

Lecture 11: Hash Algorithm

Yuan Xue

We will proceed with the topic of message authentication in this lecture. In particular, we will study how secure hash algorithms can be designed and used to provide message authentication and digital signature.

I. MOTIVATION

As we have discussed in the last lecture, without any structure that provides a hint, it is difficult for the receiver to identify the legitimate plaintext automatically. And modes of operation provide no data integrity protection. Besides, the whole message needs to be encrypted, which introduces unnecessary overhead for message authentication.

Error detection code (non-cryptographic checksum) provides redundant information for automatically data integrity checking. However, (1) Using the code directly can only provide integrity protection against data modification due to natural causes, but not malicious alteration; (2) Simply encrypting the error detection code does not work either, because the attacker is able to identify the messages that generate the same error detection code. This means that without knowing the value of the code, the attack can still change the message without being detected; (3) Encrypting the whole message still suffers from some attacks.

Message authentication code provides a cryptographic checksum which utilizes a key in generating the code. However, the use of MAC needs a shared secret key between the communicating parties. Besides MAC does not provide digital signature.

Now let's proceed to study Hash algorithms, which provide a solution for the above problems. Let's first re-examine the non-cryptographic checksum. Their main limitation is that an attack is able to construct a message that matches the checksum. Based on this observation, we would like to design a "code" where the original message can not be inferred based on its "checksum". This thinking leads to the design of hash algorithms.

II. REQUIREMENT FOR HASH FUNCTIONS

A hash function H takes a message M of variable length and transforms it into a fixed-length hash value h , as shown below.

$$h = H(M) \tag{1}$$

The 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 $H(x) = H(y)$.

III. HASH FUNCTION DESIGN

There are several similarities in the evolution of hash functions and symmetric block ciphers. Just like symmetric block ciphers that utilize the substitution-permutation network (some ciphers such as DES use Feistel cipher, a special case of S-P network) to encrypt each data block, most hash function designs share the same design structure where multiple rounds of compression functions are applied to each basic processing unit (block).

MD5 and SHA are two most widely used secure hash algorithms. In particular, the MD5 message digest algorithm was developed by Ron Rivest. The algorithm processes the variable-length input message in 512-bit blocks, and produces a 128-bit message digest as output. In this lecture we discuss the design of MD5 in detail. Students can read Section 12.1 to understand the SHA design. The Secure Hash Algorithm (SHA) was developed by National Institute of Standards and Technology (NIST). It is based on the MD4 algorithm. Based on different digest lengths, SHA includes algorithms such as SHA-1, SHA-256, SHA-384, and SHA-512. Table 12.1 in the textbook [WS] provides a comparison of these algorithms.

IV. HASH FUNCTION USAGE

A hash function is only a function of the input message. It does not use a key. Thus, hash functions themselves do not provide any security. However, hash functions can be used with either symmetric or asymmetric key (algorithms) to achieve various security goals. Fig. 1 illustrates these usages.

A. HMAC

Note that in Fig. 1, no encryption is required for message authentication. As hash functions are generally much faster than symmetric encryption algorithms such as DES, this observation shows that hash function can be used to design fast message authentication mechanisms. HMAC is such a message authentication code. Please read Section 12.3 in textbook [WS] for the detailed design of HMAC.

V. SECURITY ANALYSIS

Students are encouraged to read the following material: (1) Birthday Attacks in Section 11.4 of textbook [WS], and (2) Brute-Force Attacks and Cryptanalysis in Section 11.5.

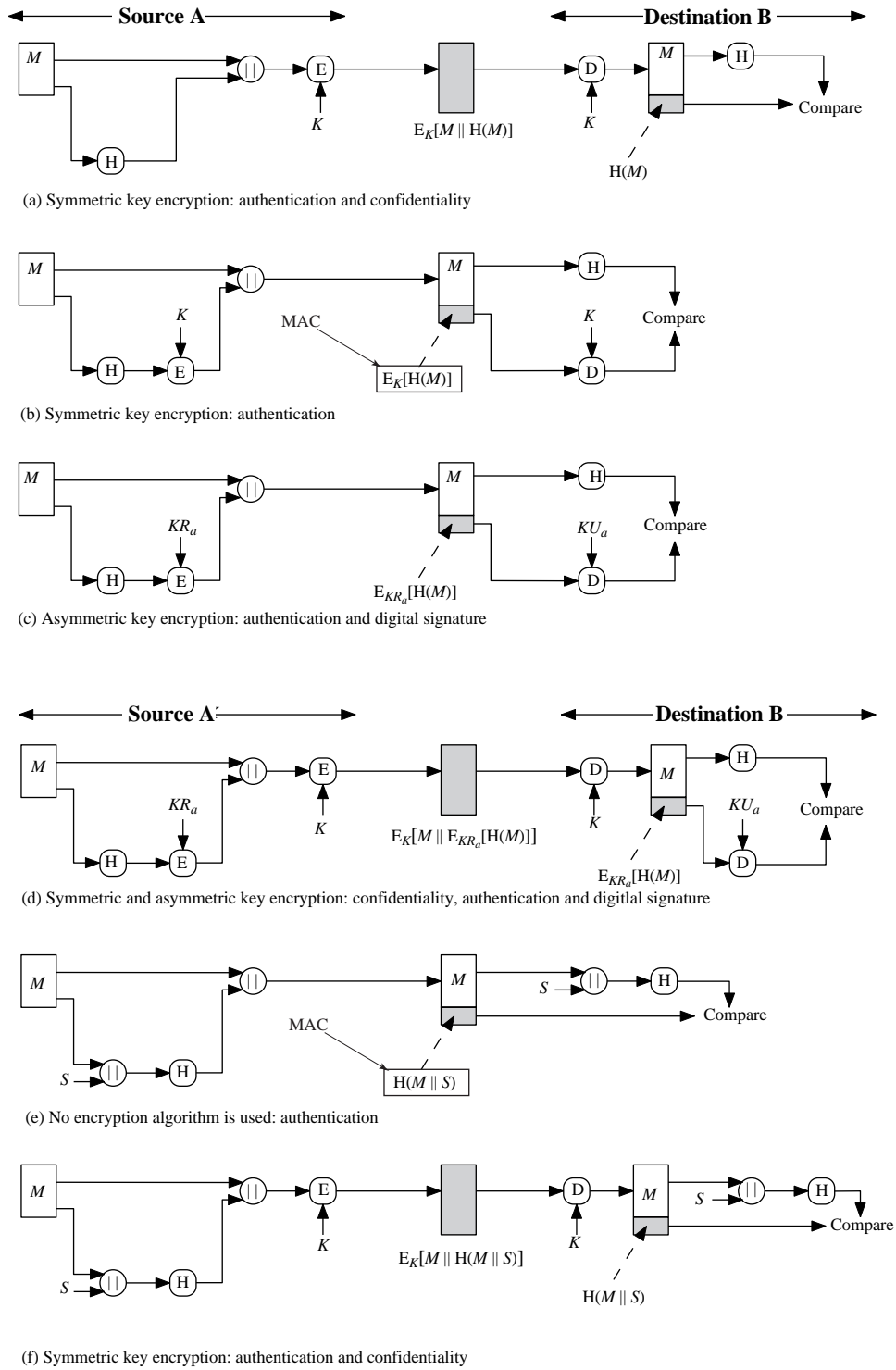


Fig. 1. Basic Usages of Hash Functions.