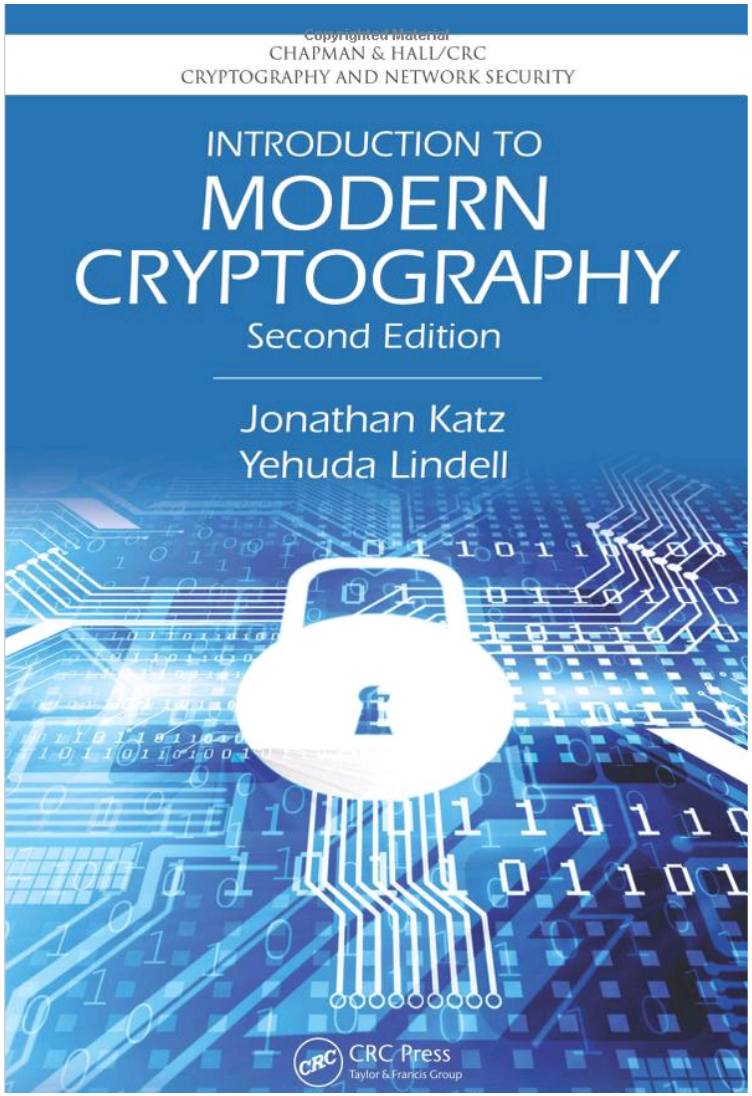
Private-Key Encryption

Federico Pintore[1]

[1]Mathematical Institute,
University of Oxford

Outline

Course Main Reference



Outline

Encryption schemes

Classical Cryptography mainly dealt with the problem of enabling two parties to communicate secretly in the presence of an eavesdropper.

Encryption schemes

Classical Cryptography mainly dealt with the problem of enabling two parties to communicate secretly in the presence of an eavesdropper.

Currently, we call encryption schemes the schemes designed for solving this problem.

Encryption schemes

Classical Cryptography mainly dealt with the problem of enabling two parties to communicate secretly in the presence of an eavesdropper.

Currently, we call encryption schemes the schemes designed for solving this problem.

Security of classical encryption schemes rely on a secret key shared in advance by the communicating parties.

Encryption schemes

Classical Cryptography mainly dealt with the problem of enabling two parties to communicate secretly in the presence of an eavesdropper.

Currently, we call encryption schemes the schemes designed for solving this problem.

Security of classical encryption schemes rely on a secret key shared in advance by the communicating parties.

The sender encrypts a message, i.e. hides it, using the secret key, obtaining the *ciphertext*.

Encryption schemes

Classical Cryptography mainly dealt with the problem of enabling two parties to communicate secretly in the presence of an eavesdropper.

Currently, we call encryption schemes the schemes designed for solving this problem.

Security of classical encryption schemes rely on a secret key shared in advance by the communicating parties.

The sender encrypts a message, i.e. hides it, using the secret key, obtaining the *ciphertext*.

The receiver decrypts the ciphertext, i.e. unhides it, using the same secret key. They recover the original message.

Syntax of Private-Key Encryption Schemes

A private-key encryption scheme consists of three algorithms $(\text{KeyGen}, \text{Enc}, \text{Dec})$:

- $k \leftarrow \text{KeyGen}(n)$: a randomised algorithm that takes the security parameter n and outputs the key k, chosen from the key space $\mathcal{K}$.
- $c \leftarrow \text{Enc}(k, m)$: an algorithm (often randomised) that takes the secret key k and the message $m \in \mathcal{M}$, and outputs the ciphertext c.
- $m \leftarrow \text{Dec}(k, c)$: a deterministic algorithm that takes the secret key k and the ciphertext c, and gives back the message m.

Caesar Cipher (100-44BC)

Example

- Plaintext: ABCD $\cdots$ WXYZ.
- Shift: $+3 \pmod{26}$ *or* $+k \pmod{26}$ $(k \in \{0, \ldots, 25\})$
- Ciphertext: DEFG $\cdots$ ZABC.

Cryptanalysis:

Caesar Cipher (100-44BC)

Example

- Plaintext: ABCD $\cdots$ WXYZ.
- Shift:$+3 \pmod{26}$ *or* $+k \pmod{26}$ $(k \in \{0, \ldots, 25\})$
- Ciphertext: DEFG $\cdots$ ZABC.

Cryptanalysis:

- Brute Force (trying every possible key): key space size is $|\mathcal{K}| = 26$.

Caesar Cipher (100-44BC)

Example

- Plaintext: ABCD $\cdots$ WXYZ.
- Shift: $+3 \pmod{26}$ *or* $+k \pmod{26}$ $(k \in \{0, \ldots, 25\})$
- Ciphertext: DEFG $\cdots$ ZABC.

Cryptanalysis:

- Brute Force (trying every possible key): key space size is $|\mathcal{K}| = 26$.
- *Sufficient key-space principle: Any secure symmetric key encryption scheme must have a key space that is sufficiently large to make an exhaustive-search attack infeasible* (e.g. $|\mathcal{K}| \geq 2^{70}$).

Caesar Cipher (100-44BC)

Example

- Plaintext: ABCD $\cdots$ WXYZ.
- Shift: $+3 \pmod{26}$ *or* $+k \pmod{26}$ $(k \in \{0, \ldots, 25\})$
- Ciphertext: DEFG $\cdots$ ZABC.

Cryptanalysis:

- Brute Force (trying every possible key): key space size is $|\mathcal{K}| = 26$.
- *Sufficient key-space principle: Any secure symmetric key encryption scheme must have a key space that is sufficiently large to make an exhaustive-search attack infeasible* (e.g. $|\mathcal{K}| \geq 2^{70}$).
- Is it a sufficient condition?

Substitution Cipher (mono-alphabetic)

Example

- Plaintext: ABCZ
- Substitution: (A$\rightarrow$ T, B$\rightarrow$ N, C$\rightarrow$ L, . . . , Z$\rightarrow$ O) *or* any one-to-one map of the alphabet
- Ciphertext:TNLO

Cryptanalysis:

Substitution Cipher (mono-alphabetic)

Example

- Plaintext: ABCZ
- Substitution: (A$\rightarrow$ T, B$\rightarrow$ N, C$\rightarrow$ L, . . . , Z$\rightarrow$ O) *or* any one-to-one map of the alphabet
- Ciphertext:TNLO

Cryptanalysis:

- Brute Force: Key space size is $|\mathcal{K}| = 26! \approx 2^{88}$.

Substitution Cipher (mono-alphabetic)

Example

- Plaintext: ABCZ
- Substitution: (A$\rightarrow$ T, B$\rightarrow$ N, C$\rightarrow$ L, . . . , Z$\rightarrow$ O) *or* any one-to-one map of the alphabet
- Ciphertext:TNLO

Cryptanalysis:

- Brute Force: Key space size is $|\mathcal{K}| = 26! \approx 2^{88}$.
- Frequency analysis:
  - Frequency of English letters

Substitution Cipher (mono-alphabetic)

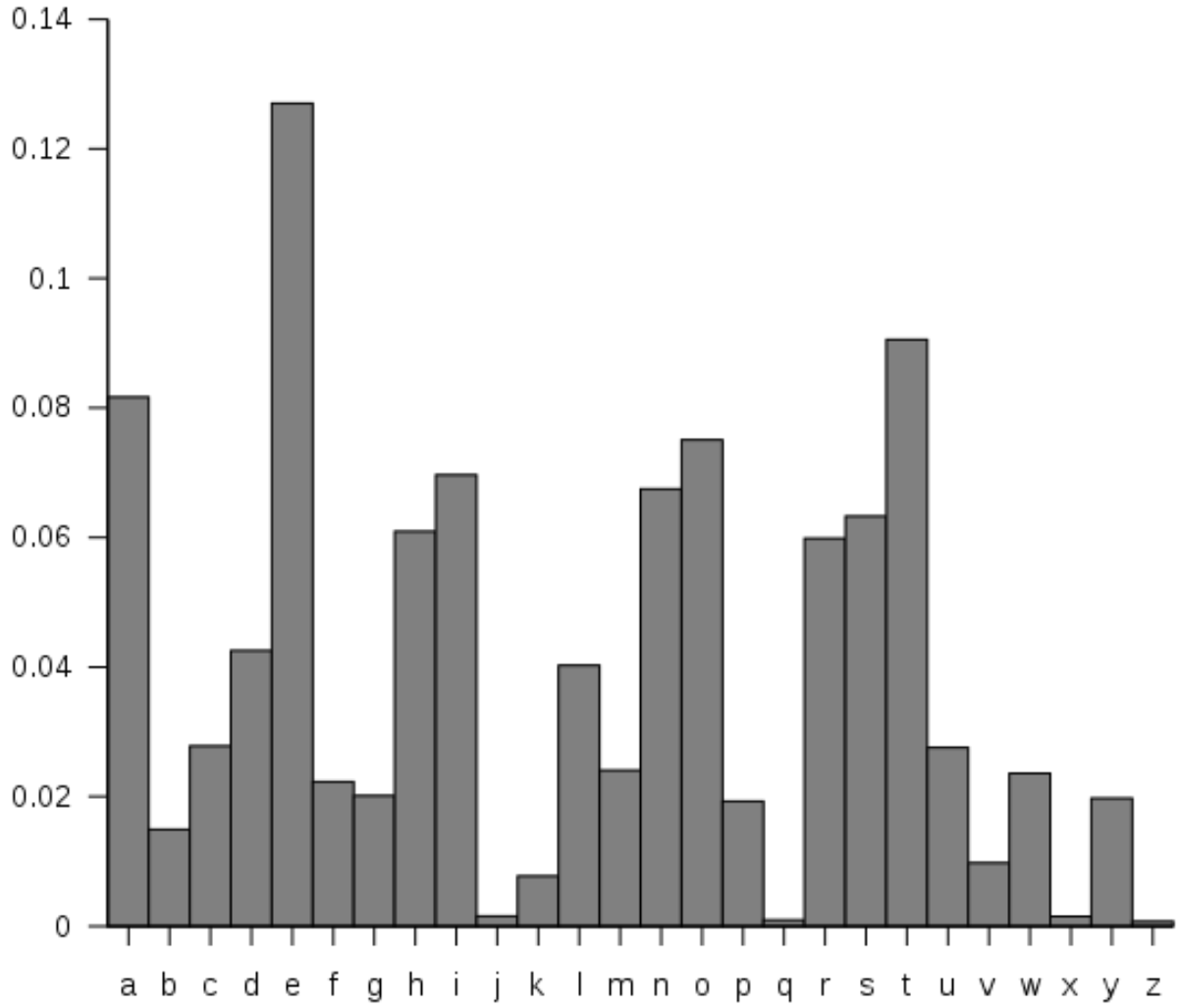
Example

- Plaintext: ABCZ
- Substitution: (A$\rightarrow$ T, B$\rightarrow$ N, C$\rightarrow$ L, . . . , Z$\rightarrow$ O) *or* any one-to-one map of the alphabet
- Ciphertext:TNLO

Cryptanalysis:

- Brute Force: Key space size is $|\mathcal{K}| = 26! \approx 2^{88}$.
- Frequency analysis:
  - Frequency of English letters
  - Frequency of pairs (or more) of letters, e.g. digrams, trigrams, etc.

Substitution Cipher (mono-alphabetic)



Vigenere Cipher (1553)

Example

- Poly-alphabetic shift:

Plaintext m:	TOBEORNOTTOBE
key k:(+ mod 26)	CRYPTOCRYPTOC
Ciphertext c:	VFZTHFPFRIHPG

- Cryptanalysis:

Vigenere Cipher (1553)

Example

- Poly-alphabetic shift:

Plaintext m:	TOBEORNOTTOBE
key k:(+ mod 26)	CRYPTOCRYPTOC
Ciphertext c:	VFZTHFPFRIHPG

- Cryptanalysis:
  - If the length of the key, say n, is known, then break ciphertext into subsets and solve each block as it was encrypted by Caesar cipher and using letter-frequency analysis.

Vigenere Cipher (1553)

Example

- Poly-alphabetic shift:

Plaintext m:	TOBEORNOTTOBE
key k:(+ mod 26)	CRYPTOCRYPTOC
Ciphertext c:	VFZTHFPFRIHPG

- Cryptanalysis:
  - If the length of the key, say n, is known, then break ciphertext into subsets and solve each block as it was encrypted by Caesar cipher and using letter-frequency analysis.
  - If n is not known, use Kasiski method (Kasiski 1863) or *index of coincidence method* to find n, and do the rest as in the first case. (What if $n = |c| = |k|?$)

Kerckhoff's Principle (1883):

Definition

The cipher must NOT be required to be secret and it must be able to fall into the hands of the enemy without inconvenience.

Modern Cryptography:

Kerckhoff's Principle (1883):

Definition

The cipher must NOT be required to be secret and it must be able to fall into the hands of the enemy without inconvenience.

Modern Cryptography:

- The encryption scheme's algorithms should be public. (Standardized, etc.)

Unofficial principle

The desired security for a cryptographic scheme should be formally defined.

Unofficial principle

The desired security for a cryptographic scheme should be formally defined.

It is composed by:

- a security guarantee
- a threat model

Security Definitions

What is a secure encryption scheme (security guarantee)?

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

What are the adversaries' abilities (threat model)?

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

What are the adversaries' abilities (threat model)?

- Ciphertext-only attack: knows one single ciphertext c (or more).

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

What are the adversaries' abilities (threat model)?

- Ciphertext-only attack: knows one single ciphertext c (or more).
- Known Plaintext attack: the adversary learns a number of pairs of (c_i, m_i) generated using some key.

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

What are the adversaries' abilities (threat model)?

- Ciphertext-only attack: knows one single ciphertext c (or more).
- Known Plaintext attack: the adversary learns a number of pairs of (c_i, m_i) generated using some key.
- Chosen-plaintext attack (CPA): same as above, but the adversary gets to choose the plaintexts this time.

Security Definitions

What is a secure encryption scheme (security guarantee)?

- Adversaries cannot compute the secret key.
- Adversaries cannot compute the plaintext.
- Adversaries cannot compute information about the plaintext.

What are the adversaries' abilities (threat model)?

- Ciphertext-only attack: knows one single ciphertext c (or more).
- Known Plaintext attack: the adversary learns a number of pairs of (c_i, m_i) generated using some key.
- Chosen-plaintext attack (CPA): same as above, but the adversary gets to choose the plaintexts this time.
- Chosen-ciphertext attack (CCA): now, they additionally get the decryption of ciphertexts of its choice.

Outline

Discrete Probability

Let Ω be a (finite) set of outcomes, known as sample space.

The event space $\mathcal{A}$ is a subset of $\mathcal{P}(\Omega)$ s.t.

- $\Omega \in \mathcal{A}$
- if $A \in \mathcal{A}$, then $\bar{A} \in \mathcal{A}$ ($\bar{A}$ denotes the complementary in Ω)
- if $A_1, A_2 \in \mathcal{A}$, then $A_1 \cup A_2 \in \mathcal{A}$

Usually, $\mathcal{A} = \mathcal{P}(\Omega)$ when Ω is finite.

Discrete Probability

A probability distribution $\Pr$ is a map from $\mathcal{A}$ to $[0,1]$ s.t.

- $\Pr[\Omega] = 1$
- if $A_1, \ldots, A_t$ are pairwise disjoint events, then

  $$\Pr[\cup_{i=1}^{t} A_i] = \sum_{i=1}^{t} \Pr[A_i]$$

Assuming Ω finite we have:

- Let $A \subseteq \Omega$, $\Pr[A] = \sum_{\omega \in A} \Pr[w]$.
- Union Formula: $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$.
- Union Bound: $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$.

Discrete Probability

- Conditional Probability: $\Pr[A|B] = \Pr[A \cap B]/\Pr[B]$ (if $\Pr[B] > 0$).
- Bayes' Theorem: $\Pr[A|B] = \dfrac{\Pr[A] \cdot \Pr[B|A]}{\Pr[B]}$ (if $\Pr[A], \Pr[B] > 0$)
- A and B are independent $\Leftrightarrow \Pr[A \cap B] = \Pr[A] \cdot \Pr[B]$.

Outline

Syntax of Private-Key Encryption Schemes

A private-key encryption scheme consists of three algorithms $(\text{KeyGen}, \text{Enc}, \text{Dec})$:

- $k \leftarrow \text{KeyGen}(n)$: a randomised algorithm that takes the security parameters n and outputs the key k, chosen from the key space $\mathcal{K}$.
- $c \leftarrow \text{Enc}(k, m)$: an algorithm (often randomised) that takes the secret key k and the message $m \in \mathcal{M}$, and outputs the ciphertext c.
- $m \leftarrow \text{Dec}(k, c)$: a deterministic algorithm that takes the secret key k and the ciphertext c, and gives back the message m.

Notation

From

- $(\mathcal{K}, \mathcal{P}(\mathcal{K}), \Pr_{\mathcal{K}})$
- $(\mathcal{M}, \mathcal{P}(\mathcal{M}), \Pr_{\mathcal{M}})$

a probability distribution $\Pr_{\mathcal{K}\times\mathcal{M}}$ over the sample space $\mathcal{K} \times \mathcal{M}$ is deduced. It is defined as:

$$\Pr_{\mathcal{K}\times\mathcal{M}}(A) = \sum_{(k,m)\in A} \Pr_{\mathcal{K}}[k] \Pr_{\mathcal{M}}[m]$$

- $\Pr[K = k] = \Pr_{\mathcal{K}\times\mathcal{M}}[\{k\} \times \mathcal{M}] = \Pr_{\mathcal{K}}[k]$
- $\Pr[M = m] = \Pr_{\mathcal{K}\times\mathcal{M}}[\mathcal{K} \times \{m\}] = \Pr_{\mathcal{M}}[m]$
- $\Pr[C = c] = \Pr_{\mathcal{K}\times\mathcal{M}}[A]$ with $A = \{(k, m) \mid \mathsf{Enc}(k, m) = c\}$

Perfect Secrecy (Shannon 1949)

Perfect Secrecy (Shannon 1949)

- *“The ciphertext should reveal no information about the plaintext”*

Perfect Secrecy (Shannon 1949)

- *"The ciphertext should reveal no information about the plaintext"*
- Also called *information theoretic security*.

Perfect Secrecy (Shannon 1949)

- *"The ciphertext should reveal no information about the plaintext"*
- Also called *information theoretic security*.

Definition (Perfect Secrecy)

For every probability distribution $\Pr_{\mathcal{M}}$ *over the message space* $\mathcal{M}$*, we have*

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

$\forall m \in \mathcal{M}, \forall c \in \mathcal{C} \;\; s.t. \;\; \Pr[C = c] > 0.$

Equivalently,

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

Perfect Indistinguishability

Perfect Indistinguishability Experiment $\text{PrivK}^{\text{perfect}-\text{ind}}_{\mathcal{A},E}$

Challenger Ch **Adversary** A

$\xleftarrow{m_0, m_1, |m_0| = |m_1|}$

$b \leftarrow\!\!\$ \{0,1\}$

$\xrightarrow{c = \text{Enc}(k, m_b)}$ Outputs his guess b'

Definition

An encryption scheme is perfectly indistinguishable if for every adversary $\mathcal{A}$ the following holds:

$$\Pr[\text{PrivK}^{\text{perfect}-\text{IND}}_{\mathcal{A},E} = 1] = 1/2$$

Where $\text{PrivK}^{\text{perfect}-\text{IND}}_{\mathcal{A},E} = 1$ if $b' = b$, and 0 otherwise.

Perfect Indistinguishability

Theorem (Perfect indistinguishability)

*An encryption scheme (*KeyGen, Enc, Dec*) has perfect secrecy iff for every probability distribution* $\Pr_{\mathcal{M}}$ *over* $\mathcal{M}$ *we have*

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

$\forall m_0, m_1 \in \mathcal{M}\ s.t.\ |m_0| = |m_1|, \forall c \in \mathcal{C}.$

Proof.

$(\Rightarrow) : \Pr[C = c|M = m_0] = \Pr[C = c] = \Pr[C = c|M = m_1]$

$(\Leftarrow) :$

$$\begin{aligned}
\Pr[C = c] &= \sum_m \Pr[C = c|M = m] \cdot \Pr[M = m] \\
&= \sum_m \Pr[C = c|M = m_0] \cdot \Pr[M = m] \\
&= \Pr[C = c|M = m_0] \cdot \sum_m \Pr[M = m] \\
&= \Pr[C = c|M = m_0]
\end{aligned}$$

which is correct for any m_0 □

Outline

One Time Pad (Vernam 1917 or some 35 years earlier!)

Fix an integer $n > 0$. Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$.

One Time Pad (Vernam 1917 or some 35 years earlier!)

Fix an integer $n > 0$. Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$.

- KeyGen(n) : it returns a random bit string (uniform distribution) of length n, i.e. $k \in \mathcal{K}$.
- Enc(k, m): it outputs the ciphertext $c = k \oplus m$.
- Dec$(k, c))$: recover the message computing $m = k \oplus c$.

One Time Pad (Vernam 1917 or some 35 years earlier!)

Fix an integer $n > 0$. Let $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$.

- KeyGen(n) : it returns a random bit string (uniform distribution) of length n, i.e. $k \in \mathcal{K}$.
- Enc(k, m): it outputs the ciphertext $c = k \oplus m$.
- Dec$(k, c))$: recover the message computing $m = k \oplus c$.

It was used between the White House and the Kremlin during the Cold War!

Security of OTP

Theorem

The one time pad (OTP) encryption scheme is perfectly secret.

Proof.

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

because the key k is a uniform n-bit string. Therefore, for any m_0, m_1, we have $\Pr[C = c | M = m_0] = \frac{1}{2^n} = \Pr[C = c | M = m_1]$

OTP has perfect secrecy, but is it practical?

Theorem

If an encryption scheme E is perfectly secret, then $|\mathcal{K}| \geq |\mathcal{M}|$.

OTP has perfect secrecy, but is it practical?

Theorem

If an encryption scheme E is perfectly secret, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof.

Assume that $|\mathcal{K}| < |\mathcal{M}|$, we will show that E is not perfectly secure. We first fix a uniform distribution $\Pr_{\mathcal{M}}$ over $\mathcal{M}$, and let

$$\mathcal{M}(c) = \{m \mid m = \mathsf{Dec}(k, c) \text{ for some } k \in \mathcal{K}\}$$

but $|\mathcal{M}(c)| \leq |\mathcal{K}|$, then there exists $m' \in \mathcal{M}$ s.t. $m' \notin \mathcal{M}(c)$. Therefore, $\Pr[M = m' | C = c] = 0 \neq \Pr[M = m']$ □

Is there a way to make OTP practical?

From Perfect to Computational Secrecy

- Perfect secrecy: No leakage of information about an encrypted message even to an eavesdropper with *unlimited computational power*.

From Perfect to Computational Secrecy

- Perfect secrecy: No leakage of information about an encrypted message even to an eavesdropper with *unlimited computational power*.
- Computational secrecy: an encryption scheme is still considered to be secure even if it leaks some information with a very small probability to eavesdroppers with *limited power*.

From Perfect to Computational Secrecy

- Perfect secrecy: No leakage of information about an encrypted message even to an eavesdropper with *unlimited computational power*.
- Computational secrecy: an encryption scheme is still considered to be secure even if it leaks some information with a very small probability to eavesdroppers with *limited power*.
- Real-world application: happy with a scheme that leaks information with probability at most 2^{-60} over 200 years using fastest supercomputers!

Further Reading (1)


Nadhem J AlFardan, Daniel J Bernstein, Kenneth G Paterson, Bertram Poettering, and Jacob CN Schuldt.
On the security of RC4 in TLS.
In *USENIX Security*, pages 305–320, 2013.


Boaz Barak and Shai Halevi.
A model and architecture for pseudo-random generation with applications to/dev/random.
In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 203–212. ACM, 2005.


Daniel J Bernstein.
The Salsa20 Family of Stream Ciphers.
In *New stream cipher designs*, pages 84–97. Springer, 2008.

Further Reading (2)


Lenore Blum, Manuel Blum, and Mike Shub.
A simple unpredictable pseudo-random number generator.
SIAM Journal on computing, 15(2):364–383, 1986.


Christian Cachin.
Entropy measures and unconditional security in cryptography.
PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 1997.


Scott Fluhrer, Itsik Mantin, and Adi Shamir.
Weaknesses in the key scheduling algorithm of RC4.
In *Selected areas in cryptography*, pages 1–24. Springer, 2001.

Further Reading (3)

Christina Garman, Kenneth G Paterson, and Thyla van der Merwe.
Attacks only get better: Password recovery attacks against RC4 in TLS.
2015.

Itsik Mantin and Adi Shamir.
A practical attack on broadcast RC4.
In *Fast Software Encryption*, pages 152–164. Springer, 2002.