Message Authentication

Network Security (NETSEC)

Yuhong Li

2022-01-28

Outline

- Message authentication techniques
  - MAC (Message Authentication Code)
  - Hash functions
    - SHA
  - HMAC

Message Authentication

Stockholm University

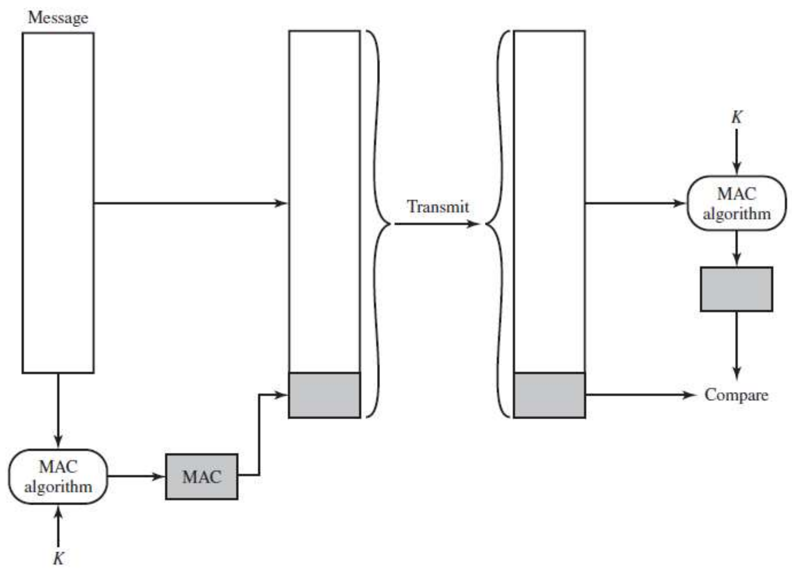
- Message authentication
  - The message (data, file, video, audio, document) is genuine and comes from its alleged source.
  - Allows communicating parties to verify that received messages are authentic.
    - The contents of the message have not been altered -> integrity
    - The source is authentic
    - Also, timeliness: has not been replayed or artificially delayed
- Used to provide integrity
  - …but not confidentiality
- Uses
  - Data transmission
  - OS system files and other stored data
  - Add-ons of web pages
  - …
- Symmetric and asymmetric encryption techniques can be used

Message Authentication Code (MAC)

Stockholm University

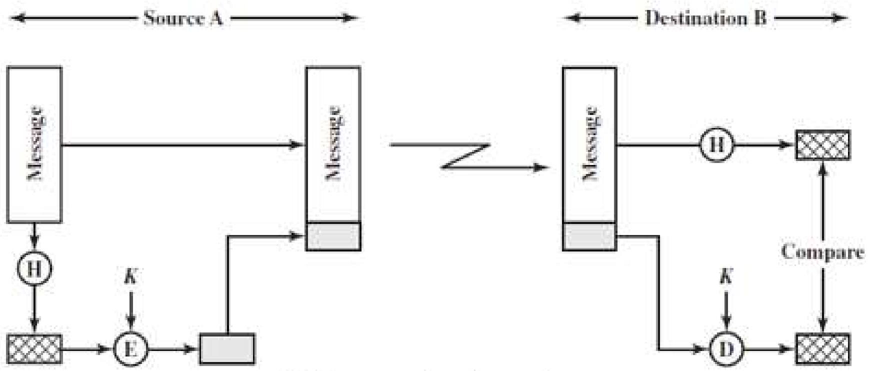
- A small block of data appended to a message
- Assume that Alice and Bob share a common secret key K_{AB}
  - Alice calculates the MAC as a function of the message M and key K_{AB}, $MAC_M = F(K_{AB}, M)$
- MAC consists of two algorithms:
  - a signing algorithm S, S(K, M) outputs in tag (t)
  - a verification algorithm V

    V(K, M, t) outputs in OK or Not OK
- Bob (receiver) is assured that
  - The message is not altered (i.e., the MAC is OK),
  - It is from the claimed sender (the only other person that knows the secret key)
- Methods to generate the code
  - The authentication algorithm need not be reversible
  - DES (NIST, FIPS PUB 113)
  - …



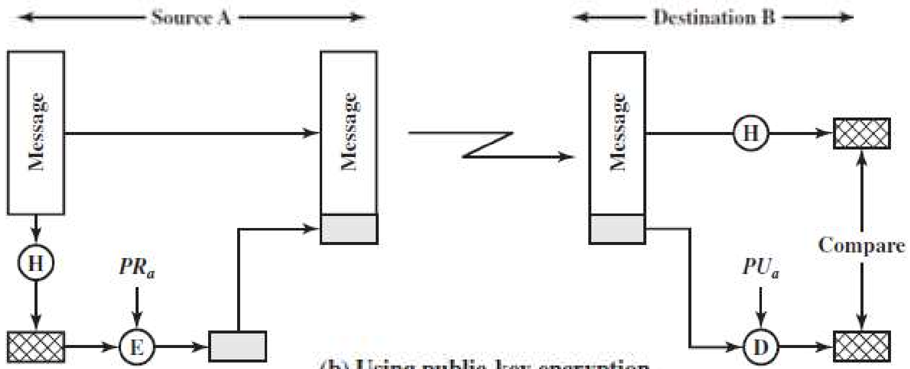
Methods for Message Authentication

Stockholm University



(a) Using conventional encryption

- Message digest
- Encrypt the message digest by using symmetric or public key



(b) Using public-key encryption

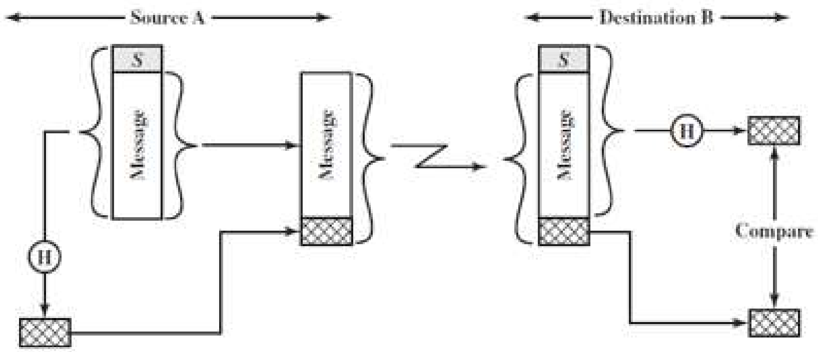
One Way Hash Functions for Authentication -1

- Authentication algorithm need not be reversible
- Encryption
  - Encryption software is quite slow
  - Encryption hardware costs, encryption hardware is optimized toward large data sizes
  - An encryption algorithm may be protected by a patent

- One way hash function: a technique that uses a hash function but no encryption for message authentication:
  - Ensures that the message has not been altered
  - Used in digital signatures and key distribution
  - Purpose: To produce a “fingerprint” of a file/message/block of data
  - Also used for authentication in HMAC and data corruption detection

One Way Hash Functions for Authentication - 2

- A technique that uses a hash function but no encryption for message authentication:
  - Assume that A and B share secret value S_{AB}.
  - A wants to send message to B and calculates the hash function over the concatenation of the secret value and the message
  - MD_{M}. = $H(S_{AB}$. || M). A sends [M||M_{DM}] to B
  - The secret value is not sent, so it is not possible to modify the message.



(c) Using secret value

Hash Functions

- Hash = message digest = hash value
- Hash function: maps a large message to a small tag, i.e. H
  - Input: variable-length block of data M
  - Output: a fixed-size hash value h= H(M)
- Does not require a secret key as additional input

Secure Hash Function Requirements

Stockholm University

- A hash function H must have the following properties:
  - H can be applied to a block of data of any size
  - H produces a fixed-length output
  - H(x) is relatively easy to compute for any given x
  - H(x) is one-way: For any given code h it is computationally infeasible to find x such that H(x)=h
  - H(x) is collision resistant:
    - Weak collision resistant: For any given block x, it is computationally infeasible to find y ≠ x with H(y) = H(x).
    - Strong collision resistant: It is computationally infeasible to find any pair (x, y) such that H(x) = H(y).

Security of Hash Functions

- There are two approaches to attack a secure hash function:
  - Cryptanalysis
    - Involves exploiting logical weaknesses in the algorithm
  - Brute-force attack
    - The strength of a hash function against this attack depends solely on the length of the hash code produced by the algorithm
- Defence: depends on the length of the hash code produced by the algorithm

Reverse Hashes

Stockholm University

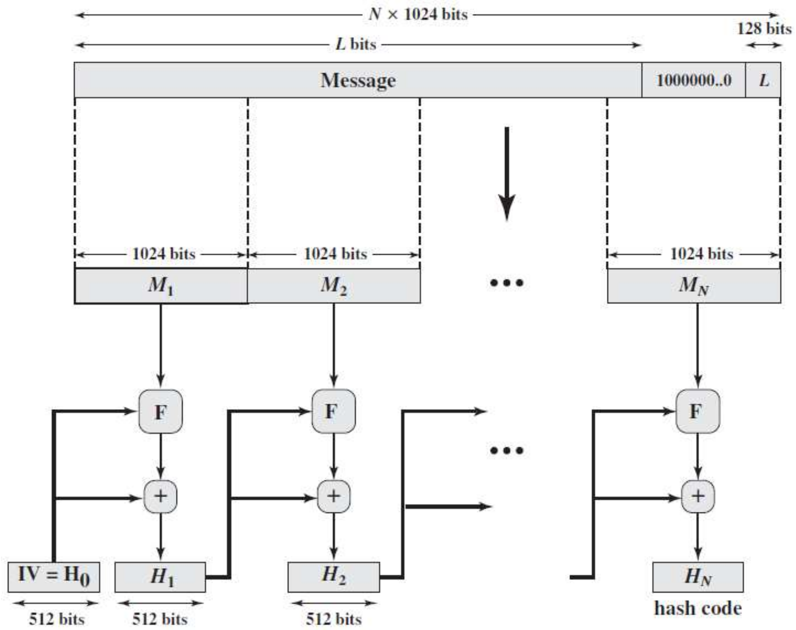
- Hash functions are one way   X -> Y
- Brute force
  - Try all possible combinations of letters & characters
  - Can take a long time, and for a long X, impossible
- Dictionary attack
  - Assume that X is a word or phrase
  - Limits the space of possible combinations dramatically, thus shortening the time to look up different combinations
- Rainbow tables
  - If we compute a lot of different X's and put all the X & Y's in a big table, then for any input Y, we can go to the table and see what X is.
  - This table is called a rainbow table.
  - Requires a lot of storage space (hundreds of gigabytes/several terabytes)

SHA (Secure Hash Algorithm)

- Originally SHA was designed by the NSA and proposed by NIST, US government standard
- **SHA-1**: improved the weakness of SHA-0
  - produces a hash value of 160 bits.
- The **SHA-2** family: have the same underlying structure, the same types of modular arithmetic and logical binary operations as SHA-1.
  - Four cryptographic hash functions: SHA2-224, SHA2-256, SHA2-384, SHA2-512
- The **SHA-3** family, a subset of Keccac, consists of:
  - Four cryptographic hash functions: SHA3-224, SHA3-256, SHA3-384, SHA3-512
  - Two extendable-output functions: SHAKE128, SHAKE256
- Output:
  - For SHA: X-bit (SHA-X) hash value
  - For SHAKE: A variable length value but with the security equivalent to X bits (ShakeX)

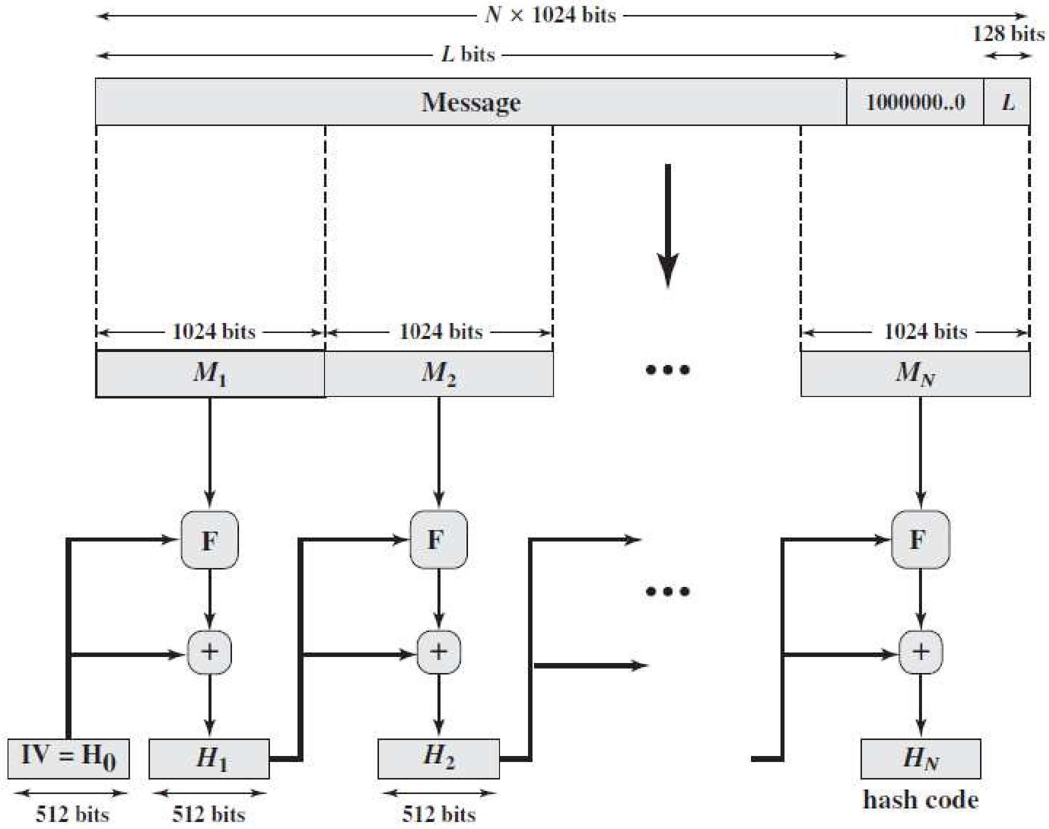
SHA-512

- Input = message with a maximum length of less than 2^{128} bits.
- Output = a 512-bit message digest.
- The input is processed in 1024-bit blocks.
- The heart of the algorithm is a module consisting of 80 rounds, labeled F.



Hash Code Generation

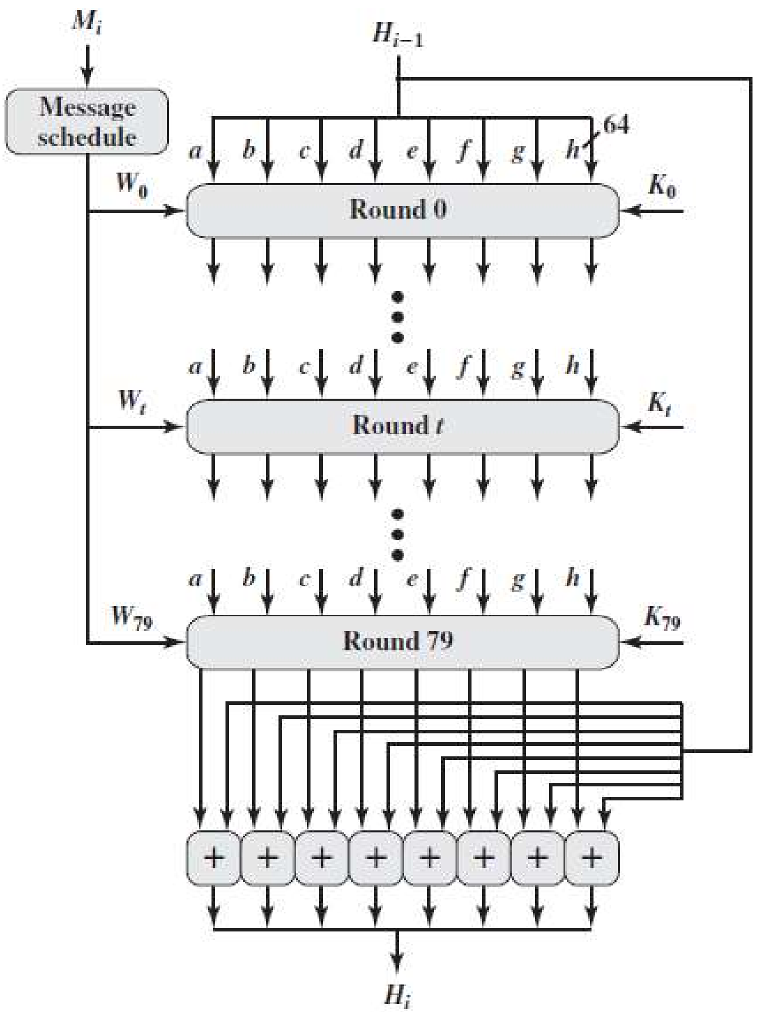
- Step 1: Append padding bits
- Step 2: Append length
- Step 3: Initialize hash buffer
  - A 512-bit buffer (eight 64-bit registers) is used to hold intermediate and final results of the hash function
  - The eight 64-bit registers (H1, H2,etc…) are always initialized to the same 64-bit words in hexadecimal notation.
- Step 4: Process the message in 1024-bit blocks
  - Consists of 80 rounds
- Step 5: Output: 512-bit message digest



Processing in each Single 1024-Bit Block

Input of each round

- A word W_t(64bit), derived from M_i
- Output from last round (512bit),
- a constant K_t(512bit)
  - the first 64 bits of the fractional parts of the cube roots of the first 80 prime numbers.
  - provide a “randomized” set of 64-bit patterns, which should eliminate any regularities in the input data.



Stockholm University

Hash-based Message Authentication Code (HMAC)

Stockholm University

- Hash function: is not designed for message authentication
- Why HMAC
  - Attacks: length extension attacks
  - Cryptographic hash functions execute faster in software than symmetric encryption algorithms such as DES
  - Software libraries for cryptographic hash functions are widely available.
- HMAC has been issued as RFC 2104, has been chosen as the mandatory-to-implement MAC for IPSec, and is used in other Internet protocols, such as
  - Transport Layer Security (TLS)
  - Secure Electronic Transaction (SET)

Stockholm University

Design Objectives of HMAC

- To use, without modifications, available hash functions.
- To allow for easy replaceability of the embedded hash function
- To preserve the original performance of the hash function without incurring a significant degradation
- To use and handle keys in a simple way
- To have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the embedded hash function

Stockholm University

Thank you.

2022-01-28

References

- Stallings. Network Security Essentials
  - Chapter 3