Department of
Computer Science and Engineering
计算机科学与工程系

CSE5014 CRYPTOGRAPHY AND NETWORK SECURITY

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room413, CoE South Tower
Email: wangqi@sustech.edu.cn

Course Information

- Instructor:

  Dr. QI WANG
  Department of Computer Science and Engineering
  Office: Room413, CoE South Tower
  Email: wangqi@sustech.edu.cn

密码学与网络安全2023
群号：458863828

- TA:

  Yuanzhao Li

- Course webpage:

  sakai.sustech.edu.cn $\rightarrow$ "CSE5014 1 1 spring2023"

Course Information

- Lectures:

  Room 206, Business School

  10:20 - 12:10 every Tuesday

Course Information

- Lectures:

  Room 206, Business School

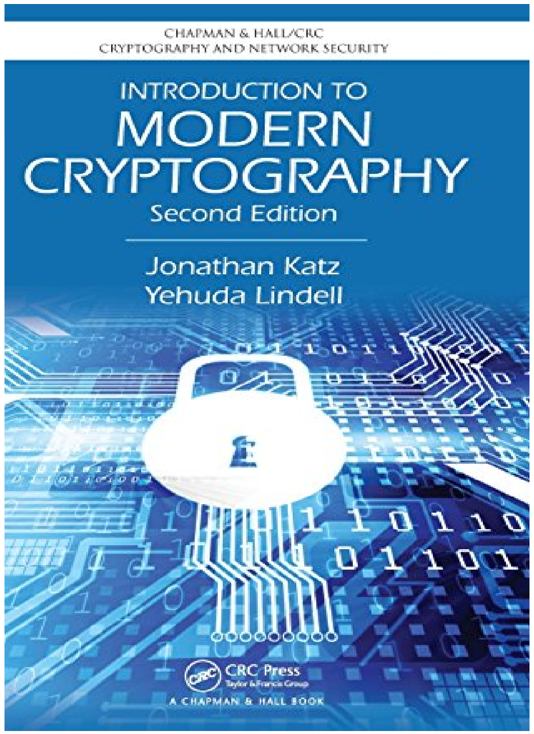
  10:20 - 12:10 every Tuesday

- Office hours:

  Tue. 16:20-18:20 or send email for appointment

Course Information

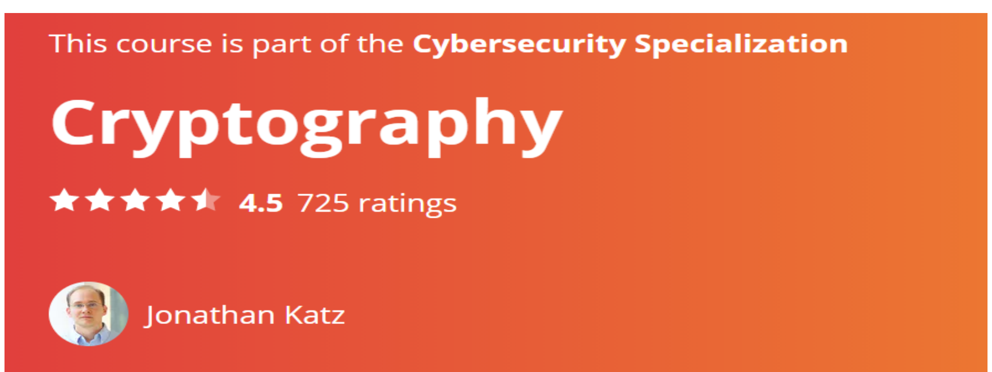
- Reading resources

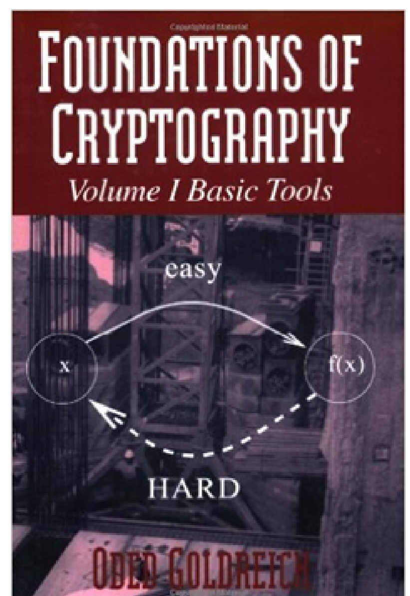


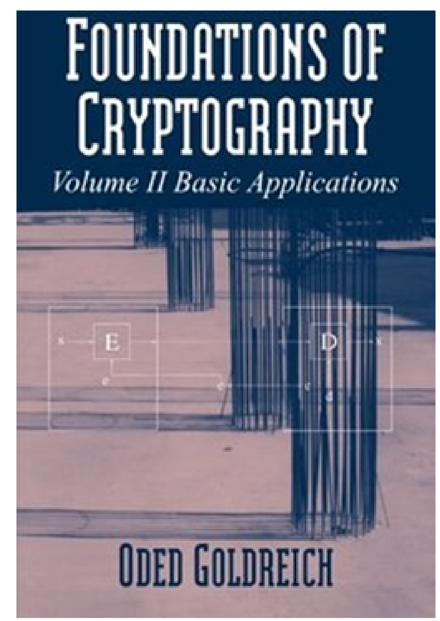
A Graduate Course in Applied Cryptography

Dan Boneh and Victor Shoup

Version 0.4, September 2017







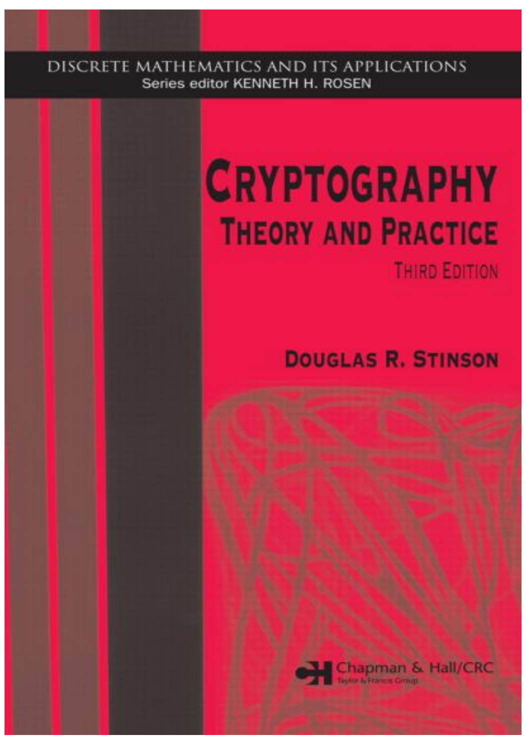
Course Information

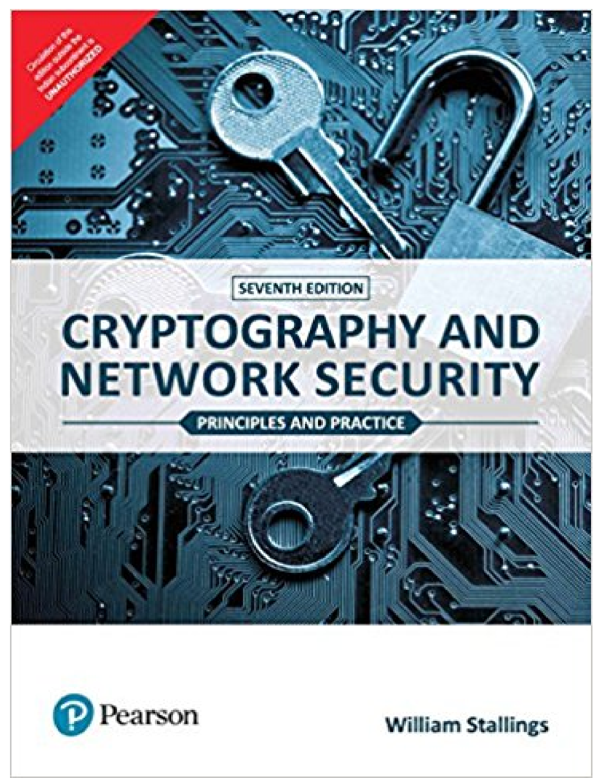
- Reading resources

Lecture Notes on Cryptography

Shafi Goldwasser[1] Mihir Bellare[2]

July 2008





This Course

- What you will learn:
  - Foundations and principles of the science
  - Basic primitives and components
  - Definitions and proofs of security
  - High-level applications

This Course

- What you will learn:
  - ◇ Foundations and principles of the science
  - ◇ Basic primitives and components
  - ◇ Definitions and proofs of security
  - ◇ High-level applications

- What you will *not* learn:
  - ◇ Buzzwords
  - ◇ The most efficient and practical versions of components
  - ◇ Designing secure systems
  - ◇ "Hacking" – breaking into systems
  - ◇ Viruses, worms, Windows/Unix bugs, buffer overflow, ...

This Course

- What you will learn:
  - Foundations and principles of the science
  - Basic primitives and components
  - Definitions and proofs of security
  - High-level applications

- What you will *not* learn:
  - Buzzwords
  - The most efficient and practical versions of components
  - Designing secure systems
  - "Hacking" – breaking into systems
  - Viruses, worms, Windows/Unix bugs, buffer overflow, ...

  But will help you avoid designing insecure systems

Marking Scheme

- Quiz in class (10%)
- Homework assignments (40%)
  - assigned in class and posted on the course webpage
  - must be submitted at the beginning of class on due date
  - no extension policy with exceptions
- Final exam (50%)
  - covers the entire semester's material
- Project (optional, 10%)

Important Messages to Students

- Main ideas will be covered in class but some details might be skipped. You are responsible for all materials in assigned sections of book, even if they are not taught in class.

- Homework assignments should be worked on and written up individually, though collaboration on homework with other students are encouraged (acknowledge this).

- Any unintellectual behavior and cheating on exams, homework assignments will be dealt with severely.

What is OK and what is not OK?

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY | 计算机科学与工程系 Department of Computer Science and Engineering

Undergraduate Students Assignment Declaration Form

This is____________ (student ID: ____________, who has enrolled in __________ course, originated the Department of Computer Science and Engineering. I have read and understood the regulations on plagiarism in assignments and theses according to "Regulations on Academic Misconduct in Assignments for Undergraduate Students in the SUSTech Department of Computer Science and Engineering". I promise that I will follow these regulations during the study of this course.

Signature:

Date:

Important Messages to Students

- Please ask questions in class

If you're having trouble understanding something, then at least 50% of the class is also having trouble: they'll *be happy* if you ask for more explanation.

- The lecture slides are still in progress

If you find mistakes or just think that something's confusing, please email me. Your classmates and future students will *thank you*.

Prerequisites of This Course

- Required:
  1. Ability to read and write mathematical proofs and definitions
  2. Familarity with algorithms - proving correctness and analyzing complexity (O notation)
  3. Familarity with basic probability theory

Prerequisites of This Course

- Required:
  1. Ability to read and write mathematical proofs and definitions
  2. Familiarity with algorithms - proving correctness and analyzing complexity (O notation)
  3. Familiarity with basic probability theory
- Helpful:
  1. Complexity NP-Completeness, reductions, P, ...
  2. Probabilistic algorithms primality testing, hashing, ...
  3. Number theory modular arithmetic, prime numbers, ...

More About This Course

- This course is HARD!
  - ◇ Emphasis on mathematical proofs
  - ◇ Counter-intuitive concepts
  - ◇ Extensive use of quantifiers/probability

More About This Course

- This course is HARD!
  - ◇ Emphasis on mathematical proofs
  - ◇ Counter-intuitive concepts
  - ◇ Extensive use of quantifiers/probability
- There are some reasons!
  - ◇ Good coverage of crypto takes one year.
  - ◇ Simulation/experimentation cannot be used to show security.
  - ◇ Need to acquire "crypto-intuition".
  - ◇ Quantifiers, proofs by contradiction, reductions, probability, etc. are inherent.

More About This Course

- Mitigating hardness
  - Avoid excessive exercises
  - Try best to explain intuiation behind proofs
  - Learn to ask questions

Cryptography

- History of almost 4000 years (from 1900 B.C.)

$$\text{Cryptography} = \text{kryptos} + \text{graphos}$$

Cryptography

- History of almost 4000 years (from 1900 B.C.)

$$\text{Cryptography} = \underset{(\text{secret})}{\text{kryptos}} + \underset{(\text{writing})}{\text{graphos}}$$

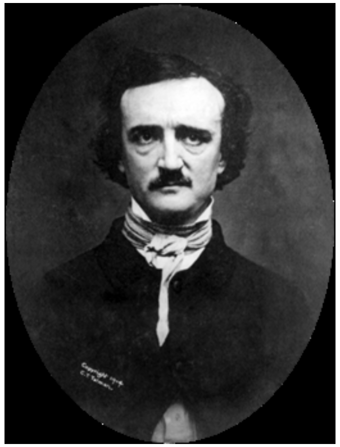
Cryptography

- History of almost 4000 years (from 1900 B.C.)

Cryptography = kryptos + graphos

(secret) (writing)

The term was first used in *The Gold-Bug*, by Edgar Allan Poe.

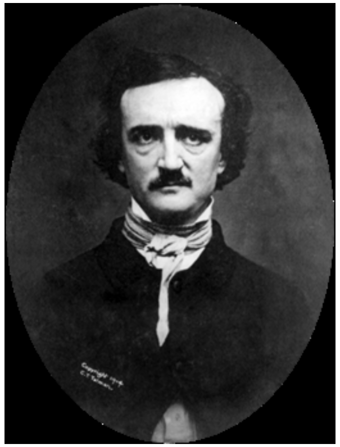


Cryptography

- History of almost 4000 years (from 1900 B.C.)

$$\text{Cryptography} = \underset{(\text{secret})}{\text{kryptos}} + \underset{(\text{writing})}{\text{graphos}}$$

The term was first used in *The Gold-Bug*, by Edgar Allan Poe.



"Human ingenuity cannot concoct a cipher which human ingenuity cannot resolve." – 1941

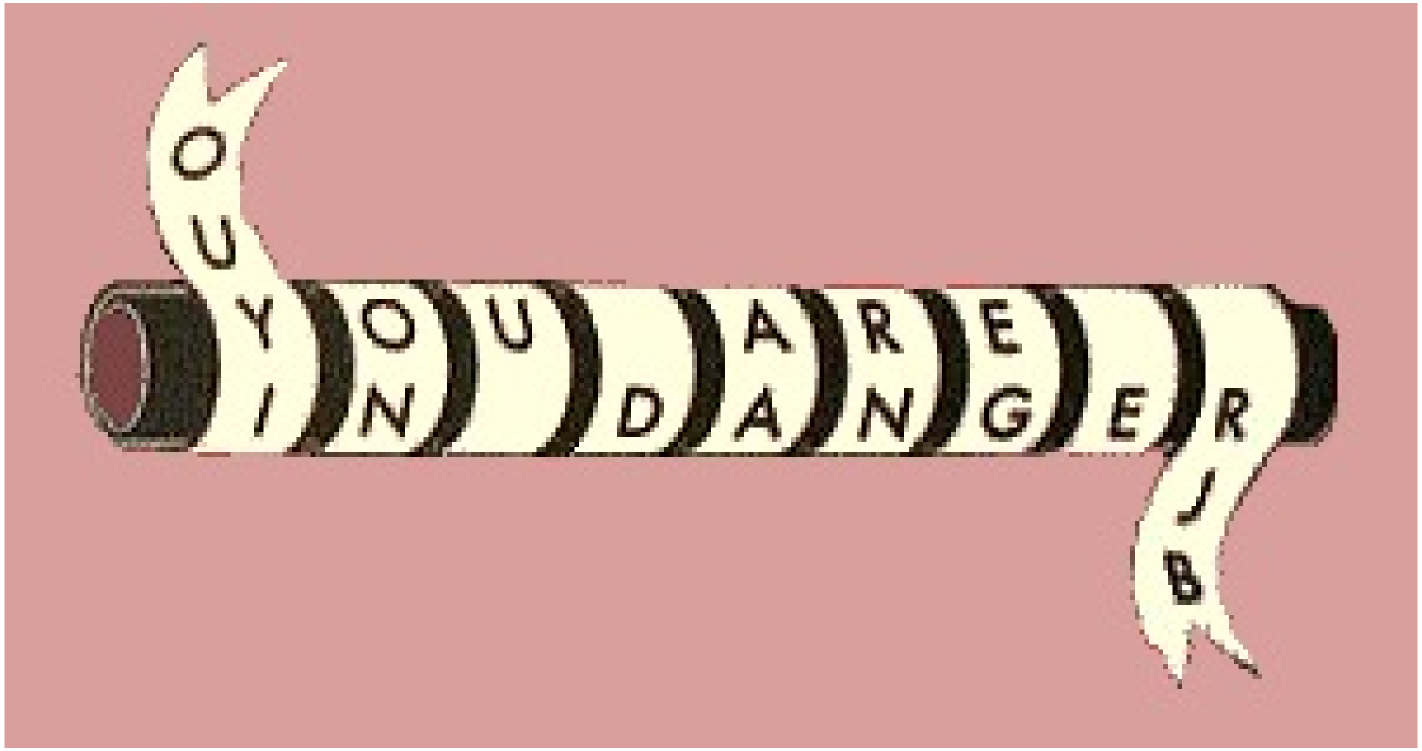
Cryptography

- In one sentence,

  "Cryptography is the practice and study of techniques for secure communication in the presence of third parties called *adversaries.*" *– Ronald L. Rivest*

Some Examples

- In 405 BC, the Greek general LYSANDER OF SPARTA was sent a coded message written on the inside of a servant's belt.



Some Examples

- The Greeks also invented a cipher which changed letters to numbers. A form of this code was still being used during *World War I*.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

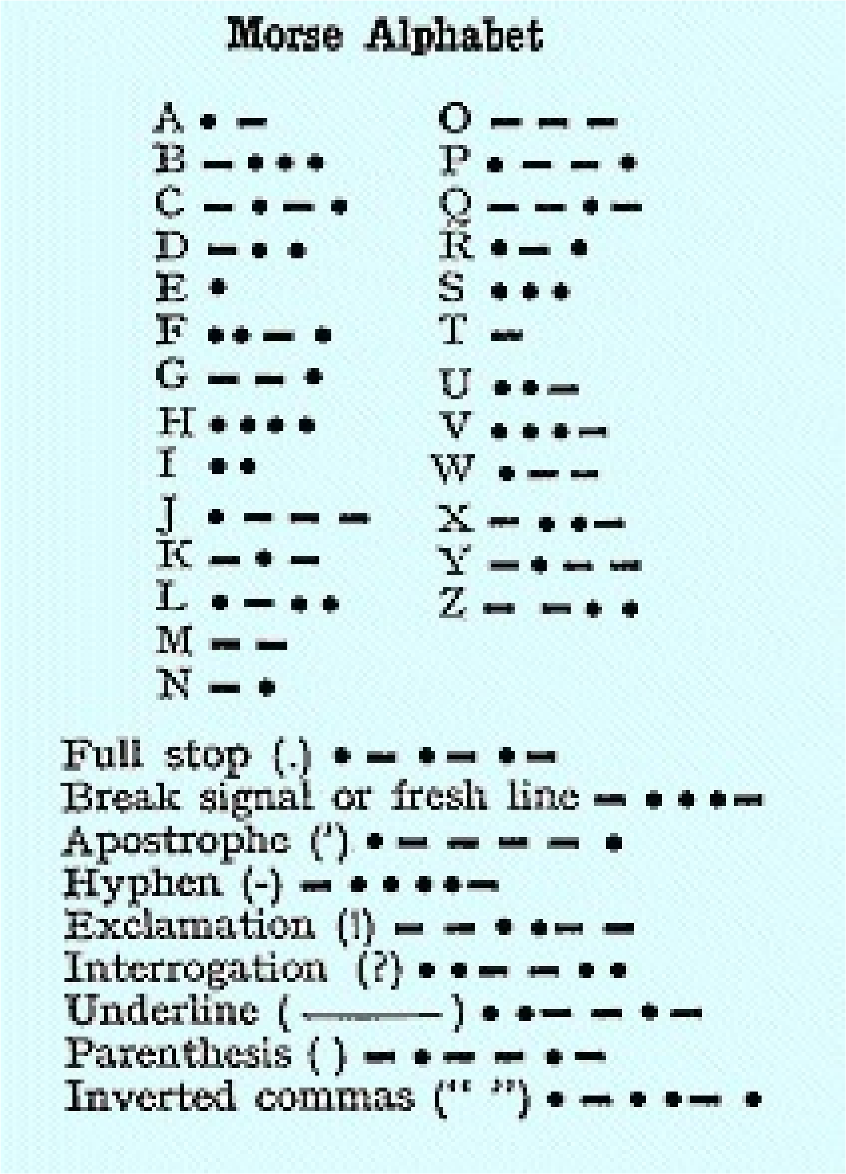
Some Examples

- Caesar Cipher (after the name of JULIUS CAESAR)

VENI, VIDI, VICI

YHQL YLGL YLFL

- Morse Code: created by Samuel Morse in 1838



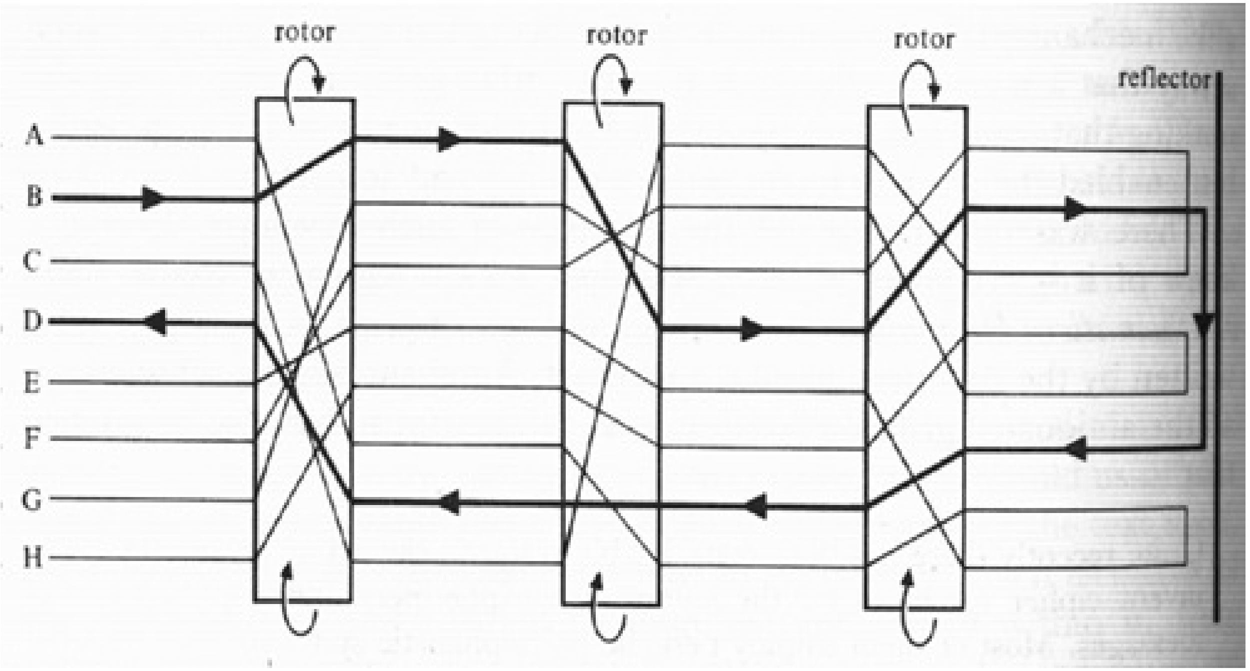
Some Examples

- Crytograms from the Chinese gold bars

http://www.iacr.org/misc/china/china.html

Some Examples

- Enigma, Germany coding machine in *World War II.*



Some Examples

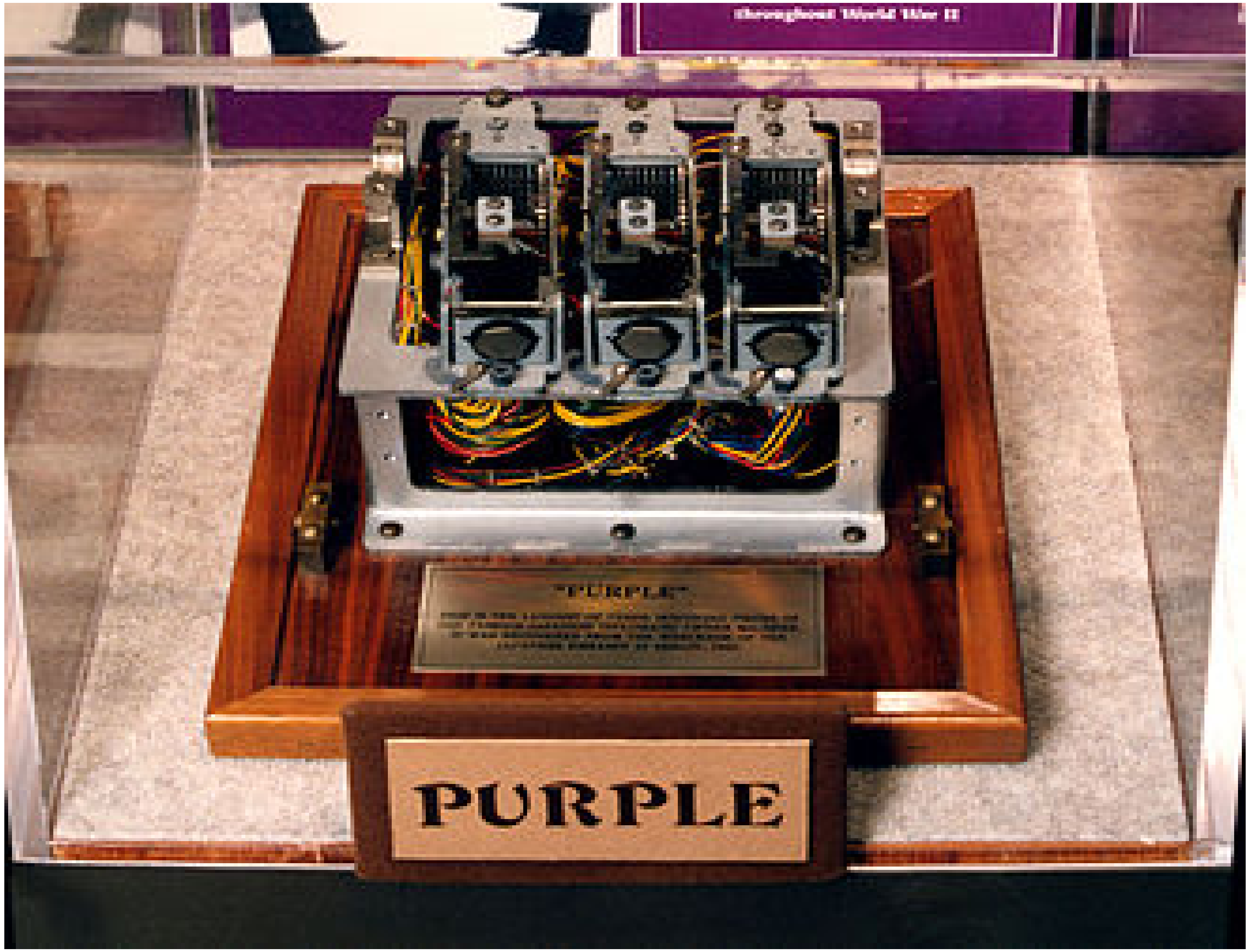
- Sigaba, used by U.S. during *World War II.*

Some Examples

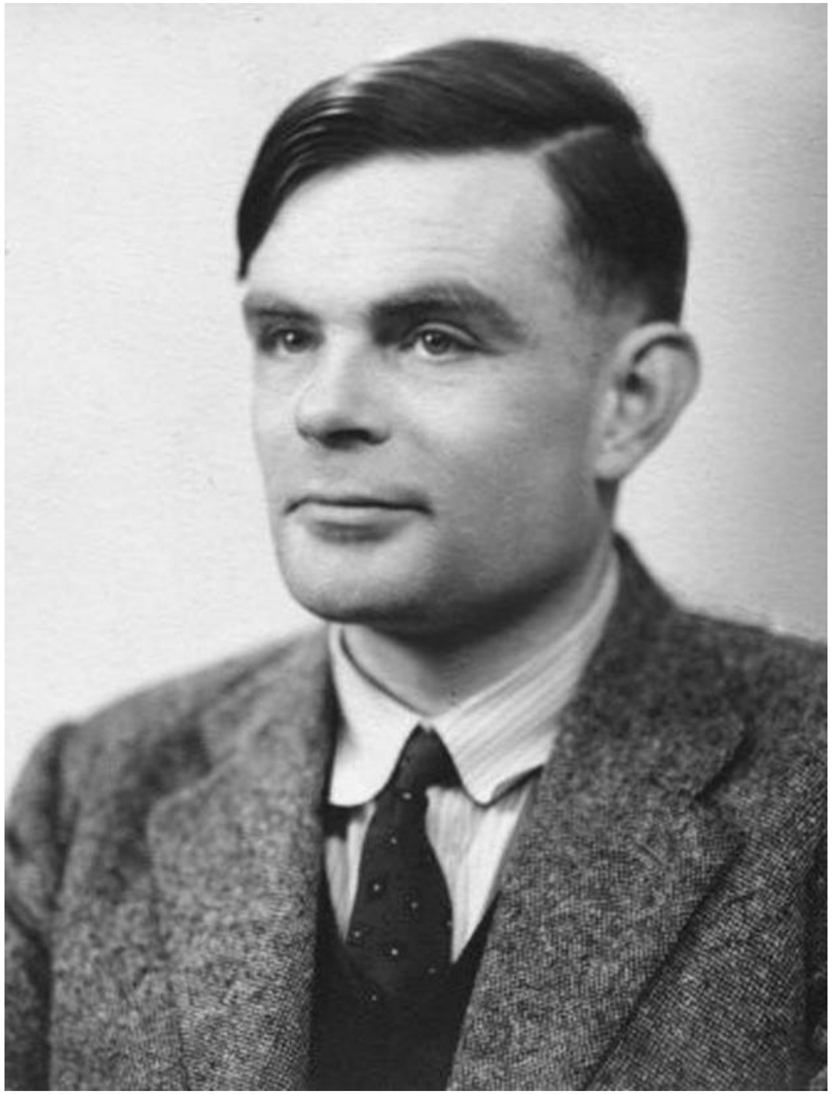
- Japanese “Enigma” Rotor Cipher Machine

Some Examples

- Japanese Purple Machine (97-shiki obun inji-ki)



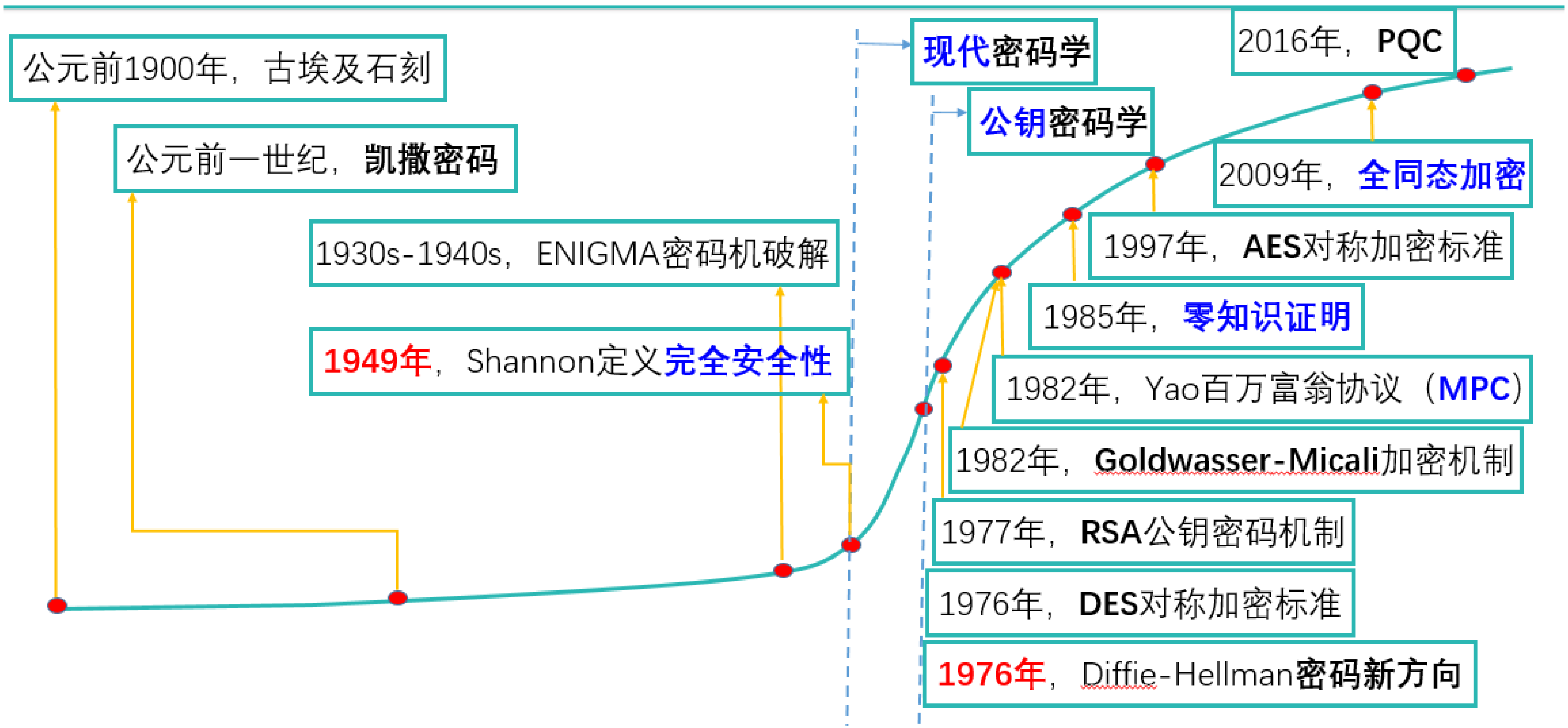
People Working in Breaking Codes



Alan Turing
(1912-1954)

Claude E. Shannon
(1916-2001)

Cryptography



Cryptography

- History (until 1970's)
  - ◇ Secret code invented
  - ◇ Typically claimed "unbreakable" by inventor
  - ◇ Used by spies, ambassadors, kings, generals for crucial tasks
  - ◇ Broken by adversaries using cryptanalysis

Cryptography

- History (until 1970's)
  - ◇ Secret code invented
  - ◇ Typically claimed "unbreakable" by inventor
  - ◇ Used by spies, ambassadors, kings, generals for crucial tasks
  - ◇ Broken by adversaries using cryptanalysis

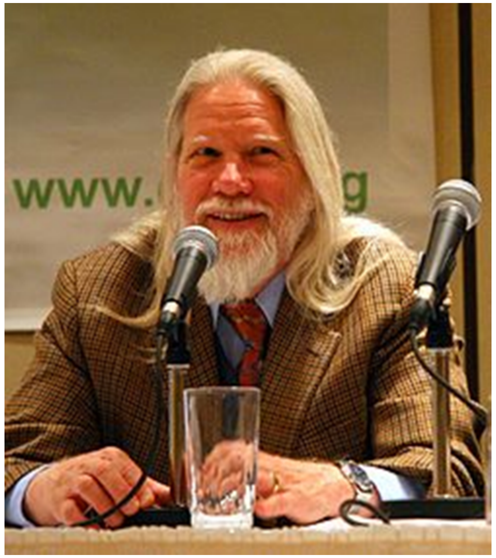
"*invent-break-tweak*" cycle

Cryptography

- History (from 1976)

  ◇ W. Diffie, M. Hellman, “New direction in cryptography”, *IEEE Transactions on Information Theory, vol. 22, pp. 644-654, 1976.*

  “We stand today on the brink of a revolution in cryptography.”

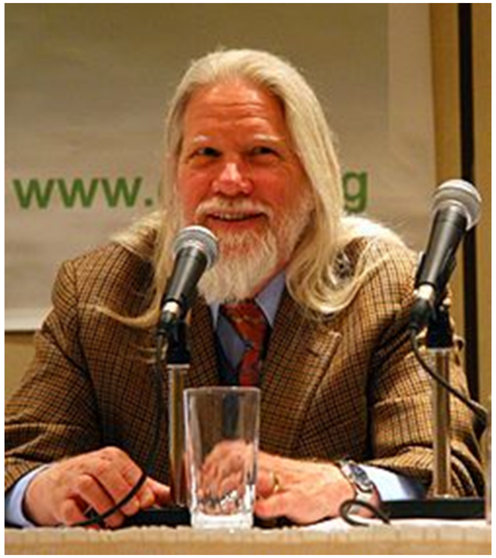


Cryptography

- History (from 1976)

  ◇ W. Diffie, M. Hellman, "New direction in cryptography", *IEEE Transactions on Information Theory, vol. 22, pp. 644-654, 1976.*

"We stand today on the brink of a revolution in cryptography."



Proposed new, more ambitious, notion of "*public-key cryptography*" based on simple to state, hard to solve, computational problem.

Modern Cryptography (post 1970's)

- Provable security
  - Perfect security (Shannon) and its limitations
  - Computational security
  - Pseudorandom generators, one-way functions

Modern Cryptography (post 1970's)

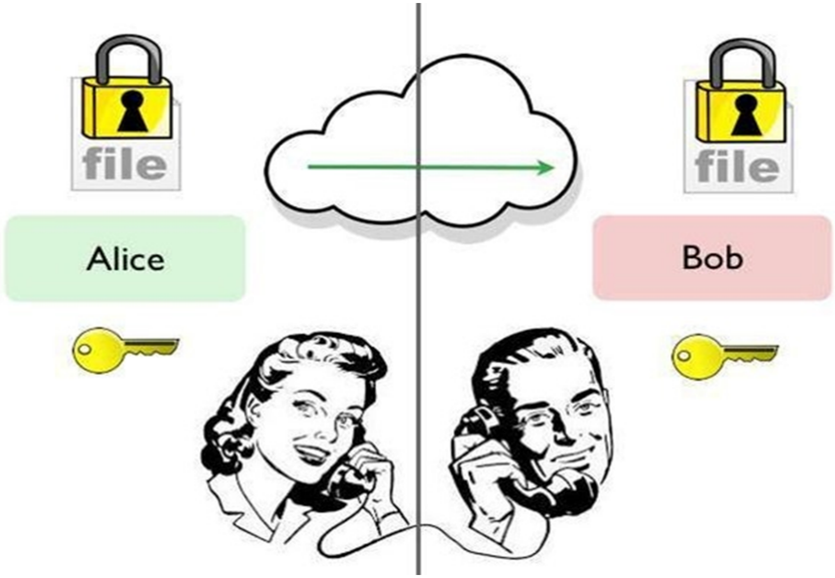
- Provable security
  - Perfect security (Shannon) and its limitations
  - Computational security
  - Pseudorandom generators, one-way functions
- Beyond encryption
  - Public-key encryption based on factoring, RSA problem
  - Digital signatures, hash functions
  - Zero-knowledge proofs

Modern Cryptography (post 1970's)

- Provable security
  - ◇ Perfect security (Shannon) and its limitations
  - ◇ Computational security
  - ◇ Pseudorandom generators, one-way functions
- Beyond encryption
  - ◇ Public-key encryption based on factoring, RSA problem
  - ◇ Digital signatures, hash functions
  - ◇ Zero-knowledge proofs
- Advanced topics*
  - ◇ The SSL protocol
  - ◇ Multi-party secure computation
  - ◇ Post-quantum cryptography
  - ◇ Fully homomorphic encryption (Gentry 2009), ...

Encryption Schemes

- Alice wants to send Bob a secret message



Encryption Schemes

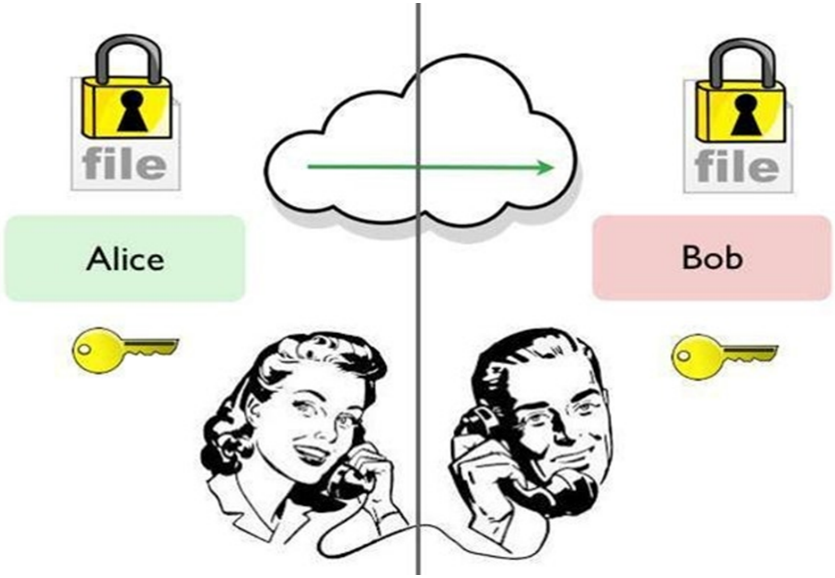
- Alice wants to send Bob a secret message

  They agree in advance on 3 components:

  ◇ Encryption algo.: E

  ◇ Decryption algo.: D

  ◇ Secret key: k

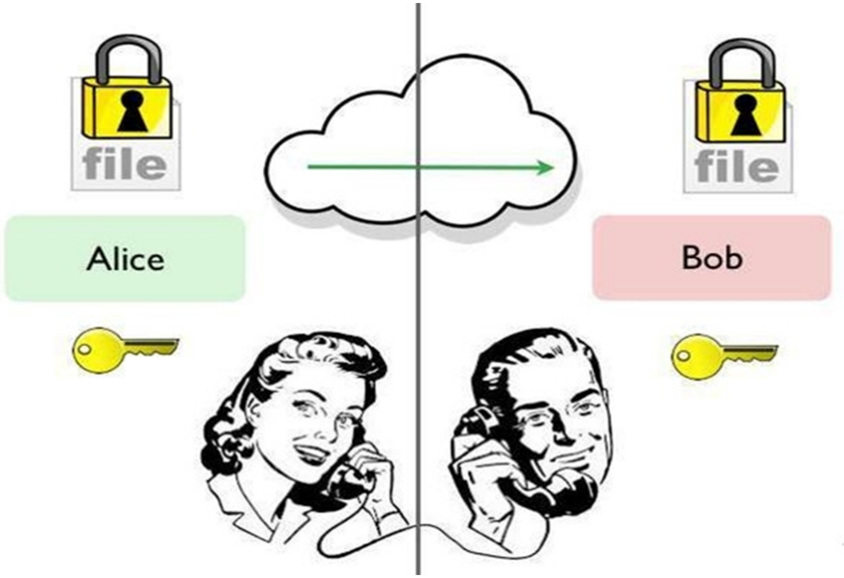


Encryption Schemes

- Alice wants to send Bob a secret message

They agree in advance on 3 components:

◇ Encryption algo.: E
◇ Decryption algo.: D
◇ Secret key: k



Encryption: $\boxed{\text{Alice}} \xrightarrow{c\,=\,\mathsf{E}(m,k)} \boxed{\text{Bob}}$

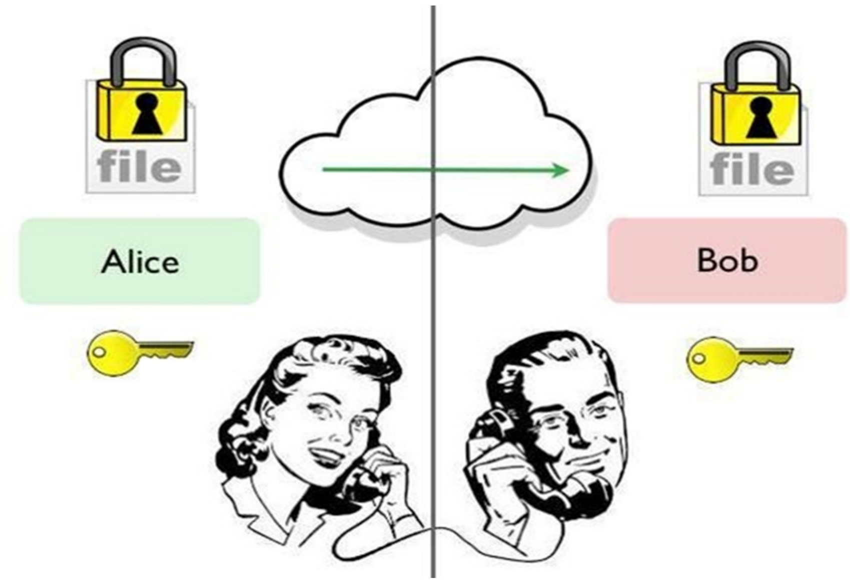
Decryption: $\boxed{\text{Bob}}$ computes $m' = \mathsf{D}(c, k)$

Encryption Schemes

- Alice wants to send Bob a secret message

They agree in advance on 3 components:

- ◇ Encryption algo.: E
- ◇ Decryption algo.: D
- ◇ Secret key: k



Encryption: $\boxed{\text{Alice}} \xrightarrow{c = \mathrm{E}(m, k)} \boxed{\text{Bob}}$

Decryption: $\boxed{\text{Bob}}$ computes $m' = \mathrm{D}(c, k)$

A scheme is valid if $m' = m$.
Intuitively, a scheme is secure if an eavesdropper cannot learn m from c.

Caesar Cipher

- Key: $k = 0, 1, \ldots, 25$

  Encryption: encode i as $(i + k) \bmod 26$

  Decryption: decode j as $(j - k) \bmod 26$

Caesar Cipher

- Key: $k = 0, 1, \ldots, 25$

  Encryption: encode i as $(i + k) \bmod 26$

  Decryption: decode j as $(j - k) \bmod 26$

  plaintext: SEND REINFORCEMENT
  Key: 2
  ciphertext: UGPF TGKPHQTEGOGPV

Caesar Cipher

- Key: $k = 0, 1, \ldots, 25$

  Encryption: encode i as $(i + k) \bmod 26$

  Decryption: decode j as $(j - k) \bmod 26$

  plaintext: SEND REINFORCEMENT
  Key: 2
  ciphertext: UGPF TGKPHQTEGOGPV

  Problem: only 26 possibilities for keys!

Caesar Cipher

- Key: $k = 0, 1, \ldots, 25$

  Encryption: encode i as $(i + k) \bmod 26$

  Decryption: decode j as $(j - k) \bmod 26$

  plaintext: SEND REINFORCEMENT
  Key: 2
  ciphertext: UGPF TGKPHQTEGOGPV

  Problem: only 26 possibilities for keys!

  Kerchoff's Principle (1883): System should be secure even if algorithms are known, as long as key is secret.

Substitution Cipher

- Key: table mapping each letter to another letter

A	B	C		Z
V	R	E		D

Substitution Cipher

- Key: table mapping each letter to another letter

A	B	C		Z
V	R	E		D

Encryption & Decryption: letter by letter according to table

Substitution Cipher

- Key: table mapping each letter to another letter

A	B	C		Z
V	R	E		D

Encryption & Decryption: letter by letter according to table

# of possible keys: $26! \approx 4 \times 10^{26}$

Substitution Cipher

- Key: table mapping each letter to another letter

A	B	C		Z
V	R	E		D

Encryption & Decryption: letter by letter according to table

# of possible keys: $26! \approx 4 \times 10^{26}$

However, substitution cipher is still insecure!

Substitution Cipher

- Key: table mapping each letter to another letter

A	B	C		Z
V	R	E		D

Encryption & Decryption: letter by letter according to table

# of possible keys: $26! \approx 4 \times 10^{26}$

However, substitution cipher is still insecure!

Key observation: can recover plaintext using *statistics* on *letter frequencies.*

Substitution Cipher

Table 1: Relative frequencies of the letters of the English language

Letter	Relative Frequency (%)	Letter	Relative Frequency (%)
a	8.167	n	6.749
b	1.492	o	7.507
c	2.782	p	1.929
d	4.253	q	0.095
e	12.702	r	5.987
f	2.228	s	6.327
g	2.015	t	9.056
h	6.094	u	2.758
i	6.966	v	0.978
j	0.153	w	2.360
k	0.772	x	0.150
l	4.025	y	1.974
m	2.406	z	0.074

Substitution Cipher

Table 2: Number of Diagraphs Expected in 2,000 Letters of English Text

th	-	50	at	-	25	st	-	20
er	-	40	en	-	25	io	-	18
on	-	39	es	-	25	le	-	18
an	-	38	of	-	25	is	-	17
re	-	36	or	-	25	ou	-	17
he	-	33	nt	-	24	ar	-	16
in	-	31	ea	-	22	as	-	16
ed	-	30	ti	-	22	de	-	16
ne	-	30	to	-	22	rt	-	16
ha	-	26	it	-	20	ve	-	16

Table 3: The 15 Most Common Trigraphs in the English Language

1	-	the	6	-	tio	11	-	edt
2	-	and	7	-	for	12	-	tis
3	-	tha	8	-	nde	13	-	oft
4	-	ent	9	-	has	14	-	sth
5	-	ion	10	-	nce	15	-	men

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

  I – *most common letter*
  LI – *most common pair*
  XLI – *most common triple*

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

I – *most common letter*	$\mathrm{I} = \mathrm{e}$
LI – *most common pair*	$\mathrm{L} = \mathrm{h}$
XLI – *most common triple*	$\mathrm{X} = \mathrm{t}$

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

I – *most common letter* — I = e

LI – *most common pair* — L = h

XLI – *most common triple* — X = t

LIVI = he?e

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

I – *most common letter*

LI – *most common pair*

XLI – *most common triple*

$\text{I} = \text{e}$

$\text{L} = \text{h}$

$\text{X} = \text{t}$

$\text{LIVI} = \text{he?e}$

$\text{V} = \text{r}$

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

I – *most common letter*
LI – *most common pair*
XLI – *most common triple*

LIVI = he?e

I = e
L = h
X = t

V = r
E = a
Y = g

Substitution Cipher

- LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFE MVEWHKVSTYLXZIXLIKIIXPIJVSZEYPERRGERIMWQL MGLMXQERIWGPSRIHMXQEREKI

I – *most common letter*

LI – *most common pair*

XLI – *most common triple*

LIVI = he?e

I = e

L = h

X = t

V = r

E = a

Y = g

HereUpOnLeGrandAroseWithAGraveAndStatelyAirAndBroug MeTheBeetleFromAGlassCaseInWhichItWasEnclosedIt-WasABe

Vigenere Cipher

- "*Multi-Caesar Cipher*" – stateful

Vigenere Cipher

- "*Multi-Caesar Cipher*" – stateful

  Key: $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ – list of m numbers in [0..25]

  Encryption: encode i as $(i + k_j)$ mod 26, if the location of i is j mod m

  Decryption: decode j as $(j - k_i)$ mod 26, if the location of j is i mod m

Vigenere Cipher

- "*Multi-Caesar Cipher*" – stateful

  Key: $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ – list of m numbers in [0..25]

  Encryption: encode i as $(i + k_j)$ mod 26, if the location of i is j mod m

  Decryption: decode j as $(j - k_i)$ mod 26, if the location of j is i mod m

  Important: Cannot break using letter frequencies alone. Because the same letter e may be mapped to $e + k_1, e + k_2, \ldots, e + k_m$ depending on different locations.

Vigenere Cipher

- "*Multi-Caesar Cipher*" – stateful

  Key: $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ – list of m numbers in [0..25]

  Encryption: encode i as $(i + k_j)$ mod 26, if the location of i is j mod m

  Decryption: decode j as $(j - k_i)$ mod 26, if the location of j is i mod m

  Important: Cannot break using letter frequencies alone. Because the same letter e may be mapped to $e + k_1, e + k_2, \ldots, e + k_m$ depending on different locations.

  Considered as "unbreakable" for 300 years (broken by Babbage, Kasiski 1850's)

Vigenere Cipher

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Vigenere Cipher

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Step 1: **Guess** the length of the key m

Vigenere Cipher

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Step 1: **Guess** the length of the key m

Step 2: Group together positions $\{1, m+1, 2m+1, \ldots\}$, $\{2, m+2, 2m+2, \ldots\}$, $\cdots$, $\{m-1, 2m-1, 3m-1, \ldots\}$

Vigenere Cipher

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Step 1: **Guess** the length of the key m

Step 2: Group together positions $\{1, m+1, 2m+1, \ldots\}$, $\{2, m+2, 2m+2, \ldots\}$, $\cdots$, $\{m-1, 2m-1, 3m-1, \ldots\}$

LIVITC
SWPIYV
EWHEVS
RIQMXL
EYVEOI
EWHRXE
XIPFEM
VEWHKV

Vigenere Cipher

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Step 1: **Guess** the length of the key m

Step 2: Group together positions $\{1, m+1, 2m+1, \ldots\}$, $\{2, m+2, 2m+2, \ldots\}$, $\cdots$, $\{m-1, 2m-1, 3m-1, \ldots\}$

Step 3: Frequency-analyze each group independently.

LIVITC
SWPIYV
EWHEVS
RIQMXL
EYVEOI
EWHRXE
XIPFEM
VEWHKV

- Breaking Vigenere:

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWHRXEXIPFEMVEWHK

Step 1: **Guess** the length of the key *m*

```
[97,  138,  60, 140,  65,  130,  130,  134,  148,  63,  124,  69,  145,  135,  127,  65,  132,  138,  144,  97]
[142, 138, 146, 125, 135,  142,  134,   59,  140, 146,  136, 125,  140,  144,    2, 127,   65,  145,  130, 146]
[146,  56, 138, 125,  57,  140,  134,  126,  133, 133,  145, 123,  134,  129,   57, 127,  149,   59,  141, 146]
[142, 126, 131, 143, 140,  131,  144,  137,  139, 133,  139,  70,   62,  104,  126,  58,  131,  144,  145,  64]
[131, 125,  62, 127, 126,  142,  149,  128,   61, 148,  132, 123,  134,  138,  136, 134,  144,  130,  134, 133]
[ 63, 122, 127, 143,   2,  127,   65,  142,  146, 146,   63, 132,  127,   60,  141, 130,   10,  138,  143, 137]
[132,  56, 131, 144,  57,  134,  134,  142,   61,   9,  147, 141,  130,  133,  122, 136,  149,  142,   73,  64]
[140, 121, 135, 143,  57,  123,  150,  142,  144, 137,   63, 132,  127,   60,  133, 123,  143,  130,  146, 133]
[ 75,  56, 142, 139, 142,  140,   65,    5,  145, 146,  132,  56,  131,  138,   57, 135,  134,  142,  146, 146]
[132,  56, 130,  67, 142,  142,  138,  135,  134, 147,  132, 138,   62,  128,  126, 141,   65,  136,    6, 148]
[135, 135, 130, 129, 140,   58,  134,  143,   61, 132,  132, 139,   62,  139,  142, 142,  138,  135,  144,  64]
[142, 134, 146,  60,   2,  142,   10,   59,  129,   9,  149, 125,  138,  139,  137, 138,   10,  142,   75,  64]
[107, 125, 145,  60,   2,  134,    9,  145,  130, 147,   63, 138,  131,    3,  136, 131,  151,  128,  139, 148]
[ 63, 141, 140,  60, 135,  131,  151,  128,  126, 149,   63,   1,  138,  129,  143,   3,   65,  127,  130,  64]
[139, 121,  62, 140, 136,  1 43,  147,  142,  146, 137,  147, 125,   62,  143,  124, 131,  134,  137,  145, 137]
```

70 40 10

EWHRXE
XIPFEM
VEWHKV

Review of Encryption Schemes

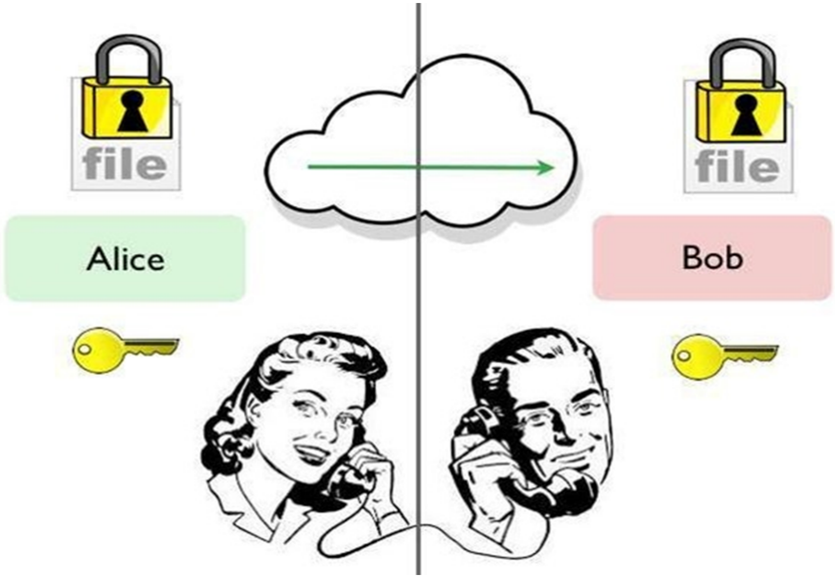
- Alice wants to send Bob a secret message

  They agree in advance on 3 components:

  ◇ Encryption algo.: E

  ◇ Decryption algo.: D

  ◇ Secret key: k



Review of Encryption Schemes

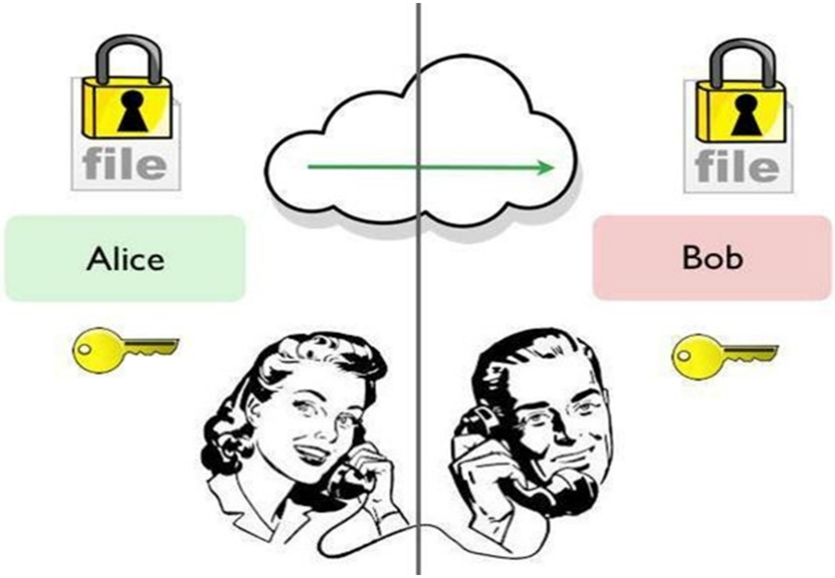
- Alice wants to send Bob a secret message

They agree in advance on 3 components:

◇ Encryption algo.: E

◇ Decryption algo.: D

◇ Secret key: k



$\mathcal{Q}$: Can Bob send Alice the secret key over the channel?

Review of Encryption Schemes

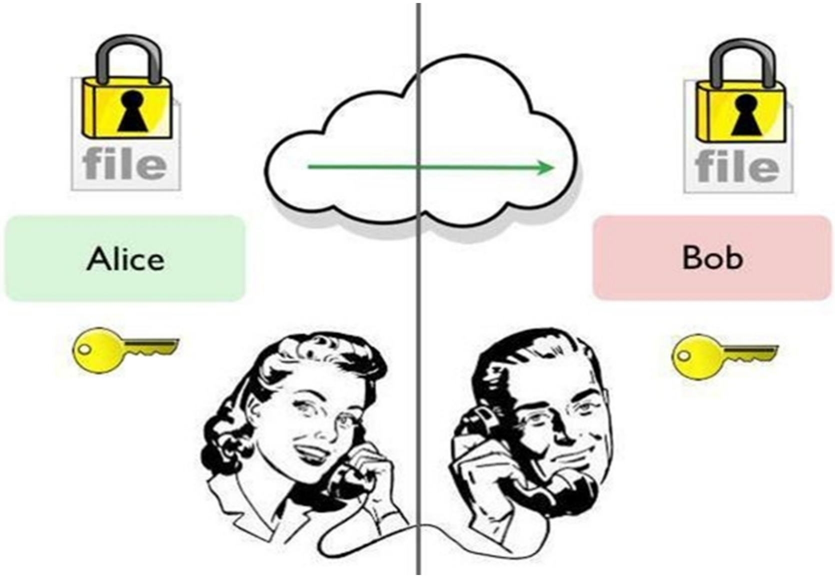
- Alice wants to send Bob a secret message

They agree in advance on 3 components:

◇ Encryption algo.: E

◇ Decryption algo.: D

◇ Secret key: k



$\mathcal{Q}$: Can Bob send Alice the secret key over the channel?

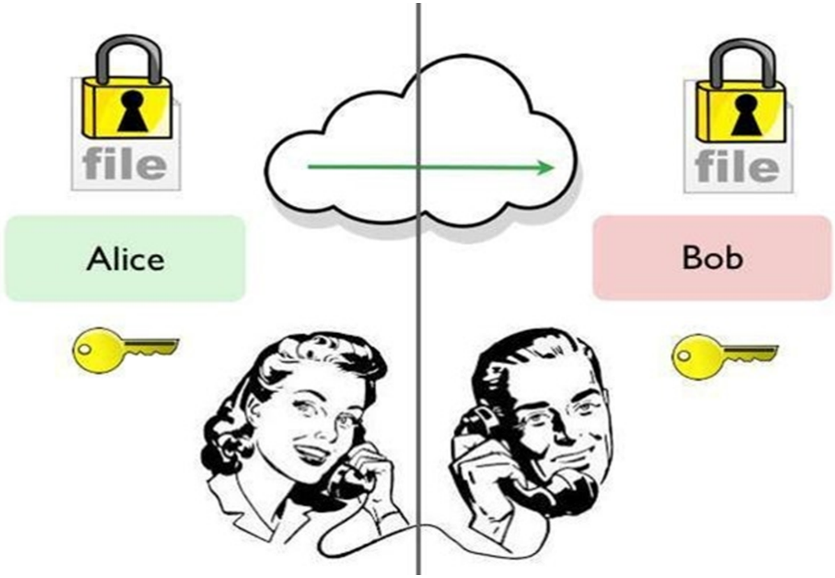
$\mathcal{A}$: Of course not! Eve could decrypt c.

Review of Encryption Schemes

- Alice wants to send Bob a secret message

They agree in advance on 3 components:

◇ Encryption algo.: E
◇ Decryption algo.: D
◇ Secret key: k



$\mathcal{Q}$: Can Bob send Alice the secret key over the channel?

$\mathcal{A}$: Of course not! Eve could decrypt c.

$\mathcal{Q}$: What if Bob could send Alice a "special key" useful only for encryption but no help for decryption?

Pulick Key Cryptography [DH76, RSA77]

- Alice wants to send Bob a secret message

  - ◇ Encryption algo.: E
  - ◇ Decryption algo.: D

Pulick Key Cryptography [DH76, RSA77]

- Alice wants to send Bob a secret message

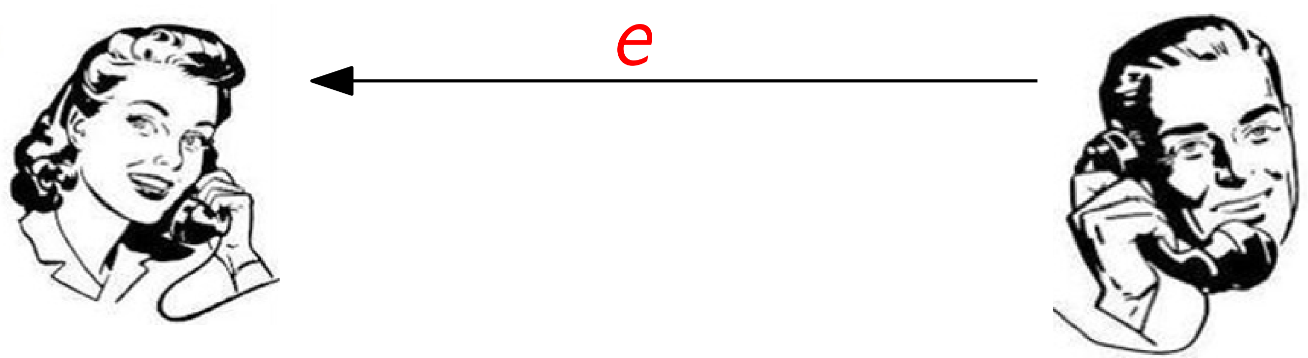
◇ Encryption algo.: E

◇ Decryption algo.: D

◇ Key: Bob chooses two keys: secret key d for decryption and public key e for encryption.

Pulick Key Cryptography [DH76, RSA77]

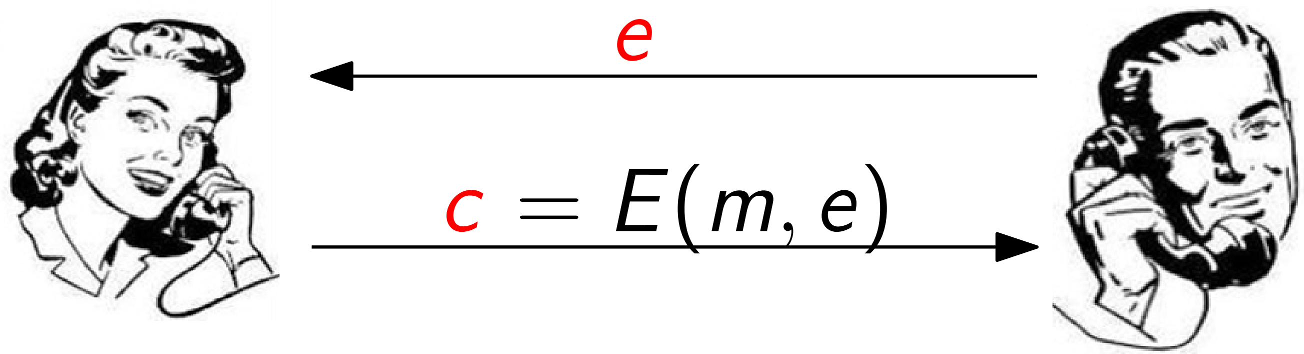
- Alice wants to send Bob a secret message



◇ Encryption algo.: E
◇ Decryption algo.: D
◇ Key: Bob chooses two keys: secret key d for decryption and public key e for encryption.

Pulick Key Cryptography [DH76, RSA77]

- Alice wants to send Bob a secret message



◇ Encryption algo.: E

◇ Decryption algo.: D

◇ Key: Bob chooses two keys: secret key d for decryption and public key e for encryption.

Pulick Key Cryptography [DH76, RSA77]

- Alice wants to send Bob a secret message

$$\xleftarrow{\quad e \quad}$$

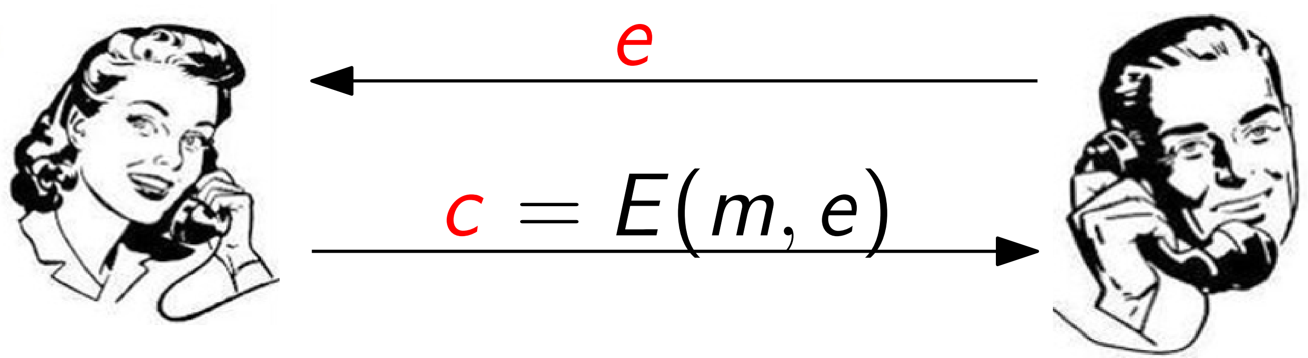
$$\xrightarrow{\quad c = E(m, e) \quad}$$

◇ Encryption algo.: E
◇ Decryption algo.: D
◇ Key: Bob chooses two keys: secret key d for decryption and public key e for encryption.

Decryption: $\boxed{\text{Bob}}$ computes $m' = \mathrm{D}(c, d)$

Pulick Key Cryptography [DH76, RSA77]

- Alice wants to send Bob a secret message



◇ Encryption algo.: E
◇ Decryption algo.: D
◇ Key: Bob chooses two keys: secret key d for decryption and public key e for encryption.

Decryption: $\boxed{\text{Bob}}$ computes $m' = \mathrm{D}(c, d)$

A scheme is valid if $m' = m$.
Intuitively, a scheme is secure if eavesdropper cannot learn m from c (even if Eve knows the key e).

Cryptography Wonders

- *Digital Signatures*. Electronically sign documents

  Zero-knowledge Proofs. Alice proves to Bob that she earns $< \$50k$ without Bob learning her income.

  Privacy-perserving data mining. Bob holds DB. Alice gets answer to one query, without Bob knowing what she asked.

  Playing poker over the net. Alice, Bob, Carol and David can play Poker over the net without trusting each other or any central server. (*E-Voting*)

  Electronic Auctions. Can run auctions s.t. no one (even not seller) learns anything other than winning party and bid.

  Fully Homomorphic Encryption. Encrypt $E(m)$ in a way that allows to compute $E(f(m))$.

Cryptography Wonders

WA

what is the most important research area in cryptography in the next 10 years?

Cryptography is a rapidly evolving field, and it is difficult to predict the most important research area in cryptography for the next 10 years with certainty. However, here are some areas that are likely to be of significant interest in the coming years:

1. Post-quantum cryptography: With the rapid progress in quantum computing, it is becoming increasingly likely that currently used public key algorithms such as RSA and Elliptic Curve Cryptography (ECC) will become vulnerable to quantum computer attacks. Hence, researchers are working on developing new cryptographic algorithms that are secure against quantum computers.
2. Secure multi-party computation: This area of research aims to develop methods for performing computations on sensitive data, while keeping the data secure and private. With the increasing amount of sensitive data being generated and stored, this research area will become increasingly important.
3. Privacy-preserving technologies: This area of research involves developing technologies that protect the privacy of users while still allowing the data to be used for useful purposes such as analytics and machine learning. This includes privacy-preserving machine learning, differential privacy, and homomorphic encryption.
4. Blockchain security: Blockchain technology has gained significant attention in recent years due to its potential to revolutionize various industries. However, ensuring the security of blockchain systems remains a major challenge, and researchers are actively working on developing secure blockchain solutions.

These are just a few examples of the many exciting research areas in cryptography. The field is rapidly evolving, and new challenges and opportunities are likely to emerge in the next 10 years.

Applications of Cryptography

- Using the crypto mindset beyond crypto

One example: preventing overfitting in machine learning

RESEARCH | *REPORTS*

STATISTICS

The reusable holdout: Preserving validity in adaptive data analysis

Cynthia Dwork,[1*] Vitaly Feldman,[2*] Moritz Hardt,[3*] Toniann Pitassi,[4*] Omer Reingold,[5*] Aaron Roth[6*]

Misapplication of statistical data analysis is a common cause of spurious discoveries in scientific research. Existing approaches to ensuring the validity of inferences drawn from data assume a fixed procedure to be performed, selected before the data are examined. In common practice, however, data analysis is an intrinsically adaptive process, with new analyses generated on the basis of data exploration, as well as the results of previous analyses on the same data. We demonstrate a new approach for addressing the challenges of adaptivity based on insights from privacy-preserving data analysis. As an application, we show how to safely reuse a holdout data set many times to validate the results of adaptively chosen analyses.

Idea: Treat training algorithm as attacker to prevent it learning "too much" about the particular sample.

Principles of Mordern Cryptography

- Principle 1 – *Formal Definitions*
- Principle 2 – *Precise Assumptions*
- Principle 3 – *Proofs of Security*

Next Lecture

- perfect secrecy ...