How to Keep a Secret

Arseniy (Senia) Sheydvasser

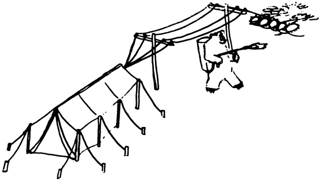
August 27, 2019

Introduction:

- Suppose I am a general looking to send a message to my troops. I don't want the enemy to be able to read my message.

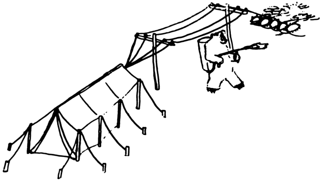
Introduction:

- Suppose I am a general looking to send a message to my troops. I don't want the enemy to be able to read my message.
- Obvious solution:

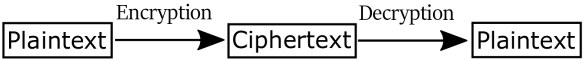


Introduction:

- Suppose I am a general looking to send a message to my troops. I don't want the enemy to be able to read my message.
- Obvious solution:



- Better solution:



Ancient Cryptography:

> If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.
>
> Suetonius, Life of Julius Caesar

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
20	21	22	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
X	Y	Z	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	V

Caesar Cipher:

Example

E	G	O	I	N	E	V	I	T	A	B	I	L	I	S
4	6	13	8	12	4	19	8	18	0	1	8	10	8	17
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0	2	9	4	8	0	15	4	14	19	20	4	6	4	13
A	C	K	E	I	A	Q	E	P	V	X	E	G	E	O

Caesar Cipher:

Example

E	G	O	I	N	E	V	I	T	A	B	I	L	I	S
4	6	13	8	12	4	19	8	18	0	1	8	10	8	17
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0	2	9	4	8	0	15	4	14	19	20	4	6	4	13
A	C	K	E	I	A	Q	E	P	V	X	E	G	E	O

- Note that this is really just modular arithmetic:
  - Encryption: $x \mapsto x + k \mod 23$ for some value k (called the *key*).
  - Decryption: $x \mapsto x - k \mod 23$.

Caesar Cipher:

Example

E	G	O	I	N	E	V	I	T	A	B	I	L	I	S
4	6	13	8	12	4	19	8	18	0	1	8	10	8	17
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	2	9	4	8	0	15	4	14	19	20	4	6	4	13
A	C	K	E	I	A	Q	E	P	V	X	E	G	E	O

- Note that this is really just modular arithmetic:
  - Encryption: $x \mapsto x + k \mod 23$ for some value k (called the *key*).
  - Decryption: $x \mapsto x - k \mod 23$.
- While clever for its time, the low number of possibilities make this encryption algorithm useless if the enemy knows the scheme.

Caesar Cipher:

Example

E	G	O	I	N	E	V	I	T	A	B	I	L	I	S
4	6	13	8	12	4	19	8	18	0	1	8	10	8	17
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	2	9	4	8	0	15	4	14	19	20	4	6	4	13
A	C	K	E	I	A	Q	E	P	V	X	E	G	E	O

- Note that this is really just modular arithmetic:
  - Encryption: $x \mapsto x + k \mod 23$ for some value k (called the *key*).
  - Decryption: $x \mapsto x - k \mod 23$.
- While clever for its time, the low number of possibilities make this encryption algorithm useless if the enemy knows the scheme.
- MORAL: Any good encryption scheme should have a large number of encryptions.

Affine Cipher:

- Slight improvement on Caesar cipher: choose two integers a_1, a_2 such that $a_1 a_2 = 1 \mod 26$, and an integer b. (We'll switch to English.)

Affine Cipher:

- Slight improvement on Caesar cipher: choose two integers a_1, a_2 such that $a_1 a_2 = 1 \mod 26$, and an integer b. (We'll switch to English.)
  - Encryption: $x \mapsto a_1 x + b \mod 26$
  - Decryption: $x \mapsto a_2(x - b) \mod 26$.

Affine Cipher:

- Slight improvement on Caesar cipher: choose two integers a_1, a_2 such that $a_1 a_2 = 1 \mod 26$, and an integer b. (We'll switch to English.)
  - Encryption: $x \mapsto a_1 x + b \mod 26$
  - Decryption: $x \mapsto a_2(x - b) \mod 26$.

Example

Take $a_1 = 5$, $a_2 = 21$, $b = 8$.

A	V	E	N	G	E	R	S	A	S	S	E	M	B	L	E
0	21	4	13	6	4	17	18	0	18	18	4	12	1	11	4
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
8	9	2	21	12	2	15	20	8	20	20	2	16	13	11	2
I	J	C	V	M	C	P	U	I	U	U	C	Q	N	L	C

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.
- For example: suppose we intercept "X BJNE SON JSZQNJ SZ ENJSAZL SON JSZQNJ. XS QNDAYL PXYYNE HN, MBS SON TZAP XJ EZQN." The most common letters here are "N" and "S."

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.
- For example: suppose we intercept "X BJNE SON JSZQNJ SZ ENJSAZL SON JSZQNJ. XS QNDAYL PXYYNE HN, MBS SON TZAP XJ EZQN." The most common letters here are "N" and "S."
  - If "E" $\mapsto$ "N", then $4 \mapsto 13$
  - If "T" $\mapsto$ "S", then $19 \mapsto 18$

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.
- For example: suppose we intercept "X BJNE SON JSZQNJ SZ ENJSAZL SON JSZQNJ. XS QNDAYL PXYYNE HN, MBS SON TZAP XJ EZQN." The most common letters here are "N" and "S."
  - If "E" $\mapsto$ "N", then $4 \mapsto 13$
  - If "T" $\mapsto$ "S", then $19 \mapsto 18$
- Thus, keys are $a_1 = 9$, $b = 3$; thus, $a_2 = 3$.

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.
- For example: suppose we intercept "X BJNE SON JSZQNJ SZ ENJSAZL SON JSZQNJ. XS QNDAYL PXYYNE HN, MBS SON TZAP XJ EZQN." The most common letters here are "N" and "S."
  - If "E" $\mapsto$ "N", then $4 \mapsto 13$
  - If "T" $\mapsto$ "S", then $19 \mapsto 18$
- Thus, keys are $a_1 = 9$, $b = 3$; thus, $a_2 = 3$.
- Decryption: "I USED THE STONES TO DESTROY THE STONES. IT NEARLY KILLED ME, BUT THE WORK IS DONE."

Frequency Analysis:

- The number of possible encryptions is larger: $26 \cdot 12 = 312$.
- However, this system is still vulnerable to frequency analysis.
- For example: suppose we intercept "X BJNE SON JSZQNJ SZ ENJSAZL SON JSZQNJ. XS QNDAYL PXYYNE HN, MBS SON TZAP XJ EZQN." The most common letters here are "N" and "S."
  - If "E" $\mapsto$ "N", then $4 \mapsto 13$
  - If "T" $\mapsto$ "S", then $19 \mapsto 18$
- Thus, keys are $a_1 = 9$, $b = 3$; thus, $a_2 = 3$.
- Decryption: "I USED THE STONES TO DESTROY THE STONES. IT NEARLY KILLED ME, BUT THE WORK IS DONE."
- MORAL: Any good encryption scheme should obfuscate any statistical properties of the original message.

Modern Cryptography:

> [A cryptographic system] should not require secrecy, and it should not be a problem if it falls into enemy hands.
>
> Kerckhoffs's principle, 1883

Modern Cryptography:

> [A cryptographic system] should not require secrecy, and it should not be a problem if it falls into enemy hands.
>
> Kerckhoffs's principle, 1883

- If secrecy is required, a single defector will render your system insecure.

Modern Cryptography:

> [A cryptographic system] should not require secrecy, and it should not be a problem if it falls into enemy hands.
>
> Kerckhoffs's principle, 1883

- If secrecy is required, a single defector will render your system insecure.
- It should be that even if an attacker learns how your system works currently, it should be easy to swap out some basic things and maintain security.

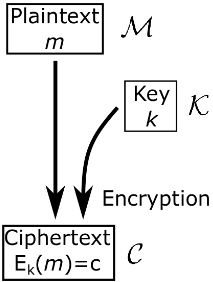
Basic Terminology:

$\boxed{\begin{array}{c}\text{Plaintext}\\ m\end{array}}$ $\mathcal{M}$

- Plaintext m
  - set of all messages is $\mathcal{M}$

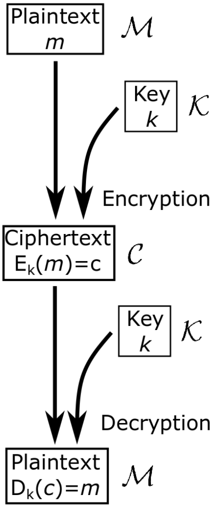
Basic Terminology:

- Plaintext m
  - set of all messages is $\mathcal{M}$
- Key k (only secret information!)
  - set of keys is $\mathcal{K}$
- Encryption algorithm E_k
- Ciphertext $c = E_k(m)$
  - set of all ciphertexts is $\mathcal{C}$



Basic Terminology:

- Plaintext m
  - set of all messages is $\mathcal{M}$
- Key k (only secret information!)
  - set of keys is $\mathcal{K}$
- Encryption algorithm E_k
- Ciphertext $c = E_k(m)$
  - set of all ciphertexts is $\mathcal{C}$
- Decryption algorithm D_k
- Require $D_k(E_k(m)) = m$.



Perfect Secrecy:

Definition

We say that $(\mathcal{M}, \mathcal{K}, \mathcal{C}, E_k, D_k)$ is *information-theoretically secure* if for any ciphertext c, the probability that it decodes to a message m is independent of m.

$$\Pr_{k \in \mathcal{K}}\left(E_k(m_1) = c\right) = \Pr_{k \in \mathcal{K}}\left(E_k(m_2) = c\right).$$

Perfect Secrecy:

Definition

We say that $(\mathcal{M}, \mathcal{K}, \mathcal{C}, E_k, D_k)$ is *information-theoretically secure* if for any ciphertext c, the probability that it decodes to a message m is independent of m.

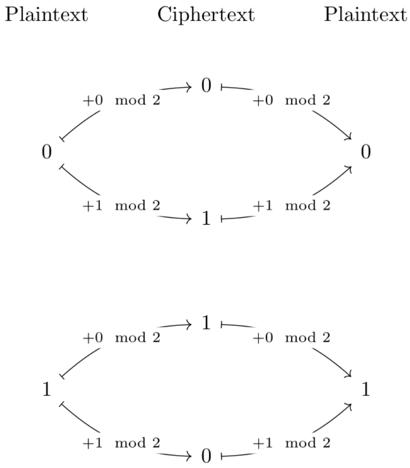
$$\Pr_{k \in \mathcal{K}}\left(E_k(m_1) = c\right) = \Pr_{k \in \mathcal{K}}\left(E_k(m_2) = c\right).$$

Question

Do there exist any information-theoretically secure cryptosystems?

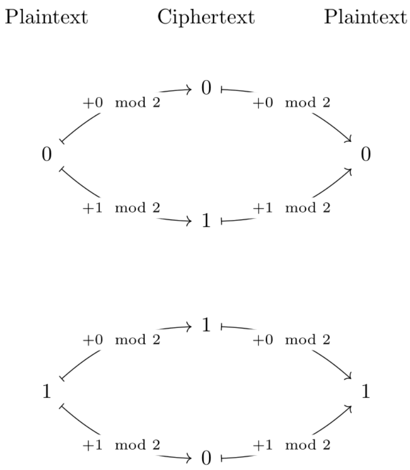
One-Time Pad (One Bit Version):

- Suppose you only have one of two possible messages, for example:
  - 0 = Don't attack
  - 1 = Attack



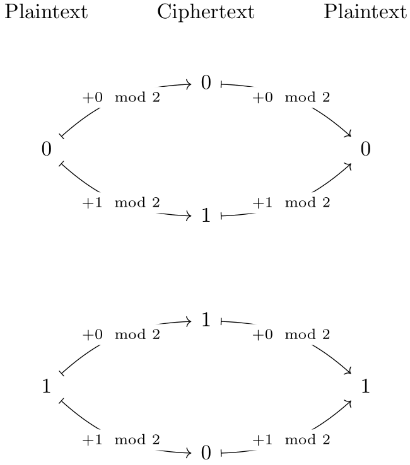
One-Time Pad (One Bit Version):

- Suppose you only have one of two possible messages, for example:
  - 0 = Don't attack
  - 1 = Attack
- Choose $k = 0$ or $k = 1$ randomly.
- Ciphertext $c = m + k \mod 2$.



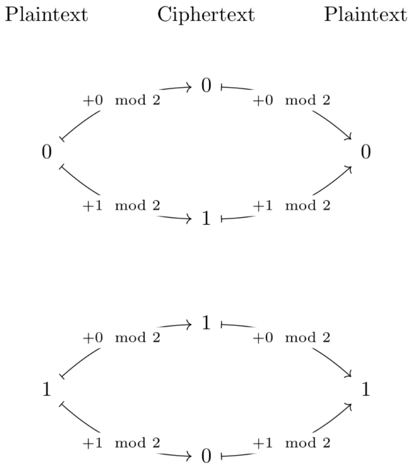
One-Time Pad (One Bit Version):

- Suppose you only have one of two possible messages, for example:
  - 0 = Don't attack
  - 1 = Attack
- Choose $k = 0$ or $k = 1$ randomly.
- Ciphertext $c = m + k \mod 2$.
- To decode: $m = c + k \mod 2$.



One-Time Pad (One Bit Version):

- Suppose you only have one of two possible messages, for example:
  - 0 = Don't attack
  - 1 = Attack
- Choose $k = 0$ or $k = 1$ randomly.
- Ciphertext $c = m + k \mod 2$.
- To decode: $m = c + k \mod 2$.
- This is information-theoretically secure!



One-Time Pad (Full Version):

- Suppose we are looking to send messages that are b bits long.
  - Example: $m = 0111000010010101$

One-Time Pad (Full Version):

- Suppose we are looking to send messages that are b bits long.
  - Example: $m = 0111000010010101$
- Frank Miller (1882): choose a key that is a random sequence of b bits.
  - Example: $k = 0000111010110001$
- Add the message and key together, mod 2, bit by bit.

$$\begin{array}{r} 0111000010010101 \\ +0000111010110001 \\ \hline 0111111000100100 \end{array}$$

One-Time Pad (Full Version):

- Suppose we are looking to send messages that are b bits long.
  - Example: $m = 0111000010010101$
- Frank Miller (1882): choose a key that is a random sequence of b bits.
  - Example: $k = 0000111010110001$
- Add the message and key together, mod 2, bit by bit.

$$\begin{array}{r} 0111000010010101 \\ +0000111010110001 \\ \hline 0111111000100100 \end{array}$$

- To decode, just add the key to the ciphertext again!

One-Time Pad (Full Version):

- Suppose we are looking to send messages that are b bits long.
  - Example: $m = 0111000010010101$
- Frank Miller (1882): choose a key that is a random sequence of b bits.
  - Example: $k = 0000111010110001$
- Add the message and key together, mod 2, bit by bit.

$$\begin{array}{r} 0111000010010101 \\ +0000111010110001 \\ \hline 0111111000100100 \end{array}$$

- To decode, just add the key to the ciphertext again!
- *Exercise:* Prove that the one-time pad is information-theoretically secure.

One-Time Pad (Practical Use):

- *Step 1*: Choose a maximum bit length b for messages you want to send.

One-Time Pad (Practical Use):

- *Step 1*: Choose a maximum bit length b for messages you want to send.
- *Step 2*: Use some encoding scheme, such as ASCII, to turn your message into a string of 0s and 1s.

01000001	A	01000011	C
01000010	B	01000100	D

One-Time Pad (Practical Use):

- *Step 1*: Choose a maximum bit length b for messages you want to send.
- *Step 2*: Use some encoding scheme, such as ASCII, to turn your message into a string of 0s and 1s.

01000001	A	01000011	C
01000010	B	01000100	D

- *Step 3*: Pad the message so that it is the right number of bits. For instance, add a bunch of 0's on the right.
  - Example: $010000010100000101000001 \mapsto 010000010100000101000001\underline{00000000}$.

One-Time Pad (Practical Use):

- *Step 1*: Choose a maximum bit length b for messages you want to send.
- *Step 2*: Use some encoding scheme, such as ASCII, to turn your message into a string of 0s and 1s.

01000001	A	01000011	C
01000010	B	01000100	D

- *Step 3*: Pad the message so that it is the right number of bits. For instance, add a bunch of 0's on the right.
  - Example: $010000010100000101000001 \mapsto 010000010100000101000001\underline{00000000}$.
- *Step 4*: Apply the one-time pad as previously described.

One-Time Pad (Practical Use):

- *Step 1*: Choose a maximum bit length b for messages you want to send.
- *Step 2*: Use some encoding scheme, such as ASCII, to turn your message into a string of 0s and 1s.

01000001	A	01000011	C
01000010	B	01000100	D

- *Step 3*: Pad the message so that it is the right number of bits. For instance, add a bunch of 0's on the right.
  - Example: $010000010100000101000001 \mapsto 010000010100000101000001\underline{00000000}$.
- *Step 4*: Apply the one-time pad as previously described.
- *Step 5*: Remove padding, undo encoding.

Pros/Cons of the One-Time Pad:

Pros:

Pros/Cons of the One-Time Pad:

Cons:

Pros:

- Easy to implement.
- Encryption is the same as decryption.
- Information-theoretically secure.

Pros/Cons of the One-Time Pad:

Pros:

- Easy to implement.
- Encryption is the same as decryption.
- Information-theoretically secure.

Cons:

- Fixed length of message.
- Key is as long as the message.
- Vulnerable to known-plaintext attacks.
- Keys cannot be reused without losing secrecy.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.
- Suppose that $\#\mathcal{K} < \#\mathcal{M}$.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.
- Suppose that $\#\mathcal{K} < \#\mathcal{M}$.
- Choose a ciphertext c, and consider all possible decryptions $D_k(c)$.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.
- Suppose that $\#\mathcal{K} < \#\mathcal{M}$.
- Choose a ciphertext c, and consider all possible decryptions $D_k(c)$.
- Since $\#\mathcal{K} < \#\mathcal{M}$, there are fewer decryptions than messages.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.
- Suppose that $\#\mathcal{K} < \#\mathcal{M}$.
- Choose a ciphertext c, and consider all possible decryptions $D_k(c)$.
- Since $\#\mathcal{K} < \#\mathcal{M}$, there are fewer decryptions than messages.
- There is some message m such that $D_k(c) \neq m$ for any key k.

Optimality of One-Time Pad:

Question

Is it possible to have an information-theoretically secure cryptosystem where the keys are shorter than the messages, or where keys can be reused?

- No.
- Suppose that $\#\mathcal{K} < \#\mathcal{M}$.
- Choose a ciphertext c, and consider all possible decryptions $D_k(c)$.
- Since $\#\mathcal{K} < \#\mathcal{M}$, there are fewer decryptions than messages.
- There is some message m such that $D_k(c) \neq m$ for any key k.
- But this means that $\Pr_{k \in \mathcal{K}}(E_k(m) = c) = 0$.

Practical Cryptography:

- It is impossible to do better than the one-time pad while maintaining information-theoretic security.

Practical Cryptography:

- It is impossible to do better than the one-time pad while maintaining information-theoretic security.
- In practice, we require a less stringent security requirement: we only ask that it be *computationally difficult* to find statistical differences between ciphertexts and randomly generated strings.

Practical Cryptography:

- It is impossible to do better than the one-time pad while maintaining information-theoretic security.
- In practice, we require a less stringent security requirement: we only ask that it be *computationally difficult* to find statistical differences between ciphertexts and randomly generated strings.
- Even so, the essential backbone of the one-time pad is still used.

High-level description of the algorithm [edit]

1. `KeyExpansion`—round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.
2. Initial round key addition:
    1. `AddRoundKey`—each byte of the state is combined with a block of the round key using bitwise xor.
3. 9, 11 or 13 rounds:
    1. `SubBytes`—a non-linear substitution step where each byte is replaced with another according to a lookup table.
    2. `ShiftRows`—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
    3. `MixColumns`—a linear mixing operation which operates on the columns of the state, combining the four bytes in each column.
    4. `AddRoundKey`
4. Final round (making 10, 12 or 14 rounds in total):
    1. `SubBytes`
    2. `ShiftRows`
    3. `AddRoundKey`