

Introduction to Information Security

— Digital Signature, One-Way Hash & MAC

Dr. Tianlei HU

Associate Professor

College of Computer Science, Zhejiang Univ.

htl@zju.edu.cn

Outlines

- Digital Signature
- One-way hash function
- Message Authentication Code, MAC
- PGP

Concepts of Digital Signature

What does Cryptography do and not do?

- Cryptography solved the issue:
 - The communication of A and B can't be seen by others!
- However, encryption/decryption can't prevent deception:
 - If Alice has sent Bob a message, the dispute between them may be:
 - Bob fabricates a different message and declares that he has received for Alice ;
 - Alice can deny sending the message, and Bob can't prove Alice has sent the message.
- How do we solve this problem in daily life?
 - How can the court accept a contract?
 - Doing something which needs the approval of the supervisor or the organization, how can we prove that we have got the approval?
- **The Signature**

Characters and Requirements of Digital Signature

- Handwritten signature features?
 - The signature is credible. The recipient believes that the signer signed the document carefully
 - A signature can't be fabricated
 - A signature can't be reusable
 - A signed document can't be changed
 - A signature is an undeniable
 - In some cases, the signature and time are bound
 - A signature can be legal evidence and can be proved
- Similar: the seal, fingerprint

Characters and Requirements of Digital Signature

- Obviously, we can't use a handwritten signature on digital documents. So, we need the **digital signature**.
- Requirements of a digital signature system:
 - Can be bound with the signed document
 - The recipient can verify the signature, and any other person can not forge a signature
 - Signer can not deny his signature
 - A third party should be able to check and confirm the signature for the settlement of disputes
 - Verify the author, date and time, content of the signature

The requirements of digital signatures

- Digital signatures must rely on the signed message
 - Relate to the contents to prevent modification.
- The digital signature must use the unique information of the sender to prevent fabrication and denial.
 - Only known by the sender, so it can't be faked, and the sender can't deny it.
- Digital signature generation, identification, and authentication must be relatively simple.
 - The signature must be able to be generated and verified in a short time.
- Fabricate a digital signature is not feasible in the calculation
 - It can't be faked.
- Keep a backup of digital signatures is feasible
 - Can be stored(e.g., can't be larger than the original documents)

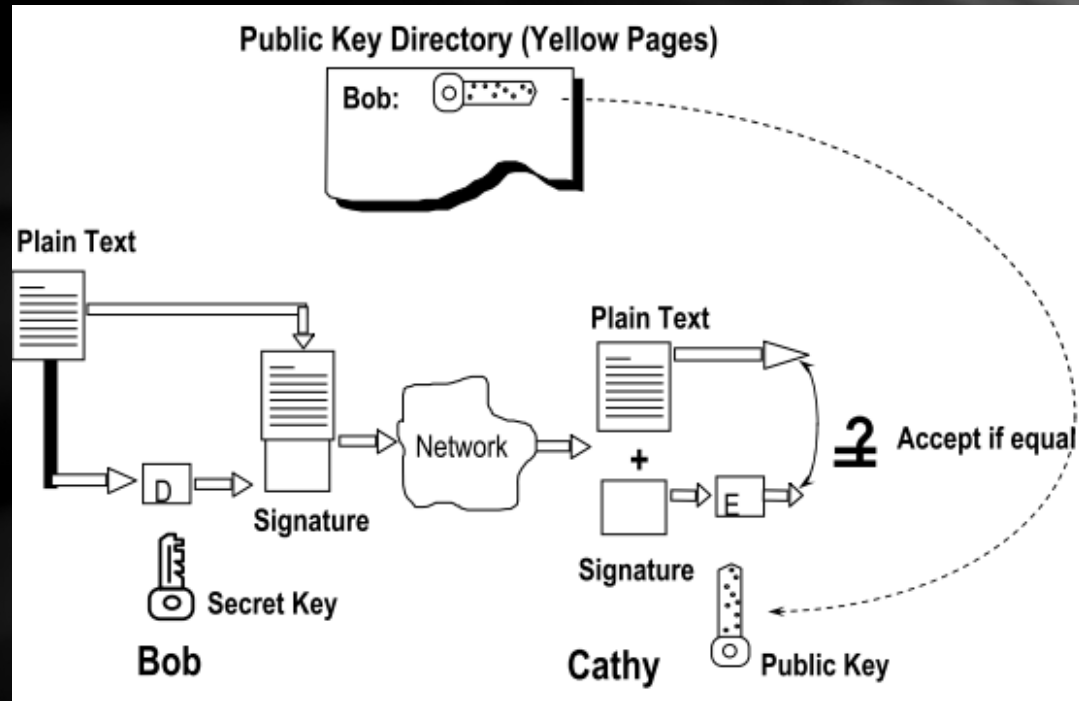
Digital Signature Algorithm

- Public key cryptography supports the “Digital Signature” natively.
- **DSS/DSA, Digital Signature Standard**
 - American National Standard, Digital Signature Standard (DSS), standardized in 1991
 - utilizing the difficulty of computing discrete logarithms
 - strongly promoted by the U.S. government
- **RSA**
 - RSA, widely supported by the industry, is the de facto industry standard
- **Elliptic curve**

Digital Signature Workflow

Attention:

- During encryption:
 - The sender **encrypts** using the **receiver's public key**
 - The receiver **decrypts** using **his private key**
- During signature
 - The sender **signs** using **his private key**
 - The receiver **verifies** the signature using the **sender's public key**



RSA signature example

- Key generation:
 - Bob:
 - Choose two prime numbers: $p = 5, q = 11, n = p \cdot q = 55, (p-1) \cdot (q-1) = 40$
 - Find $e=3$ and $d=27$, so: $3 \times 27 \equiv 1 \pmod{40}$
 - Bob's key: public key: $(3, 55)$, private key: 27
 - Bob is going to sign on a document where $m = 19$
 - He uses his private key $d = 27$ to calculate the digital signature of $m = 19$: $s = m^d \bmod n = 19^{27} \bmod 55 = 24$.
 - Attached 24 to the document, then : $(m, s) = (19, 24)$, Representing that the document is 19 and Bob's signature is 24.
 - Bob sent this document to Alice
 - Alice or a third party to verify the signature :
 - Receiving a plaintext and the signature $(m, s) = (19, 24)$
 - Check the public key directory to find Bob's public key $(e, n) = (3, 55)$
 - Computing : $t = s^e \bmod n = 24^3 \bmod 55 = 19$
 - Compare t and m whether they are equal; if equal, then $(19, 24)$ must be the document signed by Bob.

Any problems?

- In the previous example, the document m must be an integer of $[0 \dots n-1]$
 - If the document is very long, how to sign it?
- For a very long document, signing requires using the **one-way hash algorithm**.
- We do not sign the document. Instead, we get the hash value of documents, and then we sign the hash value.

One-Way Hash Algorithm

Also called Cryptographic hash function

One-way hash algorithm

- A one-way hash algorithm hashes an input document to about 100-bit output
- Given a one-way hash algorithm $H(.)$ we have:
 - Input: m — Binary string of arbitrary length
 - Output: $H(m)$ — Binary string of size L
 - Given $H(.)$, L is fixed:
 - In MD5, $L=128$
 - In SHA-1, $L=160$

One-way hash algorithm

- A good one-way hash algorithm $H(.)$ needs the following characteristics:
 - **Easy to compute**: Given any document m , $H(m)$ can be calculated quickly;
 - **Difficult to reverse computing**
 - Namely, given any hash value h , find any document m , making $H(m) = h$ is not feasible in computing.
 - Any algorithm meeting the above two requirements can be considered "**one-way**";
 - **Difficult to find a collision**
 - Finding any two documents, m_1 and m_2 , to make $H(m_1) = H(m_2)$ is computationally infeasible

Common one-way hash algorithm

- MD4、MD5 (R. Rivest, 1992)
- SHS (secure hashing standard, USA, 1992, modified in 1995):
- SHS(SHA-0, SHA-1, SHA-2)
- HAVAL (Y. Zheng, 1992)
- RIPEMD (D. Hans, 1996)
- More info: http://en.wikipedia.org/wiki/Cryptographic_hash_function



MD5 Algorithm

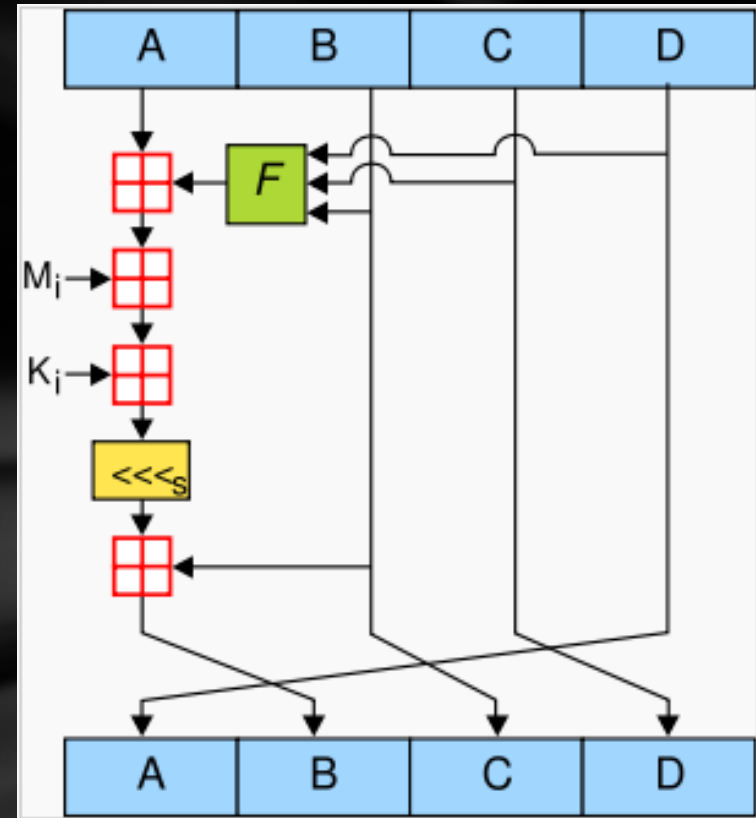
Two steps: Padding and Hashing

Padding:

- MD5 handles segments of length 512 bits, so the input string should be padded to multiple-512-sized segments(16 32-bit little-endian integers).
- Padding procedure :
 - Pad a bit '1'
 - Pad "0" until a multiple of 512 minus 64 bits
 - Pad the bit length of the original string to the last 64 bits

Hashing:

- MD5 handles a 128-bit string recursively; the initial value is the fixed constant, dividing the 128-bit string into four 32-bit-length integers(A, B, C, D)
- Using four different non-linear functions, F (downright) loops 16 times each.
 - : Addition (mod 2^{32})
 - : Shift left s bit
- Mi: 32-bit substring in the 512-bit string
- Ki: variant constant in each loop



$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X \vee \neg Z)$$

$\oplus, \wedge, \vee, \neg$ denote the XOR, AND, OR and NOT operations respectively.

Security of One-way Hash Algorithms

- Security evaluation of one-way hash algorithm:
 - **Preimage attack:** tries to find a message that has a specific hash value
 - **Preimage resistance:** for essentially all pre-specified outputs, it is computationally infeasible to find any input that hashes to that output; i.e., given y , it is difficult to find an x such that $h(x) = y$.
 - **Second-preimage resistance:** it is computationally infeasible to find any second input with the same output as a specified input; i.e., given x , it is difficult to find a second preimage $x' \neq x$ such that $h(x) = h(x')$.
 - **Collision attack:** tries to find two inputs producing the same hash value
 - Find two different messages, m_1 and m_2 , such that $\text{hash}(m_1) = \text{hash}(m_2)$.
 - More generally, **chosen-prefix collision attack:** Given two different prefixes, p_1 , and p_2 , find two appendages, m_1 , and m_2 , such that $\text{hash}(p_1 \parallel m_1) = \text{hash}(p_2 \parallel m_2)$, where \parallel denotes the concatenation operation.

Attack of One-Way Hash Algorithm

- https://en.wikipedia.org/wiki/Hash_function_security_summary
- 王小云 (2004/2005) find the collide algorithm of MD5, HAVAL-128, MD4, RIPEMD, SHA-1

Preimage resistance [\[edit \]](#)

Main article: [Preimage attack](#)

Hash function	Security claim	Best attack	Publish date
MD5	2^{128}	$2^{123.4}$	2009-04-27
SHA-1	2^{160}	45 of 80 rounds	2008-08-17
SHA256	2^{256}	43 of 64 rounds ($2^{254.9}$ time, 2^6 memory)	2009-12-10
SHA512	2^{512}	46 of 80 rounds ($2^{511.5}$ time, 2^6 memory)	2008-11-25
SHA-3	Up to 2^{512}		
BLAKE2s	2^{256}	2.5 of 10 rounds (2^{241})	2009-05-26
BLAKE2b	2^{256}	2.5 of 12 rounds (2^{481})	2009-05-26

Attack of One-Way Hash Algorithm

MD5 and SHA-1 were the most widely used one-way hash algorithms before 2009 and were used in many security-related computer products. The “cryptographic break” of them led to many severe security issues:

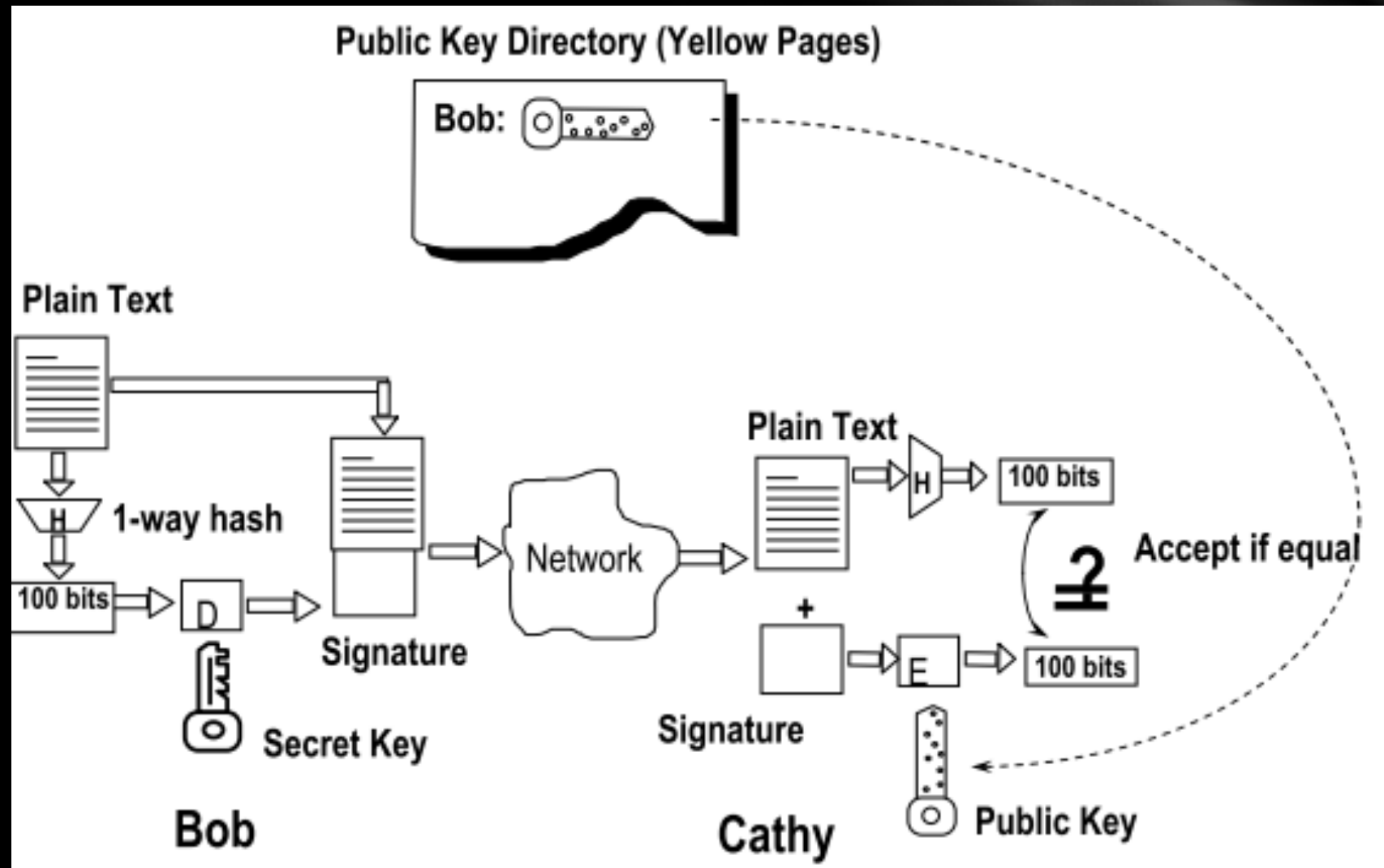
- In 2008, MD5 collision was used to attack SSL
 - <http://www.win.tue.nl/hashclash/rogue-ca/>
 - Attackers aim at the security infrastructure, PKI and CA, of SSL and can forge SSL certificates.
 - Any services, including E-commerce, E-Bank, E-trading, and so on, using HTTPS/SSH protocol will be affected by the vulnerabilities.
- In 2009, US-CERT considered that MD5 “should be considered cryptographically broken and unsuitable for further use.”
- The US government are mandated to use of SHA-2 from 2010

NIST hash function competition

http://en.wikipedia.org/wiki/NIST_hash_function_competition

- Started @2007.11.2
- During 2008: 64 algorithms were submitted before 2008.10, and 51 of them were selected as the 1st round candidates.
- During 2009: 14 algorithms were selected as the 2nd round candidates, and those algorithm were publicly reviewed for one year
- During 2010: 5 algorithms were selected for the 3rd, i.e., the last round in 2010.12
- On 2012.10.2, the Keccak algorithm was selected as the competition winner.
- On 2015.8.5, A version of this algorithm became a FIPS standard under the name SHA-3.

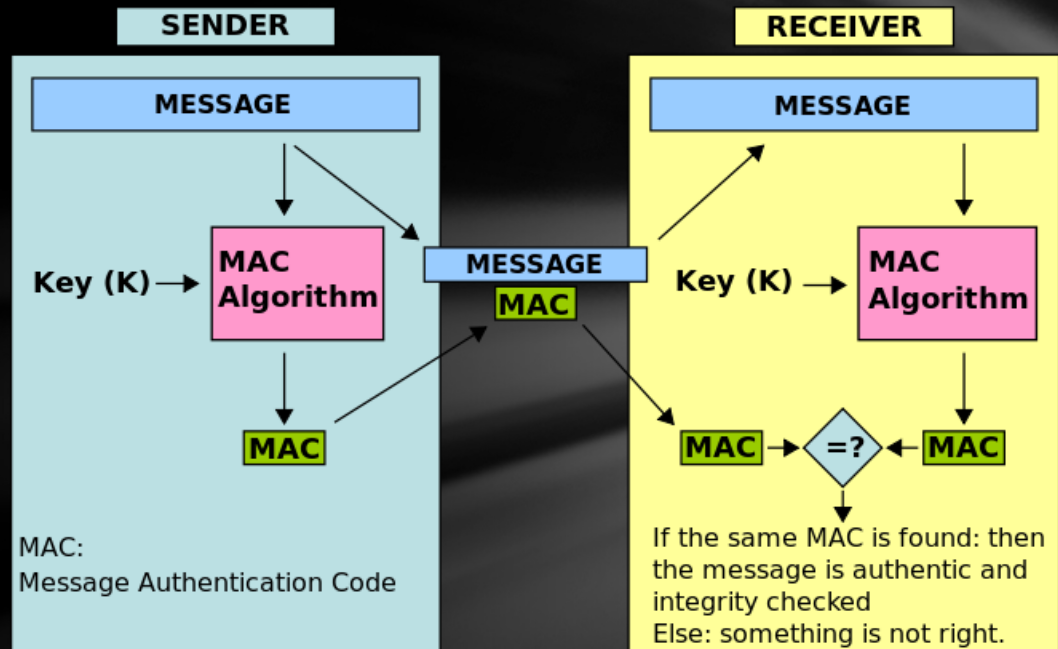
The procedure of signature for long plain text



MAC

Message Authentication Code

- MAC, Message authentication code, also called “keyed hash function.”
- Message Integrity Service
- Normal MAC algorithms:
 - HMAC
 - CBC-MAC
 - UMAC
 - CMAC
 - VMAC
 - Poly1305-AES
 - MMH-Badger MAC



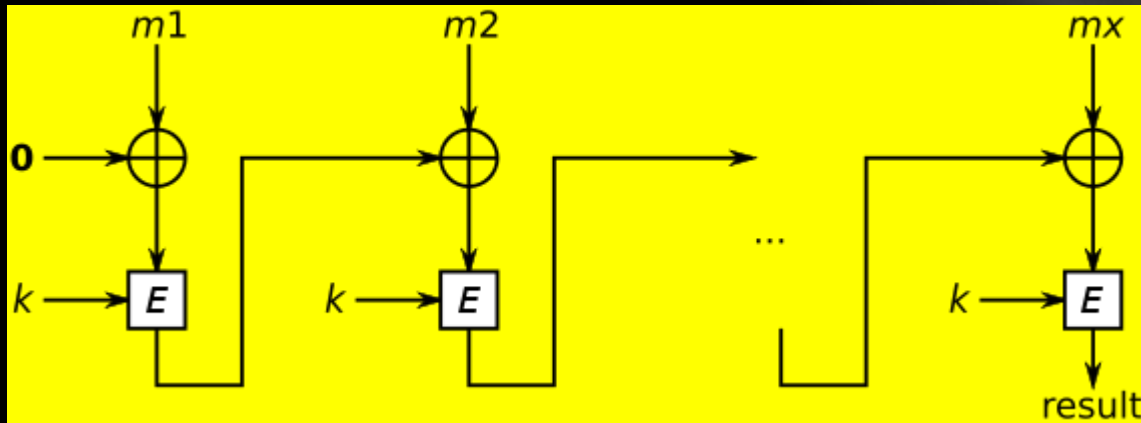
More info: http://en.wikipedia.org/wiki/Message_authentication_code

HMAC

- Hash-based message authentication code
- Definition (RFC 2014) :
 - $H()$: one-way hash function
 - K : Pad "0" until the key with the size of $H()$ input block
 - m : message want to be authenticated
 - \parallel : connect, \oplus : XOR
 - opad: outer padding (0x5c5c5c...5c5c, const in a block size)
 - ipad: inner padding (0x363636...3636, const in a block size)
 - So : $\text{HMAC}(K, m) = H((K \oplus \text{opad}) \parallel H((K \oplus \text{ipad}) \parallel m))$.
- Use different one-way hash functions to construct different HMAC algorithms:
 - HMAC-MD5
 - HMAC-SHA1
 - HMAC-SHA256
- HMAC is more difficult to occur a collision than a one-way hash function, so HMAC-MD5 and HMAC-SHA1 don't have security problems due to the vulnerability of MD5 and SHA1.
 - Until now, HMAC-MD5 and HMAC-SHA1 are safe enough and are the core components of IPSec and TLS.

CBC-MAC

- cipher block chaining message authentication code

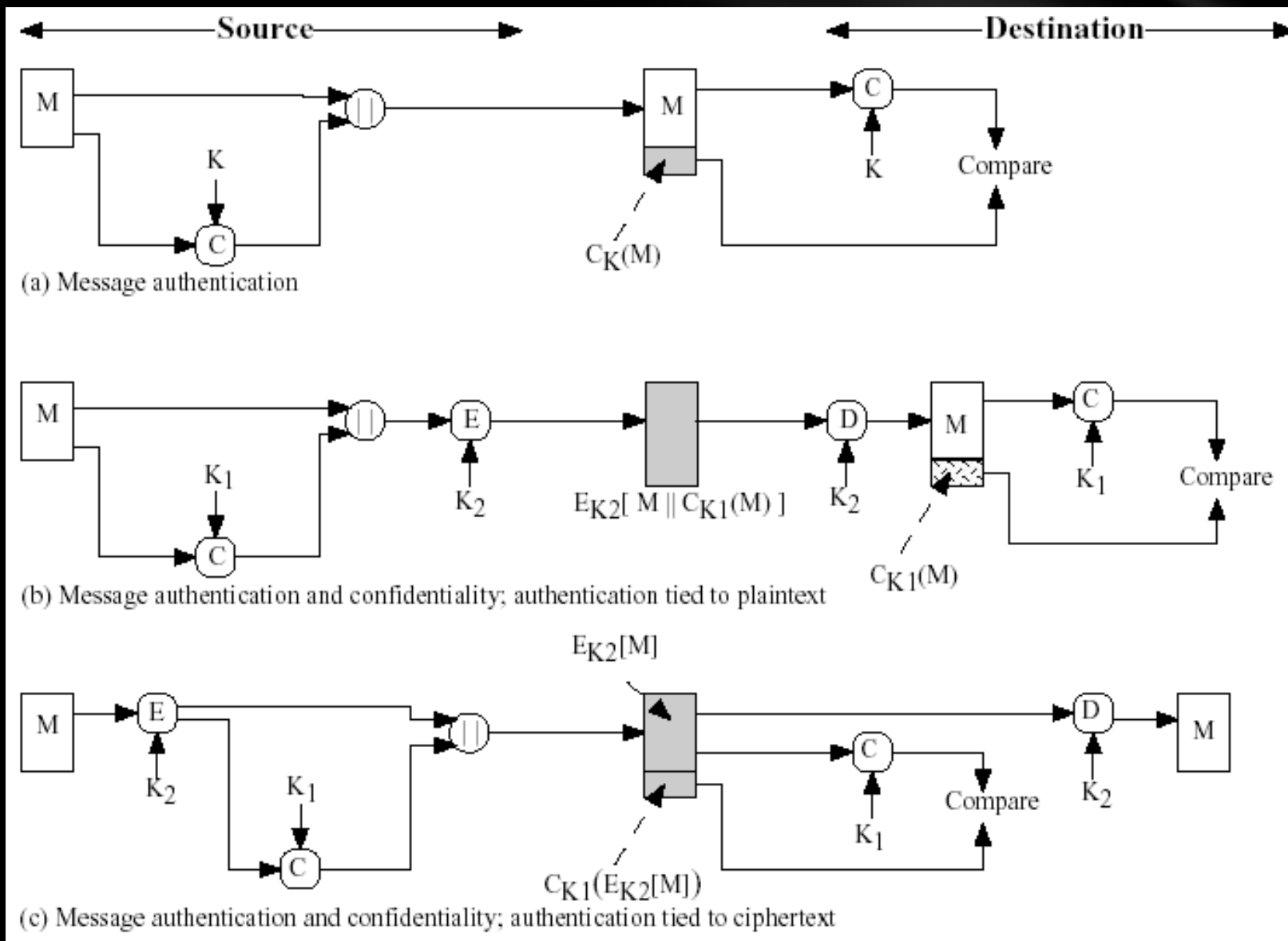


- CBC-MAC computing:
 - Encryption algorithm $E()$ uses key k to encrypt plaintext m ;
 - Divide plaintext m into x parts (m_1, \dots, m_x); the length of every part equals the input size of $E()$
 - $O_0 = 0x00000000 \dots 00$
 - For $i = 1 \dots x$:
 - $O_i = E_k(O_{i-1} \text{ XOR } m_i)$
 - O_x is the final authentication code MAC

Difference of MAC & Digital Signature

- MAC: the sender and receiver need to share a “secret” key
 - MAC can only be verified by a special receiver.
 - MAC doesn’t provide the service of “Undeniable”: all the people who can verify the confidentiality of a message can also generate a MAC.
- Digital Signature: needn’t share any “secret” information
 - Digital Signature can be verified by all the receivers who have the sender’s public key;
 - Digital Signature provides the service of “ Undeniable.”
- Why not always DS? When do we use MAC?
 - MAC is much faster than DS
 - Undeniability is not always required.

Combination of MAC & Cryptography



Introduction of PGP

PGP — Pretty Good Privacy

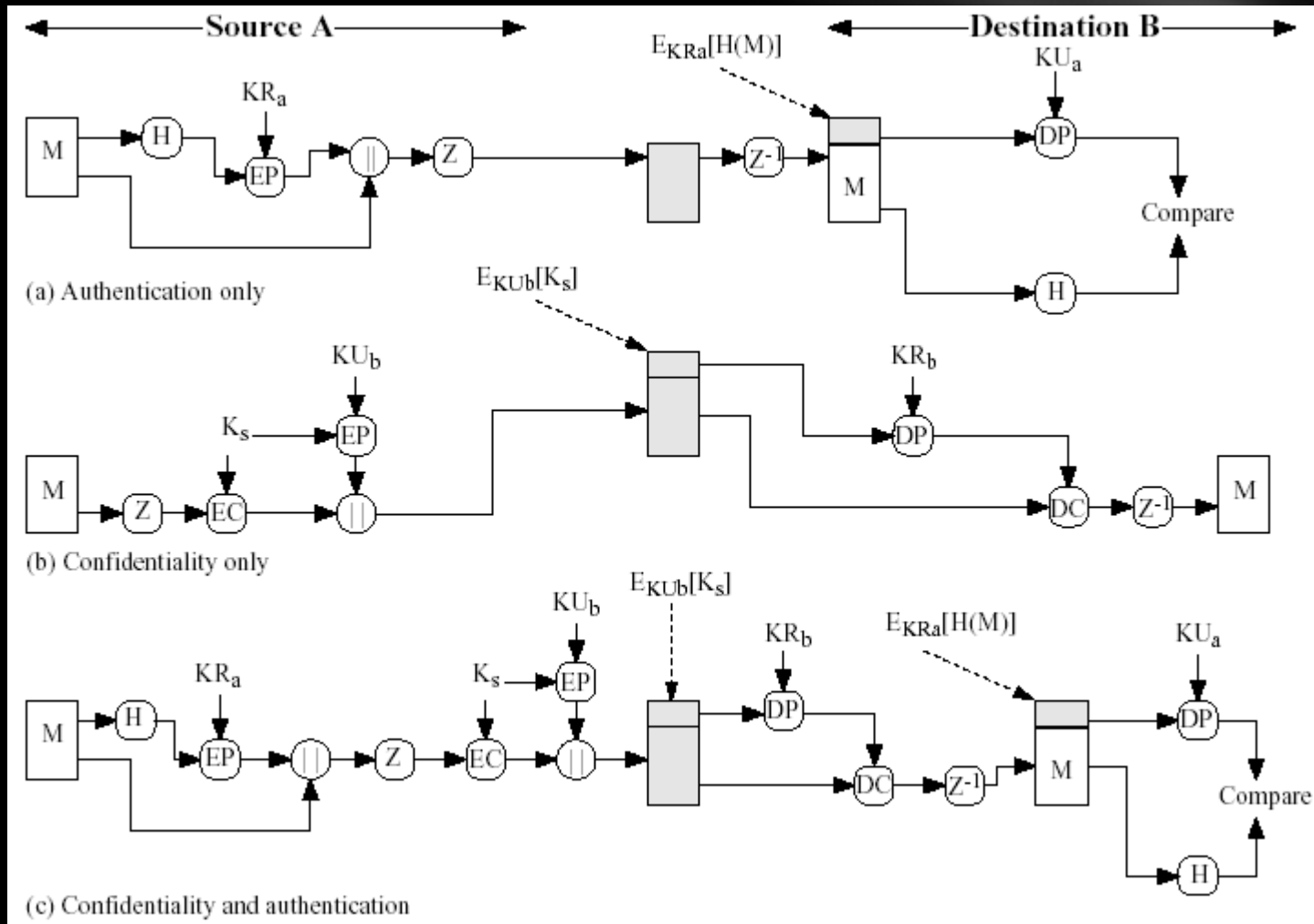
- Widely used in e-mail and file storage security applications, it provides the following services:
 - Digital signature; integrity verification; information encryption
 - Data compression; email format compatibility; data stripping
- Support multiple platforms (DOS/Windows、Unix、Macintosh etc.)
 - It used to be free in early versions, but you need to pay for use now
 - The source code is free ...
- Based on proven security-related algorithms
 - RSA、DSS、Diffie-Hellman、CAST-128、IDEA、3DES、SHA—1、MD5
 - PGP integrates these algorithms and forms common applications independent of the OS and hardware.
- Father of PGP — Phil Zimmermann
- Reference:
 - <http://www.philzimmermann.com/ZH/faq/index.html>
 - <http://www.symantec.com/pgp>
 - <https://www.gnupg.org>



History of PGP

- In 1991, Phil Zimmermann wrote the first PGP encryption software, providing the commercial version, free non-commercial version, and containing all the source code.
- In February 1993, PGP encryption software was investigated by the U.S. government for violations of export control laws --At that time, the United States export control laws did not allow software with an encryption key of more than 40 bits to be exported -- PGP never used encryption whose key less than 128 bit.
- In 1995, "PGP Source Code and Internals," MIT Press
- In July 1997, Zimmermann and the company PGP Inc. Submitted the OpenPGP standard to the IETF. In December 1997, PGP Inc. was purchased by NAI (Network Associates, Inc.) , and NAI refused to continue opening the source code
- In February 2002, NAI stopped technical support for PGP products; NAI is now McAfee
- In August 2002, the early members of the PGP development team formed a new company, PGP Corporation, and the purchased intellectual property rights of PGP from NAI
- In April 2010, PGP Corporation was acquired by Symantec for a price of U.S. \$ 370 million. PGP no longer provides separate software. Its functionality is integrated into Symantec's security software.

PGP Secure Mode



Reviews

- The concepts, characteristics and method of digital signatures
- Concepts of one-way hash algorithm
 - MD5 Algorithm
 - Attack of One-way hash algorithm
- Concepts of MAC
 - CBC-MAC
 - HMAC
- PGP