Class 5

Hash Functions

A hash function is a function that takes an arbitrary string of bits and transform them into a uniform (fixed - size) result.

h is a hash function used in digital signatures.

h(m) is signed instead of signing a message *m* directly.

Reduces the computational work, faster.

Custom outputs 128 and 1024 bits.

Hash function must be secure:

$$m_1 \rightarrow h(m_1)$$
$$m_2 \rightarrow h(m_2)$$

$$if\ h(m_1) = h(m_2)$$
$$then,$$
$$m_1 = m_2$$

Hash Functions

Hash function or *message digest.*

Can be used as in cryptographic pseudorandom number generator to create several keys from a single shared secret.

Security of Hash Function

Must be one way function

$$if\ m_1 = h(m_1)$$
$$there\ is\ not\ a\ y\ such\ as,$$
$$h(m_1) = y$$

Collision resistance property

Not collision free

$$Collision$$
$$if\ m_1 = m_2$$
$$then,$$
$$h(m_1) = h(m_2)$$

Security of Hash Function

Definition 4

The ideal hash function behaves like a random mapping from all possible input values to the set of all possible output values

Definition 5

An attack on a hash function is a non-generic method of distinguishing the hash function from an ideal hash function.

Real Hash Functions

Secure Hash (SHA) family: SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512.

SHA3 Relatively recently published standard by NIST (08/05/2015). Permutation-Based Hash and Extendable-Output Functions.

Iterative hash functions:

Using the padding rule to fill the last block divide the input into a sequence of fixed-size blocks

$m_1, \ldots, m_k$

512 bits/block length, last block shows the inputs length.

$$H_0, \textit{ fixed value}$$
$$H_i = h'(H_{i-1}, m_i)$$
$$H_k \textit{ last block hash function outcome}$$

Message Diagest 5 (MD5)

MD5 is a 128 bit hash function (Ron Rivest, 1992).

MD5 non secure it is already broken.

To compute MD5, in general:

1. Split the message into blocks of 512 bits, last block include the length and is padded.

2. 4 words of 32 bits / words result of the split each state of 128 bits

3. Uses a compression function ***h'*** mixes the message block and the state, 4 rounds in total. A combination of XOR, AND, OR and addition used to mix on 32 bits words (efficient on 32-bit CPUs).

4. The input stat and result are added together to produce the output of ***h'***

Message Diagest 5 (MD5)

If a hash size of 128 bit in MD5 is used is insufficient. Why?

$$N = 2^{128},$$

Birthday Paradox

$$\sqrt{N^{128}} = 2^{64}$$

Work from Wang and YU showed a faster collision finding.

Secure Hash Algorithm (SHA-1)

Designed by NSA and standardized by NIST.

SHA -1 release/published by NIST after NSA and fixed a weakness with SHA-0

SHA-1 main issue is the 160 bit result size. Is this bad, why?

Secure Hash Algorithm
SHA-224, SHA-256, SHA-384, SHA 512

NIST publish in 2001 and updated in 2004 by including another hash a collection of functions know as SHA-2 family of functions.

Designed to be implemented with 128, 192, 256 bit key size of AES and the 3DES 112 bit key size.

SHA-2 family slower than SHA-1.

SHA-3

Took nine years to the NIST to release SHA-3

64 submissions worldwide of proposed algorithms.

Is a new Federal Information Processing Standard FIPS (202). Permutation-based Hash and Extendable-Output Functions.

Based on KECCAK algorithm (wining algorithm).

Consist of four cryptographic hash functions and two extendable-output functions; SHA3-224 (is length 224 bits digests), SHA3-256, SHA3-384, and SHA3-512.

Extendable Output Function (XOF) is a function on bit strings in which the output can be extended to any desired length. SHAKE128 (Secure Hash Algorithm” with “KECCAK) and SHAKE256, 128 and 256 are the security strengths.

Hash Functions weaknesses

Length Extensions

For a message ***m***

$$m \rightarrow m_1 \ldots m_k$$

and hashed to value ***h(m)***

*Consider now **m'***

$$m^{'} \rightarrow m_1 \ldots m_{k+1}$$

*The first **k** blocks of the message **m** is a subset of **m'**. So:*

$$h(m') = h'(h(m), m_{k+1})$$

The Length extension is presented as there is no special processing at the end of the hash function computation.

Hash Functions weaknesses

What is Alice sends a message to Bob and wants to authenticate it by sending

$$h(m) = h(X||m), \; where :$$

X : secret know to Bob and Alice

m : the message

h is a not ideal function

Eve can append text to the message m, updated the authentication code to match the new message.

Hash Functions weaknesses

Partial-Message Collision
Inheriting the iterative structure of most hash functions.

m : the message

$h(m) = h(m||X)$, *where* :

X : the authentication key

Considering a perfect hash function of size ***n*** bits
attacker would choose *m* to get the system to authenticating
as

$$h(m) = h(m||X)$$

Using the birthday attack Eve can find m and m' with the same value

$$h(m) = h(m'||X)$$

If the attack succeeds, it is an iterative hash function; if the attack fails, it is the ideal hash function.

Weakness Fix

The goal is to have a hash function that behaves as` a random mapping.

Tradeoff between detailed design and complexity

SHA-3 addresses this as they are designed to include resistance against collision, preimage, and second preimage attacks.

Short term fix or workaround

Definition 6. Let h be an iterative hash function. The hash function h_{DBL} is defined by:

$$h_{DBL} := h(h(m)||m)$$

Disadvantages is processing slowness and message pre-stage in buffer to perform computations

Efficient short term fix

Definition 7 Let **h** be an iterative hash function, and let **b** denote the block length of the underlying compression function. The hash function $\mathbf{h_d}$ is defined by

$$h_d(m) := h(h(O^b || m))$$

and has a claimed security level of **min(k, n/2)** where k is the security level of **h** and **n** is the size of the hash result.

Choosing a hash function

Recommended by the class text authors

From

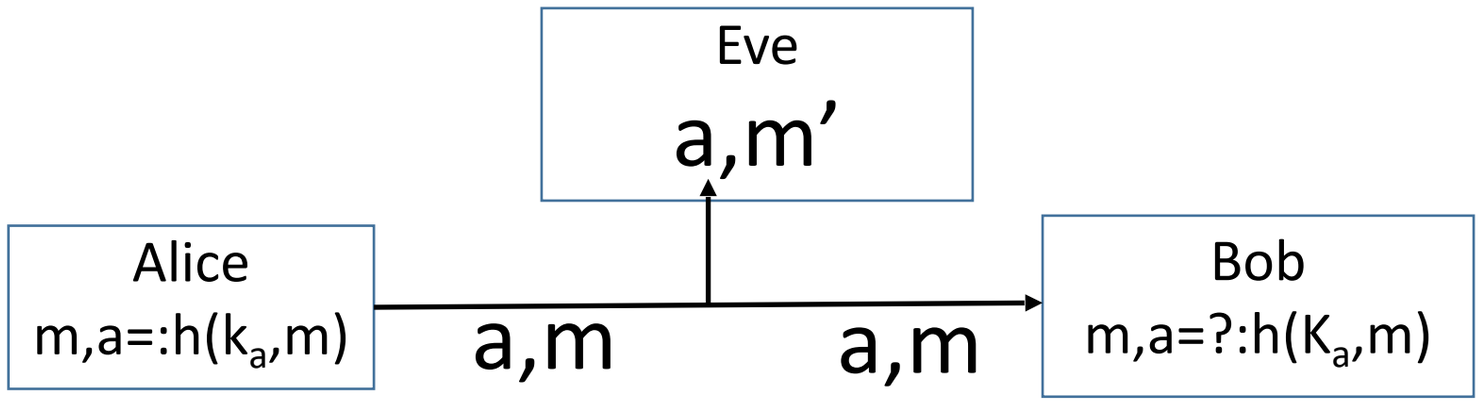
SHA-224, SHA-256, SHA-384 or SHA-512

To SHAd

In any case SHA-3 is here, so it is sensible to start using this family hash.

Message Authentication Codes

Message Authentication Code (MAC) is a system that detects tampering/modifications with messages.



k_a: authentication key.

m: plaintext message.

a: Message Authentication Code (MAC) or Tag *T*

$h(k_a,m)$: h is the MAC function or $MAC(k_a,m)$

The Ideal MAC and MAC security

Definition 8. An ideal MAC function is a random mapping from all possible inputs to n-bit outputs.

Definition 9. An attack on a MAC is a non-generic method of distinguishing the MAC from an ideal MAC function.

Chaining Block Cipher (CBC)-MAC

CBC-MAC

A block cipher (CBC) is used to create a MAC, the message m is encrypted and the last block of cipher text is kept.

So

For a message $P_1, \ldots, P_k$

$$H_0 := IV$$

$$H_i := E_k(P_i \oplus H_{i-1})$$

$$MAC := H_k$$

Common definition of CBC-MAC requires the IV to be fixed at 0

Never user the same key for encryption and authentication

CBC-MAC

Different collision attacks limit the security to half the length of the block size.

A collision attack

Let M be a CBC-MAC function. Knowing M(a)=M(b) leads to know M(a||c)=M(b||c).

c consists of a single block

$$M(a \| c) = E_k(c \oplus M(a))$$
$$M(b \| c) = E_k(c \oplus M(b))$$

So M(a)=M(b)

CBC-MAC

The attacker collect large number of MAC values for a large number of messages, based on the birthday paradox.

Attacker find **a** and **b** for which M(a)=M(b).

Attacker could get the sender to authenticate a||c, then he can replace the message with b||c and not changing the MAC value.

Implementing CBC-MAC

1. Construct a string **s** from the concatenation of **l** and **m**, where **l** is the length of **m** encoded in a fixed-length format.

2. Pad **s** until the length is a multiple of the block size.

3. Apply CBC-MAC to the padded string **s**.

4. Output the last ciphertext block, or part of that block. Do not output any of the intermediate values.

Instead of using CBC-MAC directly use CMAC

CMAC

Standardized by NIST is based on CBC-MAC.

CMAC similar to CBC-MAC with the difference that CMAC treats the last block in a different manner.

CMAC XORs one of two special values (derived from CMAC key) into the last block prior the last block cipher encryption. CMAC key dependency in the length of messages and the cipher’s block length.

Keyed-Hash based MAC

If the goal is to have a MAC behaving as a random mapping as a function of the key and the message why don't we use them with hash functions?

HMAC calculates

Here **a** and **b** are specified constants.

$$h(K \oplus a \,||\, h(K \oplus b \,||\, m))$$

Works with any iterative function

HMAC with SHA-1 less insecure than SHA-1.

HMAC avoids key recovery attacks that would reveal K to the attacker.

HMAC is limited by n/2 bit security

HMAC with SHA-256 a 128 bit security level.

Galois MAC

GMAC efficient in implementation of hardware and software.

Designed for 128 bit block ciphers.

GMAC authentication function takes three values as input:

key, message and a nonce

GMAC uses a universal hash function, especial mathematical function to compute the input message. Then encrypts the output with a block cipher in CTR mode to obtain the tag(MAC).

The IV is created using a function of its nonce.

GMAC

Only provides 64 bits of security.

Do not use GMAC for short MAC values.

Choosing a MAC

As recommended by Authors in the book[1], a possible option is to use HMAC-SHA-256 due to system constrains (64-96 bit MAC values) a security factor might need to be reduced to 64 bits.

SHA3-224, SHA3-256, SHA3-384, and SHA3-512 are NIST approved cryptographic hash functions [2].

Hash Function	SHA3-224	SHA3-256	SHA3-384	SHA3-512
Block Size (bytes)	144	136	104	72

MAC usage

Preventing the reply attack

Including state of the communication *d*

General solution is to use d||m

Horton principle “Authenticate what is meant, not what is said”

Authentication should include, protocol identifier, version number message identifier, sizes of various fields.

Layered OSI protocol and authentication isolation

Bibliography

[1] Ferguson, Niels; Schneier, Bruce; Kohno, Tadayoshi (2011-02-02). Cryptography Engineering: Design Principles and Practical Applications (p. 84). Wiley. Kindle Edition.

[2] http://www.nist.gov/manuscript-publication-search.cfm?pub_id=919061