Computer Security and Cryptography

CS381

3. Security and Complexities

来学嘉　计算机科学与工程系 电院3-423室
34205440 13564100825 laix@sjtu.edu.cn
2016-03

Organization

- Week 1 to week 16 (2016-02-24 to 2016-06-08)
- 东上院502
- Monday 3-4节; week 9-16
- Wednesday 3-4节; week 1-16
- lecture 10 + exercise 40 + random tests 40 + other 10
- Ask questions **in** class – counted as points
- Turn ON your mobile phone (after lecture)
- Slides and papers:
  - http://202.120.38.185/CS381
    - **computer-security**
  - http://202.120.38.185/references
- TA: '薛伟佳' xue_wei_jia@163.com，'黄格仕' <huang.ge.shi@foxmail.com>
- Send homework to: laix@sjtu.edu.cn and to TAs

Rule: do not disturb others!

2

Contents

- Introduction -- What is security?
- Cryptography
  - Classical ciphers
  - Today's ciphers
  - Public-key cryptography
  - Hash functions/MAC
  - Authentication protocols
- Applications
  - Digital certificates
  - Secure email
  - Internet security, e-banking

Network security
- SSL
- IPSEC
- Firewall
- VPN

Computer security
- Access control
- Malware
- DDos
- Intrusion

Examples
- Bitcoin
- Hardware
- Wireless

5

References

- W. Stallings, *Cryptography and network security - principles and practice*， Prentice Hall.
- W. Stallings, 密码学与网络安全：原理与实践（第4版），刘玉珍等译，电子工业出版社，2006
- Doug Stinson , *Cryptography Theory and Practice*, Third Edition, CRC Press, 2005
- Lidong Chen, Guang Gong, *Communication and System Security*, CRC Press, 2012.
- A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997, ISBN: 0-8493-8523-7, http://www.cacr.math.uwaterloo.ca/hac/index.html
- B. Schneier, *Applied cryptography*. John Wiley & Sons, 1995, 2nd edition.
- 裴定一,徐祥, 信息安全数学基础, ISBN 978-7-115-15662-4, 人民邮电出版社,2007.

6

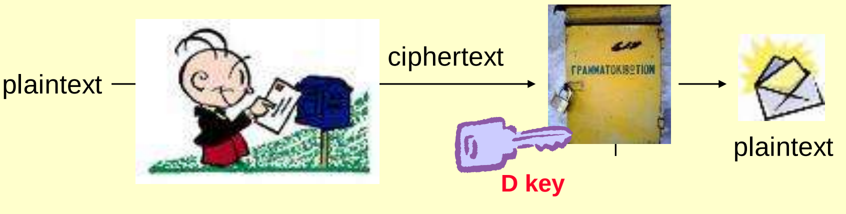
Issues -- services

- Issues in Information security
  - Confidentiality/secrecy保密 --- 防止未经授权的信息泄漏.-- information is not disclosed to unauthorized individuals, entities, or processes.
  - Authentication 认证,真实性 --- 确认身份--assurance that the communicating entity is the one claimed
  - Data Integrity 完整性-确认数据未被篡改--assurance that data received is as sent by an authorized entity
  - Non-Repudiation不可否认性,抗抵赖 – 防止否认已做过的事--protection against false denial of a taken action.
  - Access control 访问控制 --- 确定谁在什么条件下可做什么事.
- (Scientific like)

7

☆ Confidentiality

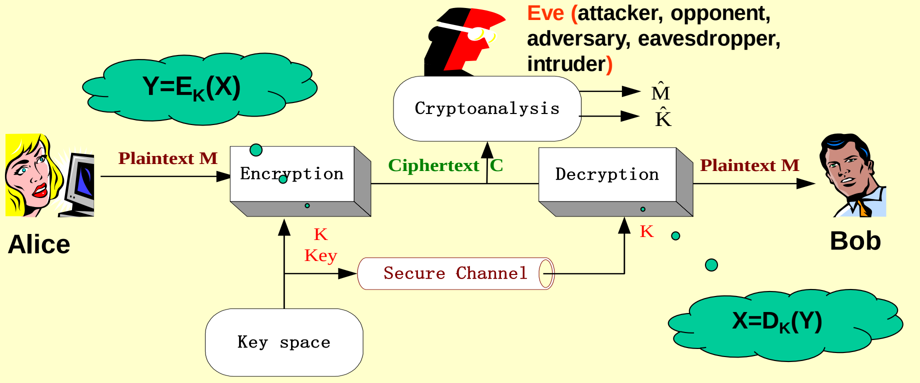
- Confidentiality : information is not disclosed to unauthorized individuals, entities, or processes. [ISO]
- Mechanism to achieve confidentiality--Encryption:



Only the user knowing the decryption key can recover plaintext

–"who can *read* the data"

Shannon's model of secret communication



- Adversary has full access to the ciphertext Y;
- Key *k* is transmitted through a secure channel
- The other parts: plain text source, encrypter, decrypter, key source and key transmission channel are all assumed to be secure;

☆ classify Attacks according to attacker's knowledge

[Kerckhoff] Attacker knows all details of E(;), D(;) except for the current key

- ciphertext-only attack
  – attacker has full access to ciphertext
- known-plaintext attack
  – attacker knows some plaintext/ciphertext pairs for the current key
- chosen-text attack
  – attacker can get ciphertexts (plaintexts) for some plaintexts (ciphertexts) of his choice
- adaptively chosen-text attack
  – attacker's choice depends on the obtained texts

10

☆

Remarks on Kerckhoff's principle

[Kerckhoff] Attacker knows all details of the cryptosystem except for the current key

- Violating Kerckhoff's principle (unknown cipher)
  - it could be more difficult to break the cipher, but
  - incompatible with other entities
  - dependence on some authorities
  - when the secret leaks, it is difficult to fix
- ➢ Security should not depend on the secrecy of the algorithm
- Standardization provide confidence in ciphers
- ➢ In open systems, one should use standard ciphers

11

☆

Public Standard vs. secret system

2 sides of a sword:

Public standard

- compatible with other entities, Independent of providers
- confidence and trust in systems
- Up to date techniques / fast reaction to incidents
- Security should not depend on the secrecy of the system

Secret System

- it could be more difficult to break the system, because adversary has less knowledge of the system in use
- Support from the provider, Insurance by the authority

Develop system : assume enemy knows everything except the key.

Use of system: hide as much as possible

☆ Classify security according to attacker's capability

- Unconditional (informational) security – depends only on enemy's knowledge, but not on enemy's computation power.
  - Perfect secrecy : ciphertext and plaintext are statistically independent
  - Strongly ideal cipher: secret-key is independent of ciphertext
  - Truly unbreakable system: secure against chosen-texts attacks
- Conditional security - Computational security: security depending on enemy's computation power.
  - Difficulty – complexity
  - Turing-machine complexity
  - Gate complexity

13

☆ Unconditional security - 1

Unconditional (informational) security – depends only on enemy's knowledge, but not on enemy's computation power.

Perfect secrecy (Shannon 1949):
- ciphertext Y and plaintext X are statistically independent
- $P(X \mid Y)=P(X)$

Example: one-time-pad (Vernam 1926): $y = x \oplus k$,

Theorem (Shannon 1949): one-time-pad can achieve perfect secrecy if k (key) is uniformly random, independent of plaintext x and used only once.

$$p(y=y_0 \mid x=x_0)=p(k=x_0 \oplus y_0 \mid x=x_0)=p(k=k_0)=1/|K|$$
$$p(y=y_0)=\Sigma_{x0}\, p(y=y_0 \mid x=x_0)p(x=x_0) = 1/|K|\, \Sigma_{x0}\, p(x=x_0) = 1/|K|$$

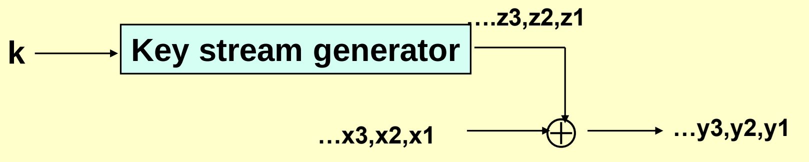
14

☆ **Perfect secrecy**

Perfect secrecy system is not practical because
#(key) ≥ #(plaintext)

Practical means to use key more than once. In this case, Vernam cipher is not secure against known plaintext attack

Stream cipher: a short key is used to produce a long key stream which appears to be a random sequence



15

熵 – Entropy， Uncertainty

- Shannon Entropy， Information Entropy.
- For a random variable X with probability distribution $p_1, p_2, \ldots, p_n$, its Entropy is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i).$$

- For uniform distribution, H(X)= log(n) bits (max)
- Also called Uncertainty

16

☆ Unicity distances

- Redundancy of (encrypted) plaintext X:

$$R = 1 - H(X) / |X|$$

- $R = 0.8$ for english text (i.e., 5-bit character has 1-bit information)
- Unicity distance [Shannon 49], n_U of a cipher is the smallest number of ciphertexts so that the key is unique.

$$n_U = |K| / R$$

- DES: $n_U = 56/0.8 = 70$ bits (2 blocks)
- 熵(entropy) $\mathbf{H}(X) = -\Sigma x \Pr[X = x] \log \Pr[X = x]$,

17

☆ Unconditional security -2

Strongly ideal cipher [Shannon 49]:

- A cipher is strongly ideal if ciphertext Y contain no information about the secret-key K, $P(K|Y)=P(K)$

- Strongly ideal cipher is unconditionally secure against ciphertext only attack
- Strongly ideal cipher can be achieved by perfect compression of plaintext source [Shannon 49]
- Impractical: perfect compression is hard to achieve
- One should compress plaintext before encryption

18

☆ Unconditional security -3

Truly unbreakable system [Massey 87]:

– for any fixed key and for any choices of plaintext/ciphertext pairs $(x_1,y_1)\ldots(x_n,y_n)$, the remaining ciphertext and plaintext are statistically independent

Example. Random cipher: encryption function $E(\ ;k)$ is uniformly chosen from the set of all invertible mappings of plaintext space (F_2^m)

- Truly unbreakable ⇔ random cipher
- Truly unbreakable system is unconditionally secure against chosen-texts attacks
- Truly unbreakable system is not practical:
  – #(key) = $2^m!$, $\log(2^m!) \sim (m-1.44)2^m$ (one bit plaintext needs $m2^m$ bits key
  – a random permutation is hard to compute – an m-bit permutation requires about 2^m binary operations for almost all permutations [Shannon 1949].

19

☆ Unconditional security

attack	ciphertext-only		chosentext
security	*perfect secrecy* X and Y are statistically independent	*strongly ideal* H(K/Y$_1$,..)=H(K) ciphertexts contain no info on key	*truly unbreakble* X and Y are stat. ind. even when $\{x_1,\ldots,x_n\}\to\{y_1,\ldots,y_n\}$ are known
example	one-time-pad $y_1,\ldots,y_n \leftarrow \otimes \leftarrow x_1,\ldots,x_n$ ↑ $k_1,\ldots,k_n$	X: i.i.d. random + any cipher	random cipher E(•,k)∈SYM(F_2^m)
impractical	#{k} ≥ #{x}	require perfect data-compression	#{k} ≥ 2^m! comp(E(•,k))≥2^m
significance	change key frequently	randomize plaintexts encrypt session-keys with a master key	random appearence of E(•,z)
constrain	key	plaintextsource	encryption function

20

Conditional security

- Conditional security - Computational security: security depends on enemy's computation power.
- Fact: attacker has only limited resource
- 'the problem of cipher design is essentially one of finding difficult problems' [Shannon 49]
- 'difficult' – 'hard' – 'infeasible'
- Historical difficulty – 'best known attack'
- Intrinsic difficulty --- provably secure – minimum resource needed to solve the problem.
- Provable security requires to define 'difficulty'

21

Difficulty and proof

- Basic approach

1. Reduce the security of a system to the difficulty of a problem
2. Determine the complexity of solving that problem

First, we need to define "difficulty"

- Complexity:
  - Turing-machine complexity
  - Gate complexity

22

Complexity classes

- Turing-Machine complexity: running time as function of input size.
- decision problems: answer YES/NO
  - class P: problems solvable in polynomial time
  - class NP: problems for which a YES can be verified in poly. time given some extra-info (certificate)
  - class co-NP: problems for which a NO can be verified in poly. time given a certificate
- Example: Problem: is n composite? i.e, are there a,b s.t. n = a*b ?. This problem is NP. Is it P?
- P $\subseteq$ NP and P $\subseteq$ co-NP
  - Is P = NP ?
  - Is NP = co-NP ?
  - Is P = NP ∩ co-NP ?
  - for all three questions, the current guess is NO

23

☆Define difficulty: Turing-Machine complexity

Turing-Machine complexity is
- uniform (one algorithm for every input length) and
- Asymptotic (complexity is about f(n) when n $\to\infty$)

- If P = NP, then we can solve many problem in polynomial time; does this means we cannot have provably secure system?
  - Answer: no
  - If we a cipher that encryption complexity is n, but attack needs at least n^3 , then this would be enough for many practices
- If P ≠ NP, do we have provably secure system?
  - Answer: no.
    - ∵ attacker works only on one fixed-size problem

      E.g. even if TM-complexity of factoring is exponential, to factor a specific integer can still be easy. To break a given RSA, you need only to factor one integer, not every integer.
    - ∵ Example: there exist problem, for which the uniform complexity is super exponential, but the fixed length complexity is linear [Cohen}.

24

uniform

- Uniform: ∃ algorithm A, such that, ∀ input x, A(x) is computable
- Non-Uniform: ∀ input x, ∃ algorithm A, such that,, A(x) is computable

- Example: sum of n1,n2,…,n100
- Uniform: n1+n2+n3+…+n100
- Non-unif. For sum of 1,2,…,100, ∃faster algorithm

Provable security

Ultimate goal – provably secure: "to break my system, you need to do at least this much work"

Limitation of Turing-Machine complexity -- Such definition of complexity=difficulty cannot solve the task of "provably secure“ system

Despite of limitation, the approach of Turing-Machine complexity is useful for proving security:

- Reduce an unknown security problem to some well known problems, e.g., integer factorization, discrete logarithm, …
- so we can concentrate on few well-studied problems.
- “security proof” today usually means the security of a system can be reduced to some known problem.
- Most algorithms we use are uniform
- Goldwasser-Michali won Turing price

Gate complexity

A *gate* is any of the 16 Boolean functions of two variables

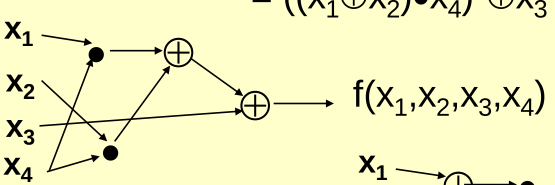
x_1	x_2	f_0	f_1 AND	f_2	f_3	f_4	f_5	f_6 XOR	f_7 OR	f_8 NOR	…	f_{14} NAND	f_{15}
0	0	0	0	0	0	0	0	0	0	1		1	1
0	1	0	0	0	0	1	1	1	1	0		1	1
1	0	0	0	1	1	0	0	1	1	0		1	1
1	1	0	1	0	1	0	1	0	1	0		0	1

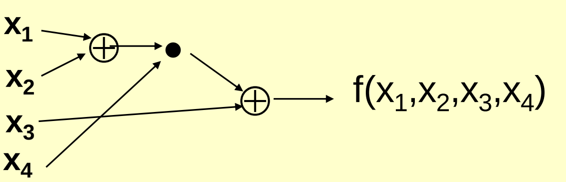
27

circuits

- a function can be computed by a circuit consists of binary gates;
- Example:
- $f(x_1,x_2,x_3,x_4) = x_1x_4 \oplus x_2x_4 \oplus x_3$ *(4 gates)*

  $= ((x_1 \oplus x_2) \bullet x_4) \oplus x_3$ *(3 gates)*





☆ Gate complexity

There can be many different circuits computing a function

Definition. Circuit/gate complexity of a function– **The minimum number of binary gates needed in the circuit computing the function**

- All functions can be computed by binary operations;
- This definition reflects the present reality
- Circuit/gate complexity is non-uniform,

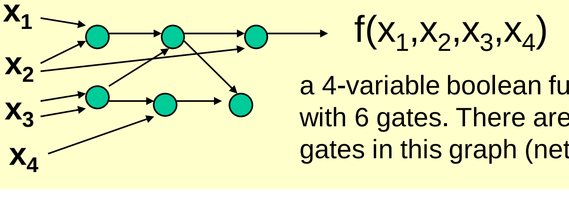
29

☆ gates needed to compute a function

- There are 2^{2^n} Boolean functions of n variables.
- Theorem. [Shannon 49]. For n>7, the number of Boolean functions of n variables that can be computed with K gates is

$$M(n.K) \le K^{2K}\, 16^{K}$$

number of connection graph=#mappings |2K| to |K|



a 4-variable boolean function computed with 6 gates. There are 16^6 choices of gates in this graph (network)

30

Existence of "hard" functions

For $n \geq 8$, $K = \frac{1}{2} 2^n/n$, $M(n.K) \leq K^{2K} 16^K < 2^{2^n}$

- The number of n-variable boolean functions that can be computed with $\frac{1}{2} 2^n/n$ gates is less than the total number of such functions. i.e.,
- Theorem. For $n \geq 8$, there exists boolean function of n variables for with gate complexity at least $\frac{1}{2} 2^n/n$.
  - (true for n>1, except n=6,7. [Hiltigen 93])
- Existence of "hard" functions (Shannon's counting argument)
  - for almost all n-bit to n-bit functions, $C_g(f) \geq 2^n$ [Shannon 1949].

  $\Rightarrow$ A randomly chosen function with $n \geq 50$ is not realizable (the largest Intel CPU has 2^{30})

Gate complexity

Almost all functions are hard to compute. However,

- We don't know how to construct a specific function with high gate complexity
- To specify a class of functions of n (=1,2,,,) variables with provable lower bound on gate complexity
  - best known lower bound: $3n$ [Paul-Blum 84]
  - The difficulty of such construction can be seen from the following:
    - Let f(x) have gate complexity C_g (f)~2^n, then F(x,y)=(f(x),f(y)) has complexity C_g (F)=?

32

Gate complexity

Almost all functions are hard to compute. However,

- We don't know how to construct a specific function with high gate complexity
- To specify a class of functions of n (=1,2,,,) variables with provable lower bound on gate complexity
  - best known lower bound: 3n [Paul-Blum 84]
  - The difficulty of such construction can be seen from the following result:
    - Let f(x) have gate complexity C_g (f)~2^n, then F(x,y)=(f(x),f(y)) has complexity ~2^n ,not $2 \bullet 2^n$, as we usually believe [Paul,76,Th.comp]

☆ Gate complexity of 1-way functions

For one-way functions, i.e., a function f for which $C_g(f) \neq C_g(f^{-1})$, we know:

- One way function exist (for n>3)
- Example [Hiltgen-Ganz 90]

$y_1= x_1 \oplus x_3$	$x_1=[(y_1 \oplus y_3) \oplus y_2)] \bullet y_4 \oplus (y_1 \oplus y_3)$
$y_2=x_2 \oplus [(x_1 \oplus x_2) \bullet x_4]$	$x_2=[(y_1 \oplus y_3) \oplus y_2)] \bullet y_4 \oplus y_2$
$y_3=x_3 \oplus [(x_1 \oplus x_2) \bullet x_4]$	$x_3=[(y_1 \oplus y_3) \oplus y_2)] \bullet y_4 \oplus y_3$
$y_4=x_4$	$x_4=y_4$

- $C_g(f)=5$ $\qquad$ $C_g(f^{-1})=6$

34

☆ Gate complexity

- Best known provable one-wayness:
  for $n \geq 4$, $C_g(f) = n+2$, $C_g(f^{-1}) = 2(n-1)$
  [Hiltigen 93]
- For $n \geq 6$, virtually all n-bit invertible function *f*, $C_g(f) / C_g(f^{-1}) \leq 2.5$ [Massey 1996]
- we are still far from the goal of provably secure system

35

☆ Difficulty and complexity

- ‘the problem of cipher design is essentially one of finding difficult problems’ [Shannon 49]
- Our goal: proving that “to break the system needs at least this much work“
- “finding difficult problems” requires to define difficulty – complexity.
- Turing machine complexity does not meet our requirement (but still useful)
- Gate-complexity – right definition but current results are not useful.
- Lots of researches and results but far from our goal.

A remark

Distinguish:

- Provably secure : “to break the system requires at least this much work”
  - today usually means the security of a system can be reduced to some known problem.
- Provable security: to show a system have some desirable property.

“if a cipher is provably secure, then it is probably breakable”
– Lars Knudsen

37

☆ 3 papers of **Claude Elwood Shannon**

- A mathematical theory of communication, *Bell System Technical Journal, 1948 July and October*
- *The Mathematical Theory of Communication,* Urbana, Illinois: University of Illinois Press, 1949.
- Communications theory of secrecy systems, *Bell System Technical Journal*, vol.28(4), page 656–715, 1949.
- The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59-98, 1949
  - "A Symbolic Analysis of Relay and Switching Circuits," thesis for Master in electrical engineering, MIT, 1940. (‘best master thesis ever written’)

38

Computational security today

- Gate complexity appears to be adequate for provable security, but
  - The known results are far from useful
  - Most of functions have high complexity
  - Lower bound on specific functions is only linear
- Known results in gate complexity is hardly usable for proving security.
- Finding non-linear complexity for specific functions would be a break-through for cryptography.

39

Computational security today

- Turing-Machine complexity has limitation for cryptosystem
  - Computation complexity is widely studied and important for security.
  - Basic idea – reduction: security of a cryptosystem can be reduced to the difficulty of a known problem
  - Proof of equivalence, or relations between difficult problems: factorization-RSA, DH-Dlog, cipher-hash-RNG,…
  - A new design is hard to be accepted without some ‘provable security’.
  - Appears to be the main area of current crypto research.

40

☆ Historical security

Historical security

- computation needed by the best algorithm known for breaking the cipher
  - e.g. factoring integer for breaking RSA
  - historical security can change overnight
- in practice, a cipher is generally considered secure if none of the known attacks can do better than the exhaustive key search:
  - for given x_0 and $y_0 = E(x_0,k_0)$, try each possible k until $E(x_0,k)=y_0$
  - about 2^{k-1} trial encryptions are required

41

☆ exhaustive key search

- **8-Byte (64-bit) Key**

Attacker	Budget	Time needed in 1998	Time needed in 2008
small company	$10K	640 days	20 days
big company	$10M	15 hours	0.5 hour
special agency	$300M (ASIC)	30 minutes	1 minutes

- **10-Byte (80-bit) Key**

Attacker	Budget	Time needed in 1998	Time needed in 2008
small company	$10K	>10 years	>10 years
big company	$10M	>10 years	2 years
special agency	$300M (ASIC)	3 years	30 days

.

16 Byte (128-bit) key: More than billions of years

42

☆ Computation power

- Heisenberg uncertainty relation
  - the number of elementary logical operations per second that can be performed by that amount of energy, E,

$$2E/(\pi * \underline{h})\ ,\ \underline{h} = 10^{-34}$$

- Using total energy in the whole universe
  - Max. number of operations: $10^{120} \approx 2^{400}$.
  - Max number of bits storage: $10^{90} \approx 2^{300}$

- Today's world computation power $< 2^{120}$
  - #operations/second $< 2^{40}$
  - #computers < 7,298,320,378 *100 $< 2^{40}$ (2016-01-12)
  - #seconds in 30,000 years $< 2^{40}$

http://www.worldometers.info/watch/world-population/

43

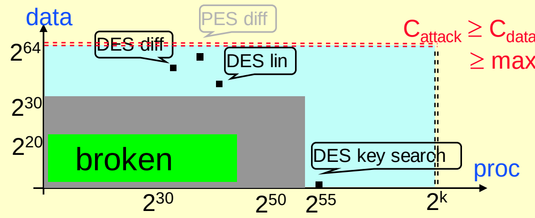
☆ Complexity of an attack

- Data complexity – amount of data needed to carry out the attack.
  - Relatively difficult to obtain
  - Not under control of the attacker
- Processing complexity
  - Computation resource needed to perform the attack.
  - Under the control of attacker
  - Time – running time of computation
  - Memory – storage needed for computation

44

☆ Data and processing complexity of an attack

	data-complexity	processing complexity
definition	*amount of input data* (difficult to obtain)	*computations needed to process the input data*
maximum	2^m (number of possible texts)	2^k (number of possible keys)
DES key search	O(1)	2^{55}
DES diff. attack	2^{47} chosentexts	2^{37}
DES linear attack	2^{43} knowntexts	2^{43}
LFSR (M-B alg) 2^k	2k	k^2
attacker	passive	active



$C_{attack} \geq C_{data} + C_{proc}$
$\geq \max\{C_{data}, C_{proc}\}$

45

Security

- providing security – our goal
- Unconditional – possible to have but impractical
- Computational – provable security
- Difference between random and sample
- Properties of secure system should have
- Specific attacks with known complexity (differential, linear, LSFR, Hash)
- Real attacks (World War 2 history, LSFR)
- Serious flaw

46

One-way function

- The intrinsic problem of information security is the "one-wayness".
- Cryptography studies one-way function
  - The measure of one-way: difficulty
  - The design of one-way function
  - The attacks on one-way function
  - The use of one-way function

47

Summary

- Unconditional:
  - Prefect secrecy; strongly ideal; truly unbreakable
  - achievable, impractical, significance
- Computational: where are we?
  - TM: not true solution
  - Gate: suitable but no usable results
  - data complexity and processing complexity

48

Exercise 3

1. Can the random cipher achieve perfect secrecy?
- Can a strongly ideal cipher achieve perfect secrecy?
- Is one-time-pad a strongly ideal cipher?

2. What are the differences between Turing-machine complexity and gate-complexity?

3. Prove that the complexity of key-search is 2^{k-1}
   --hint : define a random variable and compute the average

Sending the answer to laix@sjtu.edu.cn
Deadline: before next lecture
Next lecture: DES, test1

49