Perfect Cipher

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
gianluca.dini@unipi.it
Version: 2024-02-26

1 The informal properties talk about "difficult". It would be nice to have "impossible" instead of "difficult". Is this possible?

Towards a secure cipher

- Attacker's ability: (one) cipher-text only attack
- Security requirements
  - Attacker cannot recover the secret key
  - Attacker cannot recover the plaintext
- Intuition of perfectly secure cipher
  - Regardless of *any prior information* the attacker has about the plaintext, the cyphertext should leak *no additional information* about the plaintext

EX: Alice and Bob want to issue an appointment (prior information). Maybe the attacker has some probability distribution info. Alice and Bob exchange communication. But they are not able to extract any additional information.

feb-24 Perfect cipher

2

A probabilistic approach

- Message M is a random variable
  - Plaintext distribution
  - Example
    - Pr[M = "attack today"] = 0.7 suppose this is the distribution.
    - Pr[M = "don't attack] = 0.3
  - Prior knowledge of the attacker Attacker knows this distribution
- Gen() defines a probability distribution over **K**
  - Pr[K = k] = Pr[k ← Gen()]
- Random variables M and K are independent

feb-24 Perfect cipher 3

3

A probabilistic approach

- Ciphertext generation process
  - Choose a message m
  - Generate a key k, k ← Gen()
  - Compute c ← $E_k(m)$
- The ciphertext is a random variable C
- Encryption defines a distribution over the ciphertext **C**

feb-24 Perfect cipher 4

4

Perfect secrecy (informal)

- We formalize «information about the plaintext» in terms of probability distribution
- The adversary's *a-priori* knowledge of the plaintext distribution, i.e. before observing a ciphertext, and the adversary's *a-posteriori* knowledge of the plaintex distribution, i.e. after observing the ciphertext, must be equal

① What they know before observing c

↳ Distribution composed after computing c.

feb-24 Perfect cipher 5

5

Perfect secrecy (Shannon, 1949)

- Definition of Perfect secrecy – For every every *m* in M (SET), every *c* in C (SET), with Pr[C = c] > 0, it holds Pr[M = m | C = c] = Pr[M = m] ($m \in M$; RANDOM VARIABLE)

A-priori probability = A posteriory probability.

- An equivalent formulation
  - $\forall m, m' \in M, \forall c \in C, \Pr[E_k(m) = c] = \Pr[E_k(m') = c]$
  - The distribution of the ciphertext does not depend on the plaintex

If I have a cyphertext I cannot say which m gave me the c. The cyphertext gives no additional information.

feb-24 Perfect cipher 6

6

OFTEN ASKED

Shannon's Theorem

Hp: perfect cypher, implication is number of keys.

SETS

- Shannon's Theorem – In a perfect cipher, $|\mathbf{K}| \geq |\mathbf{M}|$ This is a necessary condition, not a sufficient one.
  - i.e., the number of keys cannot be smaller than the number of messages COROLLARY
  - Proof. By contradiction.
    a) Let $|\mathbf{K}|<|\mathbf{M}|$
    b) It must be $|\mathbf{C}| \geq |\mathbf{M}|$ or, otherwise, the cipher is not invertible → I won't be able to decrypt anymore
    c) Therefore, $|\mathbf{C}| > |\mathbf{K}|$
    d) Select m in $\mathbf{M}$, s.t., $Pr[M = m] \neq 0$; $c_i \leftarrow E(k_i, m)$ for all k_i in $\mathbf{K}$
    e) Because of c), there exists at least one c s.t. $c \neq c_i$, for all i
    f) Therefore $Pr[M = m | C = c] = 0$, that is different of $Pr[M = m]$

GIVEN THAT / CONSTRAINED TO

d) I select 1 message whose prob. is different from 0. Then I start encrypting by means of all possible keys. But $|C|>|K|$, so, there must exist another cyphertext c which is not the image of m by any key.

feb-24 Perfect cipher

7

Shannon's Theorem

- **FACT**. Let Π = (Gen, Enc, Dec) an encryption scheme with $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$. The scheme is perfectly secret iff
  1. Every $k \in \mathcal{K}$ is chosen with equal probaility $1/ |\mathcal{K}|$ by Gen
  2. For every $m \in \mathcal{M}$ and every $c \in \mathcal{C}$ there exists a unique key $k \in \mathcal{K}$ such that $E_k(m) = c$
- Useful for deciding whether a given scheme if perfectly secure
  - Condition 1 is easy to check
  - Condition 2 does not require computing any probabilities

feb-24 Perfect cipher 8

8

Unconditional security

- Perfect secrecy is equivalent to unconditional security
  - An adversary is assumed to have infinite computing resources
  - Observation of the CT provides the adversary no information whatsoever
- Necessary conditions
  - Key bits are truly randomly chosen
  - Key len $\geq$ msg len (Shannon theorem)
    Size of the set of the keys.

feb-24 Perfect cipher 9

9

Perfect indistinguishability

- Yet another definition of perfect secrecy
- **Definition** – An encryption scheme Π = (G, E, D) over ($\mathcal{K}, \mathcal{M}, \mathcal{C}$) has *perfect indistinguishability* iff
  - For all $m_1, m_2 \in \mathcal{M}$,, $|m_1| = |m_2|$
  - with k $\leftarrow$ Gen() (uniform)
  - For all $c \in \mathcal{C}$, $\Pr[E(k, m_1) = c] = \Pr[E(k, m_2) = c]$
- **Fact** – Π has perfectly indistinguishability iff it is perfectly secure

feb-24 Perfect cipher 10

10

Perfect Cipher

ONE-TIME PAD

One Time Pad PERFECT CYPHER

- Patented in 1917 by Vernam
  - Known 35 years earlier
- Proven perfect by Shannon in 1949
- Moscow-Washington “red telephone”
  - In reality a secure direct communication link
    - Teletype, fax machine, secure computer link (email)
  - Never a telephone (not even red)

Preliminary

- Or-exclusive (xor)
  - Truth table

x	y	z = x ⊕ y
0	0	0
0	1	1
1	0	1
1	1	0

  - Matematically
    - $z = x \oplus y = (x + y) \bmod 2$

One Time Pad

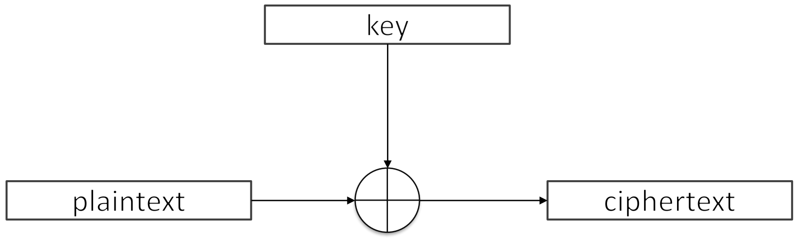
sequence of r bits

- Assumptions
  - Let x be a t-bit message, i.e., $x \in \{0,1\}^t$
  - Let k be a t-bit key stream, $k \in \{0, 1\}^t$, where each bit is truly random chosen

key is as long as the message

- Encryption
  - For all i in [1,…,t], $y_i = m_i \oplus k_i$ i.e., $y_i = m_i + k_i \bmod 2$
- Decryption
  - For all i in [1,…, t], $x_i = c_i \oplus k_i$, i.e., $x_i = y_i + k_i \bmod 2$
- Consistency property can be easily proven

One-Time Pad



15

Xor is a good encryption function

- Theorem – Let X be a random variable over $\{0, 1\}^n$, (n bits) and K an independent uniform variable over $\{0,1\}^n$. (n bits) Then, $Y = X \oplus K$ is uniform over $\{0,1\}^n$.

  ↓ generated in a perfectly random way. Every bit has 0.5 probability of being generated and doesn't depend on previous bits or on plaintext.

  – Proof (for n = 1).
    - Let Pr[X = 0] = X0, Pr[X = 1] = X1, X0 + X1 = 1
    - Pr[Y = 0] =

      = Pr[(X = 0)∧(K = 0)] + Pr[(X = 1)∧(K = 1)] =

      = Pr[X = 0]×Pr[K = 0] + Pr[X = 1] ×Pr[K = 1] =

      = X0 × 0.5 + X1 × 0.5 = 0.5 × (X0 + X1) =

      = 0.5

      ↳ Cyphertext appears as a uniform variable regardless of the input probability.

16 This is because the # of 0s and 1s in XOR is balanced.
Key has to be perfectly random.

OTP has perfect secrecy

- Theorem – OTP has perfect secrecy
  - Proof

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

a) Pr[M = m | C = c] = (Bayes law)
   = Pr[C = c | M = m] × Pr[M = m]/Pr[C = c]

b) Pr[C = c] = (Total probability law)
   = Σ_i Pr[C = c | M = m_i]×Pr[M = m_i] =
   = Σ_i Pr[K = c ⊕ m_i]×Pr[M = m_i] =
   = Σi 2^{-k} ×Pr[M = m_i] = 2^{-k}

$$P(B) = P\left(\frac{B}{A_1}\right) * P(A_1) + P\left(\frac{B}{A_2}\right) * P(A_2) + \cdots + P\left(\frac{B}{A_K}\right) * P(A_K)$$

c) Put b) into a)
   Pr[M = m | C = c] =
   = Pr[K = c ⊕ m] ×Pr[M = m]/2^{-k}
   = 2^{-k} ×Pr[M = m]/2^{-k} =
   Pr[M = m]

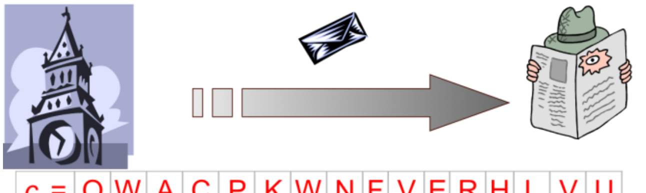
17

OTP has perfect secrecy: intuition

→ exclusive OR in base 26: addition base 26.

- **c[i] = m[i] + k[i] mod 26**
- **m = "SUPPORT JAMES BOND"**

m =	S	U	P	P	O	R	T	J	A	M	E	S	B	O	N	D
	XOR	XOR														
k =	W	C	L	N	B	T	D	E	F	J	A	Z	G	U	I	R
c =	O	W	A	C	P	K	W	N	F	V	E	R	H	I	V	U



c =	O	W	A	C	P	K	W	N	F	V	E	R	H	I	V	U
k' =	M	W	L	J	V	T	S	E	F	J	A	Z	G	U	I	R
m =	C	A	P	T	U	R	E	J	A	M	E	S	B	O	N	D

19

Spectre knows just C=16 so key is 16 chars and m is 16 chars.
By bruteforce, Spectre generates all the possible configurations of 16 characters and they don't know which one is more likely because the keys are all equally alike.

Pros and Cons

- Pros
  - Unconditionally secure → if the key is uniformly randomly generated, OTP is perfect.
    - A cryptosystem is unconditionally or information-theoretically secure if it cannot be broken even with infinite computational resources
  - OTP is optimal
    - Only one key maps m into c
    - $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$ ← I use the minimum number of bits
- Very fast enc/dec
  ↳ XOR is very fast

feb-24 Perfect cipher 20

20

Pros and Cons

- Cons
  - Long keys: unpractical!
    - Key len == msg len
    - In general, $|\mathcal{K}| \geq |\mathcal{M}|$
  - Keys must be used once: avoid two-time pad!
    - Let C1 = M1 xor K and C2 = M2 xor K ➔
    - C1 xor C2 = M1 xor M2

Consider an email

Certain emails are similar. Headers and trailers are similar.
So the areas in which the two messages are equal produce bits = 0. But $M_1 \oplus M_2 = C_1 \oplus C_2$.
By reasoning on cyphertext I get info on the plaintext.

feb-24 Perfect cipher 21

21

We are revealing info, so cypher is not perfect anymore.

Pros and Cons

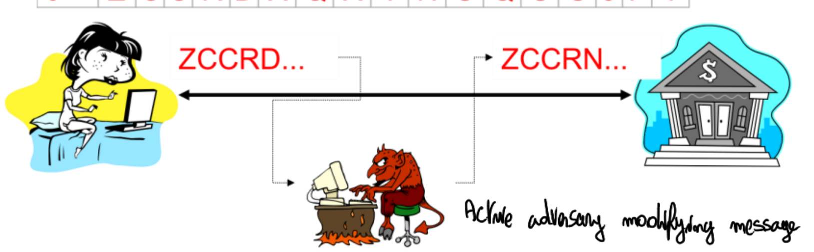
- Cons
  - A Known-PlainText attack breaks OTP
    - Given (m, c) => k = m xor c
  - OTP is malleable
    - Modifications to cipher-text are undetected and have predictable impact on plain-text

feb-24 Perfect cipher 22

22

OTP is malleable

m =	D	A	R	E	C	E	N	T	O	E	U	R	O	A	B	O	B
k =	W	C	L	N	B	T	D	E	F	J	A	Z	G	U	I	R	X
c =	Z	C	C	R	D	X	Q	X	T	N	U	Q	U	U	J	F	Y



c' =	Z	C	C	R	N	B	O	P	J	N	U	Q	U	U	J	F	Y
k =	W	C	L	N	B	T	D	E	F	J	A	Z	G	U	I	R	X
m =	D	A	R	E	M	I	L	L	E	E	U	R	O	A	B	O	B

feb-24 Perfect cipher 23

23

Malleability

- Malleability
  - A crypto scheme is said to be *malleable* if the attacker is capable of transforming the ciphertext into another ciphertext which leads to a *known* transformation of the plaintext
    - The attacker does not decrypt the ciphertext, but (s)he is able to manipulate the plaintext in a predictable manner

Malleability is different for this; manipulation is done in a PREDICTABLE manner, leading to a known transformation

feb-24 Perfect cipher 24

24

On OTP malleability

- Attack against integrity
  - Alice sends Bob: $c = p \oplus k$
  - The adversary
    - intercepts c and
    - transmits Bob $c' = c \oplus r$, with r called *perturbation*
  - Bob
    - receives c'
    - Computes $p' = c' \oplus k = c \oplus r \oplus k = p \oplus k \oplus r \oplus k$ so obtaining $p' = p \oplus r$
    - The perturbation goes undetected and
    - The perturbation has a predictable impact on the plaintext

feb-24 Perfect cipher 25

25

But strongest disadvantage is long keys and keys not reusable.
Malleability is an integrity problem you can solve with hash functions

OTP Malleabililty: example

- Assume the adversary intercepts an encrypted email. The adversary does not know anything about the email, but Bob is the sender. Furthermore, since the message comes from Bob, then the adversary knows that the first line of the message is “from: Bob”. The adversary wants to make the message to appear as coming from Eve.
- The adversary has only to apply a change to bytes 7-9 and transform the from ‘B’ ‘o’ ‘b’ to ‘E’ ‘v’ ‘e’. This is quite simple:
- ~~X~~ r = [‘B’ ‘o’ ‘b’] xor [‘E’ ‘v’ ‘e’] (byte-wise xor)
- If we consider the Ascii codes
  - B o b → 42 6F 62, E v e → 45 76 65
- ~~X~~ r = Bob xor Eve (byte-wise xor) = 07 19 07

PERTURBATION

feb-24 Perfect cipher 26

26

Not today, not tomorrow!
Not today, maybe tomorrow

Remainder of probability theory

- Random variable, probability distribution
- Conditional probability
  - Pr[A|B] = Pr[A ∧ B]/Pr[B]
- Bayes’ Theorem
  - Pr[A|B] = Pr[B|A] × Pr[A]/Pr[B]

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- Law of total probability
  - $\{E_i\}$ are a *partition* of all possible events
    - For all i, j, i ≠ j, E_i and E_j are pairwise impossible ($E_i \cap E_j = \emptyset$)
    - At least some E_i occurs
  - For any event A, $Pr[A] = \Sigma_i Pr[A \wedge E_i] = \Sigma_i Pr[A|E_i] \times Pr[E_i]$

feb-24 Perfect cipher 27

27

Pr [C='b'] = Pr[C='b'|

Example 1

- Shift cipher
  - K = {0, …, 26}, Pr[K = k] = 1/26 (random)
  - Pr[M = ‘a’] = 0.7; Pr[M = ‘z’] = 0.3 (a-priori distribution)
  - Compute Pr[C = ‘b’]
    - Result = 1/26

$$Pr[C=b] = \underbrace{Pr[C=b \mid M=a]}_{Pr[K=1]} \cdot Pr[M=a] + \underbrace{Pr[C=b \mid M=z]}_{Pr[K=2]} \cdot Pr[M=z]$$

$$Pr[C='b'] = Pr[M='a'] \cdot Pr[K='1'] + Pr[M='z'] \cdot Pr[K='2'] = \frac{1}{26} \cdot 0.7 + \frac{1}{26} \cdot 0.3 = \frac{1}{26}$$

28

Example 2

- Shift cipher
  - K = {0, …, 26}, Pr[K = k] = 1/26 (random)
  - m1 = «ONE», m2 = «TEN»
  - Pr[M = m1] = Pr[M = m2] = 0.5 (a-priori distribution)
  - Compute Pr[C = «RQH»]
    - Result = 1/52

$$Pr[C=RQH] = \underbrace{Pr[C=RQH \mid M=m_1]}_{Pr[K=K_0]} \cdot Pr[M=m_1] + \underbrace{Pr[C=RQH \mid M=m_2]}_{0} \cdot Pr[M=m_2] =$$

$$Pr[K=K_0] \cdot 0.5 + 0 \cdot 0.5 = 1/2 \cdot \frac{1}{26} = \frac{1}{52}$$

There is no key that shifts message that way.

29

Example 3

- Shift cipher
  - K = {0, …, 26}, Pr[K = k] = 1/26 (random)
  - m1 = «ONE», m2 = «TEN»
  - Pr[M = m1] = Pr[M = m2] = 0.5 (a-priori distribution)
  - Compute Pr[M=«TEN»|C = «RQH»]
    - Result = 0 that is different of Pr[M = «TEN»] ➔
  - Shift cipher is not perfect

The cypher is not perfect because $Pr[M = m_2 | C = RQH] \neq Pr[M = m_2]$

feb-24 Perfect cipher 30

30

Example 4

- Shift cipher
- Message distribution
  - Pr[M = «HI»] = 0.3
  - Pr[M = «NO»] = 0.2
  - Pr[M = «IN»] = 0.5
- Compute Pr[M = «HI»|C = «XY»]
  - Pr[M=«HI»|C=«XY»] = (Bayes' law) =
    = Pr[C = «XY»|M=«HI»]·Pr[M=«HI»]/Pr[C=«XY»]
  - Pr[C = «XY»|M=«HI»] = Pr[K = 16] = 1/26 (continue)

feb-24 Perfect cipher 31

31

Example 4 continued

- Compute Pr[M = «HI»|C = «XY»]
  - Pr[C = «XY»] = (law of total probability)
    Pr[C=«XY»|M=«HI»]·Pr[M=«HI»]+
    Pr[C=«XY»|M=«NO»]·Pr[M=«NO»]+
    Pr[C=«XY»|M=«IN»]·Pr[M=«IN»] =
    = (1/26)·0.3 + (1/26)·0.2 + 0·0.5 =
    = 1/52
  - Pr[M = «HI»|C = «XY»] = (1/26)·0.3/(1/52) = 0.6
    ≠ Pr[M = «HI»] ➔
- Shift cipher is not perfect