;
  - Operator errors;
  - Known plain-text;

# Electro-Mechanical Ciphers

- The enigma machine challenged the boundaries of computable functions;
- It is a machine to assist the usage of poly-alphabetic ciphers;
- Every time a key is pressed rotors turn to change the substitution pattern;
- The problems that led to Enigma being cracked are:
  - Characters never mapped to themselves;
  - $E() = D()$ ;
  - Operator errors;
  - Known plain-text;

# Shannon's Information Theory

- In 1948, Claude Shannon invented the basis of Information Theory in his publication “A Mathematical Theory of Communication;

# Shannon's Information Theory

- In 1948, Claude Shannon invented the basis of Information Theory in his publication “A Mathematical Theory of Communication;
- Provably secure means the key must be just as random as the cipher-text;

# Shannon's Information Theory

- In 1948, Claude Shannon invented the basis of Information Theory in his publication “A Mathematical Theory of Communication;
- Provably secure means the key must be just as random as the cipher-text;
- This would lead to an infinite number of possible meaningful decryption;

# Shannon's Information Theory

- In 1948, Claude Shannon invented the basis of Information Theory in his publication “A Mathematical Theory of Communication;
- Provably secure means the key must be just as random as the cipher-text;
- This would lead to an infinite number of possible meaningful decryption;
- One-Time Pad can be implemented with Viginere or XOR;

# Shannon's Information Theory

- In 1948, Claude Shannon invented the basis of Information Theory in his publication “A Mathematical Theory of Communication;
- Provably secure means the key must be just as random as the cipher-text;
- This would lead to an infinite number of possible meaningful decryption;
- One-Time Pad can be implemented with Viginere or XOR;
- Provably secure if you do not reuse the key;

# Stream Ciphers How-To

- Start with a secret key to seed a generator;



# Stream Ciphers How-To

- Start with a secret key to seed a generator;
- Generate a keying stream that do not repeat itself;

# Stream Ciphers How-To

- Start with a secret key to seed a generator;
- Generate a keying stream that do not repeat itself;
- The  $i$ -th bit of keying stream is a function of the key and the first  $i-1$  cipher-text bits;

# Stream Ciphers How-To

- Start with a secret key to seed a generator;
- Generate a keying stream that do not repeat itself;
- The  $i$ -th bit of keying stream is a function of the key and the first  $i-1$  cipher-text bits;
- Combine the stream with the plain-text to produce the cipher-text (typically by XOR);

# Stream Ciphers How-To

- Start with a secret key to seed a generator;
- Generate a keying stream that do not repeat itself;
- The  $i$ -th bit of keying stream is a function of the key and the first  $i-1$  cipher-text bits;
- Combine the stream with the plain-text to produce the cipher-text (typically by XOR);
- Revert it by applying the key-stream again.

# Stream Ciphers Implementations

- Most pre computer tools used stream ciphers;

# Stream Ciphers Implementations

- Most pre computer tools used stream ciphers;
- The German Enigma machine;

# Stream Ciphers Implementations

- Most pre computer tools used stream ciphers;
- The German Enigma machine;
- Linear Feedback Shift Register;

# Stream Ciphers Implementations

- Most pre computer tools used stream ciphers;
- The German Enigma machine;
- Linear Feedback Shift Register;
- A5 – encrypting GSM handset to base station communication;



# Stream Ciphers Implementations

- Most pre computer tools used stream ciphers;
- The German Enigma machine;
- Linear Feedback Shift Register;
- A5 – encrypting GSM handset to base station communication;
- RC-4 (Ron's Code) WEP Encryption.

# Stream Ciphers Implementations

- Advantages:

# Stream Ciphers Implementations

- Advantages:
  - Speed of transformation: algorithms are linear in time and constant in space;

# Stream Ciphers Implementations

- Advantages:
  - Speed of transformation: algorithms are linear in time and constant in space;
  - Low error propagation: an error in encrypting one symbol Likely will not affect subsequent symbols;

# Stream Ciphers Implementations

- Advantages:
  - Speed of transformation: algorithms are linear in time and constant in space;
  - Low error propagation: an error in encrypting one symbol Likely will not affect subsequent symbols;
- Disadvantages:

# Stream Ciphers Implementations

- Advantages:
  - Speed of transformation: algorithms are linear in time and constant in space;
  - Low error propagation: an error in encrypting one symbol Likely will not affect subsequent symbols;
- Disadvantages:
  - Low diffusion: all information of a plain-text symbol is contained in a single cipher-text symbol;

# Stream Ciphers Implementations

- Advantages:
  - Speed of transformation: algorithms are linear in time and constant in space;
  - Low error propagation: an error in encrypting one symbol Likely will not affect subsequent symbols;
- Disadvantages:
  - Low diffusion: all information of a plain-text symbol is contained in a single cipher-text symbol;
  - Susceptibility to insertions/ modifications: an active interceptor who breaks the algorithm might insert spurious text that looks authentic.

# Block Ciphers

- Encrypt a block of input to a block of output;



# Block Ciphers

- Encrypt a block of input to a block of output;
- Typically, the two blocks are of the same length;

# Block Ciphers

- Encrypt a block of input to a block of output;
- Typically, the two blocks are of the same length;
- Symmetric key systems block size vary from 64 to 256 bits;

# Block Ciphers

- Encrypt a block of input to a block of output;
- Typically, the two blocks are of the same length;
- Symmetric key systems block size vary from 64 to 256 bits;
- Has different modes for encrypting plain-text longer than a block and this affects security.

# Block Ciphers Implementations

- DES, 3-DES;

# Block Ciphers Implementations

- DES, 3-DES;
- AES;

# Block Ciphers Implementations

- DES, 3-DES;
- AES;
- RC-2, RC-5;

# Block Ciphers Implementations

- DES, 3-DES;
- AES;
- RC-2, RC-5;
- IDEA;

# Block Ciphers Implementations

- DES, 3-DES;
- AES;
- RC-2, RC-5;
- IDEA;
- Blowfish, Twofish;



# Block Ciphers Implementations

- DES, 3-DES;
- AES;
- RC-2, RC-5;
- IDEA;
- Blowfish, Twofish;

# Block Ciphers Implementations

- Advantages:

# Block Ciphers Implementations

- Advantages:
  - High diffusion: information from one plain-text symbol is diffused into several cipher-text symbols;

# Block Ciphers Implementations

- Advantages:
  - High diffusion: information from one plain-text symbol is diffused into several cipher-text symbols;
  - Immunity to tampering: difficult to insert symbols without detection;

# Block Ciphers Implementations

- Advantages:
  - High diffusion: information from one plain-text symbol is diffused into several cipher-text symbols;
  - Immunity to tampering: difficult to insert symbols without detection;
- Disadvantages:

# Block Ciphers Implementations

- Advantages:
  - High diffusion: information from one plain-text symbol is diffused into several cipher-text symbols;
  - Immunity to tampering: difficult to insert symbols without detection;
- Disadvantages:
  - Slowness of encryption: an entire block must be accumulated before encryption/decryption can begin;

# Block Ciphers Implementations

- Advantages:
  - High diffusion: information from one plain-text symbol is diffused into several cipher-text symbols;
  - Immunity to tampering: difficult to insert symbols without detection;
- Disadvantages:
  - Slowness of encryption: an entire block must be accumulated before encryption/decryption can begin;
  - Error propagation: An error in one symbol may corrupt the entire block.

# ECB Mode of Encryption

- Simple and efficient;



# ECB Mode of Encryption

- Simple and efficient;
- Parallel implementation possible;

# ECB Mode of Encryption

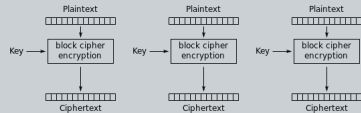
- Simple and efficient;
- Parallel implementation possible;
- Does not conceal plain-text patterns;

# ECB Mode of Encryption

- Simple and efficient;
- Parallel implementation possible;
- Does not conceal plain-text patterns;
- Active attacks are possible.

# ECB Mode of Encryption

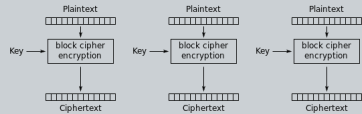
- Simple and efficient;
- Parallel implementation possible;
- Does not conceal plain-text patterns;
- Active attacks are possible.



Electronic Codebook (ECB) mode encryption

# ECB Mode of Encryption

- Simple and efficient;
- Parallel implementation possible;
- Does not conceal plain-text patterns;
- Active attacks are possible.



Electronic Codebook (ECB) mode encryption



# CBC Mode of Encryption

- It is an asynchronous stream cipher;

# CBC Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;

# CBC Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Conceals plain-text patterns into cypher-text;

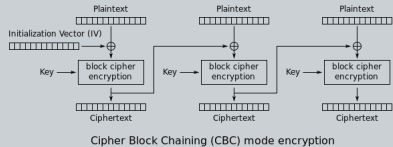


# CBC Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;

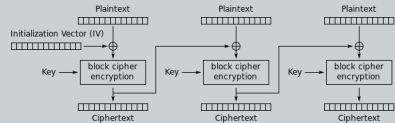
# CBC Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;
- It is difficult to manipulated Plain-text.

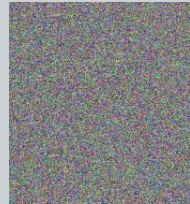


# CBC Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;
- It is difficult to manipulated Plain-text.



Cipher Block Chaining (CBC) mode encryption



# OFB Mode of Encryption

- It is an asynchronous stream cipher;

# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;

# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;

# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;
- Conceals plain-text patterns into cypher-text;

# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;

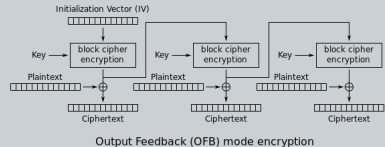


# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;
- Active attacks are possible.

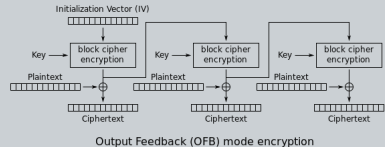
# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;
- Active attacks are possible.



# OFB Mode of Encryption

- It is an asynchronous stream cipher;
- Errors in one cipher-text block propagate to other blocks;
- Pre-processing is possible;
- Conceals plain-text patterns into cypher-text;
- Parallel implementation not known;
- Active attacks are possible.



# Hash Function

- Length-reducing function  $h$ ;

# Hash Function

- Length-reducing function  $h$ ;
- Maps an arbitrary string to a fixed-length string;

# Hash Function

- Length-reducing function  $h$ ;
- Maps an arbitrary string to a fixed-length string;
- Publicly known;

# Hash Function

- Length-reducing function  $h$ ;
- Maps an arbitrary string to a fixed-length string;
- Publicly known;
- Also known as cryptographic checksums or message digests;

# Hash Function

- Length-reducing function  $h$ ;
- Maps an arbitrary string to a fixed-length string;
- Publicly known;
- Also known as cryptographic checksums or message digests;



# Hash Properties

- Ease of computation;

# Hash Properties

- Ease of computation;
- Pre-image resistance Collision;

# Hash Properties

- Ease of computation;
- Pre-image resistance Collision;
- Collision;

# Hash Properties

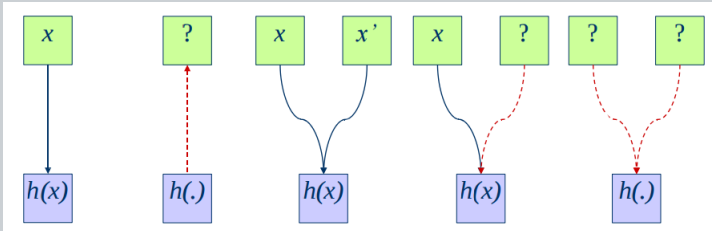
- Ease of computation;
- Pre-image resistance Collision;
- Collision;
- 2nd pre-image resistance;

# Hash Properties

- Ease of computation;
- Pre-image resistance Collision;
- Collision;
- 2nd pre-image resistance;
- Collision resistance;

# Hash Properties

- Ease of computation;
- Pre-image resistance Collision;
- Collision;
- 2nd pre-image resistance;
- Collision resistance;



# Properties of Symmetric Encryption

- Confidentiality of course!!!
- How to achieve authentication?
- And timeliness?
- And Integrity

# Questions????



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA





This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA