Introduction to Cryptography

François Koeune – Olivier Pereira

Slides 01

Goal of the course

Understand fundamental

- concepts,
- methods, and
- algorithms

used to secure information, with an emphasis on the *algorithmic* and *mathematical* aspects.

Related courses at UCL

Option in *Cryptography and Information Security*
(EPL – DATA/ELEC/INFO/MAP)

- **LELEC2760** — Secure electronic circuits and systems – F.-X. Standaert
- **LELEC2770** — Privacy Enhancing Technologies – O. Pereira, F.-X. Standaert
- **LINGI2144** — Secured systems engineering – A. Legay
- **LINGI2347** — Computer System Security – R. Sadre
- **LINGI2348** — Information theory and coding – J. Louveaux, B. Macq, O. Pereira
- **LMAT2440** — Théorie des nombres – O. Pereira, J.-P. Tignol
- **LMAT2450** — Cryptography – O. Pereira

Related courses at UCL

Other related courses

- **LELEC2870** — Machine Learning – J. Lee, M. Verleysen
- **LINGI1341** — Computer networks – O. Bonaventure
- **LINMA2111** — Discrete mathematics II : Algorithms and complexity – J.-C. Delvenne
- **LEPL2210** — Ethics and ICT – A. Gosseries, O. Pereira

Class Organisation

- Lectures/Exercises on Wednesday, 14:00 – 16:00 Exercises on Wednesday, 16:15 – 18:15
- TAs: Clément Hoffmann, Yaobin Shen
- We may offer homeworks:
  - make up to 20% of the January grade, if this helps you
  - do not count in August
- Examination: exercises. Slides and personal notes allowed Exam questions from past years often proposed as exercises

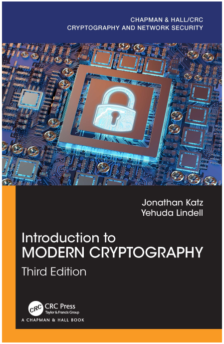
Syllabus

Expected distribution:

- Introduction (1 lecture)
- Symmetric cryptography (4 lectures)
- Asymmetric cryptography and algorithmic number theory (4 lectures)
- Protocols (2 lectures)

Support

Introduction to Modern Cryptography (2nd edition)
by J. Katz and Y. Lindell – Chapman & Hall/CRC – 2020



`http://www.cs.umd.edu/~jkatz/imc.html`

Support

Other references (see also Moodle):

- W. Mao, *Modern Cryptography, Theory and Practice*, Prentice-Hall, PTR, 2003.
- D. Stinson, *Cryptography, Theory and Practice*, 3rd edition, Chapman & Hall/CRC, 2005.
- A.J. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, 1999. Free on `http://www.cacr.math.uwaterloo.ca/hac/`.
- D. Boneh, V. Shoup, *A Graduate Course in Applied Cryptography*, Free draft on `http://toc.cryptobook.us/`
- N. Koblitz, *A Course in Number Theory and Cryptography*, Graduate Texts in Math. No. 114, 2nd edition, Springer-Verlag, 1994.

Cryptography...

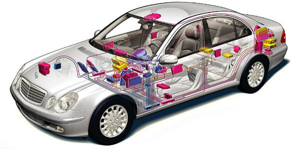
COD defines *cryptography* as:

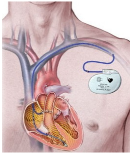
> *"the art of writing or solving codes."*

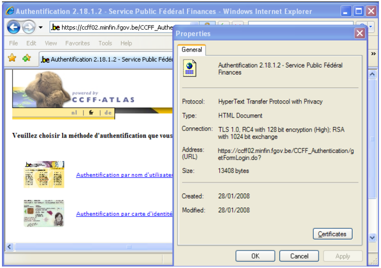
- Certainly true until mid of 20^{th} century
- Mostly used by armies and diplomats

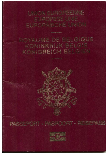
Cryptography... today

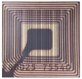
Used every day!











Cryptography… today

Much more than encryption:

- authentication
- key exchange
- identification
- elections
- Yao millionaire's problem
- ...

Cryptography... today

Much more than encryption:

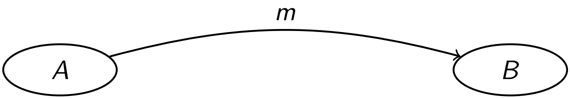
- authentication
- key exchange
- identification
- elections
- Yao millionaire's problem
- ...

From an *art*, cryptography became a *science*...

The message encryption problem

The setting:

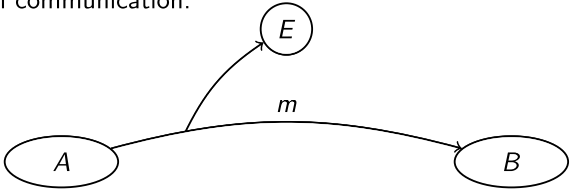
- Plain communication:



The message encryption problem

The setting:

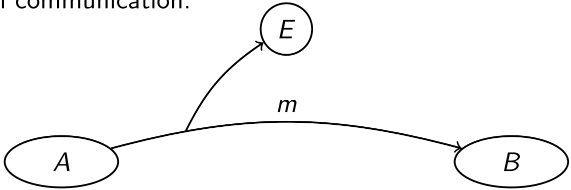
- Plain communication:



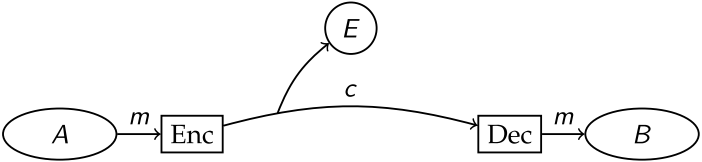
The message encryption problem

The setting:

- Plain communication:



- Encrypted communication:



Message Encryption

What is an encryption scheme? A triple $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$

Message Encryption

What is an encryption scheme? A triple $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$

- Gen probabilistically selects a key k
- Enc encrypts message m with key k, as $c := \mathrm{Enc}_k(m)$
- Dec decrypts ciphertext c with key k as $m := \mathrm{Dec}_k(c)$

Message Encryption

What is an encryption scheme? A triple $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$

- Gen probabilistically selects a key k
- Enc encrypts message m with key k, as $c := \mathrm{Enc}_k(m)$
- Dec decrypts ciphertext c with key k as $m := \mathrm{Dec}_k(c)$

Remarks:

- same key is used for encryption and decryption: symmetric/private-key encryption
- correctness requirement: $\forall k, m : m = \mathrm{Dec}_k(\mathrm{Enc}_k(m))$

Message Encryption

An example: the *Scytale* (Greece, 7th century BC (?))

Message Encryption

An example: the *Scytale* (Greece, 7$^{\text{th}}$ century BC (?))

- Gen defines the diameter of the cylinder
  ($k :=$ number of letters you can write on the circumference)
- Enc encrypts by transposing letters according to k
- Dec decrypts by performing the inverse transposition

Message Encryption

Another example: the *Caesar's cipher* (Rome, 1^{st} c. BC)

- Shift letters ($\texttt{D} \rightarrow \texttt{A}$, $\texttt{E} \rightarrow \texttt{B}$, $\texttt{F} \rightarrow \texttt{C}$, ..., $\texttt{C} \rightarrow \texttt{Z}$)
- Ex: $\texttt{HELLO} \rightarrow \texttt{EBIIL}$

Message Encryption

Another example: the *Caesar's cipher* (Rome, 1st c. BC)

- Shift letters ($\texttt{D} \rightarrow \texttt{A}$, $\texttt{E} \rightarrow \texttt{B}$, $\texttt{F} \rightarrow \texttt{C}$, ..., $\texttt{C} \rightarrow \texttt{Z}$)
- Ex: $\texttt{HELLO} \rightarrow \texttt{EBIIL}$

- Gen defines the extent $k \in [0, 25]$ of the shift
- Enc encrypts by substituting letters, applying the right shift
- Dec decrypts by performing the inverse shift

Message Encryption

Another example: the *Caesar's cipher* (Rome, 1st c. BC)

- Shift letters ($\mathtt{D} \rightarrow \mathtt{A}$, $\mathtt{E} \rightarrow \mathtt{B}$, $\mathtt{F} \rightarrow \mathtt{C}$, ..., $\mathtt{C} \rightarrow \mathtt{Z}$)
- Ex: $\mathtt{HELLO} \rightarrow \mathtt{EBIIL}$

- Gen defines the extent $k \in [0, 25]$ of the shift
- Enc encrypts by substituting letters, applying the right shift
- Dec decrypts by performing the inverse shift

For more historical examples, see, e.g.,:

`http://www.apprendre-en-ligne.net/crypto/`

Cryptanalysis

Cryptanalysis: art of code breaking/cracking

Cryptanalysis

Cryptanalysis: art of code breaking/cracking

Attacker model:

1. What should be considered as secret?
   Gen? Enc? Dec? k?
2. Which attack scenario?
   Eavesdropper? Chosen-plaintext? . . . ?

Cryptanalysis

Attacker model:

1. What should be considered as secret?
   Gen? Enc? Dec? k?

Kerckhoffs' principle (1883):
only the key should be secret

See: `http://www.petitcolas.net/fabien/kerckhoffs/`

Kerckhoffs' principle

Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas ; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants :

1° Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;

2° Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;

3° La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;

4° Il faut qu'il soit applicable à la correspondance télégraphique ;

5° Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;

6° Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

Kerckhoffs' principle

"Only the key should be secret." – Why?

Kerckhoffs' principle

"Only the key should be secret." – Why?

1. Keeping secrets is annoying:
   - A key is easier to exchange secretly than a full system
   - A key is easier to update in case of compromise
   - One encryption scheme per pair of users is not manageable
   - No need to kill the cryptographer

Kerckhoffs' principle

"Only the key should be secret." – Why?

1. Keeping secrets is annoying:
    - A key is easier to exchange secretly than a full system
    - A key is easier to update in case of compromise
    - One encryption scheme per pair of users is not manageable
    - No need to kill the cryptographer
2. Public algorithms should be safer:
    - Public algorithms can be scrutinized by friends
    - Possibility to create standards
    - No risk of code reverse-engineering

Kerckhoffs' principle

"Only the key should be secret." – Why?

1. Keeping secrets is annoying:
   - A key is easier to exchange secretly than a full system
   - A key is easier to update in case of compromise
   - One encryption scheme per pair of users is not manageable
   - No need to kill the cryptographer
2. Public algorithms should be safer:
   - Public algorithms can be scrutinized by friends
   - Possibility to create standards
   - No risk of code reverse-engineering
3. We can handle it. . .

Cryptanalysis

Attacker model:

1. Which attack scenario?

Cryptanalysis

Attacker model:

1. Which attack scenario?

Depending on the context:

1. Passive:
   - Ciphertext-only: you only see ciphertexts
   - Known-plaintext: you see some plaintext/ciphertext pairs

Cryptanalysis

Attacker model:

1. Which attack scenario?

Depending on the context:

1. Passive:
   - Ciphertext-only: you only see ciphertexts
   - Known-plaintext: you see some plaintext/ciphertext pairs
2. Active:
   - Chosen-plaintext: you can ask for the encryption of some messages
   - Chosen-ciphertext: you can also ask for the decryption of some messages

Cryptanalysis

Consider Caesar's cipher, with:

- Public algorithms
- Ciphertext only

How do we break it?

Cryptanalysis

Consider Caesar's cipher, with:

- Public algorithms
- Ciphertext only

How do we break it?

- Just try the 26 possible keys!

Cryptanalysis

Consider Caesar's cipher, with:

- Public algorithms
- Ciphertext only

How do we break it?

- Just try the 26 possible keys!

Lesson:

- Key space should not be explorable when messages are long...
  In practice: trying all keys should require at least 2^{80} computational steps
  ($2^{80} \approx 1000000000000000000000000$)

Mono-alphabetic substitution

Improvement on Caesar's cipher:

- Not just a shift: take any permutation of the alphabet
- This is $26! \approx 2^{88}$ keys

How do we break it?

- Sherlock Holmes did it!

Mono-alphabetic substitution

Improvement on Caesar's cipher:

- Not just a shift: take any permutation of the alphabet
- This is $26! \approx 2^{88}$ keys

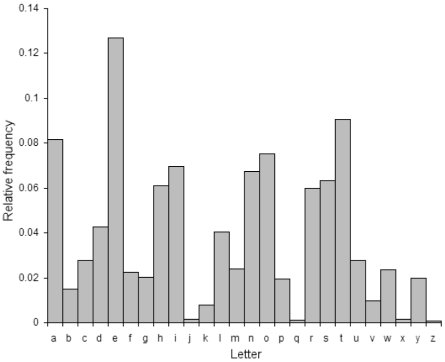
How do we break it?

- Sherlock Holmes did it!

- If you know the plaintext language, frequency analysis is possible. . .

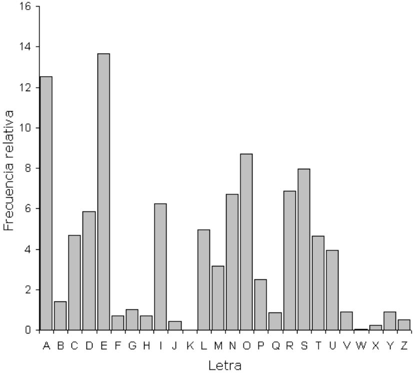
Mono-alphabetic substitution

Frequencies in English:



Mono-alphabetic substitution

Frequencies in Spanish:



Mono-alphabetic substitution

Improvement on Caesar's cipher:

- Not just a shift: take any permutation of alphabet
- This is $26! \approx 2^{88}$ keys

How do we break it?

- Just count the frequency of each symbol...

Lesson:

- Large key space is not enough!
- We need something secure independently of message distribution

Vigenère cipher

Another improvement on Caesar's cipher:

- Instead of a constant shift, use different shifts according to position
- Key is a sequence of numbers in $[0, 25]$

Example:

- Suppose key is $\langle 2, 24, 5 \rangle$
- `Cryptography is great`
  `Attnvjetvnjt gu bpgvr`

Vigenère cipher

Another improvement on Caesar's cipher:

- Instead of a constant shift, use different shifts according to position
- Key is a sequence of numbers in $[0, 25]$

Example:

- Suppose key is $\langle 2, 24, 5 \rangle$
- `Cryptography is great`
  `Attnvjetvnjt gu bpgvr`

How do we break it?

- If you have enough ciphertext material…
  Make guesses on the key length, then make frequency analysis!

Historic ciphers

Lessons:

- We can keep playing like this for a long time...
  (See, e.g., D. Kahn, "The code-breakers" (Scribner) or J. Stern, "La science du secret" (Odile Jacob))

Can we do something else?

- In many cases: yes!

Modern Cryptography

"*Modern* cryptography"

1. Definitions
2. Assumptions
3. Proofs

Modern Cryptography: Definitions

Definitions in cryptography:

1. What do we want to do?
2. Shall I use this scheme here?
3. Why choosing this scheme rather than that one?

Example of encryption

What should the definition of security say for an encryption scheme?

Example of encryption

What should the definition of security say for an encryption scheme?

1. Given any ciphertext, no adversary should be able to recover the key

Example of encryption

What should the definition of security say for an encryption scheme?

1. Given any ciphertext, no adversary should be able to recover the key
2. Given any ciphertext, no adversary should be able to recover the plaintext

Example of encryption

What should the definition of security say for an encryption scheme?

1. Given any ciphertext, no adversary should be able to recover the key
2. Given any ciphertext, no adversary should be able to recover the plaintext
3. Given any ciphertext, no adversary should be able to recover any character of the plaintext

Example of encryption

What should the definition of security say for an encryption scheme?

1. Given any ciphertext, no adversary should be able to recover the key
2. Given any ciphertext, no adversary should be able to recover the plaintext
3. Given any ciphertext, no adversary should be able to recover any character of the plaintext
4. Given any ciphertext, no adversary should be able to recover any function of the plaintext

Example of encryption

What should the definition of security say for an encryption scheme?

1. Given any ciphertext, no adversary should be able to recover the key
2. Given any ciphertext, no adversary should be able to recover the plaintext
3. Given any ciphertext, no adversary should be able to recover any character of the plaintext
4. Given any ciphertext, no adversary should be able to recover any function of the plaintext

Still need to define the adversarial model...

Modern Cryptography: Definitions

Limitations: Science vs. real world

- Check whether intuitive properties are guaranteed
- Compare with other definitions
- Compare with attack examples
- Use it during a few years. . .

Modern Cryptography: Precise Assumptions

Most schemes rely on computational assumptions

- Need to understand what we are trusting (challenges)
- Needed to write security proofs
- Useful for abstraction
- Useful for scheme comparison

Modern Cryptography: Proof of Security

Relate schemes and definitions to assumptions

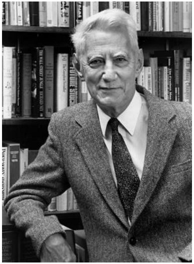
- Reductionist approach: if someone can break this scheme, (s)he is also able to falsify my assumption

Perfect encryption

We just broke encryption schemes. . . but

Perfect encryption

We just broke encryption schemes... but



Shannon (1949):
perfect encryption is possible!

Perfect encryption

What is an encryption scheme? A triple $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$

- Gen probabilistically selects a key $k \in \mathcal{K}$
- Enc encrypts message $m \in \mathcal{M}$ with key k, as $c \leftarrow \mathrm{Enc}_k(m)$
- Dec decrypts ciphertext $c \in \mathcal{C}$ with key k as $m := \mathrm{Dec}_k(c)$

Remarks:

- Enc may be probabilisitic
- $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$, always
  $\Rightarrow$ assume, wlog, Dec to be deterministic

Perfect encryption

What is an encryption scheme? A triple $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$

- Gen probabilistically selects a key $k \in \mathcal{K}$
- Enc encrypts message $m \in \mathcal{M}$ with key k, as $c \leftarrow \mathrm{Enc}_k(m)$
- Dec decrypts ciphertext $c \in \mathcal{C}$ with key k as $m := \mathrm{Dec}_k(c)$

Remarks:

- Enc may be probabilisitic
- $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$, always
  $\Rightarrow$ assume, wlog, Dec to be deterministic
- Assume $|\mathcal{M}| > 1$
- Assume $\mathcal{M}$ and $\mathcal{C}$ only contain messages and ciphertexts that may happen.

Perfect encryption

Definition: $\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[M = m | C = c] = \Pr[M = m]$$

Remarks:

- Probability distribution over $\mathcal{M}$ refers to distribution on messages

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m] = \Pr[C = c]$$

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m] = \Pr[C = c]$$

Proof of equivalence:

- Use Bayes:

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m] = \Pr[C = c]$$

Proof of equivalence:

- Use Bayes:

$$\Pr[C = c | M = m] \quad = \Pr[C = c]$$

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m] = \Pr[C = c]$$

Proof of equivalence:

- Use Bayes:

$$\begin{array}{rll} & \Pr[C = c | M = m] & = \Pr[C = c] \\ (\text{Bayes} \Rightarrow) & \frac{\Pr[M=m|C=c].\Pr[C=c]}{\Pr[M=m]} & = \Pr[C = c] \end{array}$$

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m] = \Pr[C = c]$$

Proof of equivalence:

- Use Bayes:

$$\begin{aligned}
& \Pr[C = c | M = m] &&= \Pr[C = c] \\
(\text{Bayes} \Rightarrow) \quad & \frac{\Pr[M=m | C=c].\Pr[C=c]}{\Pr[M=m]} &&= \Pr[C = c] \\
(\text{Reorganize} \Rightarrow) \quad & \Pr[M = m | C = c] &&= \Pr[M = m]
\end{aligned}$$

Perfect encryption

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if, for every $m_0, m_1 \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$:

$$\Pr[C = c | M = m_0] = \Pr[C = c | M = m_1]$$

Interpretation:

- It is impossible to distinguish the ciphertext corresponding to two plaintexts

Perfect encryption

Equivalent definition...

Given $\Pi := \langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$, and adversary $\mathcal{A}$, define the following experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$:

1. $\mathcal{A}$ outputs $m_0, m_1 \in \mathcal{M}$
2. Choose $k \leftarrow \mathcal{K}$ and $b \leftarrow \{0, 1\}$, and send $\mathrm{Enc}_k(m_b)$ to $\mathcal{A}$
3. $\mathcal{A}$ outputs b'
4. Define $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} := 1$ iff $b = b'$

Perfect encryption (cont.)

Equivalent definition:
$\langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ over message space $\mathcal{M}$ is *perfectly secret* if for every adversary $\mathcal{A}$:

$$\mathsf{Pr}[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi} = 1] = \frac{1}{2}$$

Interpretation:

- Even if $\mathcal{A}$ chooses 2 messages, it cannot decide which of them has been encrypted

One-Time Pad

One-time pad is perfectly secret!

- Fix $l > 0$. $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^l$
- Gen selects uniformly in $\mathcal{K}$
- $\mathrm{Enc}_k(m) := m \oplus k$
- $\mathrm{Dec}_k(c) := c \oplus k$

Remarks:

- $\oplus$ denotes binary XOR (exclusive OR)
- $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m \oplus k \oplus k = m$

One-Time Pad

One-time pad is perfectly secret!

Proof:

Fix any distribution over $\mathcal{M}$, any $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\Pr[C = c | M = m] \quad = \quad \Pr[M \oplus K = c | M = m]$$

One-Time Pad

One-time pad is perfectly secret!

Proof:
Fix any distribution over $\mathcal{M}$, any $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\begin{aligned} \Pr[C = c | M = m] &= \Pr[M \oplus K = c | M = m] \\ &= \Pr[m \oplus K = c] \end{aligned}$$

One-Time Pad

One-time pad is perfectly secret!

Proof:
Fix any distribution over $\mathcal{M}$, any $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\begin{aligned}
\Pr[C = c | M = m] &= \Pr[M \oplus K = c | M = m] \\
&= \Pr[m \oplus K = c] \\
&= \Pr[K = m \oplus c] = \frac{1}{2^l}
\end{aligned}$$

One-Time Pad

One-time pad is perfectly secret!

Proof:

Fix any distribution over $\mathcal{M}$, any $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

$$\begin{aligned} \Pr[C = c | M = m] &= \Pr[M \oplus K = c | M = m] \\ &= \Pr[m \oplus K = c] \\ &= \Pr[K = m \oplus c] = \frac{1}{2^l} \\ &= \Pr[C = c | M = m'] \text{ for every } m' \end{aligned}$$

One-Time Pad

One-time pad is perfectly secret!

One-Time Pad

One-time pad is perfectly secret!

One-time pad is not convenient to use...

- key needs to be as long as message!
- suppose m, m' encrypted with k
  $(m \oplus k) \oplus (m' \oplus k) = m \oplus m'$
  $\mathcal{A}$ wins if it can play $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ twice with same key!

Limits of Perfect Secrecy

Suppose $\Pi := \langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ is s.t. $|\mathcal{K}| < |\mathcal{M}|$.
Then Π is not a perfectly secret encryption scheme.

Limits of Perfect Secrecy

Suppose $\Pi := \langle \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec} \rangle$ is s.t. $|\mathcal{K}| < |\mathcal{M}|$.
Then Π is not a perfectly secret encryption scheme.

Proof:
Consider uniform distribution on $\mathcal{M}$ and any $c \in \mathcal{C}$.
Define $\mathcal{M}(c) := \{\hat{m} : \hat{m} = \mathrm{Dec}_{\hat{k}}(c)$ for some $\hat{k} \in \mathcal{K}\}$
We must have $|\mathcal{M}(c)| < |\mathcal{M}|$
Therefore, $\exists m \in \mathcal{M} - \mathcal{M}(c)$, and

$$\Pr[M = m | C = c] = 0 \neq \Pr[M = m]$$

Conclusion

Perfectly secret encryption schemes exist, but are difficult to use

Can we do better?

Conclusion

Perfectly secret encryption schemes exist, but are difficult to use

Can we do better?
Shannon's theory says: "no!"

Conclusion

Perfectly secret encryption schemes exist, but are difficult to use

Can we do better?
Shannon's theory says: "no!"

Under which assumptions?

- $\mathcal{A}$ has perfect information
- $\mathcal{A}$ has unbounded computational power

Conclusion

Perfectly secret encryption schemes exist, but are difficult to use

Can we do better?
Shannon's theory says: "no!"

Under which assumptions?

- $\mathcal{A}$ has perfect information
- $\mathcal{A}$ has unbounded computational power

Next week:
What about bounded computational power?