Introduction

Sheng Zhong Yuan Zhang

Computer Science and Technology Department
Nanjing University

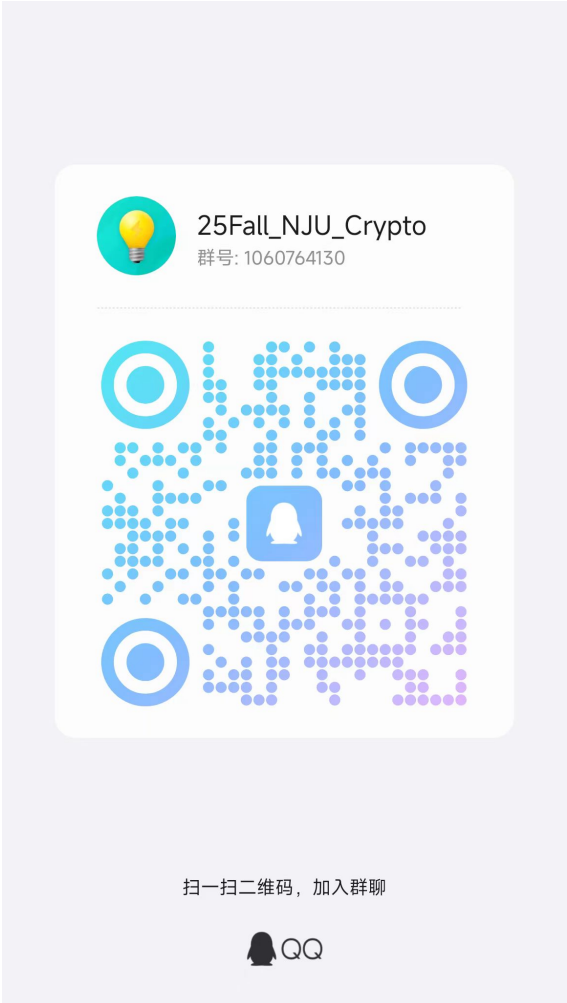
Outline

Course Info

- Lecture time:
  Thursday, 9:00-12:00.
- Classroom:
  仙 II-207.
- Instructor:
  Yuan Zhang (张渊), zhangyuan@nju.edu.cn, Rm. 505 AT CS Bldg.
  `https://zynju.github.io/zhangyuan`
- TA:
  Wenhao Wang (王文昊), wenhao.wang@smail.nju.edu.cn
  Anyi Cao (曹安毅), 502023330002@smail.nju.edu.cn
  Moyang Xie (谢模阳), 211502016@smail.nju.edu.cn
  TBA

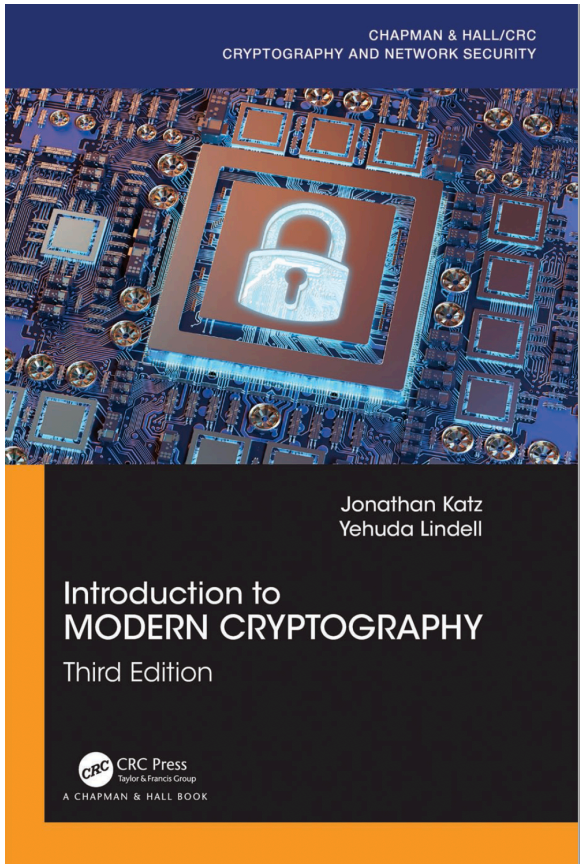
Course Info

- Course QQ Group (announcements, slides, homework, etc. here!): **1060764130**



Course Info

- Textbook:
  Introduction to Modern Cryptography (3rd edition) by J. Katz and Y. Lindell;



Course Info

- Course prerequisites:
  Being familiar with the following courses makes this course easier.
  - Probability Theory
  - Discrete Maths
  - Algorithm
- Gradings:
  Homework/Quiz (5 to 6 times) 40% + Midterm Exam 20% + Final Exam 40%
- Course Policy:
  Zero tolerance to plagiarism!!! Always give credits to other people if you use their results or works.

Outline

Have you met Crypto TODAY?

Q: Have you met Crypto TODAY ?

Have you met Crypto TODAY?

Q: Have you met Crypto TODAY ?

A: I believe you have.

Cryptography is around us

- From the moment you logged into your cellphone:

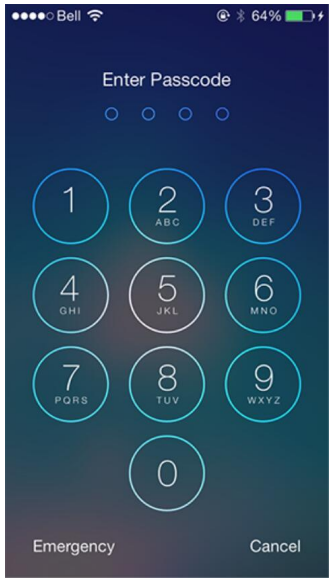


图: iphone login screenshot

Cryptography is around us

- From the moment you logged into your cellphone:

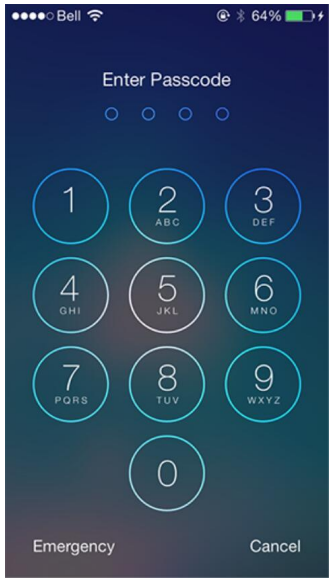


图: iphone login screenshot

- *"Every iOS device has a dedicate AES 256 crypto engine built ... Every time a file on the data partition is created, ... the hardware AES engine, which uses the key to encrypt the file"* —— iOS Security [5]

Cryptography is around us

- To the moment your cellphone logged itself to the mobile network:

图: SIM cards

Cryptography is around us

- To the moment your cellphone logged itself to the mobile network:

图: SIM cards

- 4G Sim Card is a mini-computer that stores a master key and runs AES-based MILENAGE algorithm and AKA protocol to perform mutual authentication and key agreement with the cellular tower.

Cryptography is around us

- To the moment you made an online purchase.

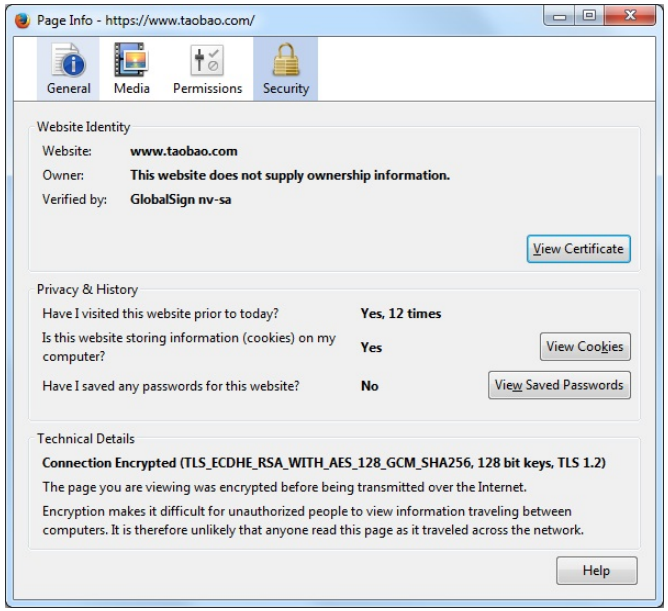


图: Page Info of www.taobao.com

Cryptography is around us

- To the moment you made an online purchase.

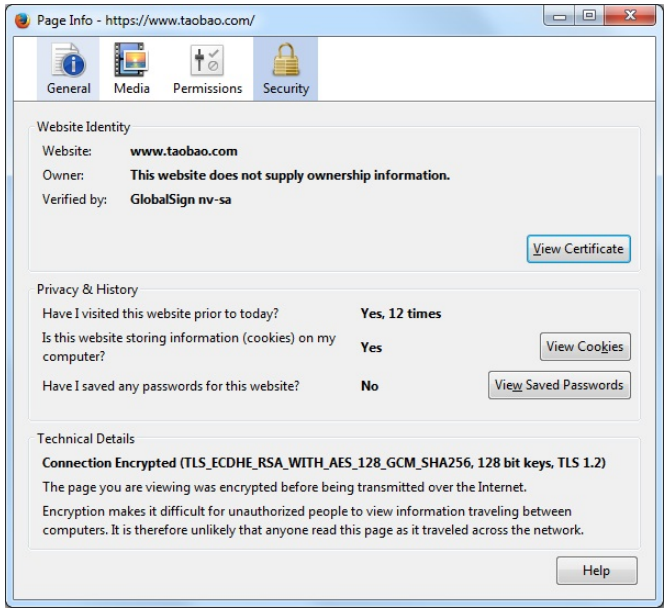


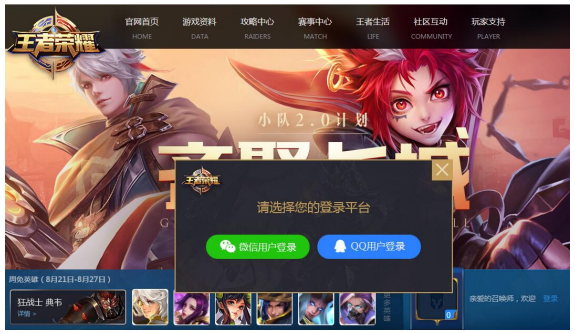
图: Page Info of www.taobao.com

- *"Website Identity Verified by GlobalSign nv-sa; Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)"*

Cryptography is around us

- To the moments you used these:

(a) NJU Campus Card



(b) Online Gaming Login

(c) Banking

(d) ID card

图: More cryptography usages

Outline

Understanding Crypto can help us to find a job

- On May 5th, 2014, US National Security Agency (NSA) sent a mysterious, garbled tweet:

图: A mysterious tweet from NSA

Understanding Crypto can help us to find a job

- On May 5th, 2014, US National Security Agency (NSA) sent a mysterious, garbled tweet:

图: A mysterious tweet from NSA

- It is a `substitution cipher`. After decryption, it says "*Want to know what it takes to work at NSA? Check back each Monday in May as we explore careers essential to protecting our nation.*"

Understanding Crypto can help us to create money

- Crypto can be used to "make" money:

图: Bitcoin' price till 08/28/2021. Pics from coinmarketcap.com

Understanding Crypto can create money? or not?

- 不要炒币!

图: Bitcoin' price till 09/04/2024. Pics from Google Finance

Understanding Crypto can help us to protect ourselves

- We often see a lot of warning messages like:

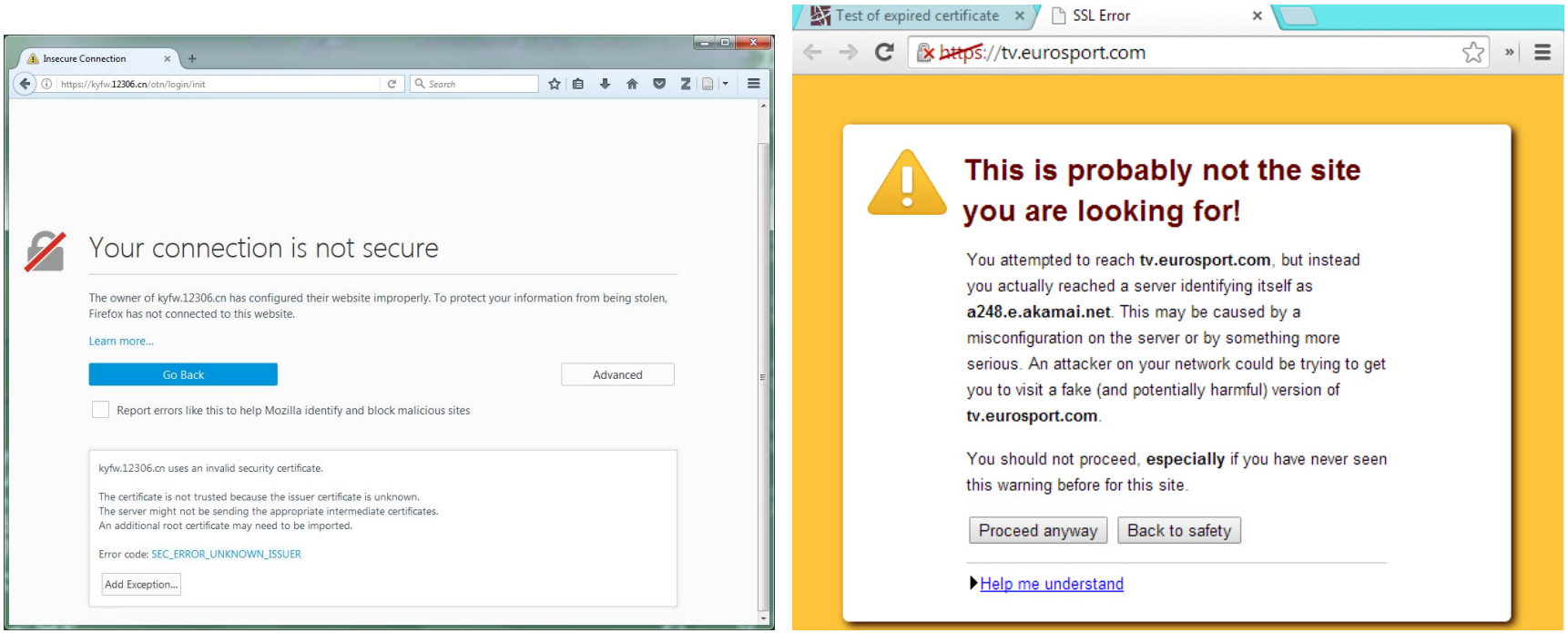


图: Warning messages

Understanding Crypto can help us to protect ourselves

- We often see a lot of warning messages like:

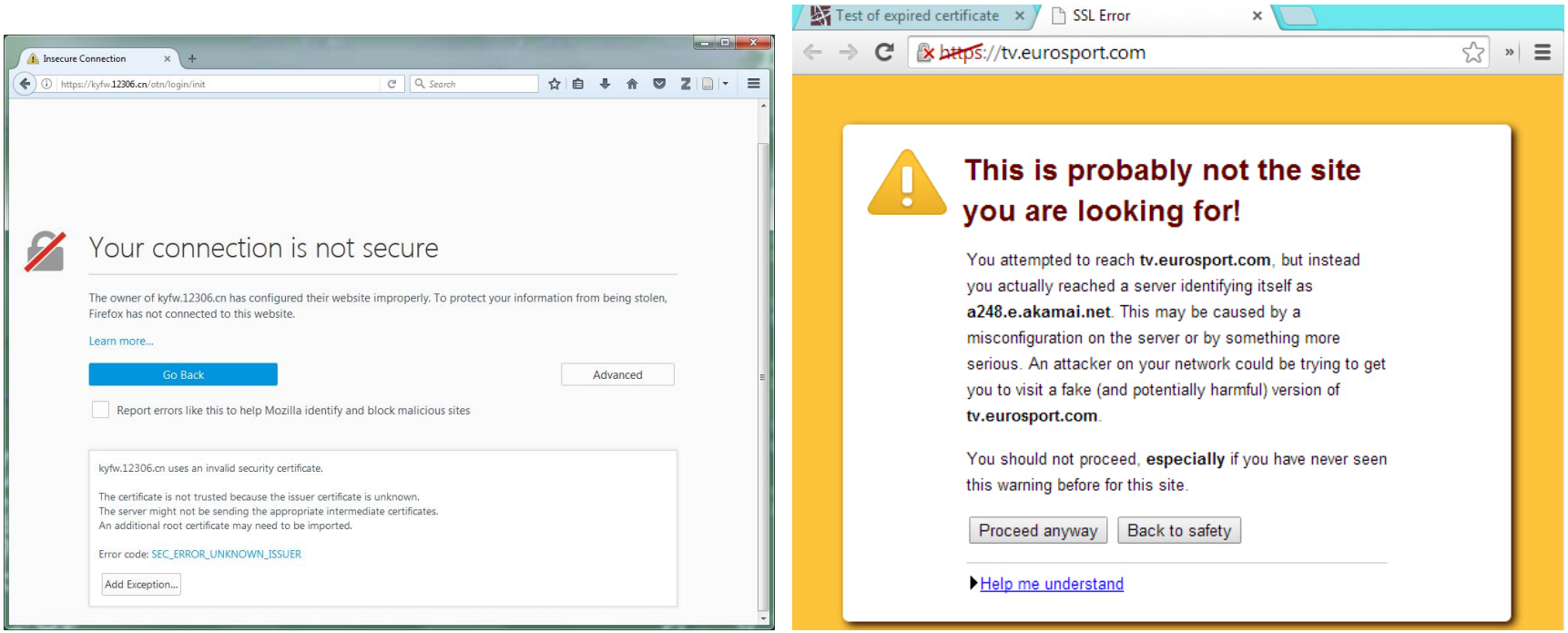
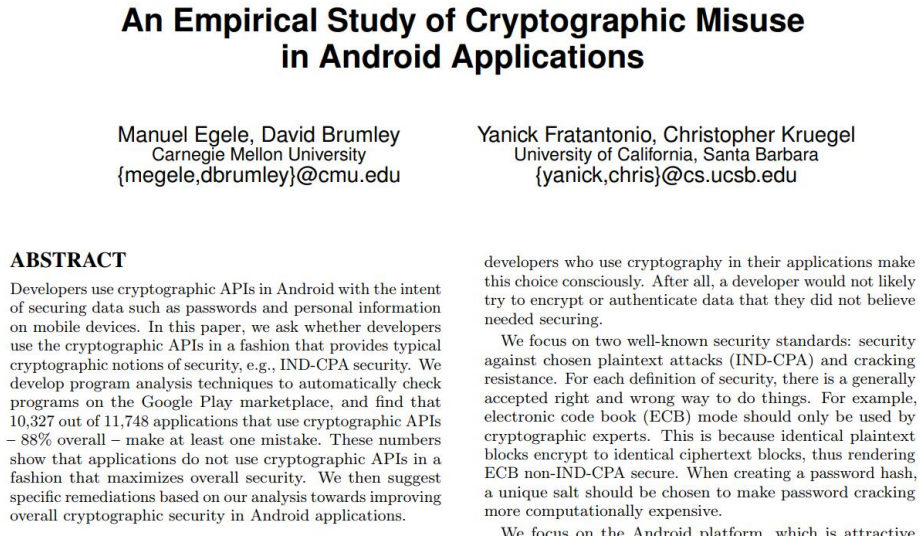


图: Warning messages

- Should we “Proceed anyway” or “Back to safety”?

Understanding Crypto can help us to build a safer world

- According to a CCS research paper, "*11,748 android apps use cryptography (encryption), and 10,327 get it wrong [Egele13]*"



An Empirical Study of Cryptographic Misuse in Android Applications

Manuel Egele, David Brumley
Carnegie Mellon University
{megele,dbrumley}@cmu.edu

Yanick Fratantonio, Christopher Kruegel
University of California, Santa Barbara
{yanick,chris}@cs.ucsb.edu

ABSTRACT

Developers use cryptographic APIs in Android with the intent of securing data such as passwords and personal information on mobile devices. In this paper, we ask whether developers use the cryptographic APIs in a fashion that provides typical cryptographic notions of security, e.g., IND-CPA security. We develop program analysis techniques to automatically check programs on the Google Play marketplace, and find that 10,327 out of 11,748 applications that use cryptographic APIs – 88% overall – make at least one mistake. These numbers show that applications do not use cryptographic APIs in a fashion that maximizes overall security. We then suggest specific remediations based on our analysis towards improving overall cryptographic security in Android applications.

developers who use cryptography in their applications make this choice consciously. After all, a developer would not likely try to encrypt or authenticate data that they did not believe needed securing.

We focus on two well-known security standards: security against chosen plaintext attacks (IND-CPA) and cracking resistance. For each definition of security, there is a generally accepted right and wrong way to do things. For example, electronic code book (ECB) mode should only be used by cryptographic experts. This is because identical plaintext blocks encrypt to identical ciphertext blocks, thus rendering ECB non-IND-CPA secure. When creating a password hash, a unique salt should be chosen to make password cracking more computationally expensive.

We focus on the Android platform, which is attractive

图: An empirical study of cryptographic misuse in android applications

- Without understanding Crypto, more and more unsafe Apps will come.

A recent example of wrong crypto

- A blogger posted a blog about hacking her/his own Hyundai car:

How I Hacked my Car

2022-05-22 :: greenluigi1

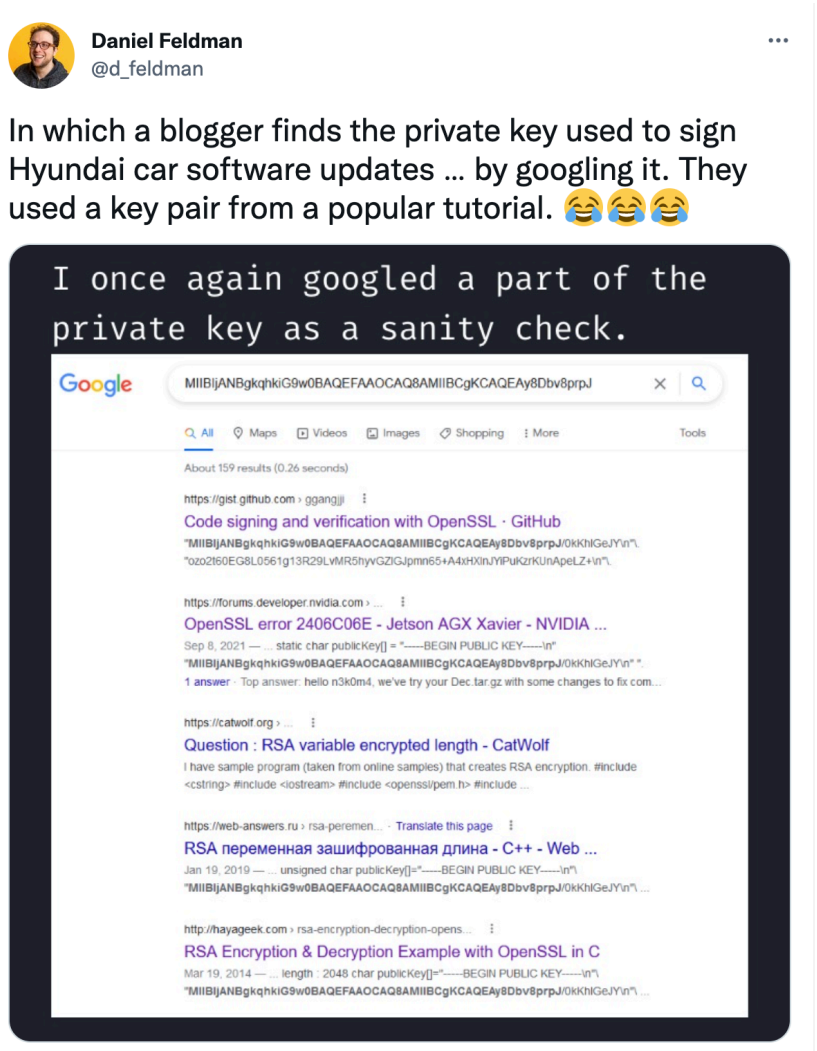
#d-audio #d-audio2 #hyundai #kia #hacking #car #IVI #howIHackedMyCar

The Car

Last summer I bought a 2021 Hyundai Ioniq SEL. It is a nice fuel-efficient hybrid with a decent amount of features like wireless Android Auto/Apple CarPlay, wireless phone charging, heated seats, & a sunroof.

One thing I particularly liked about this vehicle was the In-Vehicle Infotainment (IVI) system. As I mentioned before it had wireless Android Auto which seemed to be uncommon in this price range, and it had pretty nice, smooth animations in its menus which told me the CPU/GPU in it wasn't completely underpowered, or at least the software it was running wasn't super bloated.

As with many new gadgets I get, I wanted to play around with it and ultimately see what I could do with it.



Outline

Two major usages of Cryptography

- We often use Crypto to implement a secure communication, i.e. to achieve
  - **Message Secrecy (or Confidentiality)**
  - **Message Integrity (or Authenticity)**

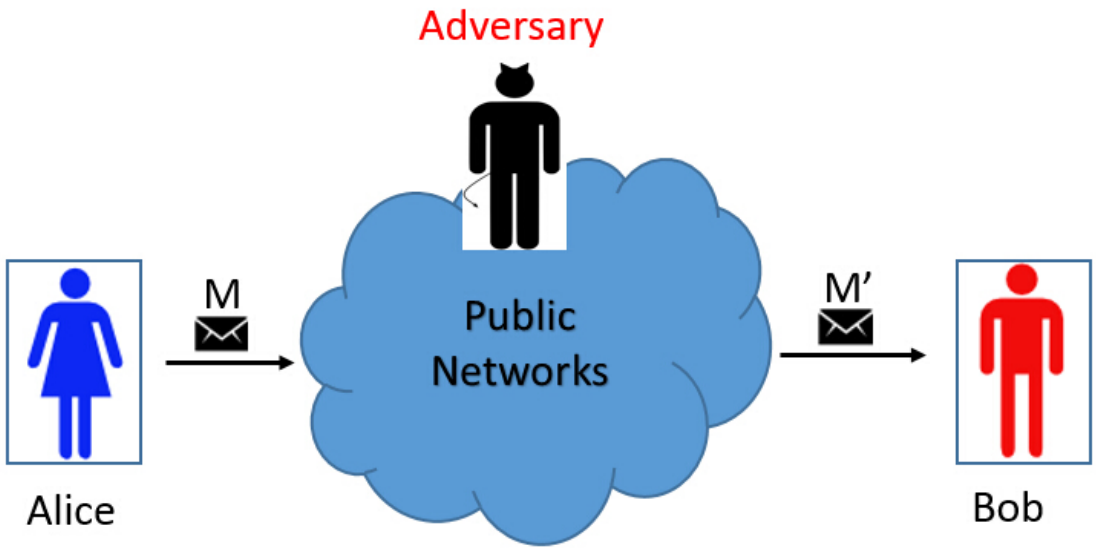


图: Communication via public networks

Protecting secrecy

- **Secrecy** generally requires that no others (e.g. the Adversary) know the message M's content except its intended receiver Bob.

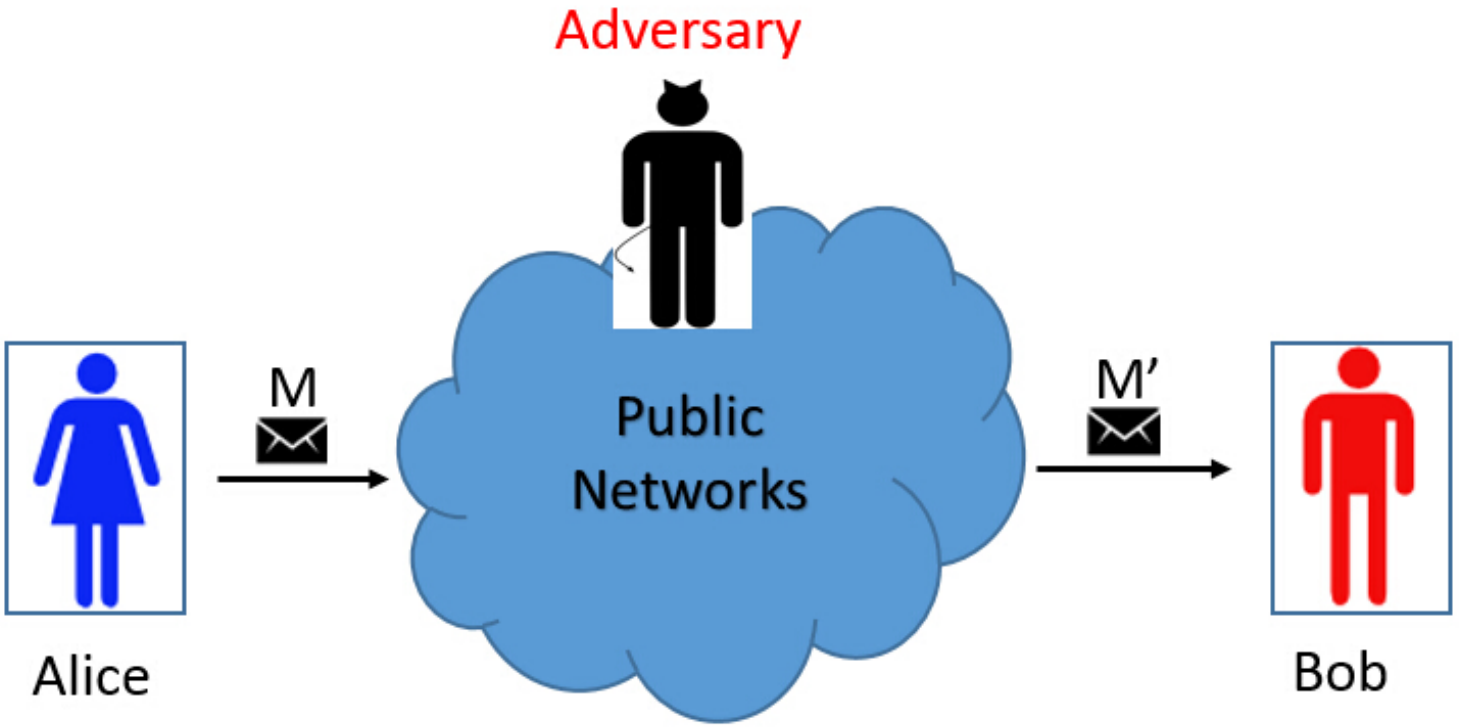


图: Secure communication via public networks

Protecting integrity

- **Integrity** requires the message M' that Bob receives is not tempered, i.e. M' = M.
- **Integrity** also requires the sender info is correct (e.g. message M' is indeed sent by Alice).[1]

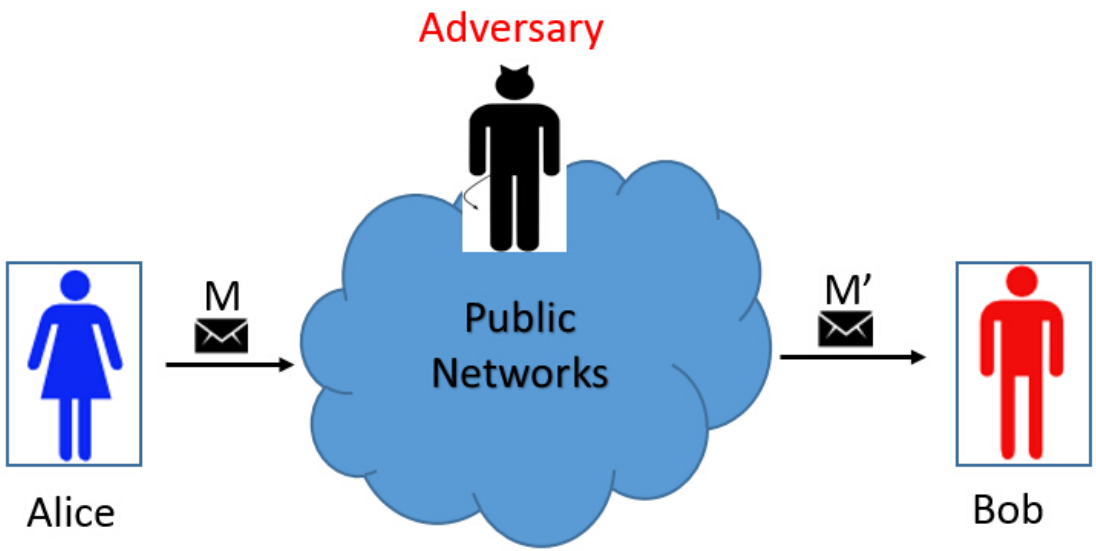
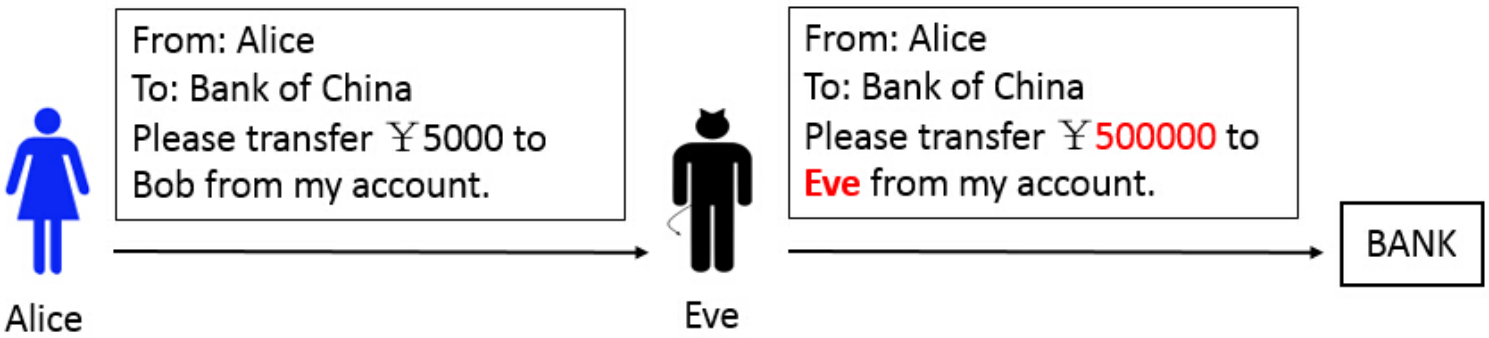


图: Secure communication via public networks

[1] When the sender info is included in the message, the second requirement is covered by the first one.

A simple example showing why integrity is important

A simple attack on the data integrity.



Outline

There's more

- More advanced usages/functionalities are provided as cryptographic protocols, e.g.
  - `Oblivious Transfer` ("不经意传输")
  - `Zero-knowledge Proof` ("零知识证明")
  - `Secure Multiparty Computation` ("多方安全计算")
  - `Digital Currency` ("数字货币")
  - ... ...

Outline

An overview of this course

- Cryptography is actually a growing area that covers a wide range of topics.

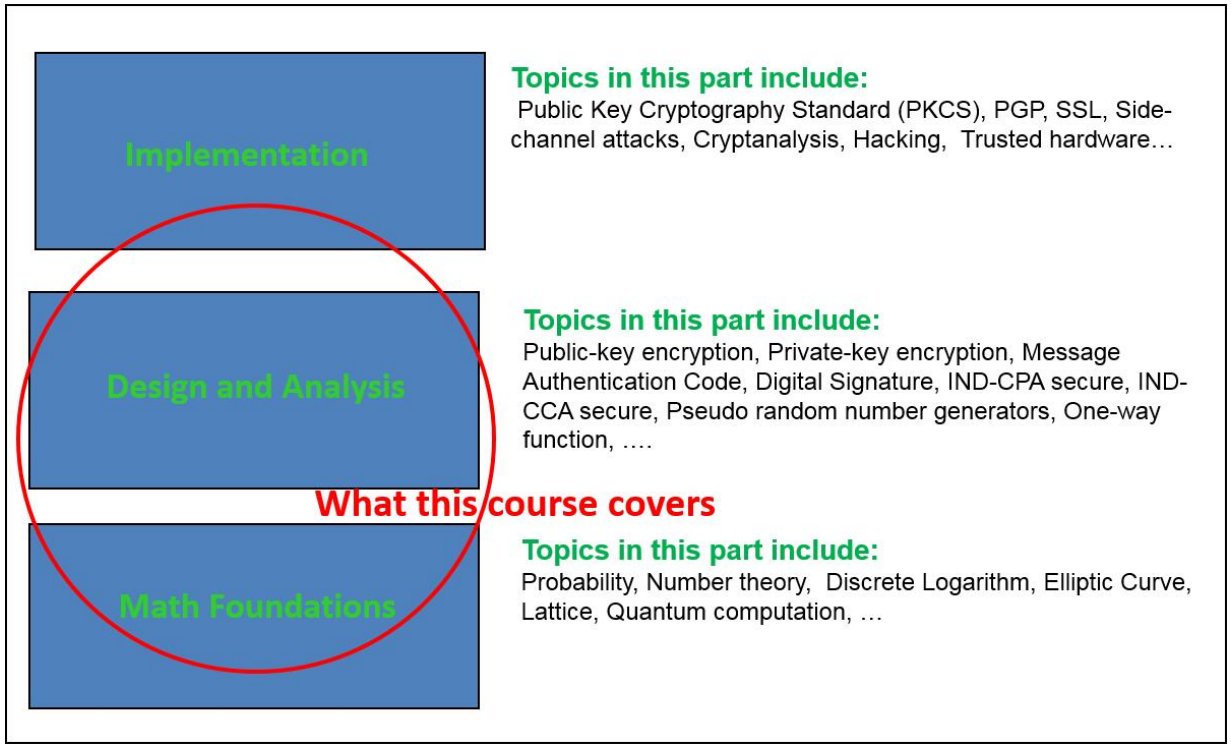


图: Topics of Cryptography

- We will see many "D&A", several "Math Foundations", and a few "Implementations".

Outline

Our focus of this course

- In this course, we will focus on:
  - Rigorous definitions of Security, e.g. *CPA-secure*, *CCA-secure*, ...
  - Cryptographic primitives, e.g. *pseudo-random generator, message authentication code, ...*
  - Cryptographic protocols, e.g. *Diffie-Hellman Key Exchange Protocol,...*

Rigorous definition of security: How can we be sure a system is secure?

- Encryption example 1: *"Only I know the encryption algorithm and keys, so it is safe."*

图: Relying on the secrecy of the encrypting mechanics

Rigorous definition of security: How can we be sure a system is secure?

- Encryption example 1: *"Only I know the encryption algorithm and keys, so it is safe."*

图: Relying on the secrecy of the encrypting mechanics

- It takes 5 seconds using an online cryptogram solver to solve it.

Rigorous definition of security: How can we're sure a system is secure?

- Encryption example 2: *"It would takes 100 years to break the system for an adversary with a currently* ***most advanced*** *computer using the* ***best known method****. "*
  Secure or not?

Rigorous definition of security: How can we're sure a system is secure?

- Encryption example 2: *"It would takes 100 years to break the system for an adversary with a currently* ***most advanced*** *computer using the* ***best known method****. "*
  Secure or not?
  NO !

Rigorous definition of security: How can we're sure a system is secure?

- Encryption example 2: *"It would takes 100 years to break the system for an adversary with a currently* ***most advanced*** *computer using the* ***best known method****. "*
  Secure or not?
  NO ! What if the adversary controls 10000 currently the most advanced computers?

Rigorous definition of security: How can we're sure a system is secure?

- Encryption example 2: *"It would takes 100 years to break the system for an adversary with a currently* ***most advanced*** *computer using the* ***best known method****. "*
  Secure or not?
  NO ! What if the adversary controls 10000 currently the most advanced computers?
- **Solutions**:
  We're sure it's secure if we can prove it with rigorous mathematical security proofs:
  computationally security,
  game-based proofs,
  simulation-based proofs
  ...

Cryptographic primitives: basic cryptographic tools

- Example 1: *"How to generate random numbers?"*

图: Pic from `http://www.moneycrashers.com`

Cryptographic primitives: basic cryptographic tools

- Example 1: *"How to generate random numbers?"*

图: Pic from `http://www.moneycrashers.com`

Solutions: Pseudo-random number generator (PRG), Pseudo-random functions (PRFs).

Cryptographic primitives: basic cryptographic tools

- Example 2: *"How to secure your data?"*

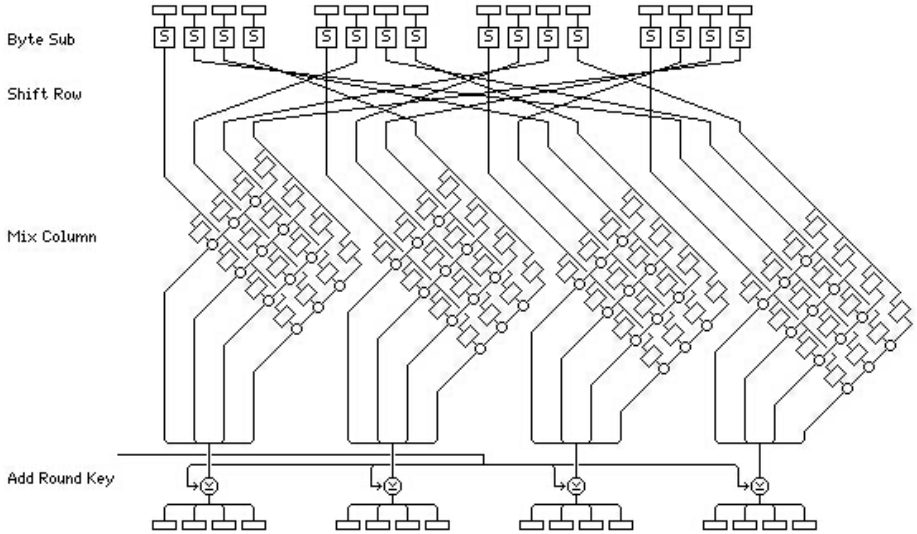


图: Pic from The Advanced Encryption Standard by Rijndael

Cryptographic primitives: basic cryptographic tools

- Example 2: *"How to secure your data?"*

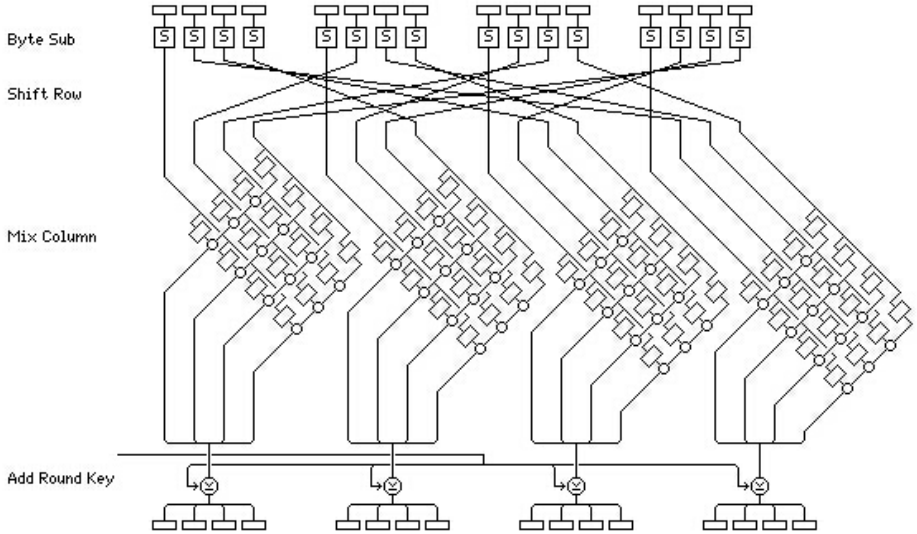


图: Pic from The Advanced Encryption Standard by Rijndael

Solutions: DES encryption, AES encryption, RSA encryption, RSA signature,...

Cryptographic protocols: tools of advanced usages

- Example 1 (*"Elsa or Anna?"*): You are allowed to get the phone number of ONLY one girl, but you have to make your choice in front of the two girls. How to get the correct number without breaking another's heart?

图: Which girl's number do you want?

Cryptographic protocols: tools of advanced usages

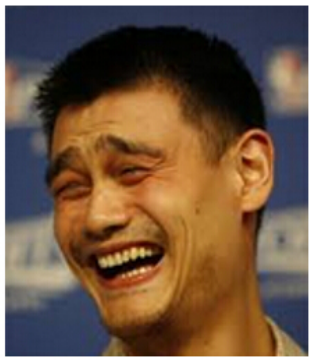
- Example 1 (*"Elsa or Anna?"*): You are allowed to get the phone number of ONLY one girl, but you have to make your choice in front of the two girls. How to get the correct number without breaking another's heart?

图: Which girl's number do you want?

Solutions: Oblivious Transfer!

Cryptographic protocols: tools of advanced usages

- Example 2 (*"Yao's Millionaire problem"*): Two millionaires want to know who is richer, but refuse to reveal their assets to each other.



X >? Y

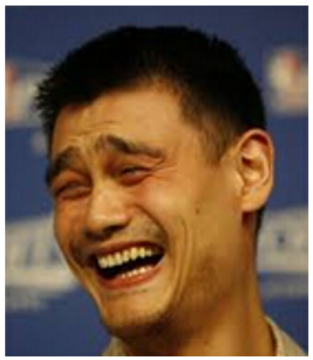
Assets: X

Assets: Y

图: Who is richer?

Cryptographic protocols: tools of advanced usages

- Example 2 (*"Yao's Millionaire problem"*): Two millionaires want to know who is richer, but refuse to reveal their assets to each other.



X >? Y

Assets: X　　Assets: Y

图: Who is richer?

Solutions: Secure comparison protocol!

Summary

- Rigorous Definition of Security
- Cryptographic Primitives
- Cryptographic Protocols

In this course, together we will witness the wits of cryptographers, and the powerfulness and miracles of Cryptography!

References I


Egele, M., Brumley, D., Fratantonio, Y., Kruegel, C..
An empirical study of cryptographic misuse in android applications.
Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013.


Katz, J. and Lindell, Y..
Introduction to modern crytography (2nd ed).
Chapman & Hall/CRC, 2014


Bellare, M.
Slides for CSE207 Introduction to Modern Cryptography
UCSD


Rijndael
The Advanced Encryption Standard (Rijndael)
`http://www.quadibloc.com/crypto/co040401.htm`


Apple Inc.
iOS Security (iOS10), March 2017
`https://www.apple.com/business/docs/iOS_Security_Guide.pdf`

Classical Ciphers (古典密码)

Sheng Zhong Yuan Zhang

Computer Science and Technology Department
Nanjing University

Outline

Caesar's cipher (凯撒密码)

- one of the simplest and most widely known encryption techniques.
- named after *Julius Caesar* for his use of this cipher writing secret letters in record.
- It is a *shift cipher* (移位密码).
- NOT secure, and can be easily broken.

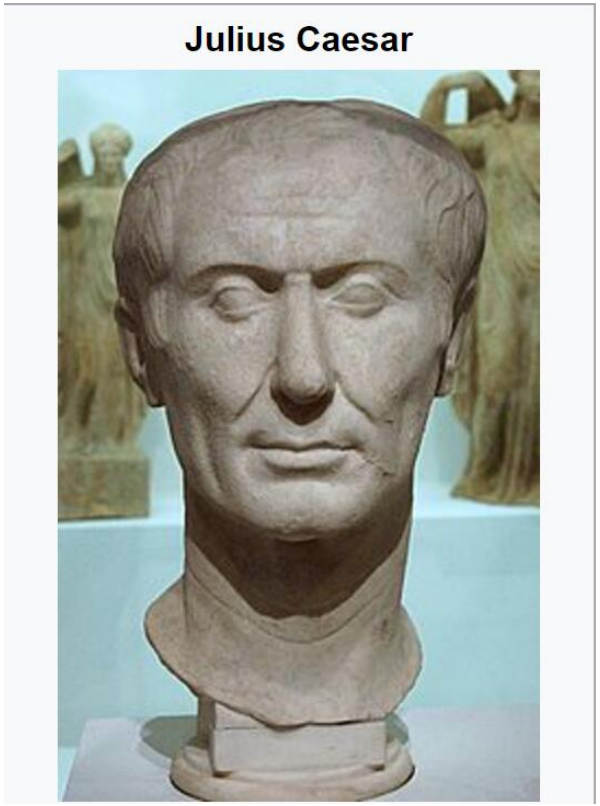


图: Julius Caesar (100BC-44BC), Dictator of the Roman Republic. Pic from Wikipedia

Caesar's cipher

Caesar's cipher (凯撒密码)

Encryption: shift the letters of the alphabet 3 places forward;
Decryption: shift the letters 3 places backwards.

Caesar's cipher

Caesar's cipher (凯撒密码)

Encryption: shift the letters of the alphabet 3 places forward;
Decryption: shift the letters 3 places backwards.

Enc: $a \to D; b \to E; \ldots; w \to Z; x \to A; y \to B; z \to C$
Dec: $D \to a; E \to b; \ldots; Z \to w; A \to x; B \to y; C \to z$
E.g.

`VHHXWRPRUURZ` (Ciphertext)

Caesar's cipher

Caesar's cipher (凯撒密码)

Encryption: shift the letters of the alphabet 3 places forward;
Decryption: shift the letters 3 places backwards.

Enc: $a \to D; b \to E; \ldots; w \to Z; x \to A; y \to B; z \to C$
Dec: $D \to a; E \to b; \ldots; Z \to w; A \to x; B \to y; C \to z$
E.g.

`VHHXWRPRUURZ` (Ciphertext)
$\to$ `seeutomorrow` (Plaintext)

Weaknesses of Caesar's cipher

- It's a fixed algorithm.

Weaknesses of Caesar's cipher

- It's a fixed algorithm.
- The key space is too small even we use a randomized version of it:

A randomized Caesar's cipher: The shift cipher (移位密码)

Enc: choose a random $k \in \mathbb{N}^+$, shift all letters $(k \mod 26)$ places forward;
Dec: shift the letters $(k \mod 26)$ places backward.

- The effective key space's size is only 26.
- Easy to be broken via the **exhaustive search** or **the brute-force attack**:

Lessons learned in Caesar's cipher and the shift cipher

Sufficient Key Space Principle (充分密钥空间原则)

Any secure encryption scheme must have a key space that is not vulnerable to exhaustive search.

Method	Date	Symmetric	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
[1] Lenstra / Verheul	2017	83	1717 1344	147	1717	157	166
[2] Lenstra Updated	2017	80	1300 1435	159	1300	159	159
[3] ECRYPT II	2016 - 2020	96	1776	192	1776	192	192
[4] NIST	2016 - 2030	112	2048	224	2048	224	224
[5] ANSSI	2014 - 2020	100	2048	200	2048	200	200
[6] IAD-NSA	-	256	3072	-	-	384	384
[7] RFC3766	-	-	-	-	-	-	-
[8] BSI	2017 - 2022	128	2000	250	2000	250	256

All key sizes are provided in bits. These are the minimal sizes for security.

图: Recommended key length from `http://www.keylength.com/`

ROT-13: A special case of Caesar's cipher

ROT-13 (Rotate by 13 places)

Enc: Shift all letters in the alphabet 13 places forward.
Dec: Shift all letters in the alphabet 13 places backward or forward.

- A special case of Caesar's cipher that is still in use, despite being NO SECURE.
- ROT-13 can be found in online forums to write spoilers, puzzle solutions, etc.
- `www.rot13.com`.

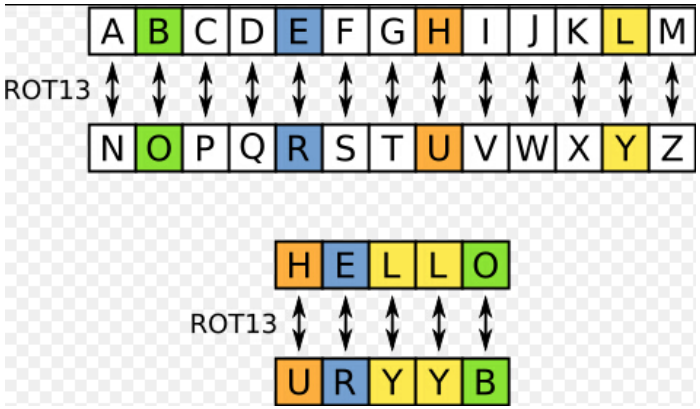


图: Pic from `https://en.wikipedia.org/wiki/ROT13`

The mono-alphabetic substitution cipher

The Mono-alphabetic Substitution Cipher (单字母替换密码)

Enc: randomly choose a permutation $\pi : \{a, b, \ldots, z\} \to \{A, B, \ldots, Z\}$, replace every letter with its image under π

Dec: substitute every letter with its image under π's inverse function.

The mono-alphabetic substitution cipher

The Mono-alphabetic Substitution Cipher (单字母替换密码)

Enc: randomly choose a permutation $\pi : \{a, b, \ldots, z\} \to \{A, B, \ldots, Z\}$, replace every letter with its image under π

Dec: substitute every letter with its image under π's inverse function.

- a permutation is a one-to-one mapping from a set to the set itself.

The mono-alphabetic substitution cipher

The Mono-alphabetic Substitution Cipher (单字母替换密码)

Enc: randomly choose a permutation $\pi : \{a, b, \ldots, z\} \to \{A, B, \ldots, Z\}$, replace every letter with its image under π

Dec: substitute every letter with its image under π's inverse function.

- a permutation is a one-to-one mapping from a set to the set itself.
- the one-to-one mapping enables correct decryption.

An example of the MAS cipher

- The chosen permutation π:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	E	U	A	D	N	B	K	V	M	R	O	C	Q	F	S	Y	H	W	G	L	Z	I	J	P	T

An example of the MAS cipher

- The chosen permutation π:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	E	U	A	D	N	B	K	V	M	R	O	C	Q	F	S	Y	H	W	G	L	Z	I	J	P	T

- Ciphertext:
  KDOOFUHPSGF

An example of the MAS cipher

- The chosen permutation π:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	E	U	A	D	N	B	K	V	M	R	O	C	Q	F	S	Y	H	W	G	L	Z	I	J	P	T

- Ciphertext:
  `KDOOFUHPSGF`
- After decryption:
  `hellocrypto`

A few security analyses

Q: What is the size of the MAS cipher's key space?

$\pi(a) = a, b, ..., z$;	so 26 possible choices;
$\pi(b) = a, b, ..., z$, but not $\pi(a)$;	so 25 possible choices;
...	...
$\pi(z) = a, b, ..., z$, but not $\pi(a), \pi(b), \ldots,$ or $\pi(y)$	so 1 possible choices;

In total, $26 \times 25 \times 24 \times \ldots \times 1 = 26! \approx 2^{88}$.

A few security analyses

Q: What is the size of the MAS cipher's key space?

A: It equals the total number of permutations on $\{a, b, \ldots, z\}$.
There are approximately 2^{88} different permutations in total:

$\pi(a) = a, b, ..., z$;	so 26 possible choices;
$\pi(b) = a, b, ..., z$, but not $\pi(a)$;	so 25 possible choices;
...	...
$\pi(z) = a, b, ..., z$, but not $\pi(a), \pi(b), \ldots,$ or $\pi(y)$	so 1 possible choices;

In total, $26 \times 25 \times 24 \times \ldots \times 1 = 26! \approx 2^{88}$.

A few security analyses

Q: How good is this 2^{88}?

[1] From Wiki: the age of our universe since the big bang is $13.799 \pm 0.0021 \times 10^9$ years.

A few security analyses

Q: How good is this 2^{88}?

A: Suppose an adversary uses a 4GHz computer to run exhaustive search, the time it needs to finish searching the entire key space is at least $2^{88}/2^{32}$ secs $\approx 2.28341614 \times 10^9$ yrs.[1]

图: A 6000+ yuan laptop in 2015

[1] From Wiki: the age of our universe since the big bang is $13.799 \pm 0.0021 \times 10^9$ years.

A few security analyses

Q: How good is this 2^{88}?

A: Suppose an adversary uses a 4GHz computer to run exhaustive search, the time it needs to finish searching the entire key space is at least $2^{88}/2^{32}$ secs $\approx 2.28341614 \times 10^9$ yrs.[1]

图: A 6000+ yuan laptop in 2015

Q: Secure or not?

[1] From Wiki: the age of our universe since the big bang is $13.799 \pm 0.0021 \times 10^9$ years.

A few security analyses

Q: How good is this 2^{88}?

A: Suppose an adversary uses a 4GHz computer to run exhaustive search, the time it needs to finish searching the entire key space is at least $2^{88}/2^{32}$ secs $\approx 2.28341614 \times 10^9$ yrs.[1]

图: A 6000+ yuan laptop in 2015

Q: Secure or not?

A: Maybe secure against the exhaustive search, but highly insecure against other attacks.

[1] From Wiki: the age of our universe since the big bang is $13.799 \pm 0.0021 \times 10^9$ years.

A major drawback of MAS cipher

- The encryptions of a letter are always the same in MAS cipher. E.g.

  `noclassthisfriday` $\rightarrow$ `QFUOXWWGKVWNHVAXP`

- So, statistics or letter frequencies are the same in plaintexts and in ciphertexts. E.g.

  # of appearance (`s`) in plaintext

  = # of appearance (`W`) in ciphertext.

- This drawback allows adversary to perform statistic attacks.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	E	U	A	D	N	B	K	V	M	R	O	C	Q	F	S	Y	H	W	G	L	Z	I	J	P	T

Statistical attacks on the MAS cipher

- What if we know a few letters' appearing times in the plaintext?
  For example, given the ciphertext "QFUOXWWGKVWNHVAXP", somehow the adversary knows "letter `s` appears 3 times in the plaintext" and "letter `i` and `a` appear 2 times."

Statistical attacks on the MAS cipher

- What if we know a few letters' appearing times in the plaintext?
  For example, given the ciphertext "QFUOXWWGKVWNHVAXP", somehow the adversary knows "letter s appears 3 times in the plaintext" and "letter i and a appear 2 times."
  Then, we know the plaintext should be:
  $****ass*is**i*a*$ or
  $****iss*as**a*i*$

Q: Now how many possible plaintexts can we have?

Statistical attacks on the MAS cipher

- What if we know a few letters' appearing times in the plaintext?
  For example, given the ciphertext "QFUOXWWGKVWNHVAXP", somehow the adversary knows "letter s appears 3 times in the plaintext" and "letter i and a appear 2 times."
  Then, we know the plaintext should be:
  $****ass*i*s**i*a*$ or
  $****iss*a*s**a*i*$

Q: Now how many possible plaintexts can we have?

A: 26*25*24*23*22*21*20*19*18*17

Statistical attacks on the MAS cipher

- What if we know a few letters' appearing times in the plaintext?
  For example, given the ciphertext "QFUOXWWGKVWNHVAXP", somehow the adversary knows "letter s appears 3 times in the plaintext" and "letter i and a appear 2 times."
  Then, we know the plaintext should be:
  $****ass*i*s**i*a*$ or
  $****iss*a*s**a*i*$

Q: Now how many possible plaintexts can we have?

A: 26*25*24*23*22*21*20*19*18*17 **Much smaller than 26!**

Statistical attacks on the MAS cipher

- What if we know a few letters' appearing times in the plaintext?
  For example, given the ciphertext "QFUOXWWGKVWNHVAXP", somehow the adversary knows "letter s appears 3 times in the plaintext" and "letter i and a appear 2 times."
  Then, we know the plaintext should be:
  $****ass*i*s**i*a*$ or
  $****iss*a*s**a*i*$

Q: Now how many possible plaintexts can we have?

A: 26*25*24*23*22*21*20*19*18*17 **Much smaller than 26!**

- The search can be further expedited when English words are encrypted:

  "u generally follows q", "h is likely to appear between t and e", ...

The Statistical Attack on the MAS cipher

- Although in practice, we may not know the actual appearing times of each letter in the plaintext, we know the approximate frequency of each letter in a general plaintext (**assuming english-language text is encrypted**):

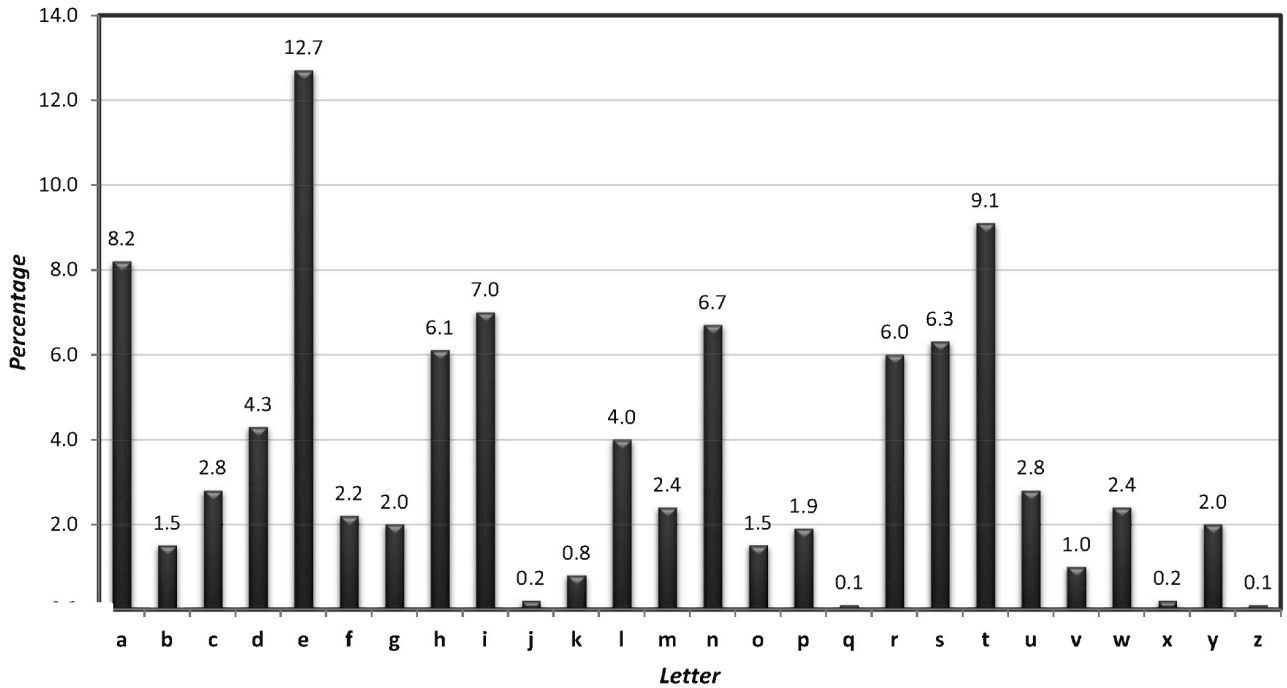


图: Average letter frequencies for English-language text

Lessons learned from statistical attacks on MAS cipher

Q: Why statistical attacks work for MAS cipher?

Lessons learned from statistical attacks on MAS cipher

Q: Why statistical attacks work for MAS cipher?

A: Because the key defines a fixed mapping that is applied letter-by-letter to the plaintext.

Lessons learned from statistical attacks on MAS cipher

Q: Why statistical attacks work for MAS cipher?

A: Because the key defines a fixed mapping that is applied letter-by-letter to the plaintext.

Q: Better alternatives?

Lessons learned from statistical attacks on MAS cipher

Q: Why statistical attacks work for MAS cipher?

A: Because the key defines a fixed mapping that is applied letter-by-letter to the plaintext.

Q: Better alternatives?

A: Try to avoid this fixed letter-by-letter mapping. For example,

Poly-alphabetic Substitution cipher (多字母替换密码)

The encryption/decryption is defined with a mapping/inverse-mapping which is applied on blocks of plaintext characters.

e.g. **Enc**: ab→DZ, and ac→TY.

Lessons learned from statistical attacks on MAS cipher

Q: Why statistical attacks work for MAS cipher?

A: Because the key defines a fixed mapping that is applied letter-by-letter to the plaintext.

Q: Better alternatives?

A: Try to avoid this fixed letter-by-letter mapping. For example,

Poly-alphabetic Substitution cipher (多字母替换密码)

The encryption/decryption is defined with a mapping/inverse-mapping which is applied on blocks of plaintext characters.

e.g. **Enc**: ab→DZ, and ac→TY.

The Vigenère cipher

Let's see an example of the Vigenère cipher:

Plaintext:	t	e	l	l	h	i	m	a	b	o	u	t	m	e
Key(repeated):	c	a	f	e	c	a	f	e	c	a	f	e	c	a
Ciphertext:	V	E	Q	P	J	I	R	E	D	O	Z	X	O	E

- The short phrase "`cafe`" is used as the key.
- Notice now the two "`l`"s in the plaintext are mapped to two different letters "`Q`" and "`P`".
- Similarly, the two "`m`"s are mapped to two different different letters "`R`" and "`O`".

The Vigenère (poly-alphabetic shift) cipher

The Vigenère cipher (維吉尼亞密碼 or 多字母移位密碼)

Given an alphabet $\mathcal{A}$ of s letters, choose a secret key $k = \{\pi_0, \ldots, \pi_{m-1}\}$ consists of m shiftings on $\mathcal{A}$. Given a plaintext x that consists of letters $x_0, x_1, \ldots,$

Enc: Apply $\pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots$ to $x_0, x_1, \ldots$, i.e.,

$$y_i = \pi_{i \bmod m}(x_i) \text{ for } i = 0, 1, \ldots$$

Dec: Apply $\pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots$ to $y_0, y_1, \ldots$

The Vigenère (poly-alphabetic shift) cipher

The Vigenère cipher (維吉尼亞密碼 or 多字母移位密碼)

Given an alphabet $\mathcal{A}$ of s letters, choose a secret key $k = \{\pi_0, \ldots, \pi_{m-1}\}$ consists of m shiftings on $\mathcal{A}$. Given a plaintext x that consists of letters $x_0, x_1, \ldots,$

Enc: Apply $\pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots$ to $x_0, x_1, \ldots,$ i.e.,

$$y_i = \pi_{i \bmod m}(x_i) \text{ for } i = 0, 1, \ldots$$

Dec: Apply $\pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots$ to $y_0, y_1, \ldots$

- named after French cryptographer Blaise de Vigenère, but first invented by the Italian cryptographer Giovan Battista Bellaso in 1553.

The Vigenère (poly-alphabetic shift) cipher

The Vigenère cipher (維吉尼亞密碼 or 多字母移位密碼)

Given an alphabet $\mathcal{A}$ of s letters, choose a secret key $k = \{\pi_0, \ldots, \pi_{m-1}\}$ consists of m shiftings on $\mathcal{A}$. Given a plaintext x that consists of letters $x_0, x_1, \ldots,$

Enc: Apply $\pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots$ to $x_0, x_1, \ldots,$ i.e.,

$$y_i = \pi_{i \bmod m}(x_i) \text{ for } i = 0, 1, \ldots$$

Dec: Apply $\pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots$ to $y_0, y_1, \ldots$

- named after French cryptographer Blaise de Vigenère, but first invented by the Italian cryptographer Giovan Battista Bellaso in 1553.
- considered unbreakable for over three centuries. (The first published attack is by Wilhelm Kasiski in 1863)

The Vigenère (poly-alphabetic shift) cipher

The Vigenère cipher (維吉尼亞密碼 or 多字母移位密碼)

Given an alphabet $\mathcal{A}$ of s letters, choose a secret key $k = \{\pi_0, \ldots, \pi_{m-1}\}$ consists of m shiftings on $\mathcal{A}$. Given a plaintext x that consists of letters $x_0, x_1, \ldots,$

Enc: Apply $\pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots$ to $x_0, x_1, \ldots,$ i.e.,

$$y_i = \pi_{i \bmod m}(x_i) \text{ for } i = 0, 1, \ldots$$

Dec: Apply $\pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots$ to $y_0, y_1, \ldots$

- named after French cryptographer Blaise de Vigenère, but first invented by the Italian cryptographer Giovan Battista Bellaso in 1553.
- considered unbreakable for over three centuries. (The first published attack is by Wilhelm Kasiski in 1863)
- It is a special case of the poly-alphabetic substitution cipher.

The Vigenère (poly-alphabetic shift) cipher

The Vigenère cipher (維吉尼亞密碼 or 多字母移位密碼)

Given an alphabet $\mathcal{A}$ of s letters, choose a secret key $k = \{\pi_0, \ldots, \pi_{m-1}\}$ consists of m shiftings on $\mathcal{A}$. Given a plaintext x that consists of letters $x_0, x_1, \ldots,$

Enc: Apply $\pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots, \pi_{m-1}, \pi_0, \ldots$ to $x_0, x_1, \ldots,$ i.e.,

$$y_i = \pi_{i \bmod m}(x_i) \text{ for } i = 0, 1, \ldots$$

Dec: Apply $\pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots, \pi_{m-1}^{-1}, \pi_0^{-1}, \ldots$ to $y_0, y_1, \ldots$

- named after French cryptographer Blaise de Vigenère, but first invented by the Italian cryptographer Giovan Battista Bellaso in 1553.
- considered unbreakable for over three centuries. (The first published attack is by Wilhelm Kasiski in 1863)
- It is a special case of the poly-alphabetic substitution cipher.
- It is a method of encrypting alphabetic text by performing a series of different Caesar's ciphers.

One attack on the Vigenère cipher

We can break the Vigenère cipher in the following two steps:

- Determine the key length m.
- Break m shift ciphers.

To determine the key length

- One possible method to determine the key length is by looking for repeated patterns in the ciphertext.

Plaintext:	the man and the woman retrieved the letter from the post office
Key:	bea dsb ead sbe adsbe adsbeadsb ead sbeads bead sbe adsb eadsbe
Ciphertext:	ULE PSO ENG LII WREBR RHLSMEYWE XHH DFXTHJ GVOP LII PRKU SFIADI

图: A Vigenère cipher with the key "beads"

For example, by inspecting the ciphertext above, we can know 30, which is the distance between two "LII"s in the ciphetext, is a multiple of the key length.

To break m shift ciphers

After we know possible key lengths, for each key length m, we can break the Vigenère cipher by breaking m shift ciphers with m ciphertexts

$$y_0, y_m, y_{2m}, \cdots$$

$$y_1, y_{m+1}, y_{2m+1}, \cdots$$

...

$$y_{m-1}, y_{2m_1}, \cdots$$

independently using statistics attacks.

An easy-to-automate statistical attack on shift ciphers

- To automate attacks with computers, a quantitative metric that measures the correctness of a "guess" needs to be computed.

An easy-to-automate statistical attack on shift ciphers

- To automate attacks with computers, a quantitative metric that measures the correctness of a "guess" needs to be computed.
- The following frequency graph can help.

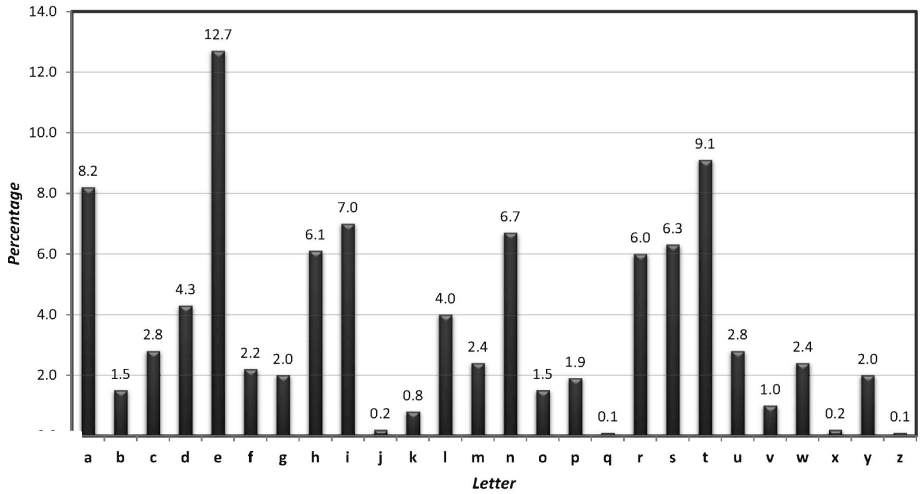


图: Average letter frequencies for English-language text

An easy-to-automate statistical attack on shift ciphers

- Let p_i denote the frequency of the i-th letter in normal English text. According to the figure, we know:

$$\sum_{0}^{25} p_i^2 \approx 0.065.$$

- Let q_i denote the frequency of the i-th letter of the alphabet in the ciphertext:

$$q_i = \frac{\#\text{ of occurrences of the } i\text{-th letter in the ciphertext}}{\text{length of the ciphertext}}.$$

An easy-to-automate statistical attack on shift ciphers

- Let p_i denote the frequency of the i-th letter in normal English text. According to the figure, we know:

$$\sum_{0}^{25} p_i^2 \approx 0.065.$$

- Let q_i denote the frequency of the i-th letter of the alphabet in the ciphertext:

$$q_i = \frac{\#\text{ of occurrences of the } i\text{-th letter in the ciphertext}}{\text{length of the ciphertext}}.$$

- Next, we compute I_j for all $j \in \{0, 25\}$, and find the j for which I_j is the closest to 0.065:

$$I_j = \sum_{i=0}^{25} p_i \cdot q_{i+j}.$$

Classical ciphers

- Classical cipher design is more of an art than a science.
- The security of many classical ciphers heavily rely on hiding their details e.g. how encryptions are done.
- All historical classical ciphers have been proven to be NOT SECURE.

Modern Cryptography

Over the past several decades, cryptography (modern cryptography) has developed into more of a science.

- Ultimate goal: to give a rigorous proof that a given construction is secure.

Modern Cryptography

Over the past several decades, cryptography (modern cryptography) has developed into more of a science.

- Ultimate goal: to give a rigorous proof that a given construction is secure.
- Emphasize the security definition.

Modern Cryptography

Over the past several decades, cryptography (modern cryptography) has developed into more of a science.

- Ultimate goal: to give a rigorous proof that a given construction is secure.
- Emphasize the security definition.
- Emphasize the unproven assumptions about the algorithmic hardness of certain mathematical problems.

Modern Cryptography

Over the past several decades, cryptography (modern cryptography) has developed into more of a science.

- Ultimate goal: to give a rigorous proof that a given construction is secure.
- Emphasize the security definition.
- Emphasize the unproven assumptions about the algorithmic hardness of certain mathematical problems.

"*An emphasis on definitions, assumptions, and proofs distinguishes modern cryptography from classical cryptography.*"

Modern Cryptography's Principle 1-Formal Definitions

- One of the key contributions of modern cryptography: the recognition that formal definitions of security are essential for proper design, study, evaluation, and usage of cryptography primitives.

Modern Cryptography's Principle 1-Formal Definitions

- One of the key contributions of modern cryptography: the recognition that formal definitions of security are essential for proper design, study, evaluation, and usage of cryptography primitives.
- To formally define the security, you must first understand what you are trying to achieve, i.e. the security goal, and what kind of adversaries you are dealing with, i.e. the threat model.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is our security goal?

- It should be impossible for an attacker to recover the key from the ciphertext.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is our security goal?

- It should be impossible for an attacker to recover the key from the ciphertext.
- It should be impossible for an attacker to recover the entire plaintext from the ciphertext.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is our security goal?

- It should be impossible for an attacker to recover the key from the ciphertext.
- It should be impossible for an attacker to recover the entire plaintext from the ciphertext.
- It should be impossible for an attacker to recover any character of the plaintext from the ciphertext.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is our security goal?

- It should be impossible for an attacker to recover the key from the ciphertext.
- It should be impossible for an attacker to recover the entire plaintext from the ciphertext.
- It should be impossible for an attacker to recover any character of the plaintext from the ciphertext.
- Regardless of any information an attacker already has, a ciphertext should leak no additional information about the underlying plaintext.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is the threat model?

- Ciphertext-only attack: The adversary just observes ciphertexts, and guess.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is the threat model?

- Ciphertext-only attack: The adversary just observes ciphertexts, and guess.
- Known-plaintext attack: The adversary is allowed to know some plaintext/ciphertext pairs before trying to attack on other ciphertexts.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is the threat model?

- Ciphertext-only attack: The adversary just observes ciphertexts, and guess.
- Known-plaintext attack: The adversary is allowed to know some plaintext/ciphertext pairs before trying to attack on other ciphertexts.
- Chosen-plaintext attack: The adversary can obtain plaintext/ciphertext pairs for plaintexts of its own choice.

Modern Cryptography's Principle 1-Formal Definitions

An example: secure encryption.
What is the threat model?

- Ciphertext-only attack: The adversary just observes ciphertexts, and guess.
- Known-plaintext attack: The adversary is allowed to know some plaintext/ciphertext pairs before trying to attack on other ciphertexts.
- Chosen-plaintext attack: The adversary can obtain plaintext/ciphertext pairs for plaintexts of its own choice.
- Chosen-ciphertext attack: The adversary can obtain the corresponding plaintexts for ciphertexts of its own choice.

Modern Cryptography's Principle 2-Precise Assumptions

- Most modern cryptographic constructions cannot be proven secure unconditionally.
- Proofs often require resolving questions in the theory of computational complexity that seem far from being answered today.

Modern Cryptography's Principle 2-Precise Assumptions

- Most modern cryptographic constructions cannot be proven secure unconditionally.
- Proofs often require resolving questions in the theory of computational complexity that seem far from being answered today.

For example. Factoring $N = p \times q$, where p, q are large prime numbers, is assumed to be difficult without knowing p or q.

Modern Cryptography's Principle 3-Proofs of Security

Provable Security and Real-world Security

Provable security of a scheme does not necessarily imply security of that scheme in the real world.

Modern Cryptography's Principle 3-Proofs of Security

Provable Security and Real-world Security

Provable security of a scheme does not necessarily imply security of that scheme in the real world.

This difference is not a drawback:

- To attack, we only need to focus on the definition or the underlying assumption.
- In return, we can refine the security definitions to more closely match the real world, and work on the assumptions.

Security through obscurity

- The very idea based on which most classic ciphers are designed and believed to be secure...
- Keeping encryption algorithms secret improves security? Use "home-brewed" algorithm?

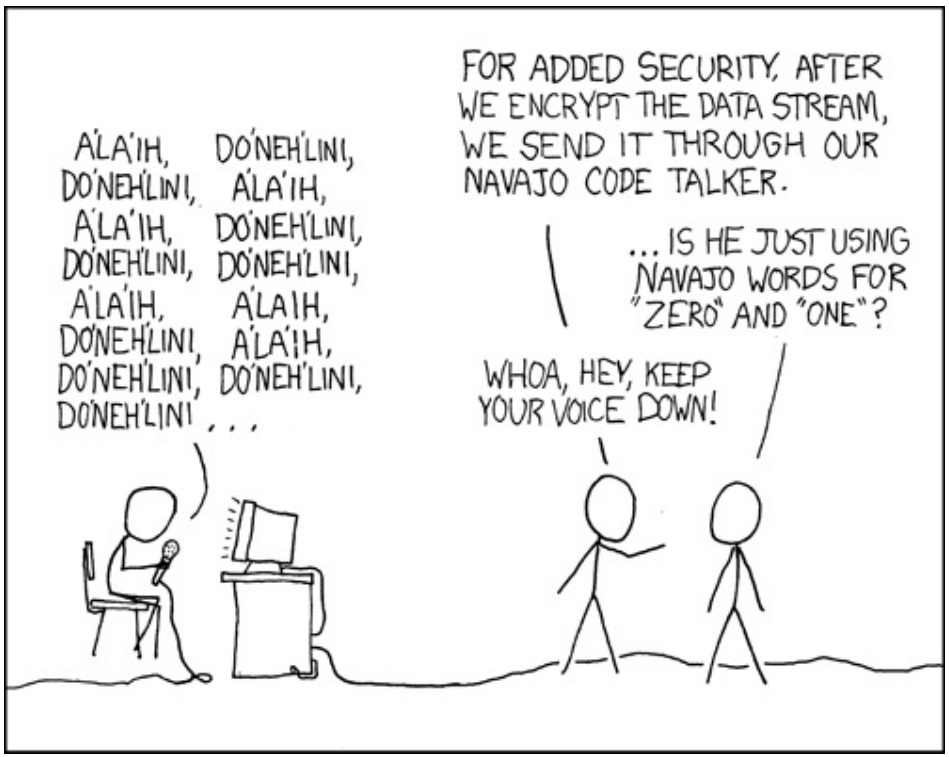


图: "Code Talkers" (pic courtesy of xkcd.com)

Kerckhoffs' Principle(柯克霍夫原则)

- "*The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.*"

  by Auguste Kerckhoffs in 1883 (Dutch, 1835-1903)

Kerckhoffs' Principle(柯克霍夫原则)

- "*The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.*"
  by Auguste Kerckhoffs in 1883 (Dutch, 1835-1903)
- Need to be secure even all details (except the encryption key) are known by adversary.
- **Security relies solely on secrecy of the key, rather than secrecy of the encryption method.**

Discussion

Q: Which do you support/prefer/choose?

Discussion

Q: Which do you support/prefer/choose?

- Read Chapter 1.2 to see authors' opinions.

References I


Katz, J. and Lindell, Y..
Introduction to modern crytography (2nd ed).
Chapman & Hall/CRC, 2014


Joachim von zur Gathen.
Classical Cryptography.
Version: July 14, 2008