

Hash Algorithms

Yuan Xue

Oct 19 2006



Outline

- ◆ Review of MAC
 - Why we need hash function
- ◆ Requirement for hash function
- ◆ Hash function overview
- ◆ Hash function usage
- ◆ GnuPG help session



Message Authentication – Lessons learned

◆ Using encryption only

- Legitimate plaintext can not be identified automatically without a structure
- Modes of operation provides no data integrity protection
- Encrypting the whole message introduces unnecessary overhead

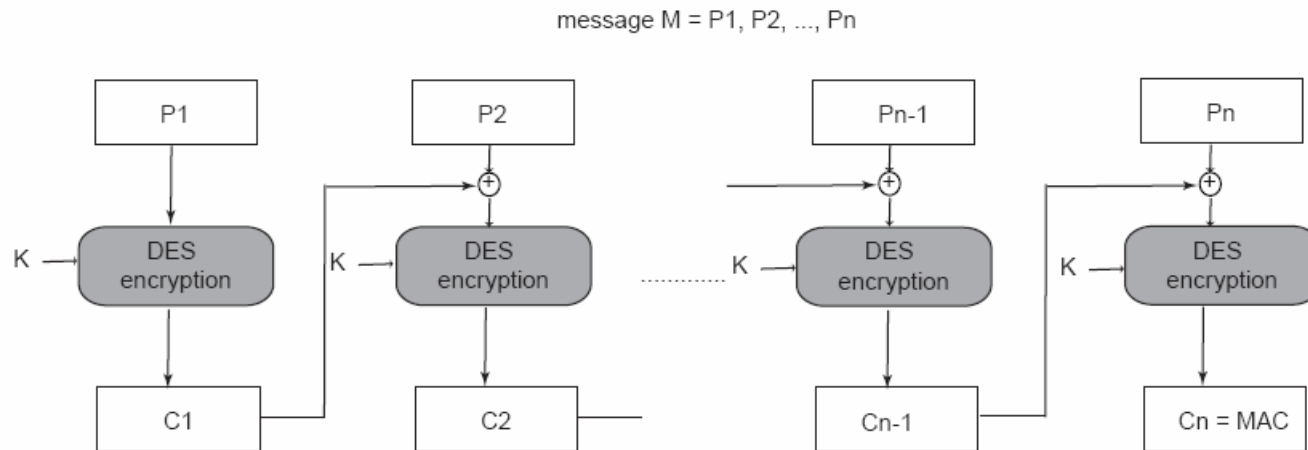
◆ Error detection code (non-cryptographic checksum)

- Provides redundant information for automatically data integrity checking
- Using the code directly can only provide integrity protection against data modification due to natural causes, but not malicious alteration
- Encrypting the error detection code does not work either
 - ◆ Attackers can identify the messages that generate the same error detection code
 - ◆ Attackers can still change the message without being detected even without knowing the value of the code
- Encrypting (message + EDC) still suffers from some attacks.

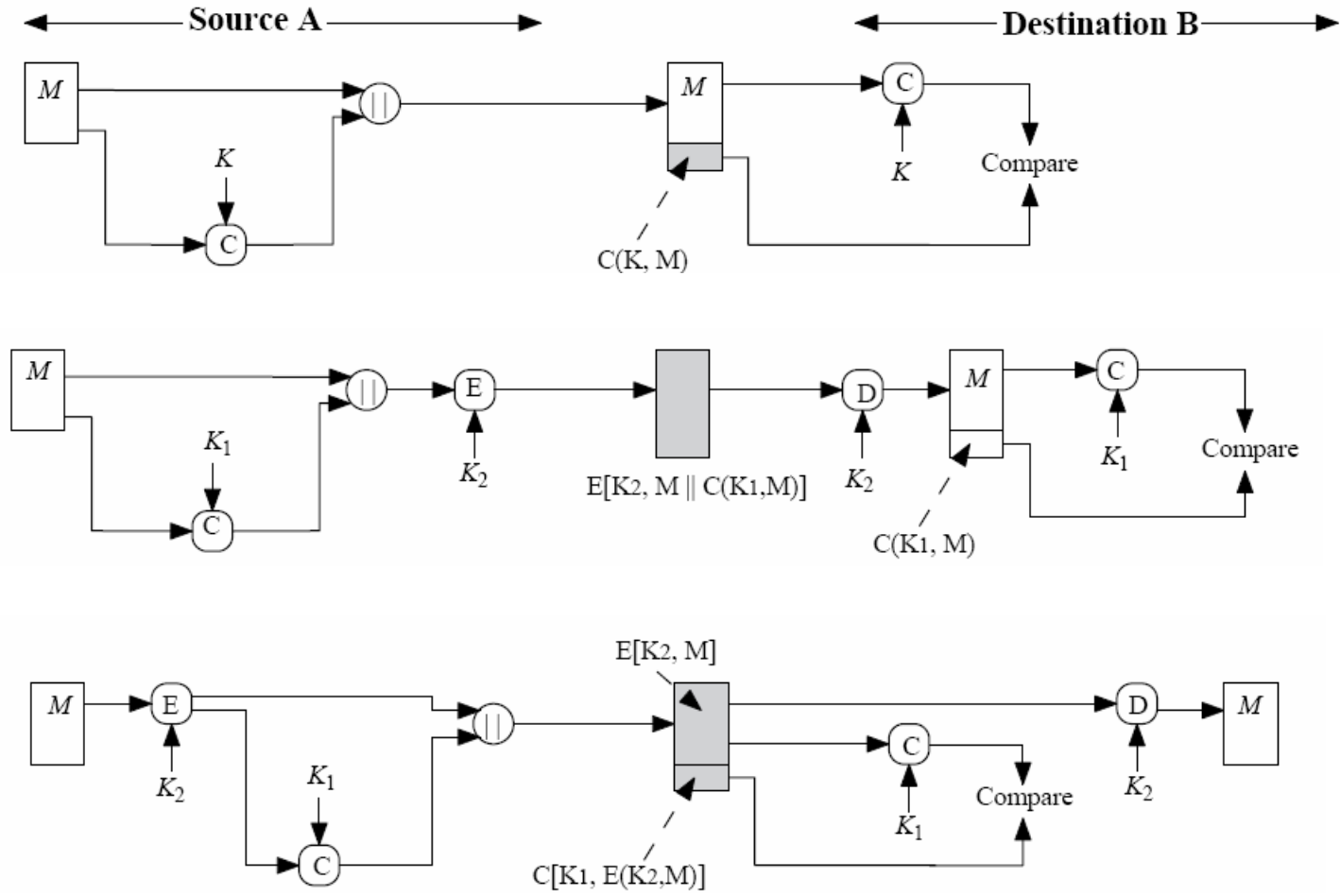
Message Authentication Code

◆ Cryptographic checksum

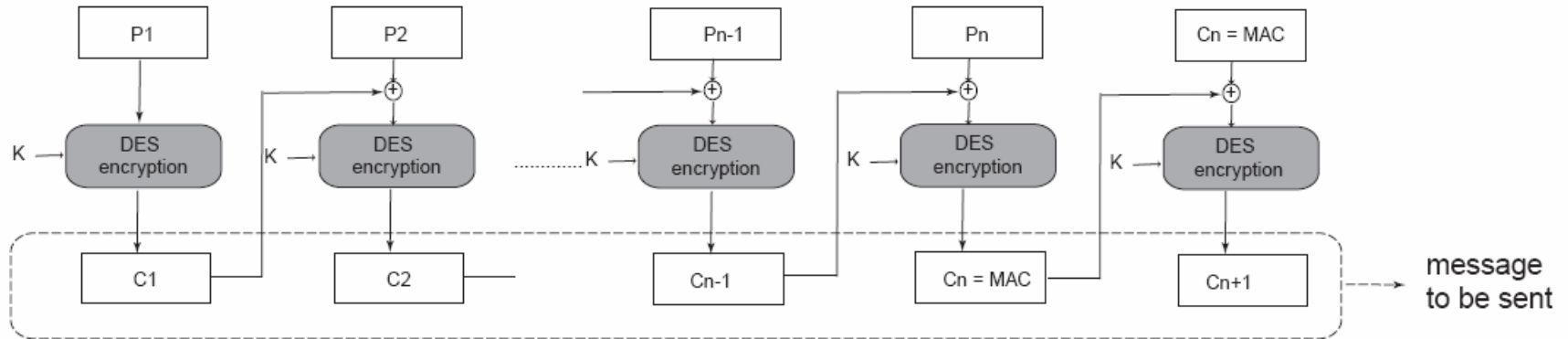
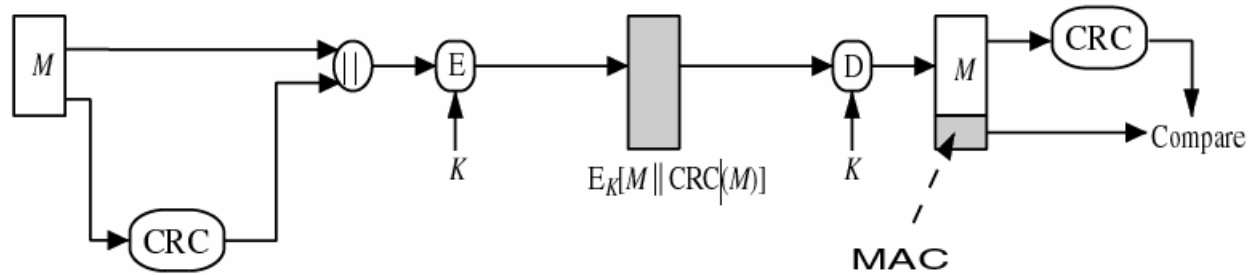
- Utilize a key in generating the code
- $MAC = C(K, M)$
- CBC-based MAC generation algorithm



Review Usage of MAC



How about these two?



Message Authentication Code

◆ Summary of Limitations

- Use of MAC needs a shared secret key between the communicating parties
- MAC does not provide digital signature
- CBC-based MAC generation still involves high computation overhead



Motivation for Hash Algorithms

◆ Intuition

- Re-examine the non-cryptographic checksum.
- Main Limitation
 - ◆ An attack is able to construct a message that matches the checksum

◆ Goal

- Design a code where the original message can not be inferred based on its checksum → design of hash algorithms.

Requirements for Hash function

- ◆ A hash function H takes a message M of variable length and transforms it into a fixed-length hash value h
 - $h = H(M)$
- ◆ A hash function H must have the following properties:
 - One-way property: for any given value h , it is computationally infeasible to find x such that $H(x) = h$.
 - Weak collision resistance: for any given message x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$
 - Strong collision resistance: it is computationally infeasible to find any pair (x,y) , such that $H(x) = H(y)$.



Overview of Hash Algorithms

◆ MD5

- Message-Digest algorithm 5
 - ◆ By Ronald Rivest in 1991 based on MD4
- Digest length: 128-bit
- Weak collision resistance
- Vulnerable to collision attack (no strong collision resistance)

◆ SHA hash functions (all by NSA)

- SHA-0 in 1993; 160-bit hash value
- SHA-1 in 1995; 160-bit hash value
 - ◆ widely used, once considered as the successor to MD5
- SHA-2
 - ◆ SHA-224, SHA-256; SHA384; SHA512
 - ◆ Digest length (based on name)
- SHA-0 and SHA-1 are vulnerable to collision attacks
 - ◆ Recent result on SHA-1: collision attack on SHA-1 that would allow an attacker to select at least parts of the message.

Ref: <http://en.wikipedia.org/wiki/MD5>
http://en.wikipedia.org/wiki/SHA_hash_functions



Hash Function Usage

◆ Used Alone

- File integrity verification
- Public key fingerprint
- Passwd storage

◆ Combined with encryption functions



