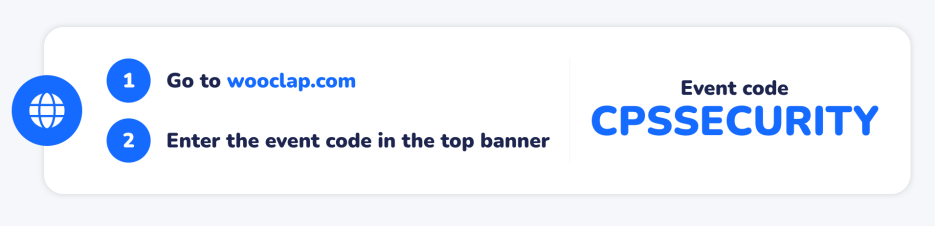
CE/CZ4055 Cyber Physical System Security

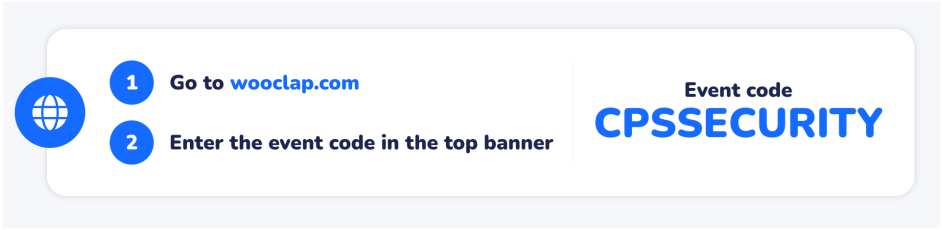
Anupam Chattopadhyay
CCDS, NTU

NANYANG TECHNOLOGICAL UNIVERSITY



Contents

- Discussions from Last Week



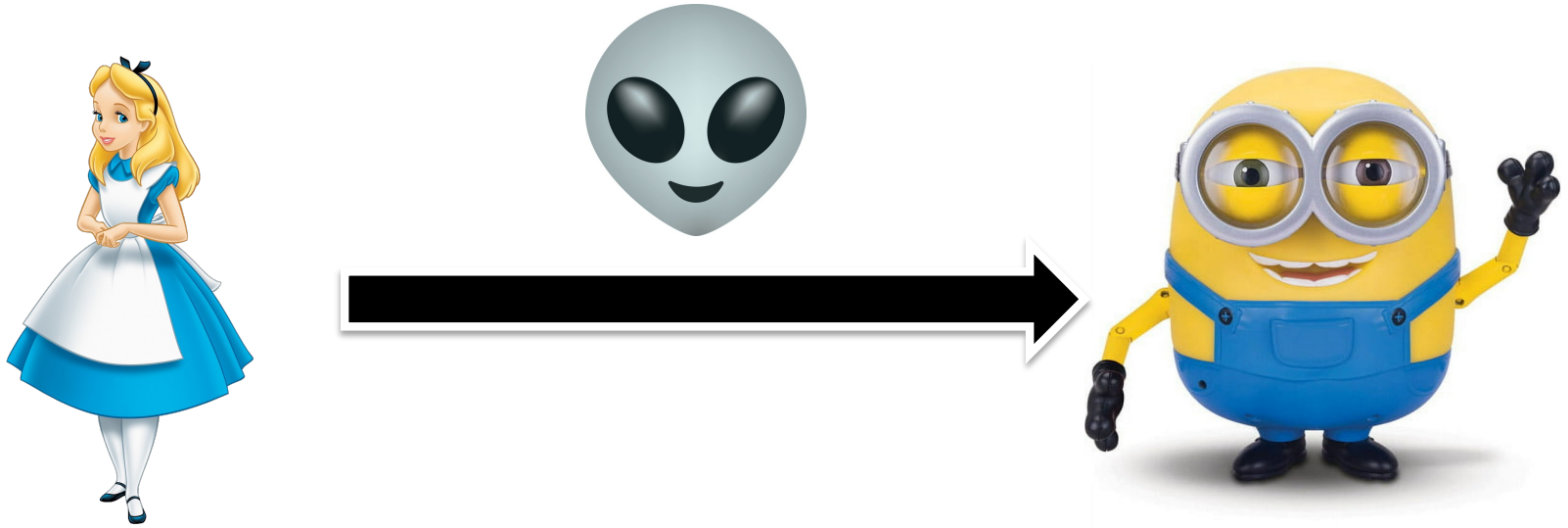
NANYANG TECHNOLOGICAL UNIVERSITY

Further Details

HASH, MAC

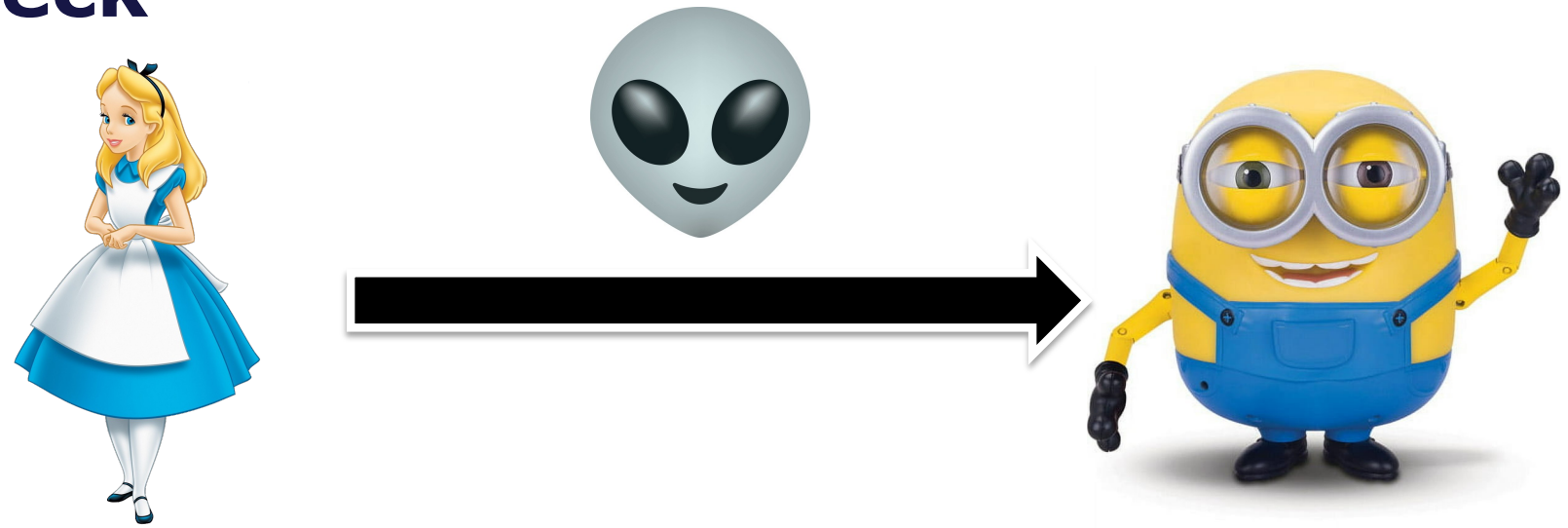
HASH FOR INTEGRITY
MAC FOR AUTHENTICITY

Hash-based Message Integrity Check



- Alice sends Message M and Hash H (calculated using Hash function over M) to Bob
- Bob recalculates Hash H' using M
- If $H == H'$ then, the integrity check passes
- Eve attempts to modify M, and yet pass integrity. This requires simultaneous modification of M and H, which is hard.

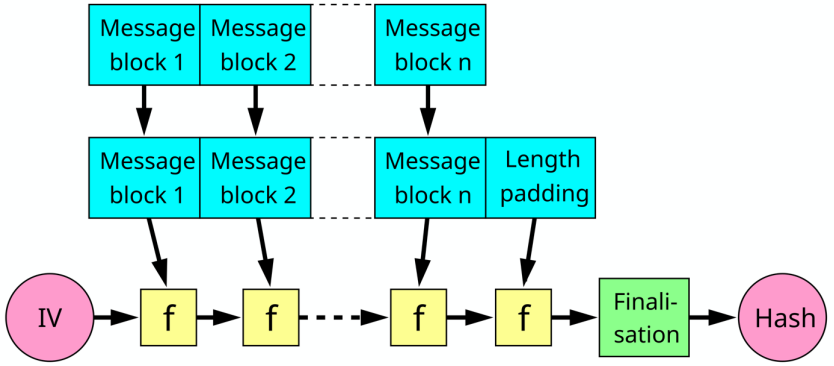
MAC-based Authenticity and Integrity Check



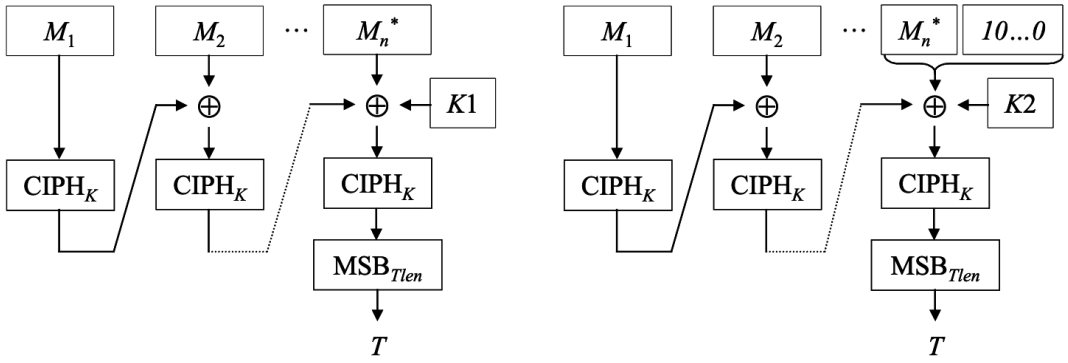
- Alice sends Message M and MAC H (calculated using MAC function over M, K) to Bob, where K is a secret key shared between Alice and Bob
- Bob recalculates MAC H' using M, K
- If $H == H'$ then the integrity and authenticity check passes
- Eve attempts to modify M and yet pass integrity and authenticity. This requires Eve to possess K, which is not possible.

Hash, HMAC, CMAC, KMAC

- **Hash**: *Integrity*. Basic construction from Merkle–Damgård



- **MAC**: *Integrity* and *Authenticity*. We discussed the construction based on Block cipher.

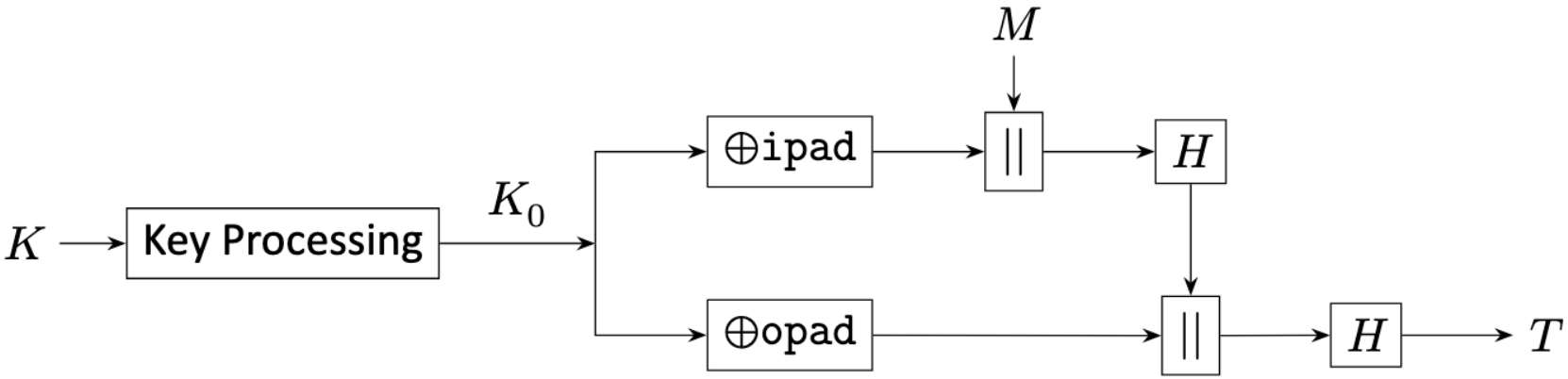


https://csrc.nist.gov/Projects/Message-Authentication-Codes

NANYANG TECHNOLOGICAL UNIVERSITY

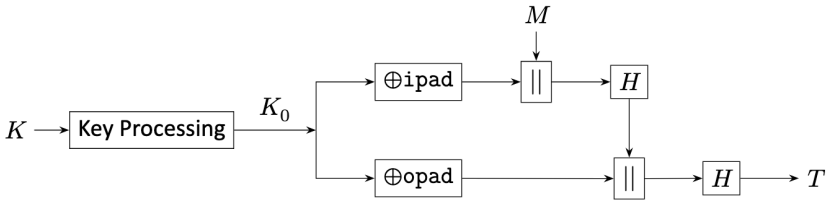
HMAC

- MAC generated using key-ed Hash functions

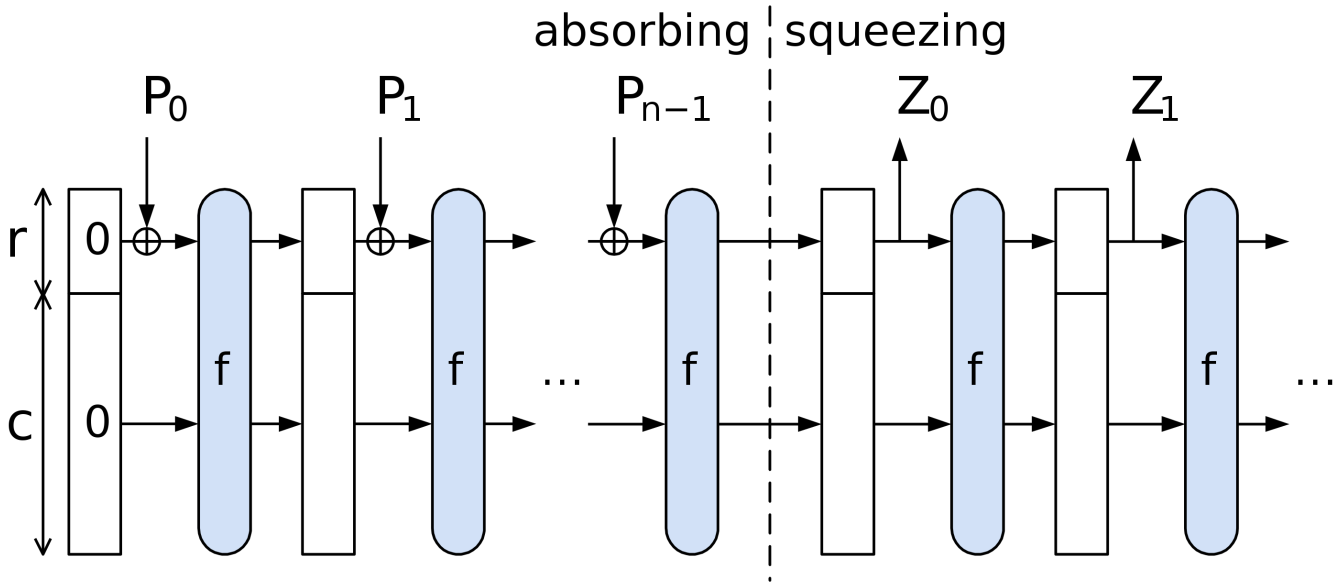


https://csrc.nist.gov/Projects/Message-Authentication-Codes

KMAC



- MAC generated using key-ed Hash functions, where the **Hash function is Keccak** (currently standardized as SHA-3)

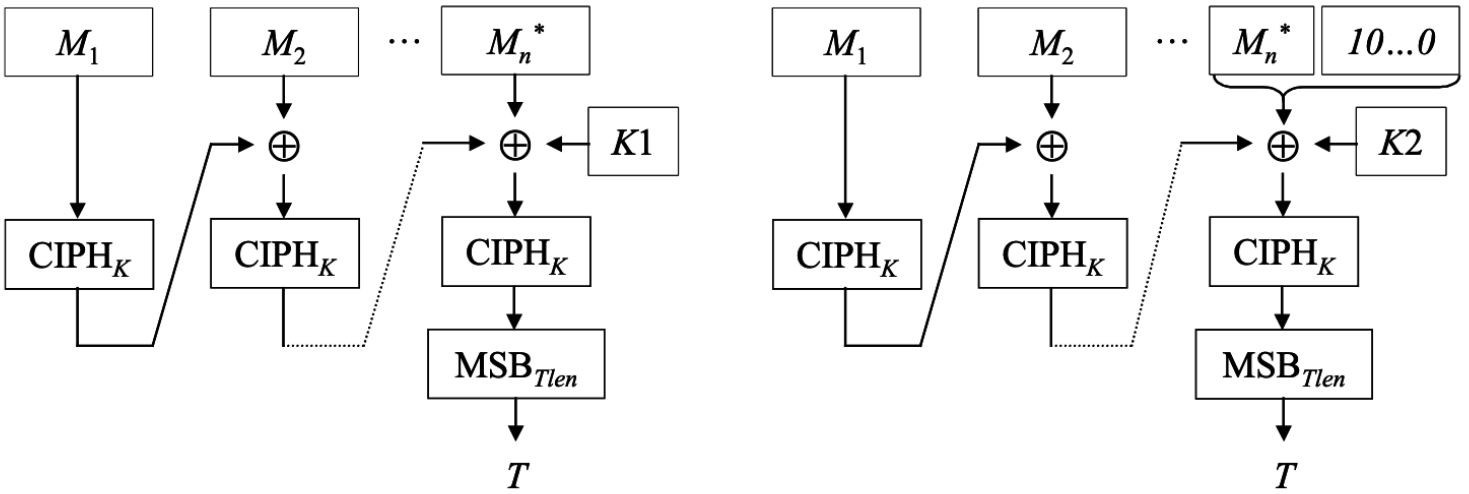


NANYANG TECHNOLOGICAL UNIVERSITY

https://csrc.nist.gov/Projects/Message-Authentication-Codes

CMAC

- **Cipher Block Chaining MAC**
  - Can be based on AES
  - Also referred to as a ‘mode of operation’ for the cipher



https://csrc.nist.gov/Projects/Message-Authentication-Codes

Further Details

802.11B WEP

Valid Tag Generation

(3) (The 802.11b insecure MAC). Consider the following MAC (a variant of this was used for WiFi encryption in 802.11b WEP). Let F be a PRF defined as $F : \{0,1\}^{128} \times \{0,1\}^{128} \rightarrow \{0,1\}^{32}$. Let $CRC32$ be a simple and popular error-detecting code meant to detect random errors; $CRC32(m)$ takes inputs $m \in \{0,1\}^*$ and always outputs a 32-bit string. For this exercise, the only fact you need to know is that $CRC32(m_1) \oplus CRC32(m_2) = CRC32(m_1 \oplus m_2)$. Define the following MAC system $(Sign, Verify)$ as:

$$Sign(k, m) = r \leftarrow \{0,1\}^{128},\ t = F(k, r) \oplus CRC32(m),\ \text{Output: } (r, t)$$

$$Verify(k, m, (r, t)) = \{\text{accept if } t = F(k, r) \oplus CRC32(m) \text{ or reject otherwise}\}$$

Show that this MAC system is insecure.

- It is possible to generate a valid tag by $t_1 \oplus t_2$, where t_1 and t_2 are two tags from m_1 and m_2, respectively.
- Regardless of what is contained in r, the tag validity leads to a denial-of-service (DoS) attack.

Brute Force Attack

(3) (The 802.11b insecure MAC). Consider the following MAC (a variant of this was used for WiFi encryption in 802.11b WEP). Let F be a PRF defined as $F : \{0,1\}^{128} \times \{0,1\}^{128} \rightarrow \{0,1\}^{32}$. Let $CRC32$ be a simple and popular error-detecting code meant to detect random errors; $CRC32(m)$ takes inputs $m \in \{0,1\}^*$ and always outputs a 32-bit string. For this exercise, the only fact you need to know is that $CRC32(m_1) \oplus CRC32(m_2) = CRC32(m_1 \oplus m_2)$. Define the following MAC system $(Sign, Verify)$ as:

$$Sign(k,m) = r \leftarrow \{0,1\}^{128},\ t = F(k,r) \oplus CRC32(m),\ \text{Output: } (r,t)$$

$$Verify(k,m,(r,t)) = \{\text{accept if } t = F(k,r) \oplus CRC32(m) \text{ or reject otherwise}\}$$

Show that this MAC system is insecure.

- $F(k,r)$ produces a 32-bit output, which can be brute-forced at the receiver side.
- Attacker, without possessing the key k, can claim to have received a valid message and tag.

Message Tampering

(3) (The 802.11b insecure MAC). Consider the following MAC (a variant of this was used for WiFi encryption in 802.11b WEP). Let F be a PRF defined as $F : \{0,1\}^{128} \times \{0,1\}^{128} \rightarrow \{0,1\}^{32}$. Let $CRC32$ be a simple and popular error-detecting code meant to detect random errors; $CRC32(m)$ takes inputs $m \in \{0,1\}^*$ and always outputs a 32-bit string. For this exercise, the only fact you need to know is that $CRC32(m_1) \oplus CRC32(m_2) = CRC32(m_1 \oplus m_2)$. Define the following MAC system $(Sign, Verify)$ as:

$$Sign(k, m) = r \leftarrow \{0,1\}^{128},\ t = F(k, r) \oplus CRC32(m),\ \text{Output: } (r, t)$$

$$Verify(k, m, (r, t)) = \{\text{accept if } t = F(k, r) \oplus CRC32(m) \text{ or reject otherwise}\}$$

Show that this MAC system is insecure.

- Since $CRC32$ is a linear function over the message space, it is possible to XOR a δ (tamper) with the message and generate the correct tag by XOR-ing $CRC32(\delta)$ with the tag.
- $t' = F(k,r) \oplus CRC32(m) \oplus CRC32(\delta) = F(k,r) \oplus CRC32(m \oplus \delta)$

Contents

- *Security Triad*
- Cryptographic Primitives
- Discussion

NANYANG TECHNOLOGICAL UNIVERSITY

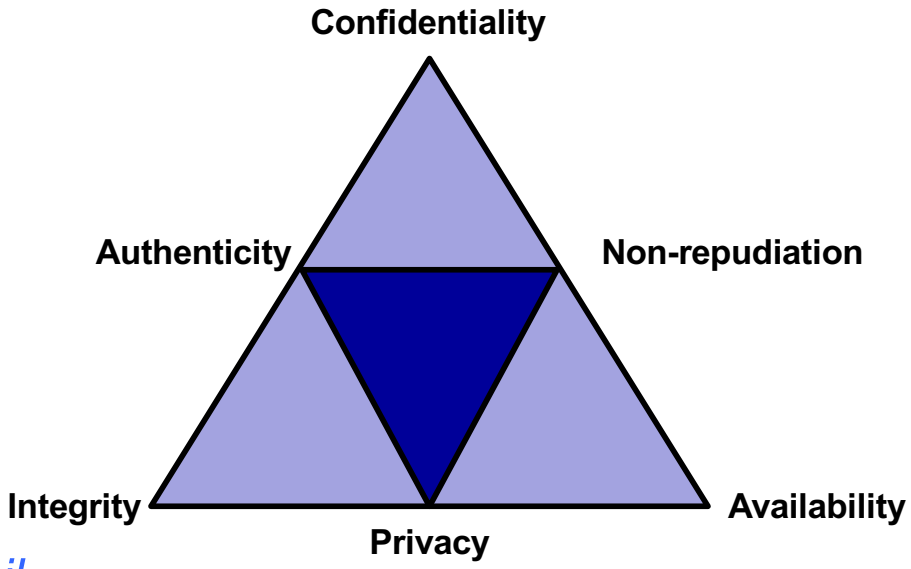
1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
CPSSECURITY

Security Triad

- IT security
  - Confidentiality
    - *Data is secured*
  - Integrity
    - *Data is trusted*
  - Availability
    - *Data is accessible*
  - Non-repudiation
    - *Service has a trusted audit-trail*
  - Authenticity
    - *Components have provable identity*
  - Privacy
    - *Service does not see customer data*



Contents

- Security Triad

Cryptographic Primitives

  - Private-Key Cryptography
  - Hash Function, Message Authentication Code
  - ***Public-Key Cryptography***
  - Digital Signature
  - PKI

- Discussion

NANYANG TECHNOLOGICAL UNIVERSITY

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
CPSSECURITY

Motivation

- Until early 70s, cryptography was mostly owned by government and military
  - Key distribution is more manageable and better funded
- Symmetric cryptography not ideal for commercialization
  - Enormous key distribution problem
  - Most parties may never meet physically
- Few researchers (Diffie, Hellman, Merkle) started exploring public-key cryptography realizing its importance in the forthcoming digital world
  - Privacy, Effective commercial relations, Payment, Voting

Public-Key Cryptography

- Idea: use separate keys to encrypt and decrypt
  - First proposed by Diffie and Hellman
  - Independently proposed by Merkle (1976)
- Pair of keys for each user
  - generated by the user
  - Public key is advertised
  - Private key is kept secret, and is computationally infeasible to discover from the public key and ciphertexts
  - Each key can decrypt messages encrypted using the other key
- Applications:
  - Encryption, Authentication (Digital Signature), Key Exchange (to establish Session Key)

Crossword Puzzles

- Ralph Merkle's Key Exchange Algorithm
  - Alice generates MANY crossword puzzles and sends to Bob
  - Bob chooses ONE and solves it
  - The solution includes an identifier, and the key
  - Bob communicates the identifier to Alice
  - Alice and Bob communicate using the key

- **Important observation: Eve would have to solve ALL puzzles to identify the right one and the key.**

- First attempt, cumbersome, but revolutionary at the time

CLUES ACROSS

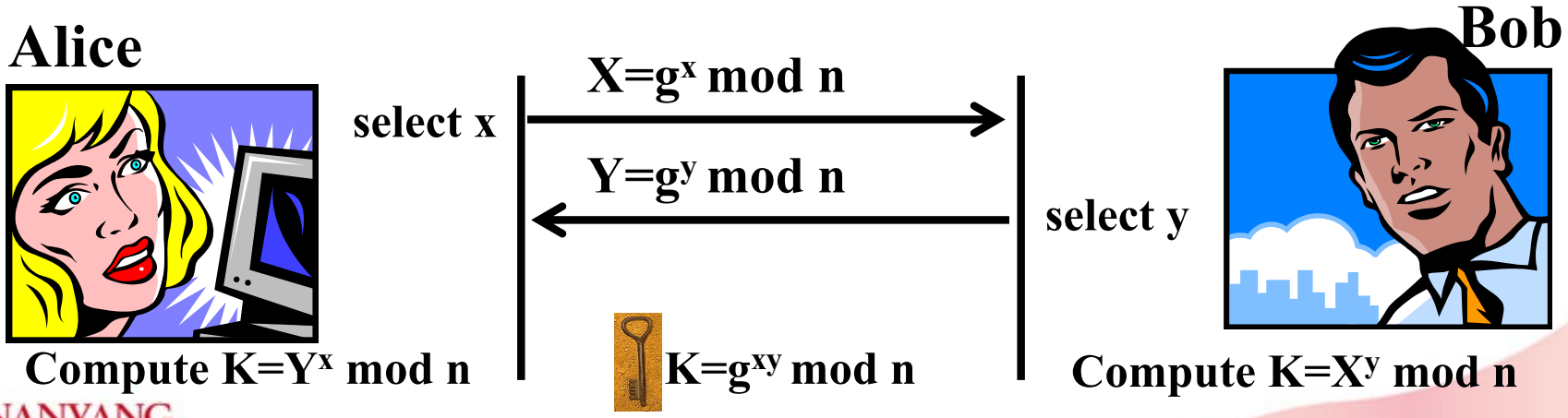
1 "as he ___ upon the mount" (Matt. 24)
4 "the poison of ___ is under their lips" (Rom. 3)
8 "righteousnesses are as filthy ___" (Isa. 64)
12 "the price of his ___ shall be…the number of years" (Lev. 25)
13 "ye are so ___ removed" (Gal. 1)
14 "For that which I do I ___ not" (Rom. 7)
16 "rule them with a rod of ___" (Rev. 19)
17 "breathed ___ his nostrils the breath of life" (Gen. 2)
18 "be holy and without ___ before him in love" (Eph. 1)
19 "saw Jesus afar off, he ___ and worshipped him" (Mark 5)
20 "shall be astonished, and ___ his head" (Jer. 18)
21 "weavest the seven locks of my head with the ___" (Judg. 16)
23 "secretly as a lion in his ___" (Ps. 10)
24 "the queen of ___ heard of the fame of Solomon" (I Kings 10)
26 "the ___, which the LORD God had taken from man" (Gen. 2)
28 "when ye pray, ___ not vain repetitions" (Matt. 6)
30 "if any man will ___ thee at the law" (Matt. 5)
32 "As the partridge sitteth on ___" (Jer. 17)
36 "hair is fallen off his head, he is ___" (Lev. 13)
39 "were as swift as the ___" (I Chron. 12)
41 "nations are as a ___ of a bucket" (Isa. 40)
42 "repented long ___ in sackcloth" (Matt. 11)
43 "the days when the judges ___" (Ruth 1)
45 "Blessed ___ the poor in spirit" (Matt. 5)
46 "let him ___ himself" (Matt. 16)
48 "children of Edom…said, Rase it, ___ it" (Ps. 137)
49 "Saul __ David from that day" (I Sam. 18)
50 "fall by the ___ of the sword" (Luke 21)
51 "he planteth an ___" (Isa. 44)
52 "harder to be ___ than a strong city" (Prov. 18)
54 "find an ___ tied, and a colt" (Matt. 21)

4

NANYANG TECHNOLOGICAL UNIVERSITY

Diffie-Hellman Key Exchange

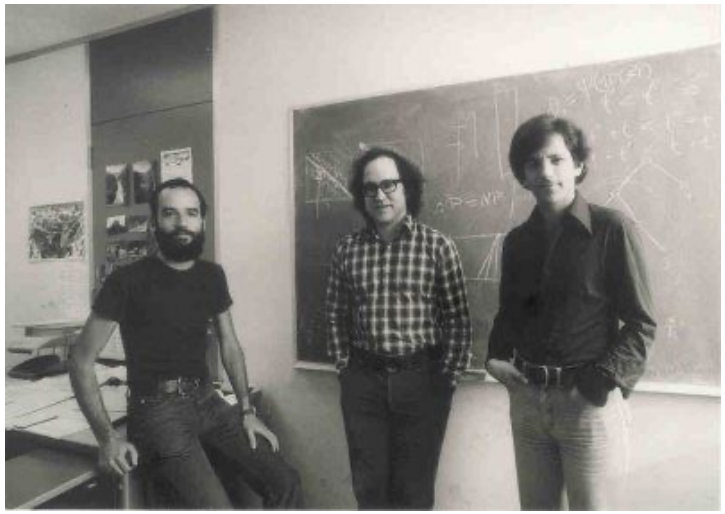
- First public-key algorithm, based on the difficulty of computing discrete logarithms modulo n
- Protocol:
  - Use key exchange protocol to establish session key
  - Use session key to encrypt actual communication
- Algorithm:
  - Choose a large prime n, and a primitive root g



NANYANG TECHNOLOGICAL UNIVERSITY

RSA

- Developed by Rivest, Shamir, and Adleman (1977)
  - Security comes from the difficulty of factoring large numbers
  - Discovered earlier by UK GCHQ (Ellis and Cocks) in 1973 !



NANYANG TECHNOLOGICAL UNIVERSITY

Key-Generation for RSA

- Generate two large random distinct primes
  p and q, each roughly the same size
- Compute $n = pq$ and $\varphi(n) = (p-1)(q-1)$
- Select random integer e:

$$1 < e < \varphi(n) \mid \gcd\big(e, \varphi(n)\big) = 1$$

- Compute unique integer d:

$$1 < d < \varphi(n) \mid ed = 1 \bmod \varphi(n)$$

- Public key is (n, e); Private key is d

Why RSA works?

- **Encryption** (with public key): $C = M^e$
- **Decryption** (with private key): $M = C^d$

- $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+h\varphi(n)} \equiv M \pmod{n}$

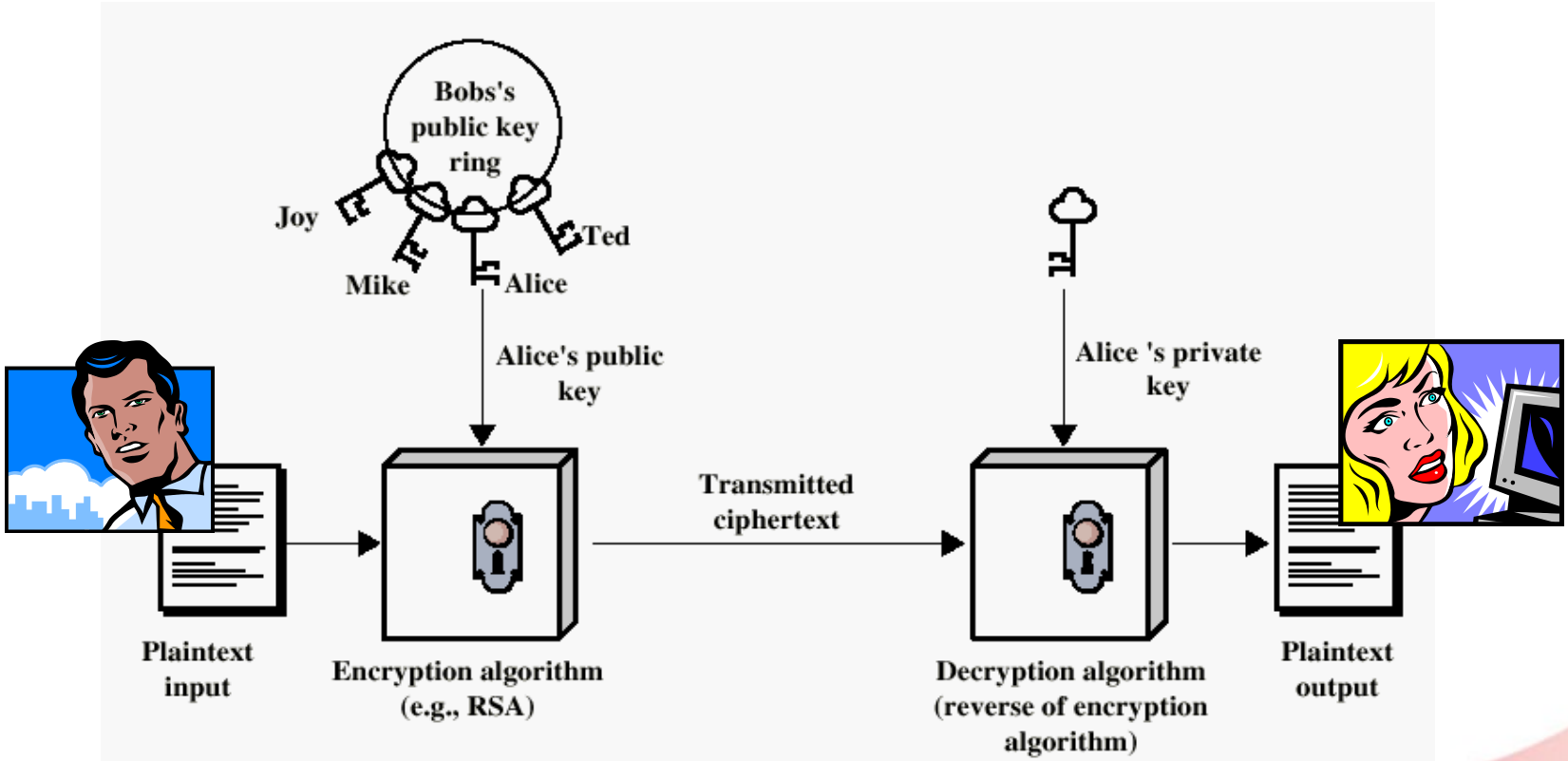
- We have $ed \equiv 1 \pmod{\varphi(n)}$. Hence we can write $ed = h(p-1)(q-1) + 1$ for some (non negative) integer h
- Now, $(M^e)^d = M^{ed} = M^{ed-1}M = M^{h(p-1)(q-1)}M = (M^{p-1})^{h(q-1)}M$.
- Fermat's Little Theorem → $M^{p-1} \equiv 1 \pmod{p}$

- $(M^e)^d = \cdots \equiv 1^{h(q-1)}M \pmod{p} = M \pmod{p}$
- $(M^e)^d = \cdots \equiv 1^{h(p-1)}M \pmod{q} = M \pmod{q}$

→ $(M^e)^d = M \pmod{pq} = M \pmod{n}$ [Euler's Lemma]

Two-key Public-Key Encryption

- Sender uses the public key of the receiver to encrypt
- Receiver uses her private key to decrypt



Threat from Quantum Computers

- Universal Quantum computers (IBM, Intel, Google, IonQ)
  - Akin to a general-purpose processor
  - Relevant to the security threats

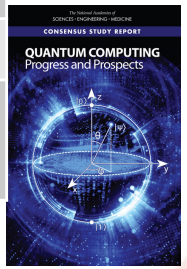
- Quantum Annealing (D-Wave)
  - Designed to solve a hard optimization problem
  - No evidence of quantum speed-up over entire dataset[1]
  - Demonstrated speed-up over Simulated Annealing, and Quantum Monte Carlo[2]

1. T. F. Ronnow et al, “Defining and detecting quantum speedup”, Science 2014
2. H. Neven, “When can Quantum Annealing win?”, Google AI Blog, 2015

NANYANG TECHNOLOGICAL UNIVERSITY

Attack Complexity Estimates

Cryptosystem	Category	Key Size	Quantum Algorithm	# Logical Qubits Required	# Physical Qubits Required	Time Required to Break System
AES-GCM	Symmetric-Key Encryption	128	Grover's Algorithm	2,953	4.61×10^6	2.61×10^{12} years
		192		4,449	1.68×10^7	1.97×10^{22} years
		256		6,681	3.36×10^7	2.29×10^{32} years
RSA	Asymmetric-Key Encryption	1024	Shor's Algorithm	2,050	8.05×10^6	3.58 hours
		2048		4,098	8.56×10^6	28.63 hours
		4096		8,194	1.12×10^7	229 hours
ECC Discrete-log problem	Asymmetric-Key encryption	256	Shor's Algorithm	2,330	8.56×10^6	10.5 hours
		384		3,484	9.05×10^6	37.67 hours
		521		4,719	1.13×10^6	55 hours



1. Quantum Computing: Progress and Prospects (2019). Consensus Study Report. National Academies Press, 2019.

NANYANG TECHNOLOGICAL UNIVERSITY

Post-Quantum Cryptography

- Instead of waiting for a Quantum computer ***to actually break*** the current e-commerce, we prepare for that by designing new public-key cryptographic primitives that are resistant against Quantum-enabled attackers
- **Idea: Base security on a hard computational problem 'without efficient Quantum algorithm'**

- Lattice-based: Closest-vector problem, Shortest-vector problem
- Hash-based: Security of one-way hash functions
- Multivariate Cryptography: Multivariate Quadratic Equation Solving problem
- Code-based: Syndrome decoding problem

- Multiple proposals
  - NIST Standardized PKE/KEM → CRYSTALS-Kyber
  - NIST Standardized Digital Signature → CRYSTALS-Dilithium, Falcon, Sphincs+

Contents

- Security Triad
- *Cryptographic Primitives*
  - Private-Key Cryptography
  - Hash Function, Message Authentication Code
  - Public-Key Cryptography
  - ***Digital Signature***
  - PKI
- Discussion

NANYANG TECHNOLOGICAL UNIVERSITY

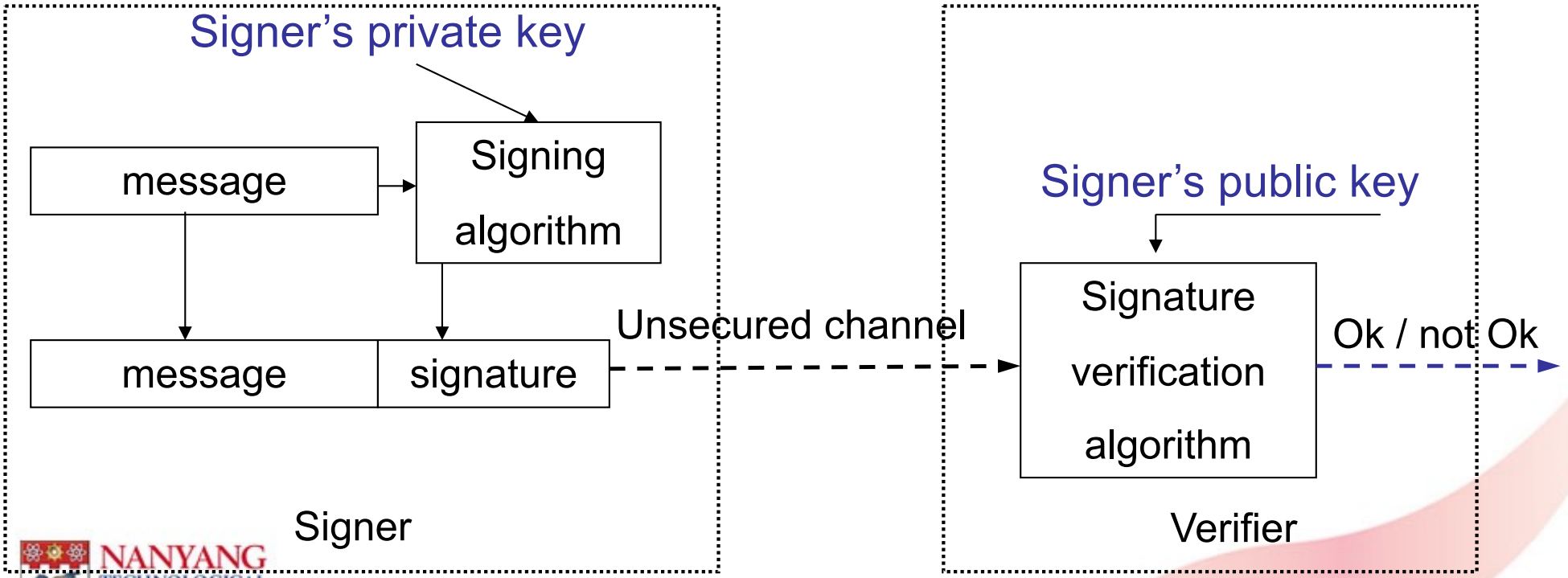
1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
CPSSECURITY

Digital Signature Scheme

- Used to provide
  - Data integrity
  - Message authentication
  - Non-repudiation



RSA signature generation and verification

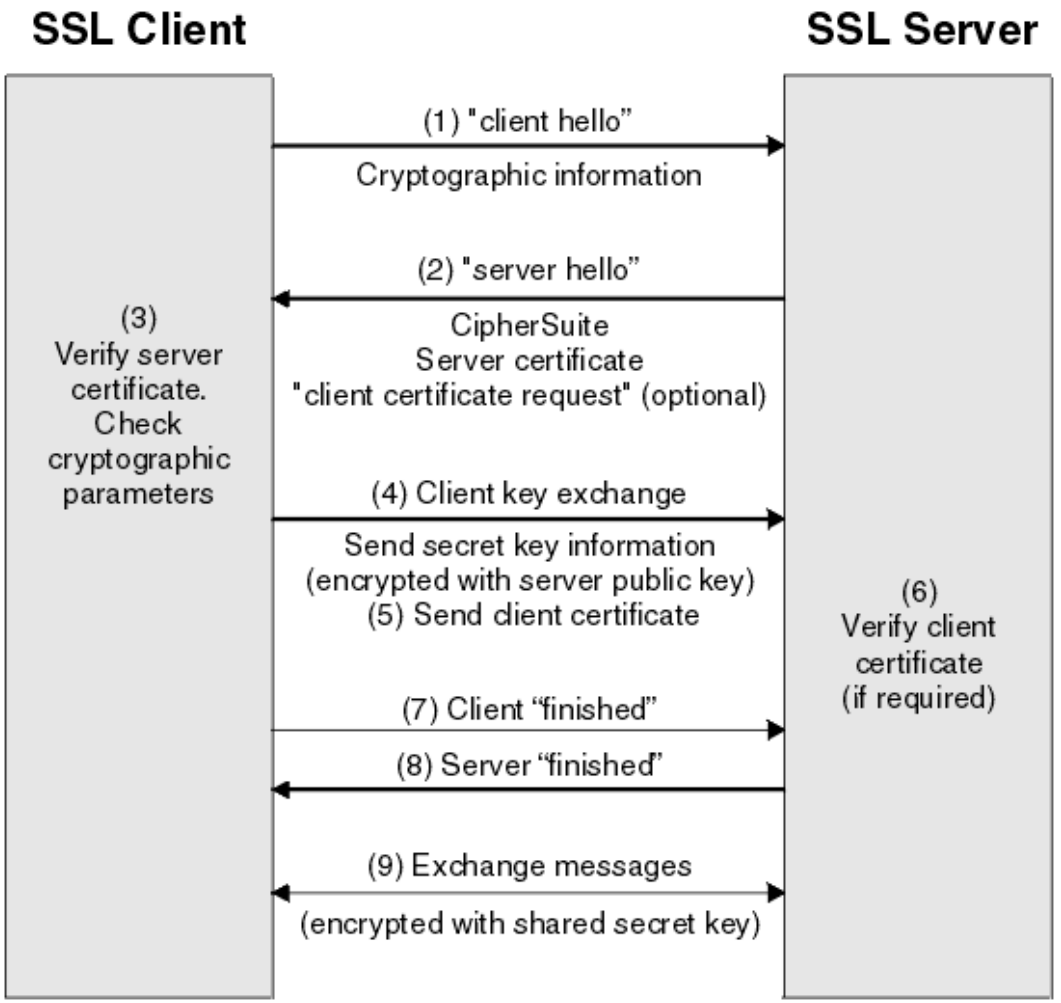
- To sign a message
  - *Generate a tag from the message, e.g., using a Hash*
  - *Encrypt the Hash (H) with sender's private key, to obtain Ciphertext (C)*

- *To verify the signature*
  - *Compute the Hash (H') from the message M*
  - *Decrypt the received Ciphertext (C) to obtain Hash (H) using sender's public key*
  - *Match H and H'*

Difference between MAC and digital signature

- To prove the validity of a MAC to a third party, you need to reveal the key
- If you can verify a MAC, you can also create it
- MAC does not allow a distinction to be made between the parties sharing the key
- Computing a MAC is (usually) much faster than computing a digital signature
  - Important for devices with low computing power

SSL/TLS Handshake



NANYANG TECHNOLOGICAL UNIVERSITY

A Few More **Crypto Building Blocks**

- **Fully Homomorphic Encryption**
  - Allows computing on encrypted dataset
  - Open-source implementations from Microsoft, Intel
  - https://en.wikipedia.org/wiki/Microsoft_SEAL
  - Alternative: **Searchable Symmetric Encryption**

- **Secure Multiparty Computation**
  - Allows shared computing without any party knowing the inputs
  - Example: $F(x, y, z) = \max(x, y, z)$ being computed by 3 parties, where none gets to know the other inputs.

- **Function Secret Sharing**
  - Allows sharing a given function over multiple parties, where no party gets to know the complete function
  - Example: Computing inference of a given data using a 'secret' Neural Network by multiple parties.

Contents

- Security Triad
- *Cryptographic Primitives*
  - Private-Key Cryptography
  - Hash Function, Message Authentication Code
  - Public-Key Cryptography
  - Digital Signature
  - ***PKI***
- Discussion

NANYANG TECHNOLOGICAL UNIVERSITY

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
CPSSECURITY

Why do we need a PKI?

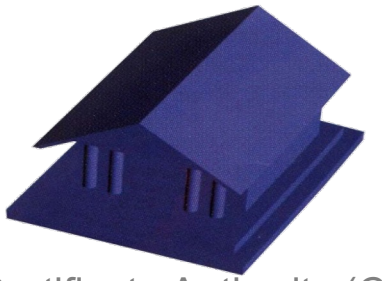
- Public key security issues:
  - Users can generate their own public/private key pairs and exchange them – but how do other parties trust them?
  - If you receive a public key from Alien Pkie, how do you know it's Pkie's key and not the human spy's?

- Solution: Digital Certificates
  - Bind the user's public key with a digital certificate signed by a *trusted third party*.
  - The trusted third party is called the certification authority (CA).
  - CA will vouch for its subscribers.

Entities of PKI

Corporations

Issues Certificates to



Certificate Authority (CA)

Can validate companies by

Issue certificates

Do business with

Relying Parties

Trust each other

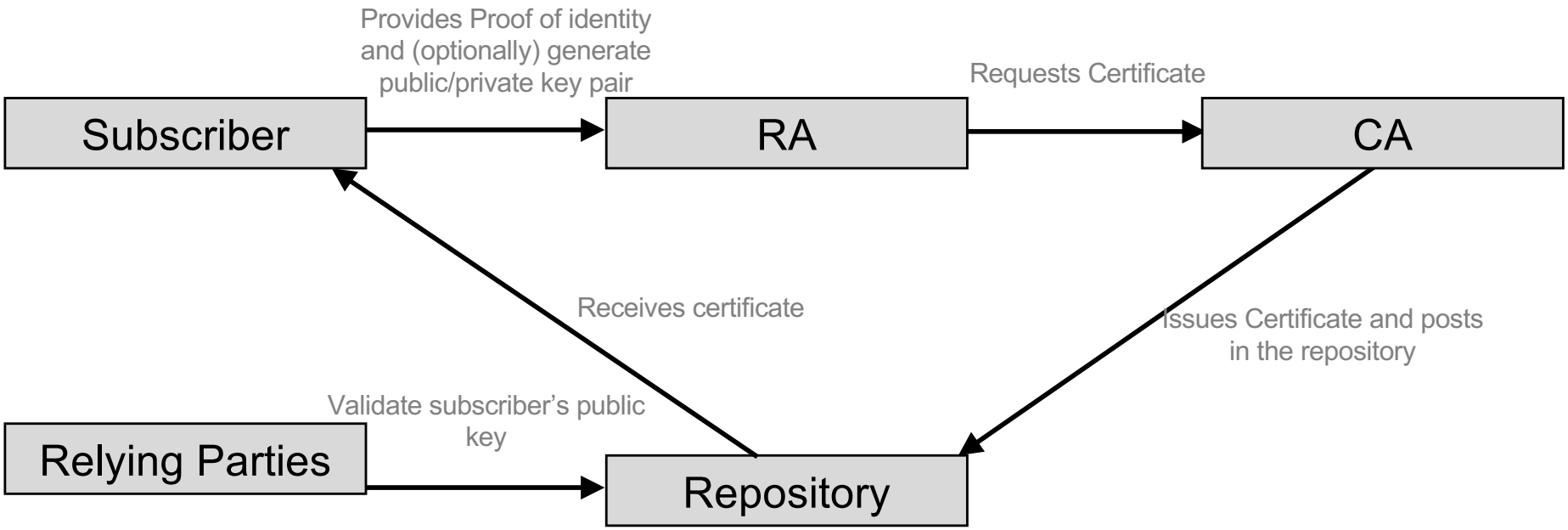
Individual Subscribers

NANYANG TECHNOLOGICAL UNIVERSITY

Certification Authority

- **Registration Authority (RA)** – Registers subscribers into the system
- **Certification Authority (CA)** – Creates digital certificates by binding user identity to public key.
- **Certificate Repository** – a directory service to store certificates for subscribers.
- **Certificate Revocation System** – Service to invalidate any certificates that has been compromised.

The steps in subscribing to a CA



NANYANG TECHNOLOGICAL UNIVERSITY

Contents

- Security Triad
- Cryptographic Primitives
- *Discussion*

NANYANG TECHNOLOGICAL UNIVERSITY

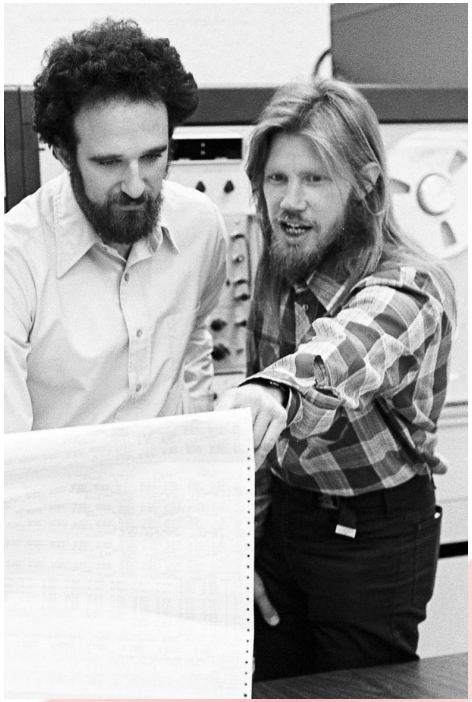
1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
CPSSECURITY

What did we learn?

- **What is Security?**
  - Confidentiality, Integrity, Availability
  - *Authenticity, Privacy, Non-repudiation*



- **What are the building blocks of security?**
  - Private-Key Cryptography, Hash, MAC
  - Public-Key Cryptography, Digital Signature
  - *A few more* functions

NANYANG TECHNOLOGICAL UNIVERSITY

Further Reading

- Security Engineering
  - by Ross Anderson, available online - http://www.cl.cam.ac.uk/~rja14/book.html
- Handbook of Cryptography
  - by Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, available online - http://cacr.uwaterloo.ca/hac/
- Applied Cryptography
  - by Dan Boneh and Victor Shoup, available online - http://toc.cryptobook.us/
- Leisure Reading - Simon Singh: *The Code Book*, Fourth Estate 1999

NANYANG TECHNOLOGICAL UNIVERSITY