Introduction to Cryptology

7.2 - Hash functions: Constructions and Applications

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

How to Design a Hash Function?

Is it possible to derive a collision-resistant hash function from a collision-resistant, fixed-length hash function?

How to Design a Hash Function?

Is it possible to derive a collision-resistant hash function from a collision-resistant, fixed-length hash function?

Merkle-Damgård transform is a very famous approach for domain extension.

How to Design a Hash Function?

Is it possible to derive a collision-resistant hash function from a collision-resistant, fixed-length hash function?

Merkle-Damgård transform is a very famous approach for domain extension.

- Used for MD5 and the SHA family.
- Theoretical implication: if you can compress by a single bit, then you can compress by an arbitrary amount of bits!

The Merkle-Damgård Transform

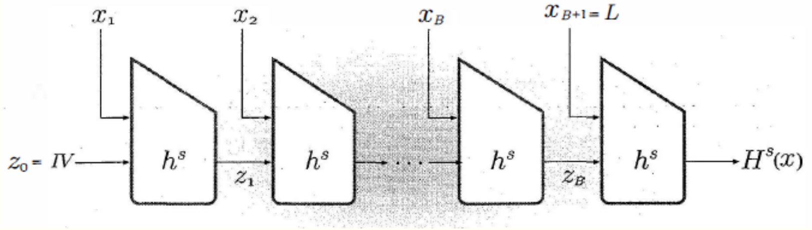
Let (KeyGen, h) be a fixed-length hash function, with $\ell'(n) = 2n$ and $\ell(n) = n$. Define an arbitrary-length hash function

$$(\text{KeyGen}, H)$$

as follows:

- $s \leftarrow \text{KeyGen}(n)$: KeyGen is the same for the two hash functions.
- $H^s(x) \leftarrow H(s, x)$: on input a key s and a string $x \in \{0, 1\}^*$ of length $L < 2^n$, it proceeds as follows:
  - x is padded with zeros to get a string of length $B \cdot n$;
  - $x = (x_1, \ldots, x_B)$ and $x_{B+1} := L$;
  - z_0 (also called IV) is set to 0^n;
  - $z_i := h^s(z_{i-1} || x_i)$, for $i = 1, \cdots, B+1$;
  - $H^s(x) := z_{B+1}$.

The Merkle-Damgård Transform



From [Katz-Lindell].

Theorem

If (KeyGen, h) *is collision-resistant, then so is* (KeyGen, H)*.*

The Merkle-Damgård Transform

Proof.

Let x and x' be two distinct strings s.t. $H^s(x) = H^s(x')$.

Assume $|x| = L$ and $|x'| = L'$. After the padding, $x = x_1, \cdots \cdots, x_{B+1}$ and $x' = x'_1, \cdots, x'_{B+1}$, with $x_{B+1} = L$ and $x'_{B'+1} = L'$.

- $\underline{L \neq L'}$: then $H^s(x) = z_{B+1} = h^s(z_B||L) = h^s(z'_{B'}||L') = z'_{B'+1} = H^s(x')$. Hence $z_B||L \neq z'_{B'}||L'$ is a collision for h^s.
- $\underline{L = L'}$: in this case $B = B'$. Consider $J_i = z_{i-1}||x_i$ and $J'_i = z'_{i-1}||x'_i$ for $i = 1, \ldots, B+2$, where

$$J_{B+2} = z_{B+1} = z'_{B+1} = J'_{B+2}.$$

  Let N be the largest integer s.t. $I_N \neq I'_N$ (which exists since $x \neq x'$). Since $N \leq B+1$, then $h^s(I_N) = z_N = z'_N = h^s(I'_N)$.

$\square$

MACs using Hash Functions

MAC using Hash Functions

Hash functions offer an alternative to CBC-MAC to construct MACs for arbitrary-length messages.

MAC using Hash Functions

Hash functions offer an alternative to CBC-MAC to construct MACs for arbitrary-length messages.

The general idea is simple and widely used.

MAC using Hash Functions

Hash functions offer an alternative to CBC-MAC to construct MACs for arbitrary-length messages.

The general idea is simple and widely used.

- Step 1: a collision-resistant hash function (KeyGen, H) is used to hash a message m into a fixed-length string $H^s(m)$.
- Step 2: a fixed-length MAC is applied to $H^s(m)$.

Hash-and-MAC

Let $S_{MAC} = (\text{KeyGen}_M, \text{Mac}, \text{Verify})$ be a fixed-length MAC for messages of length $\ell(n)$, and (KeyGen_H, H) a hash function with output length $\ell(n)$.

A MAC for arbitrary length messages

$$S'_{MAC} = (\text{KeyGen}', \text{Mac}', \text{Verify}')$$

can be defined as follows.

- $(k, s) \leftarrow \text{KeyGen}'(1^n)$: given a security parameter n, it runs KeyGen_M and KeyGen_H on input n, obtaining two keys, k and s. The output is (k, s).
- $t \leftarrow \text{Mac}'((k, s), m \in \{0, 1\}^*)$: $t := \text{Mac}_k(H^s(m))$.
- $1/0 \leftarrow \text{Verify}'((k, s), m, t)$: it outputs 1 if $\text{Verify}_k(H^s(m), t)$ is equal to 1, 0 otherwise.

Hash-and-MAC

Theorem

If S_{MAC} is a secure MAC for messages of length $\ell(n)$ and (KeyGen_H, H) is a collision-resistant hash function, then S'_{MAC} is a secure MAC for arbitrary-length messages.

HMAC

HMAC is a standardised secure message authentication code that uses two layers of hashing.

It can be viewed as an instantiation of the hash-and-MAC technique.

HMAC is very efficient and widely used in practice.

HMAC

Let (KeyGen_H, H) be a hash function obtained from a fixed-length hash function (KeyGen_h, h), with $\ell'(n) = 2n$ and $\ell(n) = n$, applying the Merkle-Damgård transform. Let opad and ipad be two fixed strings of length n.

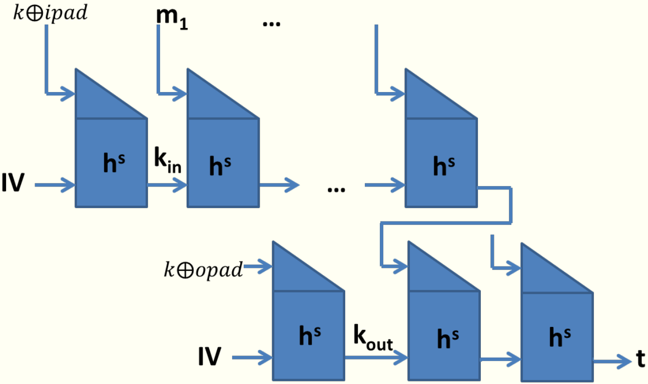
A MAC $S = (\text{KeyGen}, \text{Mac}, \text{Verify})$ for arbitrary-length messages can be defined as follows:

- $(s, k) \leftarrow \text{KeyGen}(n)$: given a security parameter n, it samples a uniform $k \in \{0,1\}^n$ and runs KeyGen_H on input n, obtaining s. It outputs the key (s, k).
- $t \leftarrow \text{Mac}((s,k), m \in \{0,1\}^*)$: it returns the output

$$H^s((k \oplus \text{opad}) || H^s((k \oplus \text{ipad}) || m)).$$

- $1/0 \leftarrow \text{Verify}((s,k), m, t)$: it is the canonical verification.

HMAC



Further Reading I

Mihir Bellare and Phillip Rogaway.
Random oracles are practical: A paradigm for designing efficient protocols.
In Proceedings of the 1st ACM conference on Computer and communications security, pages 62–73. ACM, 1993.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Keccak sponge function family main document.
Submission to NIST (Round 2), 3:30, 2009.

Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya.
Merkle-Damgård revisited: How to construct a hash function.
In Advances in Cryptology–CRYPTO 2005, pages 430–448. Springer, 2005.

Further Reading II

- Morris J Dworkin.
  SHA-3 standard: Permutation-based hash and extendable-output function.
  No. Federal Inf. Process. Stds.(NIST FIPS)-202, 2015.
- Pierre Karpman, Thomas Peyrin, and Marc Stevens.
  Practical free-start collision attacks on 76-step SHA-1.
  In Advances in Cryptology–CRYPTO 2015, pages 623–642. Springer, 2015.
- Neal Koblitz and Alfred J Menezes.
  The random oracle model: a twenty-year retrospective.
  Designs, Codes and Cryptography, pages 1–24, 2015.

Further Reading III

Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone.
Handbook of applied cryptography.
CRC press, 1996.

Marc Stevens.
New collision attacks on SHA-1 based on optimal joint local-collision analysis.
In Advances in Cryptology–EUROCRYPT 2013, pages 245–261. Springer, 2013.

Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov.
The first collision for full SHA-1.
In Annual International Cryptology Conference–CRYPTO 2017, pages 570–596. Springer, CHam, 2005.