Cryptography Engineering

Jerry J. R. Shieh, PhD

February 29 2024

Housekeeping

Write Critiques with a group (1-5 people).

Critique should contain the following:

1. **Summary** – answering these four questions in your own words:

   What problem is the paper trying to solve?

   Why does the problem matter?

   What is the approach used to solve the problem?

   What is the conclusion drawn from this work?

2. **Strength(s) of the paper**

3. **Weakness(es) of the paper**

4. **Your own reflection, which can include but not limited to:**

   What did you learn from this paper?

   How would you improve or extend the work if you were the author?

   What are the unsolved questions that you want to investigate?

   What are the broader impacts of this proposed technology?

5. **Realization of a technical specification or algorithm as a experiment.**

Critique 1

- Password Managers: Attacks and Defenses
  David Silver, Suman Jana, and Dan Boneh, Stanford University;
  Eric Chen and Collin Jackson, Carnegie Mellon University.
- Text may with figures, about 2000+ words
- With experiment lab.

Grading Components

- Quiz and homework assignments (6+*5%)
- Reading critiques (3+*8%)
- Midterm exam (16%)
- Group project (20%)
  - Written proposal (5%)
  - Final oral presentation (5%)
  - Final report (10%)
- Class participation (10+%)

Outline

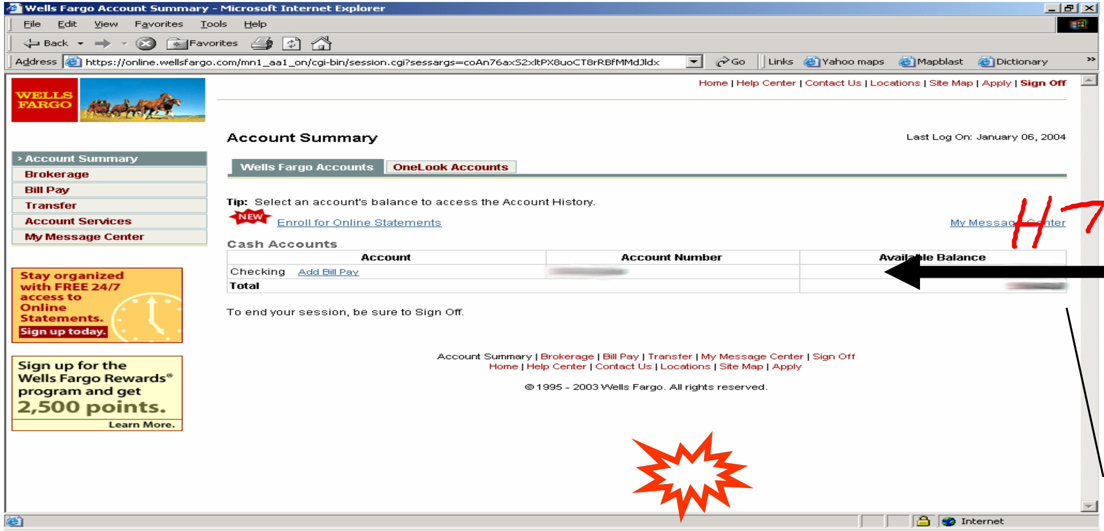
1. Cryptography Engineering Application
2. Symmetry Key and Public Key Encryption
3. Open security vs Obscurity for Encryption Algorithm
4. Privacy from anonymity
5. How to solve Quiz 1

Cryptography is everywhere

- **Secure communication**:
  - web traffic: HTTPS SSL
  - wireless traffic: 802.11i WPA2 (and WEP) WPA3, GSM, Bluetooth
  - IPsec
- **Encrypting files on disk (Data at Rest)**: EFS, TrueCrypt
- **Content protection** (e.g. DVD, Blu-ray): CSS, AACS
- **User authentication**
- … and much much more

1.Secure communication

Alice



Wells Fargo Account Summary - Microsoft Internet Explorer

File Edit View Favorites Tools Help
Back Favorites
Address https://online.wellsfargo.com/mn1_aa1_on/cgi-bin/session.cgi?sessargs=coAn76axS2xltPX8uoCT8rRBfMMdJldx
Go Links Yahoo maps Mapblast Dictionary
WELLS FARGO

Home | Help Center | Contact Us | Locations | Site Map | Apply | **Sign Off**

- › Account Summary
- Brokerage
- Bill Pay
- Transfer
- Account Services
- My Message Center

Stay organized with FREE 24/7 access to Online Statements. Sign up today.

Sign up for the Wells Fargo Rewards® program and get 2,500 points. Learn More.

Account Summary

Last Log On: January 06, 2004

Wells Fargo Accounts | OneLook Accounts

Tip: Select an account's balance to access the Account History.

NEW Enroll for Online Statements

My Message Center

Cash Accounts

Account	Account Number	Available Balance
Checking Add Bill Pay	[redacted]	
Total		[redacted]

To end your session, be sure to Sign Off.

Account Summary | Brokerage | Bill Pay | Transfer | My Message Center | Sign Off
Home | Help Center | Contact Us | Locations | Site Map | Apply

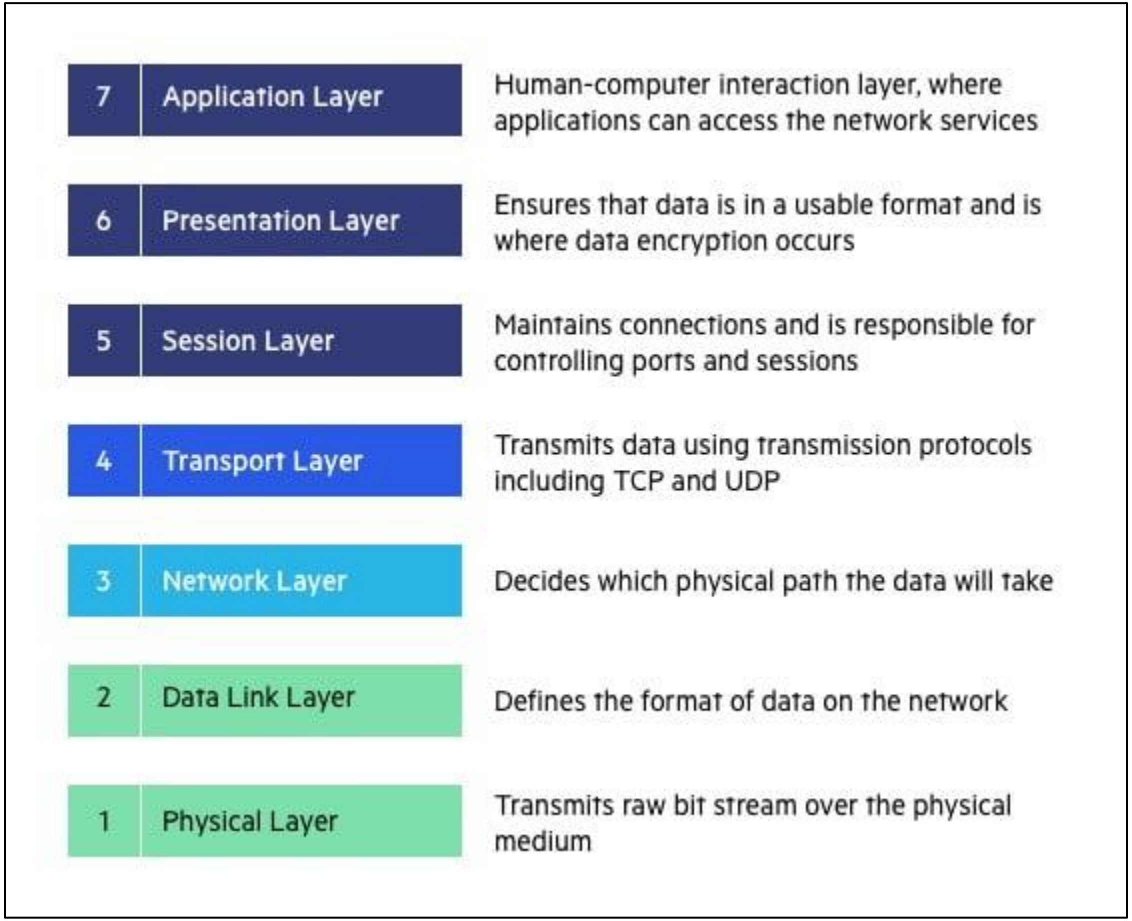
© 1995 - 2003 Wells Fargo. All rights reserved.

HTTPS

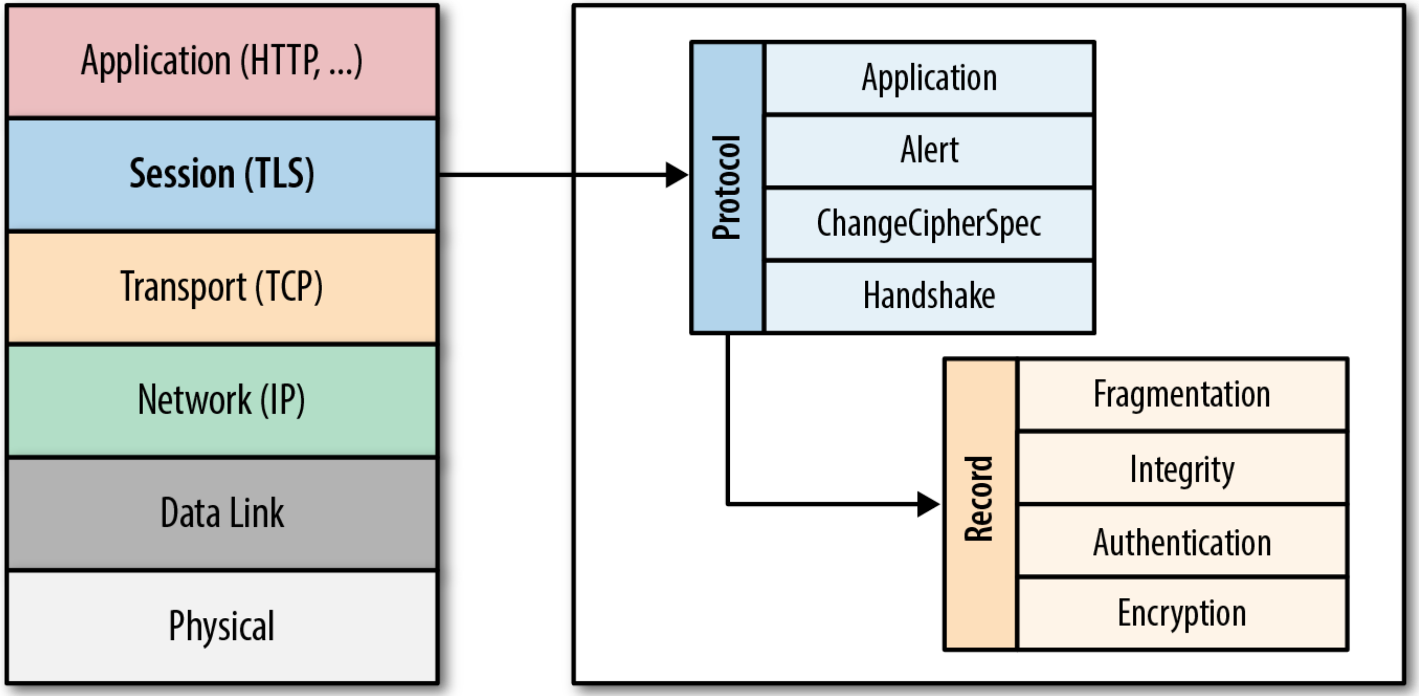
Bob

no eavesdropping
no tampering

不能竊聽、也不能篡改要完整傳遞



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium



Application (HTTP, ...)

Session (TLS)

Transport (TCP)

Network (IP)

Data Link

Physical

Protocol

Application

Alert

ChangeCipherSpec

Handshake

Record

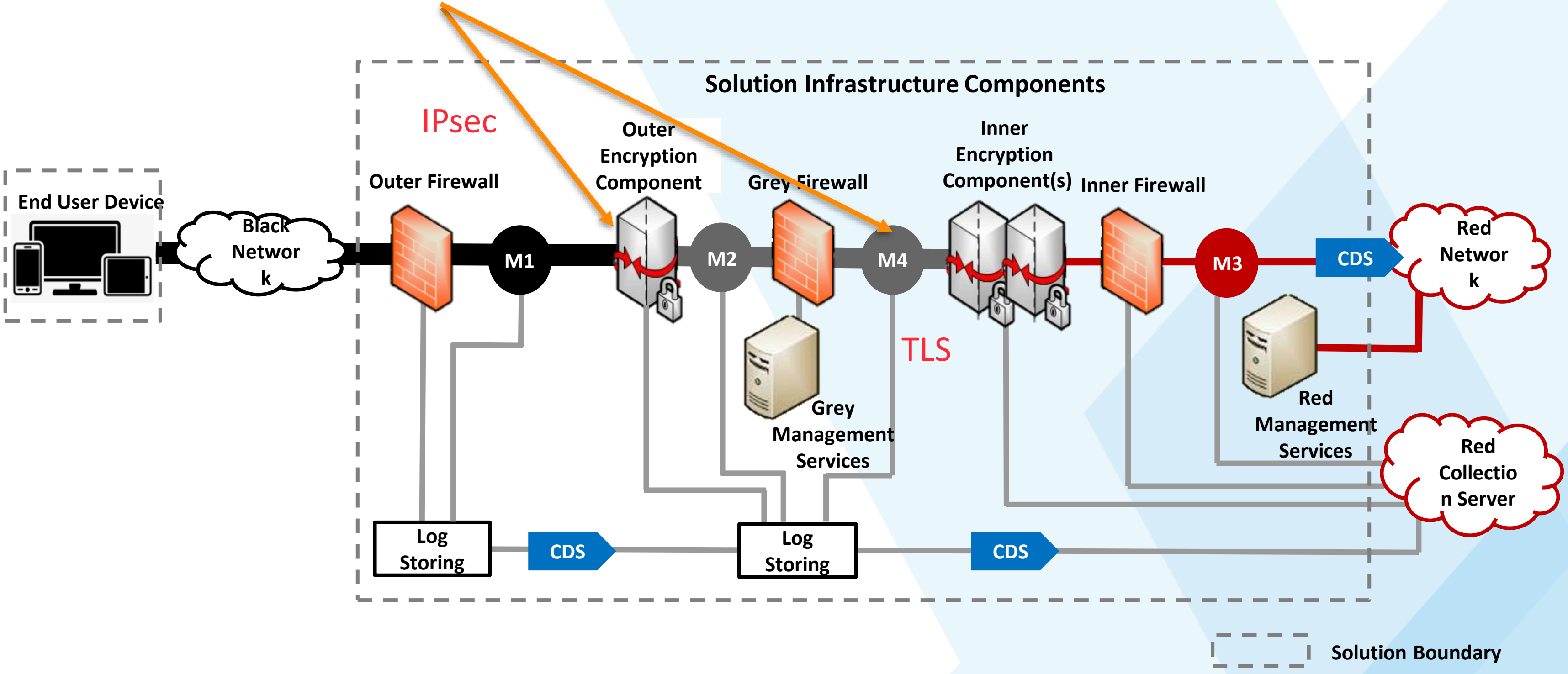
Fragmentation

Integrity

Authentication

Encryption

In real world, applied two layered Secure Communication (IPsec, TLS)



https://www.nsa.gov/Portals/70/documents/resources/everyone/csfc/capability-packages/MACPv2_1.pdf

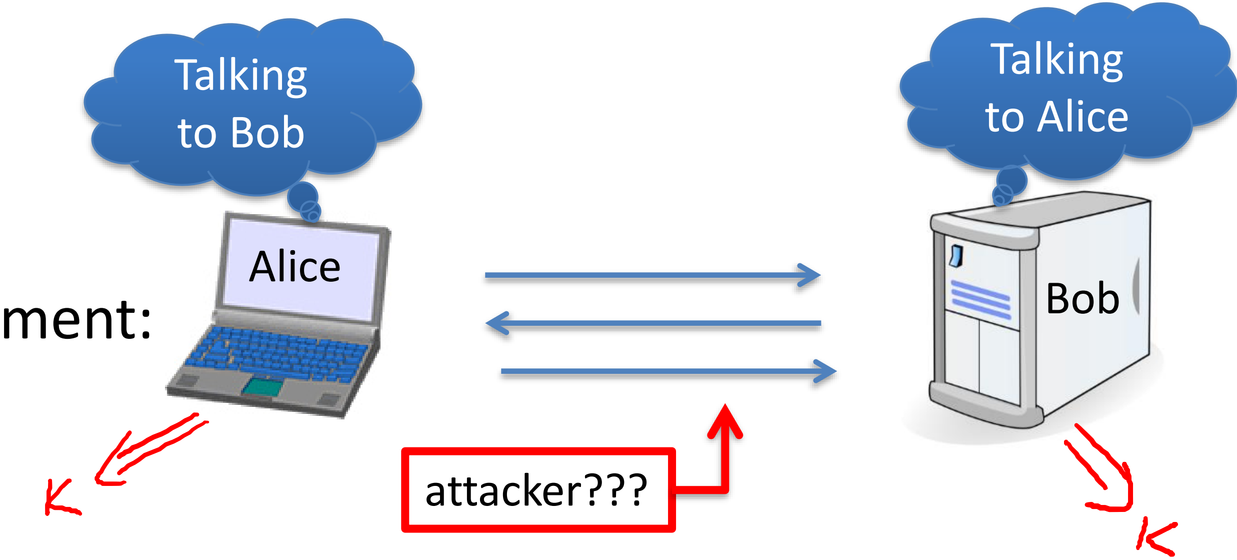
LINE Encryption

- **LINE** employs various encryption technologies to protect user information.
- Two Layered Encryption, In addition to transport layer encryption, which is used to protect traffic between LINE clients and servers.
- **LINE** apply **Letter Sealing encryption** to supported ONLY message types and voice/video calls.
- Check your own **LINE!**

Crypto core

1.Secret key establishment:

安全的密鑰建立



Talking to Bob

Alice

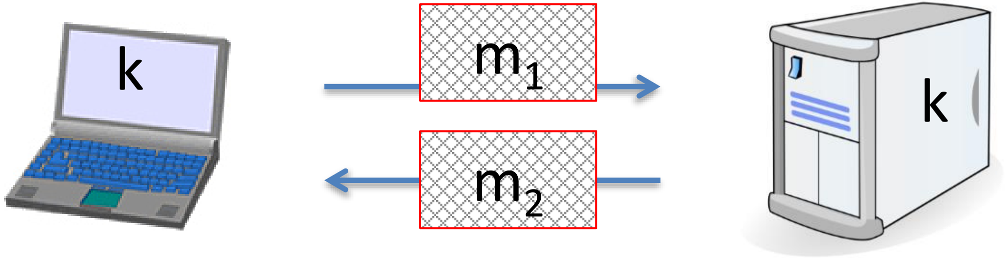
Talking to Alice

Bob

attacker???

2.Secure communication:

安全地交換訊息



k

m_1

m_2

k

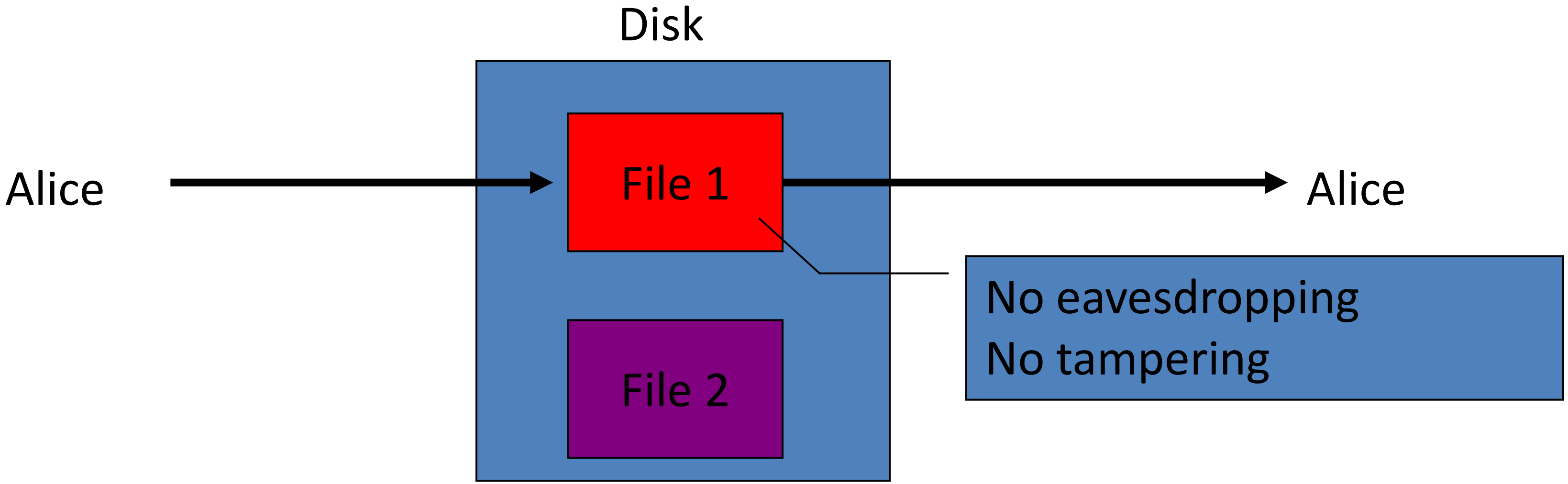
confidentiality and integrity

Secure Sockets Layer / TLS

Two main parts:

1. Handshake Protocol: **Establish shared secret key using public-key cryptography** (2nd part of course)
2. Record Layer: **Transmit data using shared secret key** ensure confidentiality and integrity (1st part of course)

2. Data at Rest-Protected Files on Disk



密碼工程的另一個應用是保護磁碟上的文件 Data at Rest

Analogous to secure communication:

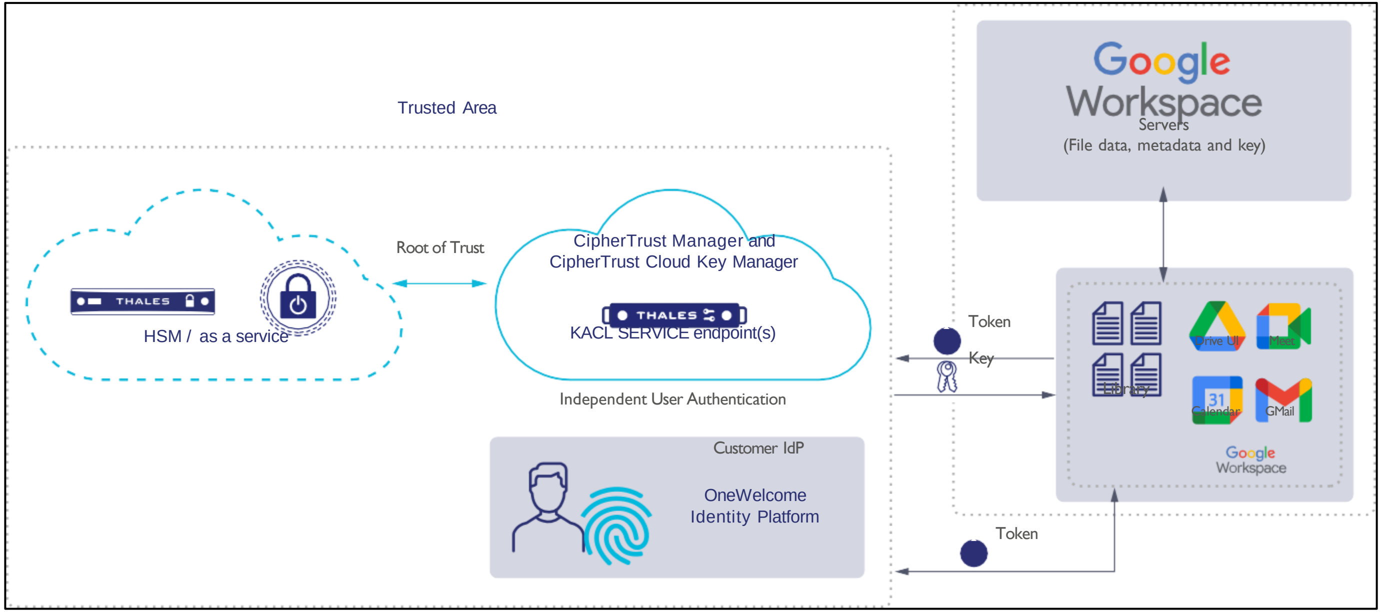
Alice today sends a message to Alice tomorrow

Using Public Cloud to Back Up!

- Xu Yanjun, an officer in China's Ministry of State Security (MSS), was caught with the help of an **iCloud backup of his iPhone.**
- A federal jury convicted Yanjun on November 5, 2021, of industrial espionage against GE Aviation and Honeywell.

Xu Yanjun Passport Seized from Xu Heng

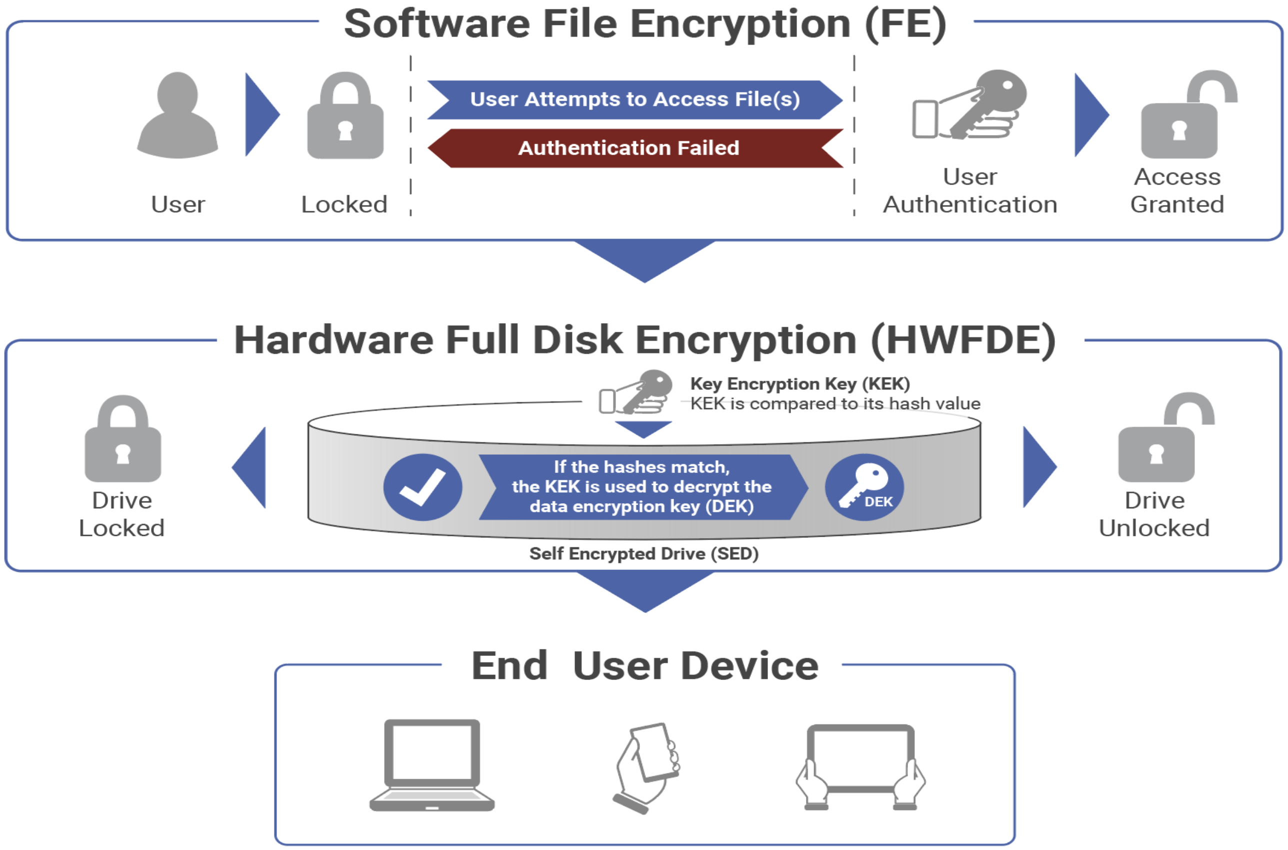
Security Solutions for Google Workspace Enhanced Privacy and Confidentiality using Google Workspace Client-side encryption



Layer 2
Software Encryption

Layer 1
Hardware Encryption

Layer 0
Physical Possession



Software File Encryption (FE)

User

Locked

User Attempts to Access File(s)

Authentication Failed

User Authentication

Access Granted

Hardware Full Disk Encryption (HWFDE)

Drive Locked

Key Encryption Key (KEK)
KEK is compared to its hash value

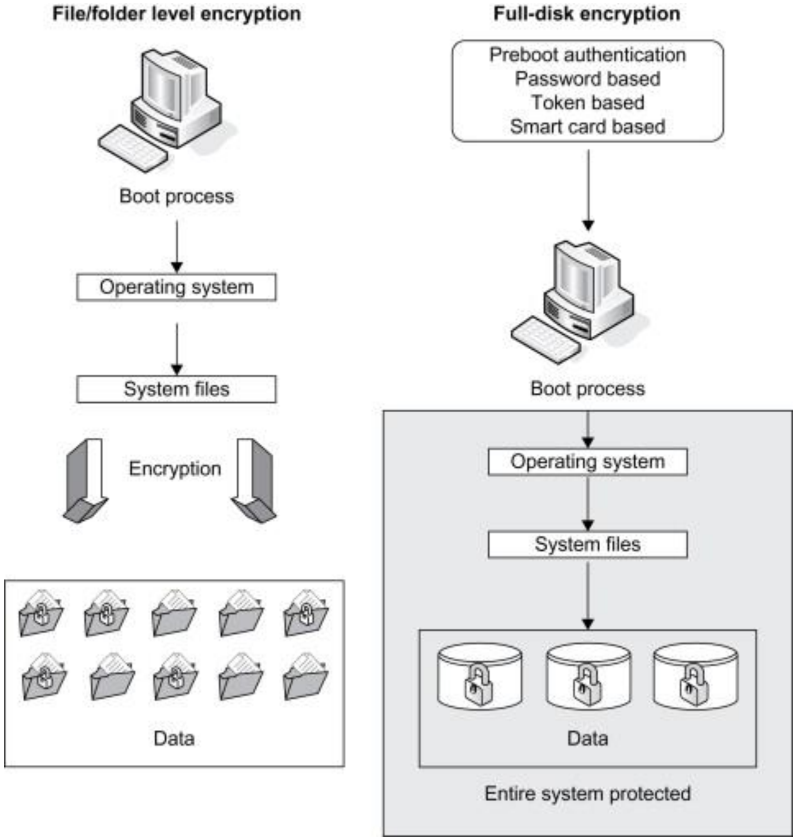
If the hashes match, the KEK is used to decrypt the data encryption key (DEK)

DEK

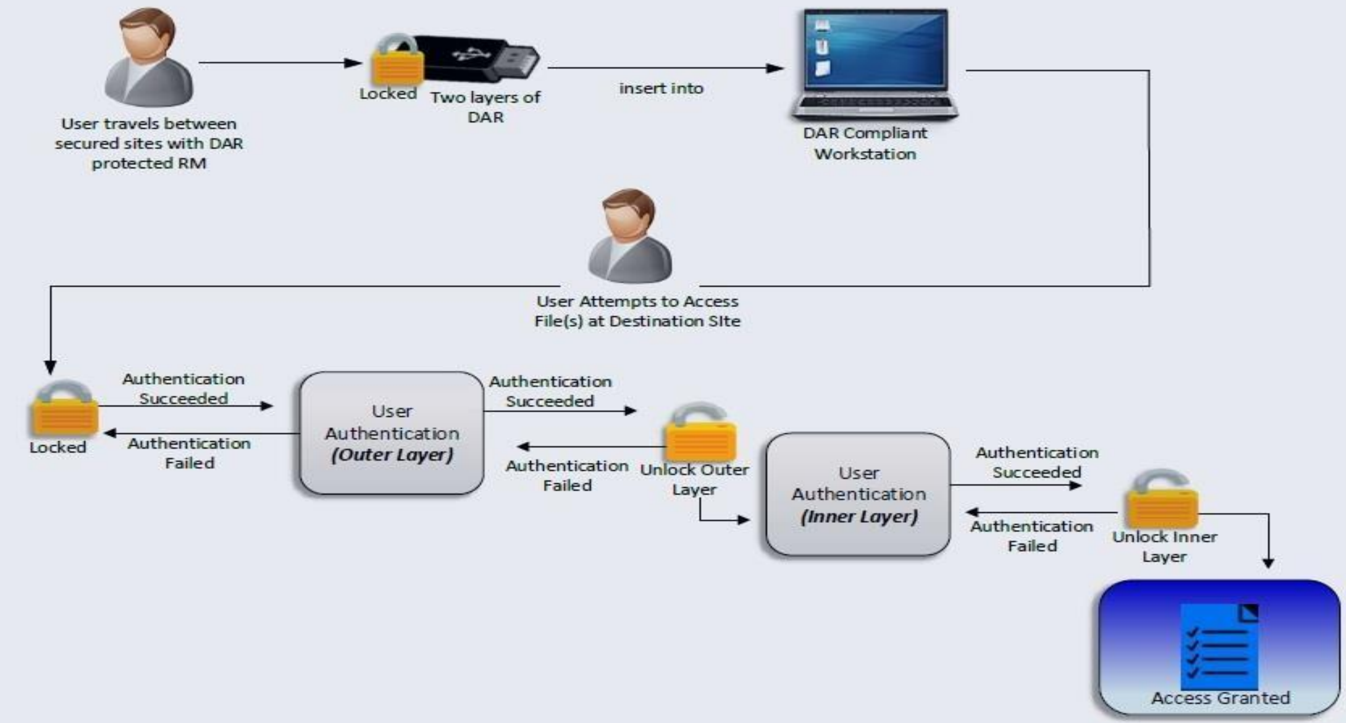
Self Encrypted Drive (SED)

Drive Unlocked

End User Device


File/folder level encryption
Boot process
Operating system
System files
Encryption
Data
Full-disk encryption
Preboot authentication
Password based
Token based
Smart card based
Boot process
Operating system
System files
Data
Entire system protected

Removable Media (RM)

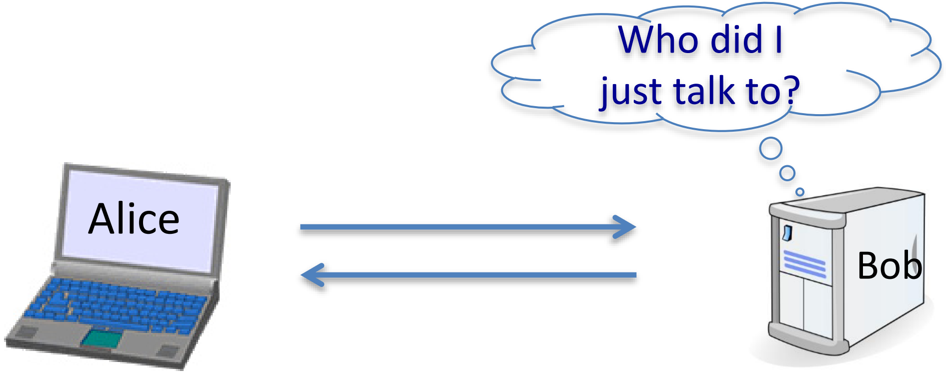


3.But Crypto Can Do Much More

- **Digital signatures**

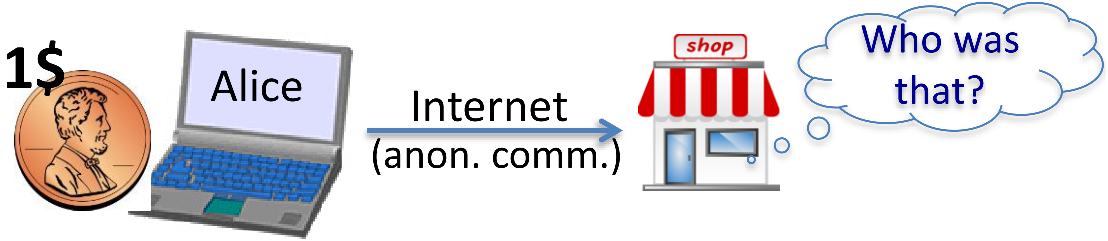
因為如果攻擊者剛剛從我那裡獲得一個簽名的文檔，
他可以剪切並粘貼我的簽名到其他一些我可能不想簽署的文檔

- **Anonymous communication**



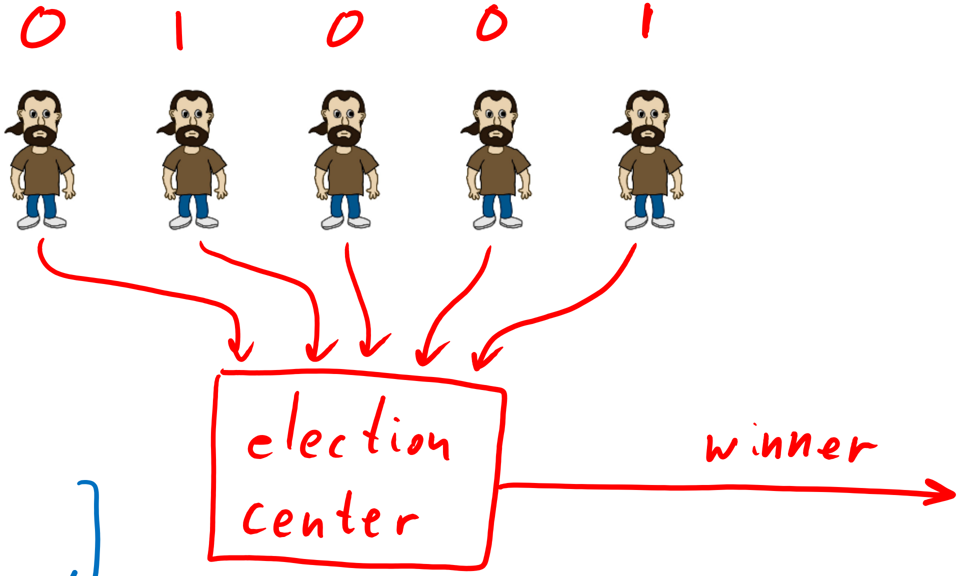
But crypto can do much more

- **Digital signatures**
- **Anonymous communication (Tor)**
- **Anonymous digital cash (Blockchain)**
    - Can I spend a “digital coin” without anyone knowing who I am?
    - How to prevent double spending?



Protocols

- Elections
- Private auctions

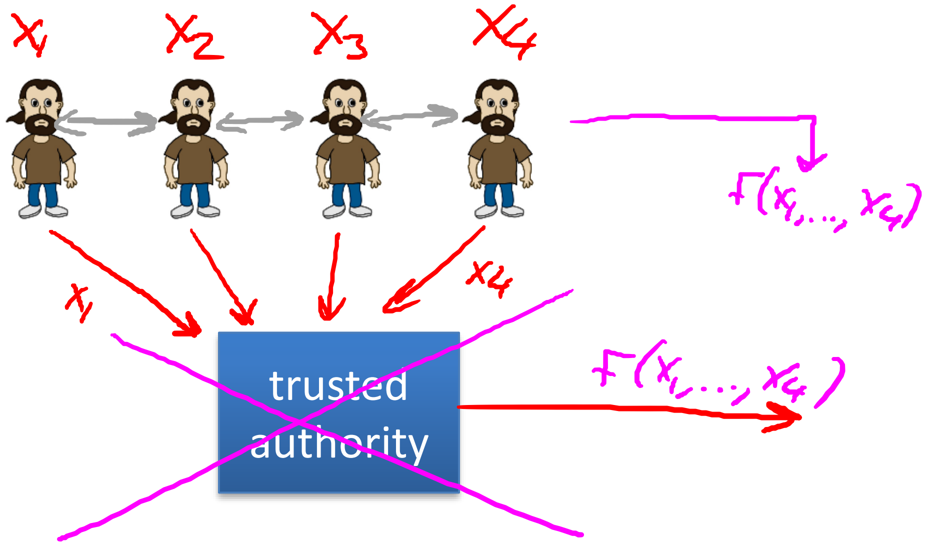


winner = MAJ[votes]

auction winner = [highest bidder, pays 2nd highest bid]

Protocols

- **Elections**
- **VICKEY** Private auctions

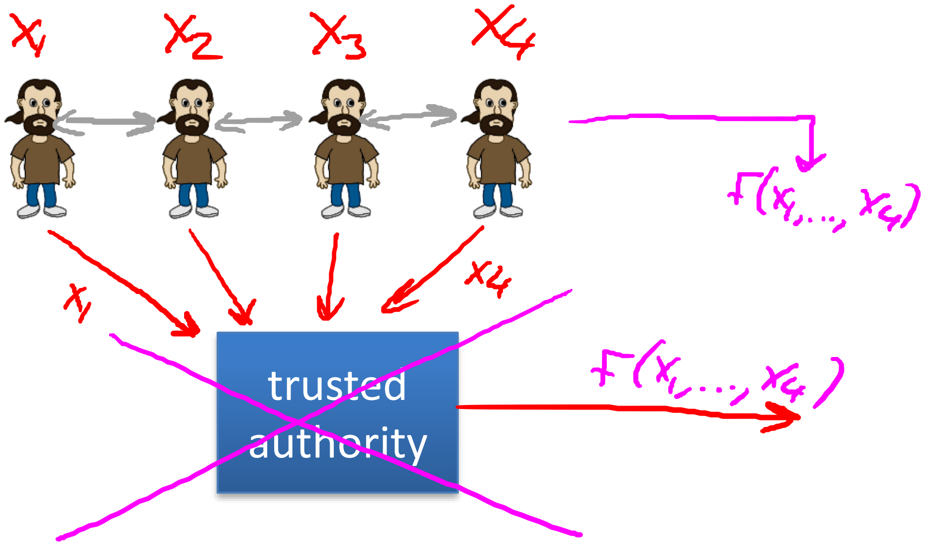


Goal: compute $f(x_1, x_2, x_3, x_4)$

“Thm:” Anything that can done with trusted auth, can also be done without secure multi-party computation.

Protocols

- Elections
- **VICKEY** Private auctions



Goal: compute $f(x_1, x_2, x_3, x_4)$

“Thm:” Anything that can done with trusted auth, can also be done without secure multi-party computation.

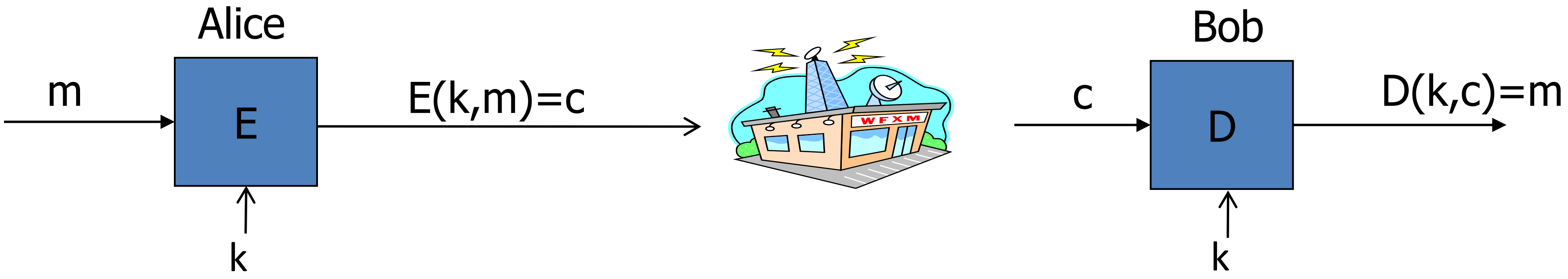
Vickery auction

- **維克里拍賣的優點就在於「講真話」是最優策略**，所有的投標人都會顯示他對拍賣物的真實評價。這點其實很好理解，因為中標者沒有價格影響力。
  舉例說明：一個物件，**最後的中標人的出標價是10元**，未中標的最高價格是8元，按照維克里拍賣，中標人只需要出8元。中標者的成交價不是由自己而是由出價最高的未中標者決定的，**出標的高低只決定輸贏而不決定具體價格，只有用真實價格競拍才能保證最後中標的那個價格是自己想要支付的價格，所以參加維克里拍賣的人沒有扭曲自己價格的激勵**。維克里拍賣的最著名的衍生就是eBay的競價代理（Proxy bidding）拍賣，除了不是密封式拍賣外，它規則基本和維克里拍賣相同，自然而然的擁有維克里拍賣的優點。
- 當然，任何經濟機制的設計都無法做到完美，即使是維克里拍賣這樣在學術界頗有好評的設計也有以下的幾個缺點：
  - 未中標者會結成同盟
  - 很難避免一個投標人用多個身份投標的行為
  - 難以避免買方串謀，互相揭示價格，以降低
  - 賣方會僱傭「托兒」來抬高價格
  - 不一定最大化賣方利潤

Outline

1. Cryptography Engineering Application
2. Symmetry Key and Public Key Encryption
3. Open security vs Obscurity for Encryption Algorithm
4. Privacy from Anonymity

Building Block: Sym. Encryption



E, D: cipher   k: secret key (e.g. 128 bits)

m, c: plaintext, ciphertext

Encryption algorithm is publicly known

- Never use a proprietary cipher?

The Binary Version of Sym. Encryption

Plaintext space = Ciphertext space = Key space = $\{0,1\}^n$

Key is chosen randomly

For example: **Alice encrypted**

• Plaintext is	11011011	m
• Key(Random) is	01101001	k
• Then ciphertext is	10110010	c

Building Block: Sym. Encryption

Plaintext space = Ciphertext space = Key space = $\{0,1\}^n$

Key is chosen randomly

For example: **Bob decrypted**

- Ciphertext is 10110010 c
- Key(Random) is 01101001 k
- Plaintext is 11011011 m

Bit Operators

- Bit AND

  $0 \wedge 0 = 0$ $0 \wedge 1 = 0$ $1 \wedge 0 = 0$ $1 \wedge 1 = 1$

- Bit OR

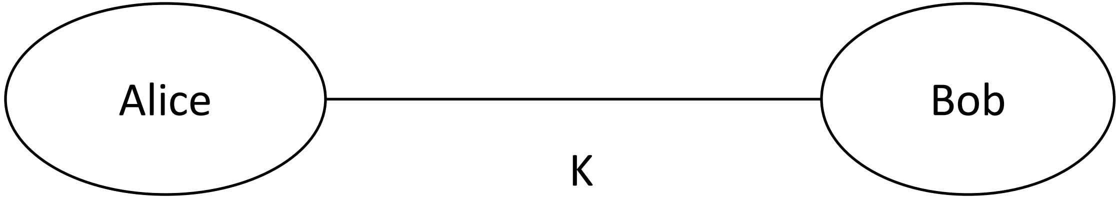
  $0 \vee 0 = 0$ $0 \vee 1 = 1$ $1 \vee 0 = 1$ $1 \vee 1 = 1$

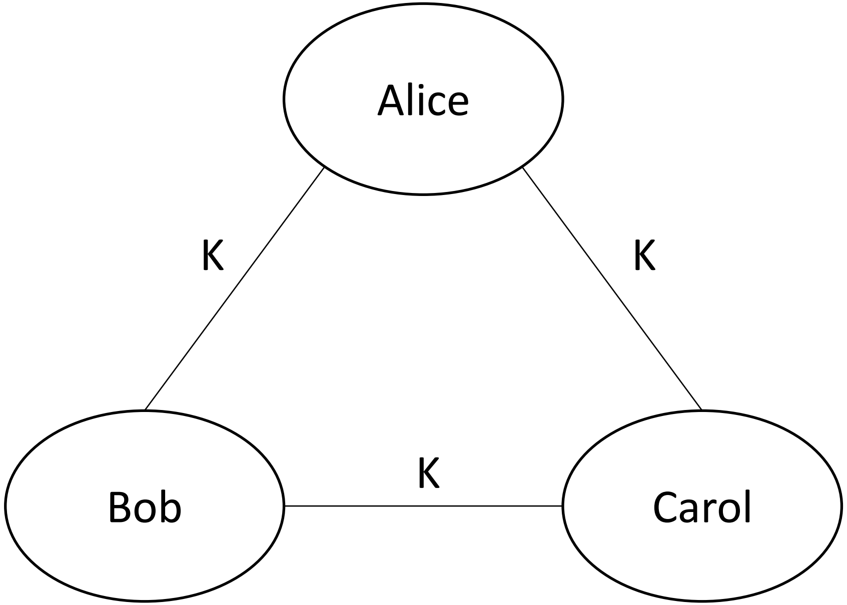
- Addition mod 2 (also known as Bit XOR)

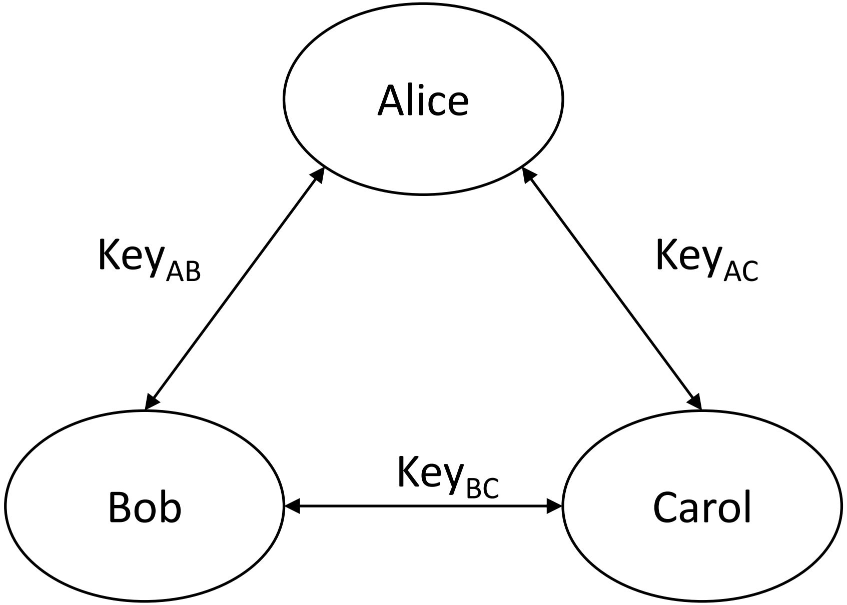
  $0 \oplus 0 = 0$ $0 \oplus 1 = 1$ $1 \oplus 0 = 1$ $1 \oplus 1 = 0$

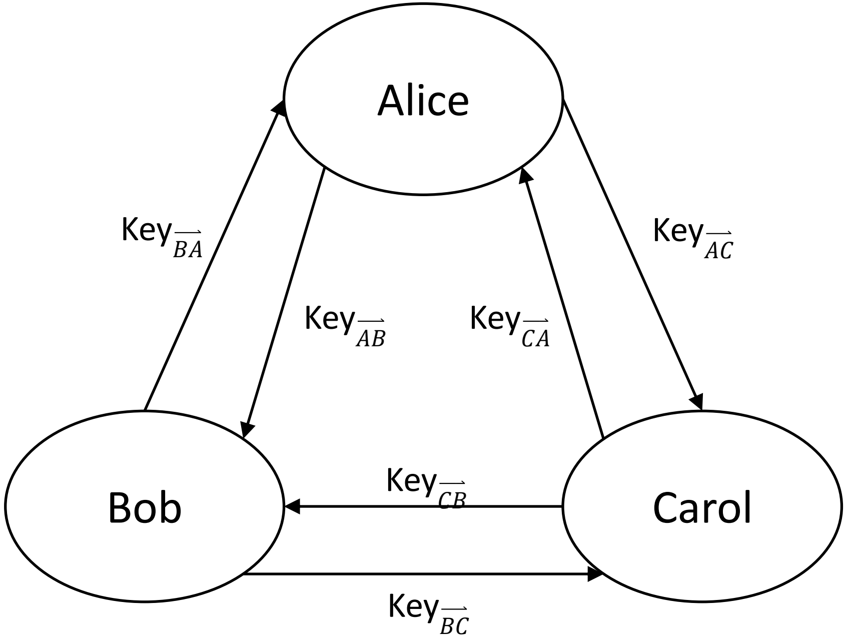
- Can we use operators other than Bit XOR for binary version of One-Time Pad?

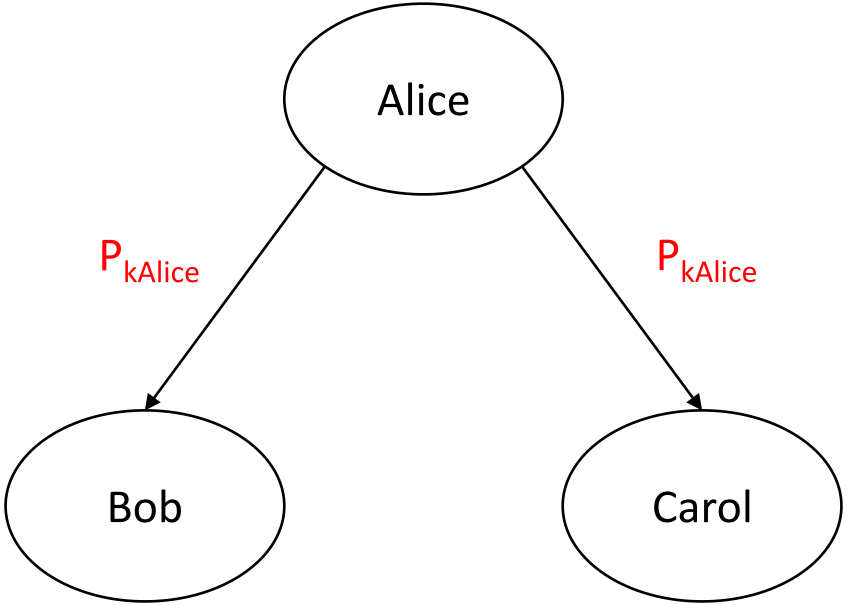
- Before the mid-1970s, all cipher systems used symmetric key algorithms, in which the same cryptographic key is used with the underlying algorithm by both the sender and the recipient, who must both keep it secret. Of necessity, the key in every such system had to be exchanged between the communicating parties in some secure way prior to any use of the system – for instance, via a secure channel.
- This requirement is never trivial and very rapidly becomes unmanageable as the number of participants increases, or when secure channels are not available, or when, (as is sensible cryptographic practice), keys are frequently changed.
- In particular, if messages are meant to be secure from other users, a separate key is required for each possible pair of users.
- By contrast, in a public key system, the public keys can be disseminated widely and openly, and only the corresponding private keys need be kept secret by its owner.

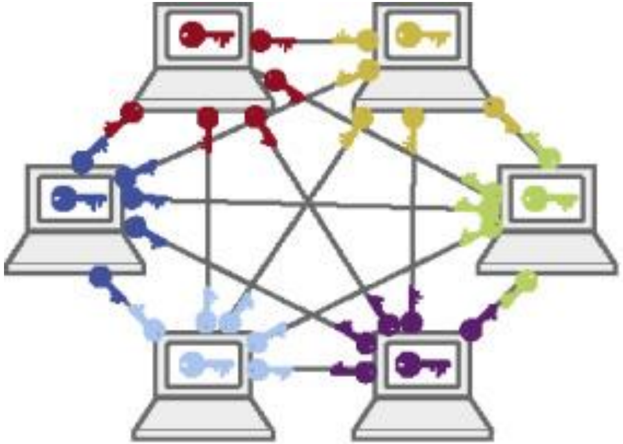

Alice
Bob
K


Alice
K
K
Bob
K
Carol


Alice
Key$_{AB}$
Key$_{AC}$
Key$_{BC}$
Bob
Carol

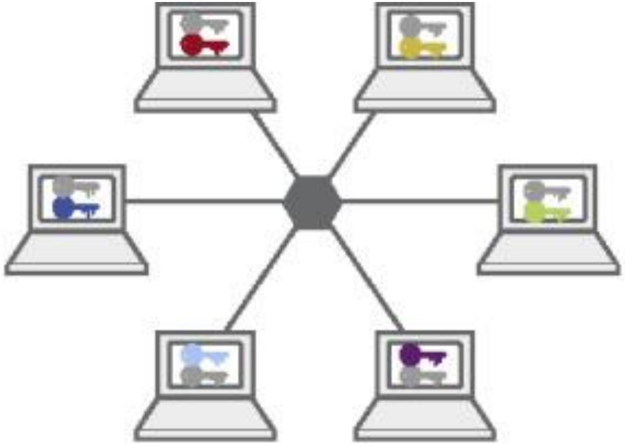

Alice
Key$_{\overrightarrow{BA}}$
Key$_{\overrightarrow{AC}}$
Key$_{\overrightarrow{AB}}$
Key$_{\overrightarrow{CA}}$
Bob
Key$_{\overrightarrow{CB}}$
Carol
Key$_{\overrightarrow{BC}}$


Alice
P_{kAlice}
P_{kAlice}
Bob
Carol



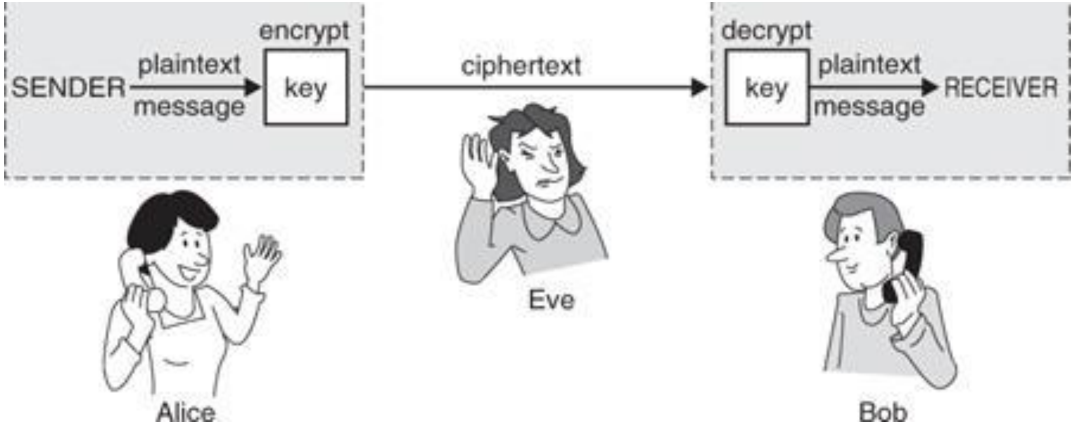
SYMMETRIC

Symmetric cryptography has an equation of $\frac{nxn-1}{2}$ for the number of keys needed. In a situtaion with 1000 users, that would mean **499,500 keys.**



ASYMMETRIC

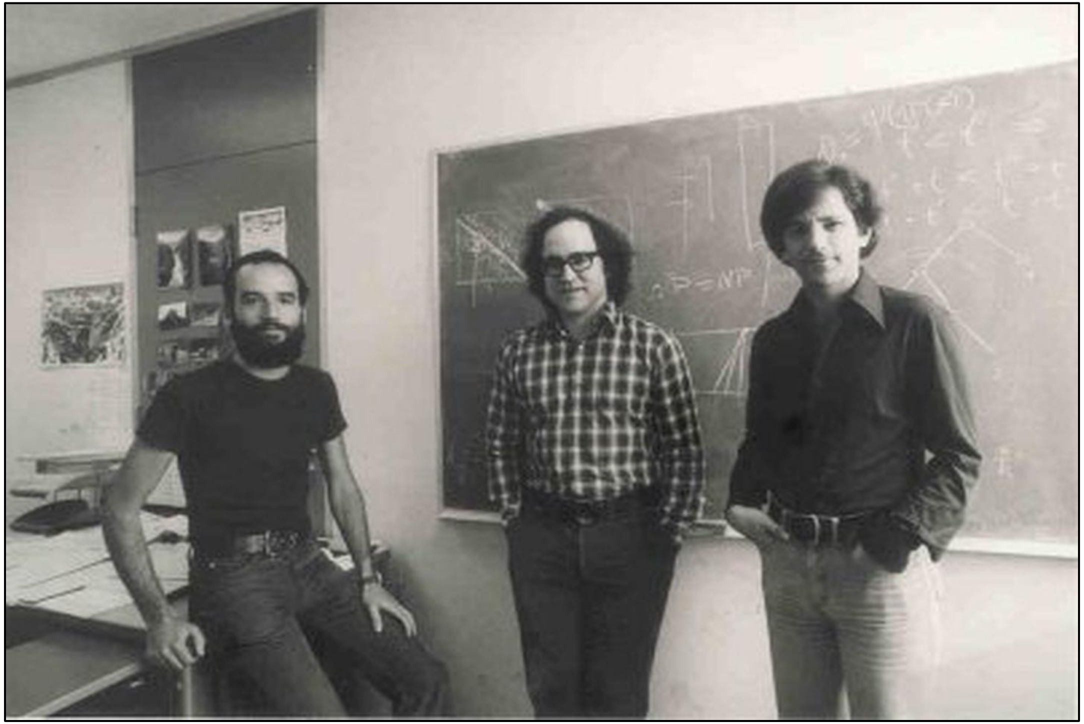
Asymmetric cryptography, using key pairs for each of its users,has n as the number of key pairs needed. In a situation with 1000 users, that would mean **1000 key pairs.**



Alice, Bob and Eve diagram used to explain Cryptography

The birth of Alice and Bob (February 1978)

- But what we are going to talk about today is not the RSA cryptosystem, but Alice and Bob. Since the paper "**A Method for Realizing Digital Signature and Public Key Cryptography**" was published in February 1978, February 1978 became the **birthday of the protagonists of this answer, Alice and Bob**. Why do you say this way? Because Alice and Bob are used to describe the scheme for the first time in this paper.



From left to right are Shamir, Rivest, and Adleman. Image source: https://cryptologicfoundation.org/

Non-Secret Encryption

- **In 1970, James H. Ellis, a British cryptographer** at the UK Government Communications Headquarters (GCHQ), conceived of the possibility of "non-secret encryption", (now called public key cryptography), but could see no way to implement it. In 1973, his colleague **Clifford Cocks** implemented what has become known as the RSA encryption algorithm, giving a practical method of "non-secret encryption", and in 1974 another GCHQ mathematician and cryptographer, **Malcolm J. Williamson**, 1968 International Mathematical Olympiad (IMO) in Moscow developed what is now known as Diffie–Hellman key exchange. **The scheme was also passed to the USA's National Security Agency.**
- **Both organizations had a military focus and only limited computing power was available in any case; the potential of public key cryptography remained unrealized by either organization**:

- I judged it most important for military use ... if you can share your key rapidly and electronically, you have a major advantage over your opponent. Only at the end of the evolution from Berners-Lee designing an open internet architecture for CERN, its adaptation and adoption for the Arpanet ... did public key cryptography realize its full potential.—Ralph Benjamin
- These discoveries were not publicly acknowledged for 27 years, until the research was declassified by the British government in 1997.

GCHQ

PUBLIC-KEY CRYPTOGRAPHY:

a way for Alice and Bob to agree on a secret key
without ever meeting and
through messages that are completely public!

Whit Diffie and Marty Hellman,
New Directions in Cryptography,
1976

Clifford Cocks and Malcolm Williamson,
secret work in the British GCHQ,
1973-74, revealed in 1997

A Public Key Encryption Example (RSA)

Choose **p = 3** and **q = 11** (secret)

Compute n = p * q = 3 * 11 = 33

Compute ϕ(n) = (p - 1) * (q - 1) = 2 * 10 = 20

Choose e such that 1 < e < ϕ(n) and e and n are coprime.

Let e = 7

Compute a value for d such that (d * e) mod ϕ(n) = 1.

One solution is **d = 3** [(3 * 7) mod 20 = 1]

Bob chooses two primes p,q and compute n=pq

Bob chooses e with gcd(e,(p-1)(q-1))= gcd(e, ϕ(n))=1

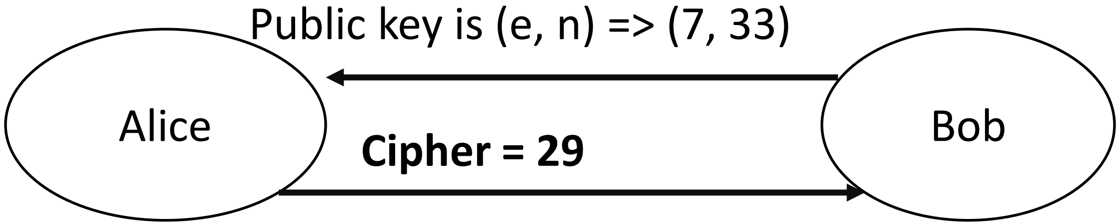
Bob solves ed≡1 (mod ϕ (n))

Bob makes (e,n) public and (p,q,d) secret

Public key is (e, n) => (7, 33)

Private key is (d, n) => (3, 33)

The decryption of $c = 29$ is $m = 29^3$ mod $33 = 2$



Public key is (e, n) => (7, 33)

Cipher = 29

The decryption of $c = 29$ with private key 3

is $m = 29^3$ mod $33 = 2$ (Plaintext)

Bob decrypts by computing m≡C^d (mod n)

The encryption of $m = 2$ is $c = \boldsymbol{2^7}$ mod $33 = 29$

Alice encrypts m as c = m^e (mod n)

Proof for the RSA Algorithm

- $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\phi(n)} \equiv M^1\ M^{k\phi(n)}\ (\mathrm{mod}\ n)$
  $\equiv M^1\ \mathbf{M}^{\boldsymbol{\phi}\mathbf{(n)}\ k}\ (\mathrm{mod}\ n)$

- By Euler's theorem $M^{\phi(n)}\ (\mathrm{mod}\ n)=1$

$$ed \equiv 1\ (\mathrm{mod}\ \phi\ (n)) \quad \Rightarrow \quad ed=1+k\phi(n)$$

- p=885320963, q=238855417,
- n=p · q=211463707796206571
- Let e=9007, ∴ d=116402471153538991
- M="cat"=30120, C=113535859035722866

NEW CRYPTOGRAPHIC STANDARDS for PQC RELEASED WITH TIMELINES (SEPTEMBER 2022)

Public-key

CRYSTALS-Dilithium

CRYSTALS-Kyber

Symmetric-key

Advanced Encryption Standard (AES)

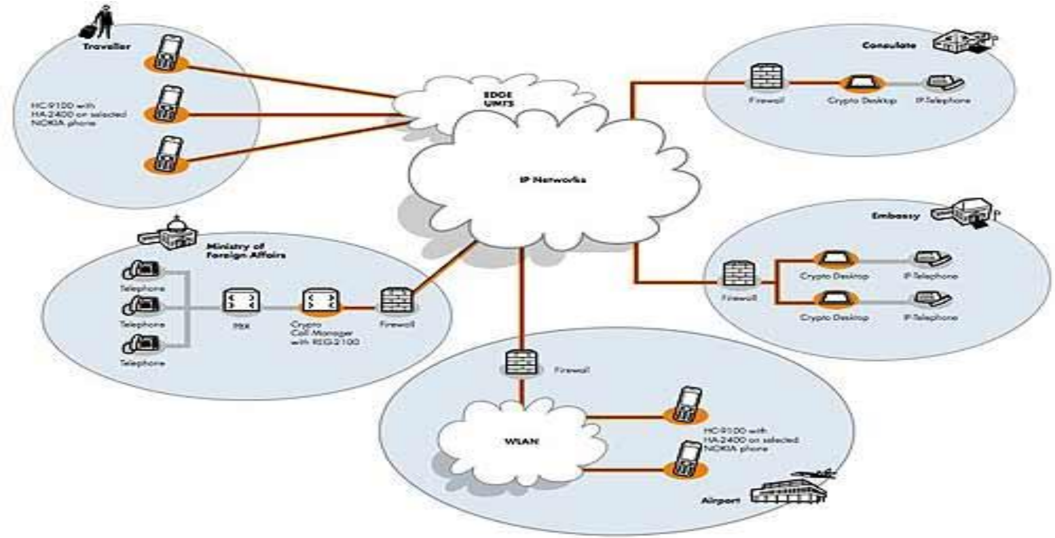
Secure Hash Algorithm (SHA)

Software and Firmware Updates

Xtended Merkle Signature Scheme (XMSS)

Leighton-Micali Signature (LMS)

The Real World is more complicated than theoretical models



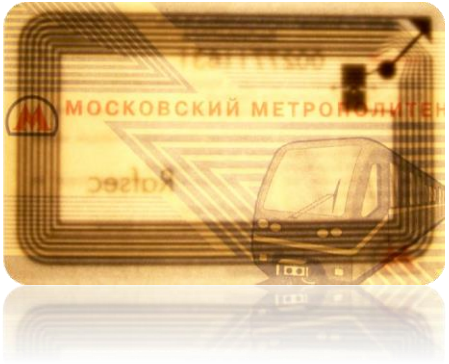
Outline

1. Cryptography Engineering Application
2. Symmetry Key and Public Key Encryption
3. Open vs Obscurity for Encryption Algorithm
4. Privacy from Anonymity

Open Algorithm- Kerckhoffs Principle

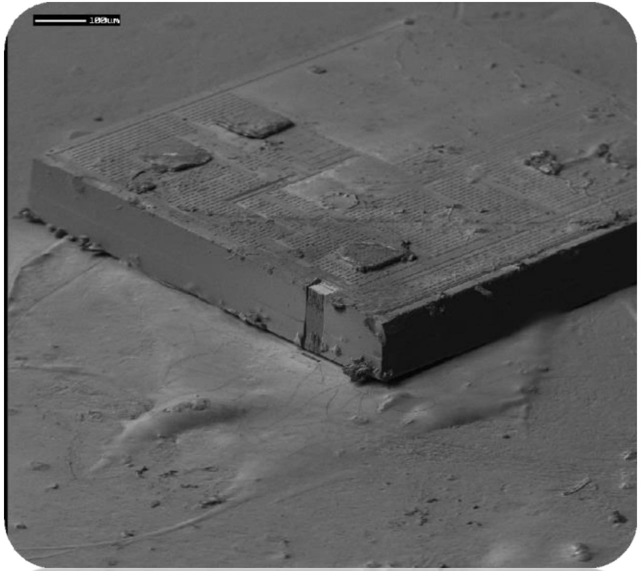
- 密碼學理上的柯克霍夫原則（Kerckhoffs’s Principle，也稱為柯克霍夫假說、公理、或定律）：即使密碼系統的任何細節已為人悉知，只要key，又稱金鑰或密鑰未洩漏，它也應是安全的。
- 它和使用隱密(Obscurity)的設計、實作、或其他等等來提供加密的隱晦式安全想法相對。
- 依據柯克霍夫原則，大多數民間加密都使用公開的演算法。 但相對地，用於政府或國防的密碼密式通常是不公開的。

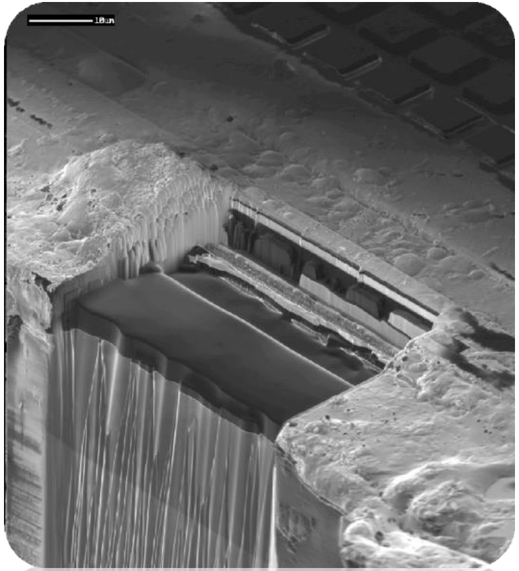
Obscurity Algorithm – 早期悠遊卡(Obscurity)設計案例



- Chemically extract chips:
  - Acetone
  - Fuming nitric acid
- Shortcut: buy blank chips

Mifare Classic RFID tag





Polishing

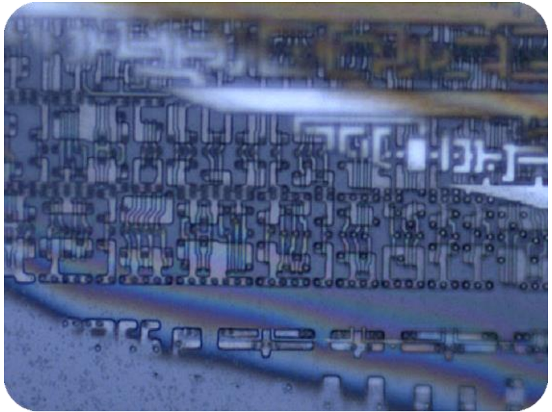
- Embed chip in plastic
  - Downside: chip is tilted

- Automated polishing with machine –or–

  manually with sand paper

- “On your kitchen table”

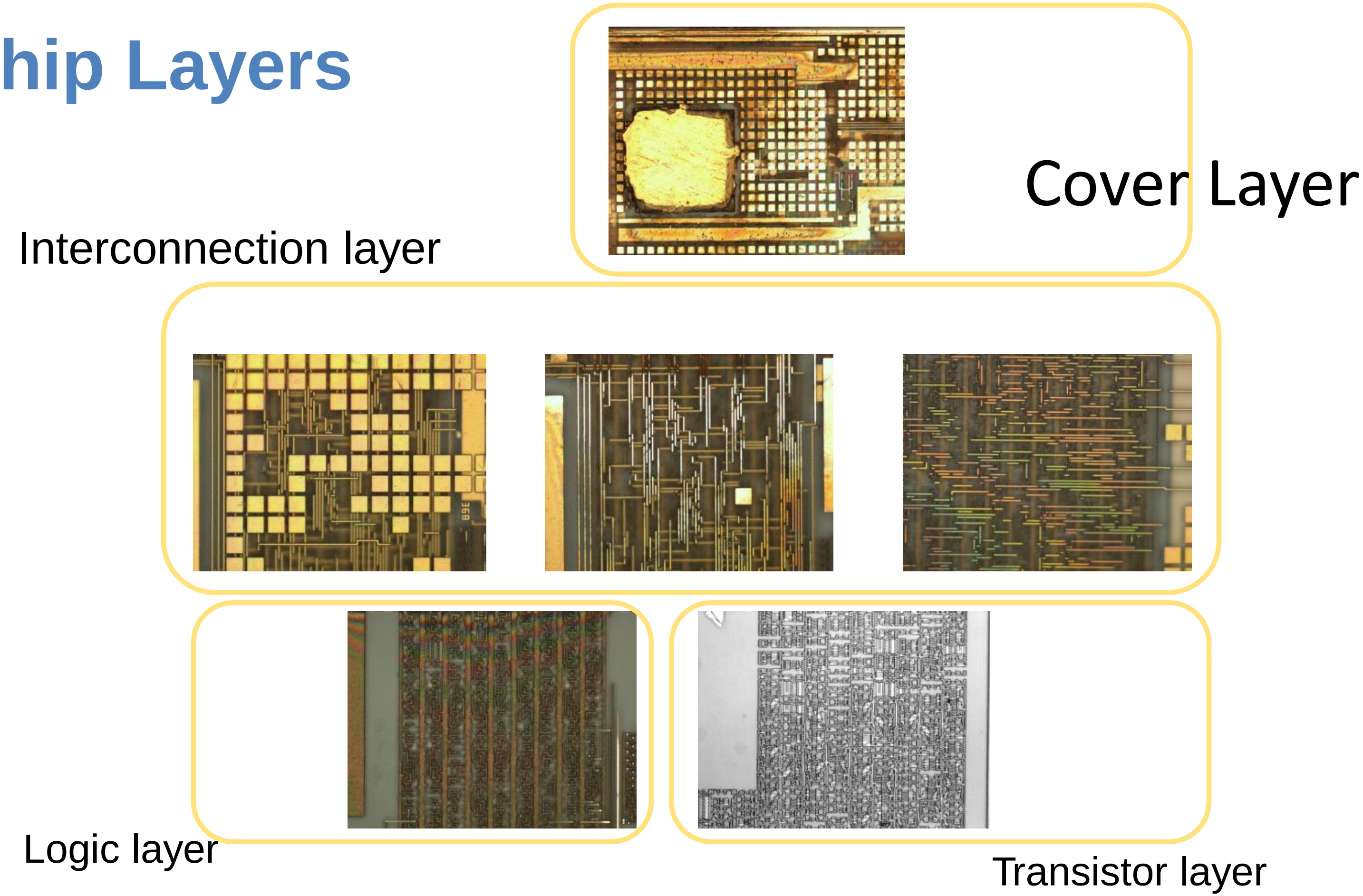
  -Starbug



Imaging Chip

- Simple optical microscope
  - 500x magnification
  - Camera 1 Mpixel
  - Costs < $1000, found in most labs
- Stitching images
  - Panorama software (hugin)
  - Each image ~100x100 µm
- Align different layers

Chip Layers



Cover Layer

Interconnection layer

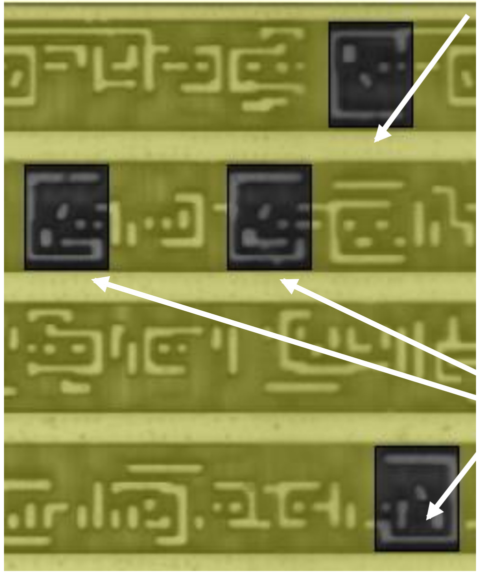
Logic layer

Transistor layer

Logic Gates Library

Chip has several thousand gates on logic layer

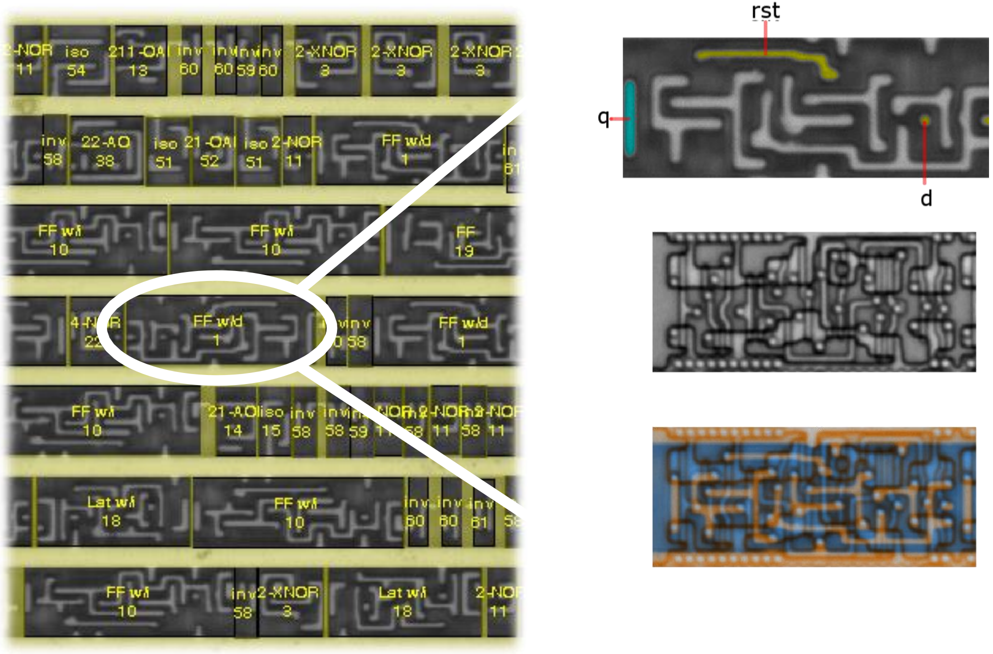
But only ~**70 different types detection can be automated through template matching**



Select

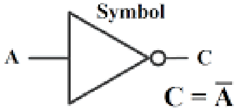
Detect

Logic Gates


2-NOR 11
iso 54
211-OAI 13
inv 60
inv 60
inv 59
inv 60
2-XNOR 3
2-XNOR 3
2-XNOR 3
inv 58
22-AO 38
iso 51
21-OAI 52
iso 51
2-NOR 11
FF w/d 1
in 61
FF w/ 10
FF w/ 10
FF 19
4-NOR 22
FF w/d 1
inv 58
FF w/d 1
FF w/ 10
21-AOI 14
iso 15
inv 58
inv 58
inv 59
2-NOR 11
inv 58
2-NOR 11
inv 58
2-NOR 11
Lat w/ 18
FF w/ 10
inv 60
inv 60
inv 61
FF w/ 10
inv 58
2-XNOR 3
Lat w/ 18
2-NOR 11
rst
q
d

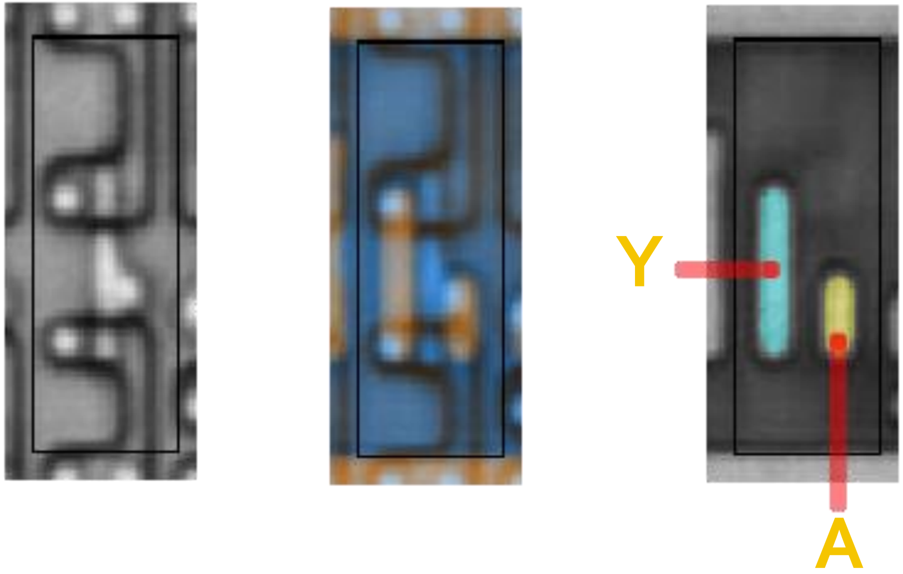
Logic Gates-Inverter

NOT Gate

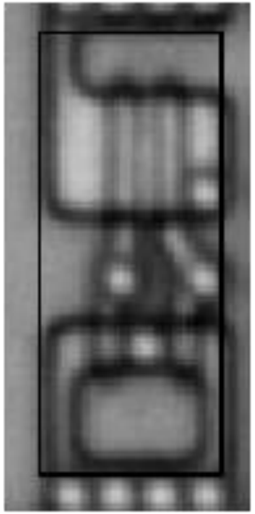


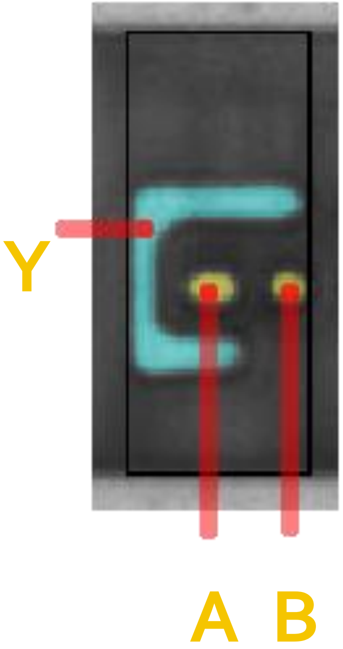
Truth Table

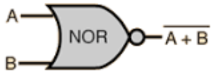
INPUT	OUTPUT
A	NOT A
0	1
1	0



Logic Gates - 2NOR



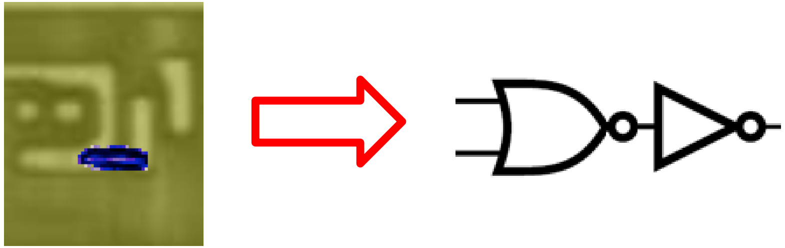




A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

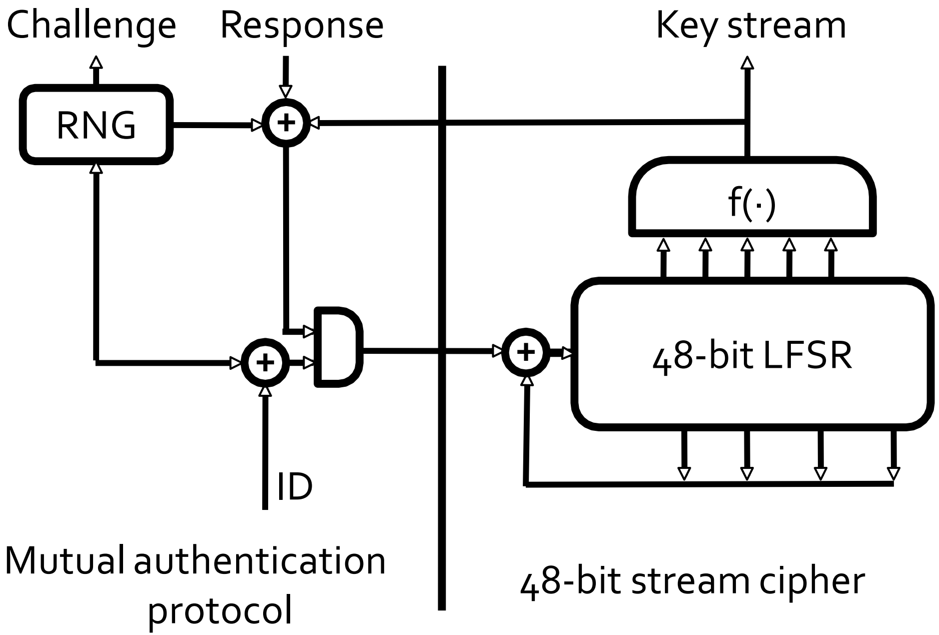
Logic Gates Interconnect

- Connections across all layers



- Traced 1500 (!) connections manually
  - Tedious, time consuming
  - Error-prone (but errors easily spottable)
  - Tracing completely automated by now

Mifare Crypto-1



Use Cases

Single use key: (one time key)

- Key is only used to encrypt one message
  - encrypted email: new key generated for every email

Multi use key: (many time key)

- Key used to encrypt multiple messages
  - encrypted files: ZR same key used to encrypt many files
- Need more machinery than for one-time key

Things to remember

Cryptography is:

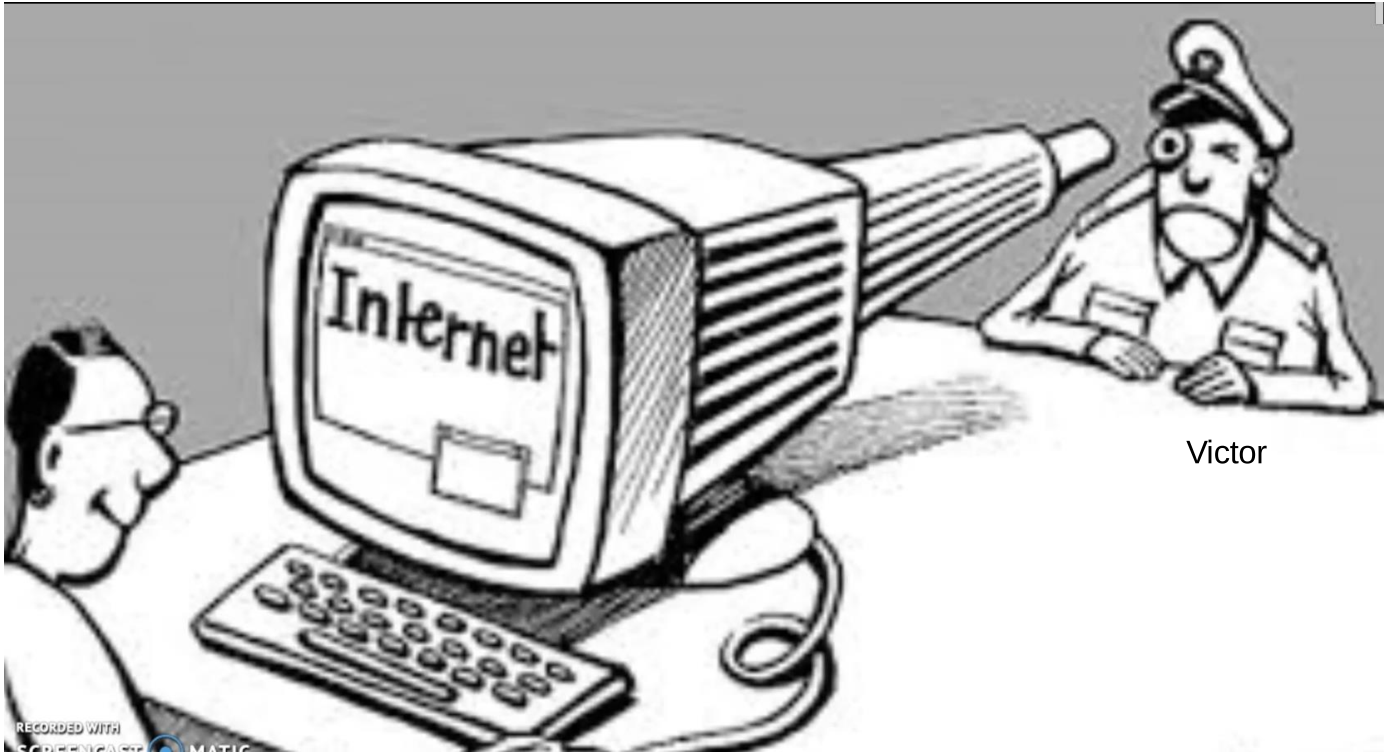
- A tremendous tool
- The basis for many security mechanisms

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
  - many many examples of broken ad-hoc designs

Outline

1. Cryptography Engineering Application
2. Symmetry Key and Public Key Encryption
3. Open security vs Obscurity for Encryption Algorithm
4. Privacy from Anonymity



Victor

Peggy

Zero-knowledge proof

- 所謂的Zero-knowledge proof
- 簡單來說就是:

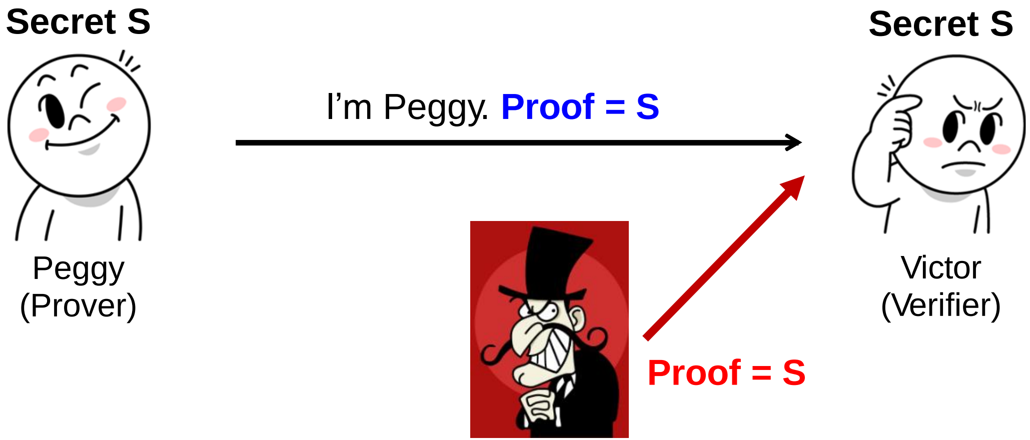
我今天有一個秘密
我要讓你知道我知道這個秘密
但是如果我把秘密告訴你了
那這就不是一個秘密了

那我要如何向你證明我知道這個秘密
卻又不能讓你知道這個秘密呢?

就是Zero-knowledge proof

Sending the secret directly

- Recall that Peggy wants to prove the possession of **S** (e.g., a password)
- This assumes a shared secret with a secure channel



Secret S

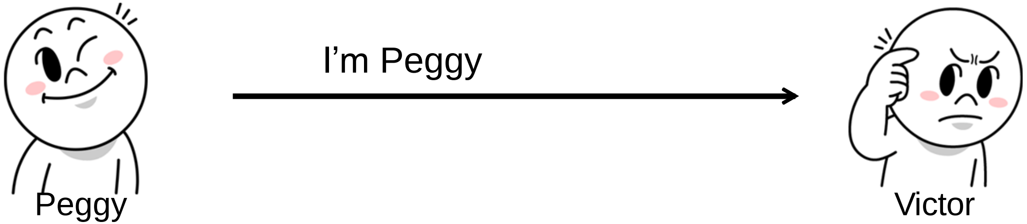
Peggy
(Prover)

I'm Peggy. **Proof = S**

Secret S

Victor
(Verifier)

Proof = S

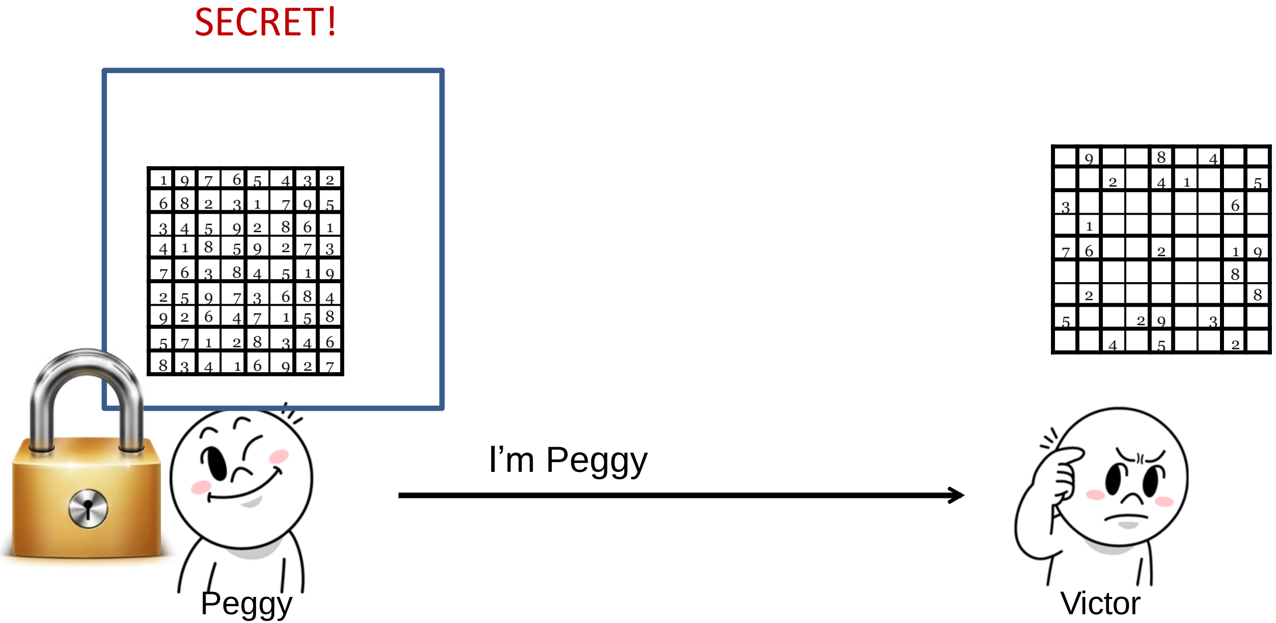


I’m Peggy

Peggy

Victor

- https://manishearth.github.io/sudoku-zkp/zkp.html



SECRET!

1	9	7	6	5	4	3	2	
6	8	2	3	1	7	9	5	
3	4	5	9	2	8	6	1	
4	1	8	5	9	2	7	3	
7	6	3	8	4	5	1	9	
2	5	9	7	3	6	8	4	
9	2	6	4	7	1	5	8	
5	7	1	2	8	3	4	6	
8	3	4	1	6	9	2	7	

Peggy

I’m Peggy

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

Victor

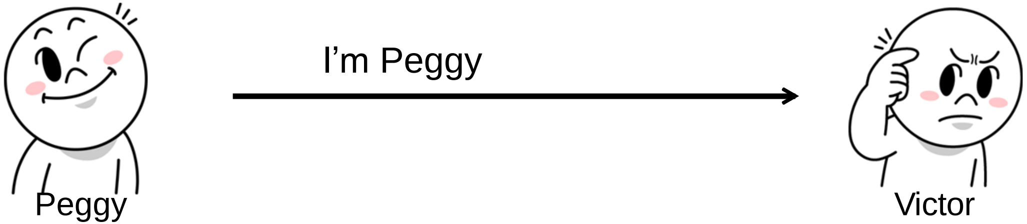
First, Peggy chooses a random permutation σ of {1,…,9} say σ(1)=2, σ(2)=8, σ(3)=6, σ(4)=5, σ(5)=4, σ(6)=9, σ(7)=1, σ(8)=7 and σ(9)=3.

1	9	7	6	5	4	3	2	
6	8	2	3	1	7	9	5	
3	4	5	9	2	8	6	1	
4	1	8	5	9	2	7	3	
7	6	3	8	4	5	1	9	
2	5	9	7	3	6	8	4	
9	2	6	4	7	1	5	8	
5	7	1	2	8	3	4	6	
8	3	4	1	6	9	2	7	

⇨

2	3	1	9	7	4	5	6	8
9	7	8	6	5	2	1	3	4
6	5	4	3	1	8	7	9	2
5	2	7	4	9	3	8	1	6
1	9	6	7	8	5	4	2	3
8	4	3	1	2	6	9	7	5
3	8	9	5	6	1	2	4	7
4	1	2	8	3	7	6	5	9
7	6	5	2	4	9	3	8	1

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	



2	3	1	9	7	4	5	6	8
9	7	8	6	5	2	1	3	4
6	5	4	3	1	8	7	9	2
5	2	7	4	9	3	8	1	6
1	9	6	7	8	5	4	2	3
8	4	3	1	2	6	9	7	5
3	8	9	5	6	1	2	4	7
4	1	2	8	3	7	6	5	9
7	6	5	2	4	9	3	8	1

First proof!

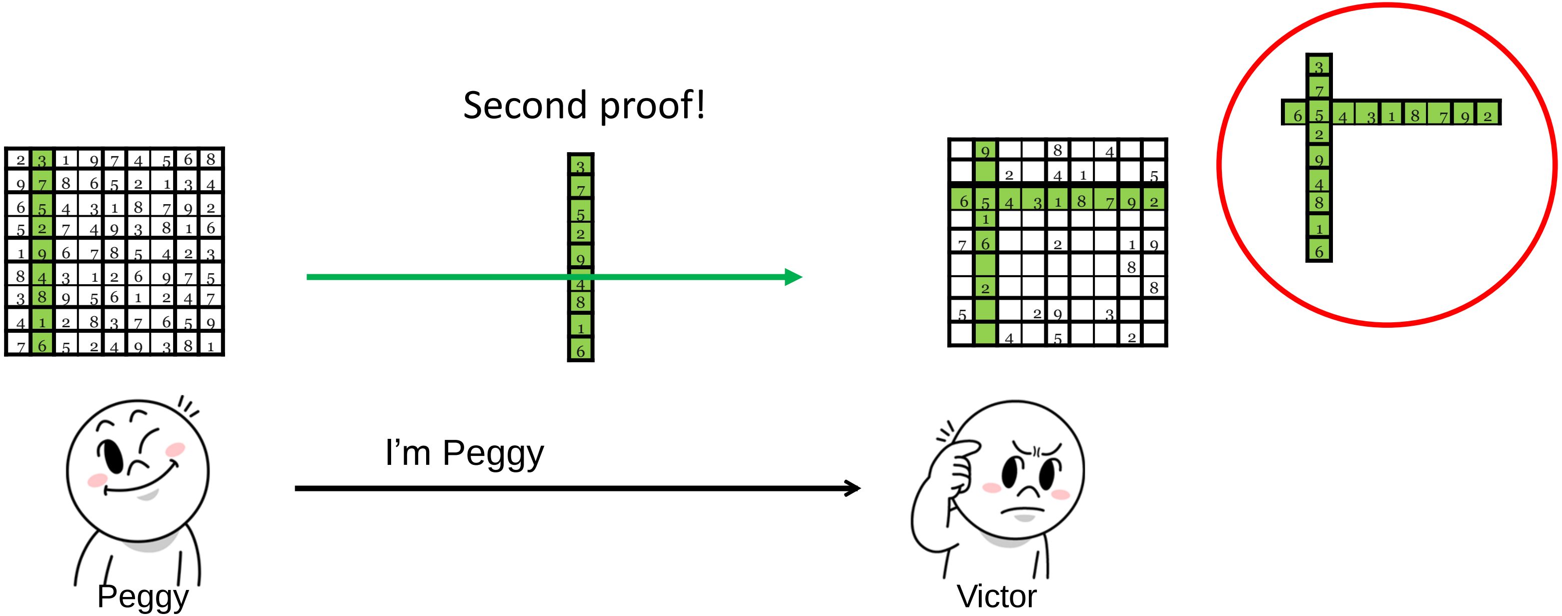
6	5	4	3	1	8	7	9	2

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

I'm Peggy

chooses a random permutation σ of {1,...,9} say σ(1)=2, σ(2)=8, σ(3)=6, σ(4)=5, σ(5)=4, σ(6)=9, σ(7)=1, σ(8)=7 and σ(9)=3.

Second proof!



2	3	1	9	7	4	5	6	8
9	7	8	6	5	2	1	3	4
6	5	4	3	1	8	7	9	2
5	2	7	4	9	3	8	1	6
1	9	6	7	8	5	4	2	3
8	4	3	1	2	6	9	7	5
3	8	9	5	6	1	2	4	7
4	1	2	8	3	7	6	5	9
7	6	5	2	4	9	3	8	1

	9			8		4		
		2		4	1			5
6	5	4	3	1	8	7	9	2
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

chooses a random permutation σ of {1,…,9} say σ(1)=2, σ(2)=8, σ(3)=6, σ(4)=5, σ(5)=4, σ(6)=9, σ(7)=1, σ(8)=7 and σ(9)=3.

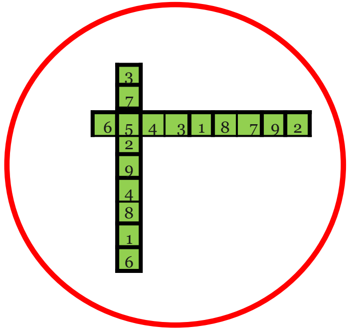
Choose a random permutation σ of {1,…,9} say σ(1)=2, σ(2)=8, σ(3)=6, σ(4)=5, σ(5)=4, σ(6)=9, σ(7)=1, σ(8)=7 and σ(9)=3.

2	3	1	9	7	4	5	6	8
9	7	8	6	5	2	1	3	4
6	5	4	3	1	8	7	9	2
5	2	7	4	9	3	8	1	6
1	9	6	7	8	5	4	2	3
8	4	3	1	2	6	9	7	5
3	8	9	5	6	1	2	4	7
4	1	2	8	3	7	6	5	9
7	6	5	2	4	9	3	8	1

Third proof!

3	8	9
4	1	2
7	6	5

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

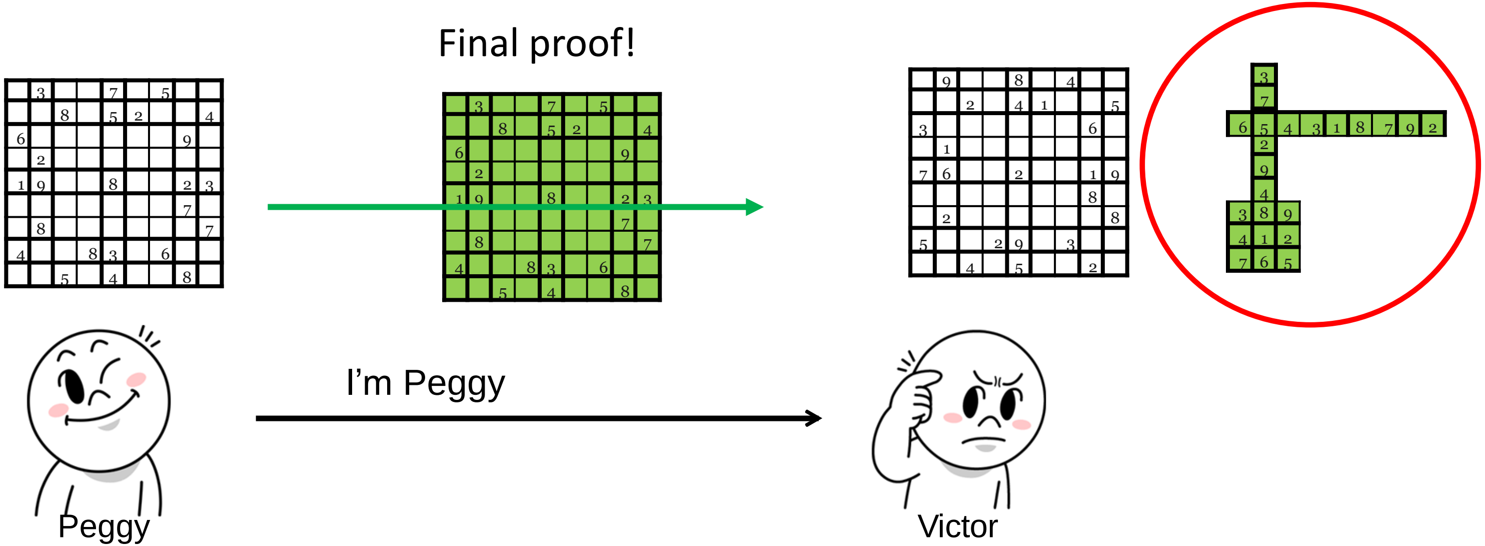


Peggy

I’m Peggy

Victor

Choose a random permutation σ of {1,…,9} say σ(1)=2, σ(2)=8, σ(3)=6, σ(4)=5, σ(5)=4, σ(6)=9, σ(7)=1, σ(8)=7 and σ(9)=3.



	3			7		5		
		8		5	2			4
6							9	
	2							
1	9			8			2	3
							7	
	8							7
4			8	3		6		
		5		4			8	

	3			7		5		
		8		5	2			4
6							9	
	2							
1	9			8			2	3
							7	
	8							7
4			8	3		6		
		5		4			8	

	9			8		4		
		2		4	1			5
3							6	
	1							
7	6			2			1	9
							8	
	2							8
5			2	9		3		
		4		5			2	

Cryptography Engineering is not just a rigorous science but an art.

A rigorous science

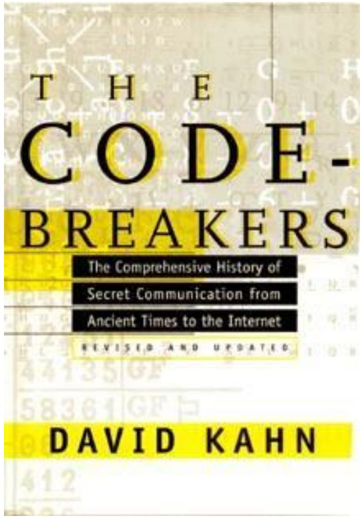
The three steps in cryptography:

- Precisely specify threat model
- Propose a construction
- Prove that breaking construction under threat mode will solve an underlying hard problem

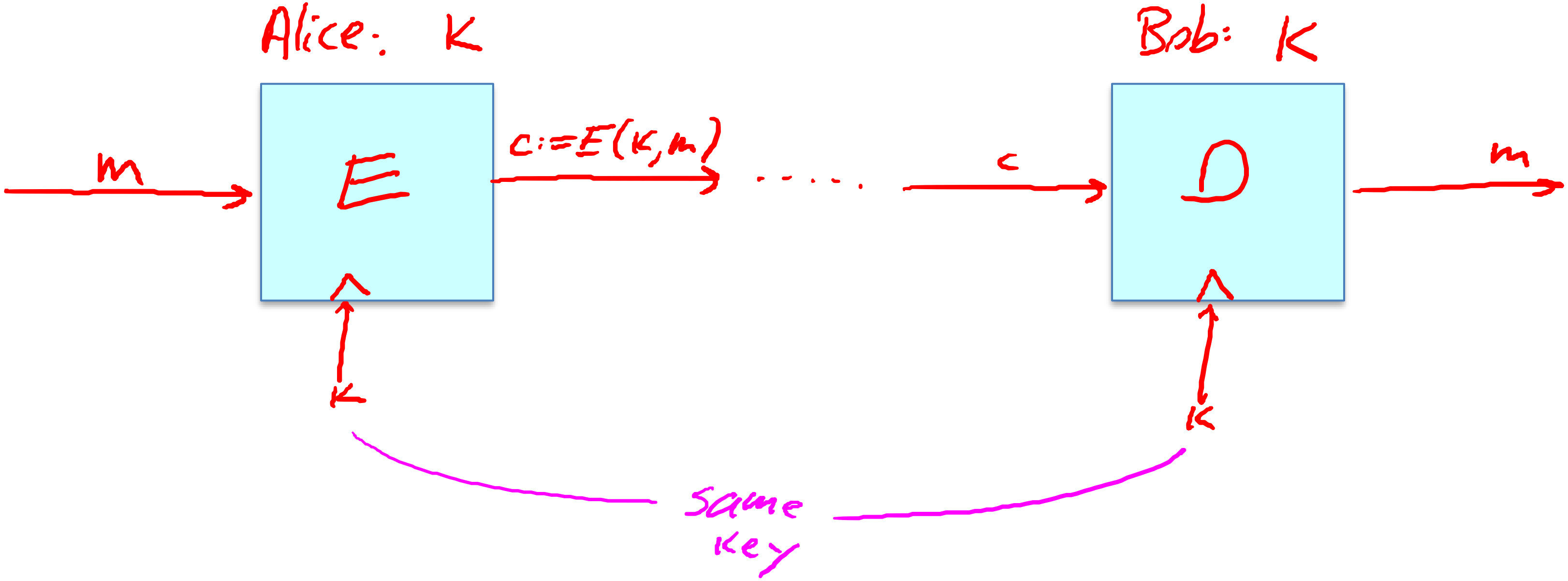
End of Segment

History

David Kahn, “The code breakers”   (1996)


THE
CODE-
BREAKERS
The Comprehensive History of
Secret Communication from
Ancient Times to the Internet
REVISED AND UPDATED
DAVID KAHN

Symmetric Ciphers



Few Historic Examples (all badly broken)

1. Substitution cipher

$$k := \boxed{\begin{array}{l} a \longrightarrow c \\ b \longrightarrow w \\ c \longrightarrow n \\ \quad\vdots \\ z \longrightarrow a \end{array}}$$

$c := E(k, \text{"bcza"}) = \text{"wnac"}$

$D(k, c) = \text{"bcza"}$

Caesar Cipher (no key)

shift by 3 :

a → d

b → e

c → f

⋮

y → b

z → c

What is the size of key space in the substitution cipher assuming 26 letters?

$|\mathcal{K}| = 26$

$|\mathcal{K}| = 26!$ (26 factorial) ⟸

$|\mathcal{K}| = 2^{26}$

$26! \approx 2^{88}$

$|\mathcal{K}| = 26^2$

How to break a substitution cipher?

What is the most common letter in English text?

“X”

“L”

[✓ “E”]

“H”

How to break a substitution cipher?

(1) Use frequency of English letters

"e": 12.7% , "t": 9.1% , "a": 8.1%

(2) Use frequency of pairs of letters (digrams)

"he", "an", "in", "th"

⟹ CT only attack!!

An Example

UKBYBIPOUZBCUFEEBORUKBYBHOBBRFESPVKBWFOFERVNBCVBZPRUBOFERVNBCVBPCYYFVUFO
FEIKNWFRFIKJNUPWRFIPOUNVNIPUBRNCUKBEFWWFDNCHXCYBOHOPYXPUBNCUBOYNRVNIWN
CPOJIOFHOPZRVFZIXUBORJRUBZRBCHNCBBONCHRJZSFWNVRJRUBZRPCYZPUKBZPUNVPWPCYVF
ZIXUPUNFCPWRVNBCVBRPYYNUNFCPWWJUKBYBIPOUZBCUIPOUNVNIPUBRNCHOPYXPUBNCUB
OYNRVNIWNCPOJIOFHOPZRNCRVNBCUNENVVFZIXUNCHPCYVFZIXUPUNFCPWZPUKBZPUNVR

B	**36**	→ E
N	**34**	
U	**33**	→ T
P	**32**	→ A
C	**26**	

NC	**11**	→ IN
PU	**10**	→ AT
UB	**10**	
UN	**9**	

digrams

UKB	**6**	→ THE
RVN	**6**	
FZI	**4**	

trigrams

https://www.dcode.fr/substitution-cipher

Backup Slides