Introduction to Cryptology

8.3 - Hash functions: Practical Constructions

Federico Pintore

Mathematical Institute, University of Oxford (UK)

UNIVERSITY OF OXFORD

Michaelmas term 2020

Constructing hash functions

Hash functions used in practice are commonly constructed in two steps:

- a fixed-length collision-resistant hash function h is constructed;
- a techniques (e.g. the Merkle-Damgård transform) is applied to extend h to arbitrary-length inputs.

Hash Functions From Block Ciphers

Block ciphers can be used to build fixed-length collision-resistant hash functions h.

Hash Functions From Block Ciphers

Block ciphers can be used to build fixed-length collision-resistant hash functions h.

There exist different constructions. Davies-Meyer construction is one of the most common.

Hash Functions From Block Ciphers

Block ciphers can be used to build fixed-length collision-resistant hash functions h.

There exist different constructions. Davies-Meyer construction is one of the most common.

Given a block cipher with n as key length and ℓ as block- ength, h is defined as follows:

$$\begin{aligned} h : \{0,1\}^{n+\ell} &\rightarrow \{0,1\}^{\ell} \\ (k,x) &\mapsto h(k,x) = F_k(x) \oplus x. \end{aligned}$$

Hash Functions From Block Ciphers

Not known how to prove the collision resistance of h relying on the assumption that F is a strong pseudorandom permutation.

Something similar to the random oracle model is used.

- F is modelled as an ideal cipher (each key specifies a uniform permutation F_k).
- Each party has to query an oracle to compute $F(k, x)$ or $F^{-1}(k, y)$.

MD5

MD5 was designed in 1991. It is now totally broken, as collisions can be found in less than a minute on a PC!

MD5

MD5 was designed in 1991. It is now totally broken, as collisions can be found in less than a minute on a PC!

MD5 has output length equal to 128.

MD5

MD5 was designed in 1991. It is now totally broken, as collisions can be found in less than a minute on a PC!

MD5 has output length equal to 128. Given an input x:

- the bit 1 is appended;
- the bit 0 is appended until the length of the string is congruent to 448 modulo 512;
- the length of the original input is appended, encoded as a 64-bit string;
- the padded string is divided into blocks of length 512. Each block is divided into 16 chunks M_i of 32 bits each.

MD5

For each block, 64 operations are executed. They are grouped into 4 rounds, each of 16 operations.

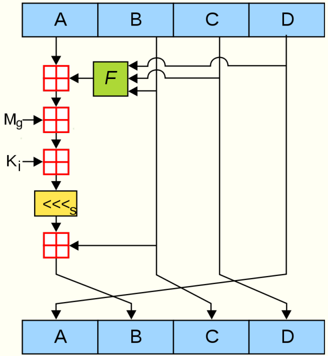
At the very beginning, four 32-bit strings, A, B, C and D, are initialised to constant values.

- The i-th operation sends B in C, C in D and D in A. The new value of B is obtained as

$$B + \lll_{s[i]} (F(B, C, D) + K[i] + M_{g(i)});$$

- $\lll_{s[i]} (\cdot)$ denotes a bit rotation to the left of $s[i]$ places, where s is a fixed list;
- the non-linear function $F(\cdot)$ and the index $g(\cdot)$ vary depending on the round;
- $K[i]$ denotes a 32-bit constant, different for each operation.
- Addition is done modulo 2^{32}.

MD5



One MD5 operation (from wikipedia)

The string obtained after processing a block is added to the value of $(A||B||C||D)$ at the beginning of the four rounds, obtaining a new initial state for the next block.

MD5

Round 1

- $F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$;
- $g(i) = i$.

Round 2

- $F(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$;
- $g(i) = 5i + 1 \pmod{16}$.

Round 3

- $F(B, C, D) = B \oplus C \oplus D$;
- $g(i) = 3i + 5 \pmod{16}$.

Round 4

- $F(B, C, D) = C \oplus (B \vee \neg D)$;
- $g(i) = 7i \pmod{16}$.

Secure Hash Algorithms: SHA-1 and SHA-2

A family of cryptographic hash functions standardised by NIST.

Secure Hash Algorithms: SHA-1 and SHA-2

A family of cryptographic hash functions standardised by NIST.

First, the Davies-Meyer construction is used to obtain a fixed-length collision-resistant hash function from a block cipher.

Secure Hash Algorithms: SHA-1 and SHA-2

A family of cryptographic hash functions standardised by NIST.

First, the Davies-Meyer construction is used to obtain a fixed-length collision-resistant hash function from a block cipher.

The block ciphers were specifically designed for this purpose.

- SHACAL-1 - with block length equal to 160 - for SHA1.
- SHACAL-2 - with block length equal to 256 - for SHA2.
- Keys are 512-bit strings in both block ciphers.

Secure Hash Algorithms: SHA-1 and SHA-2

A family of cryptographic hash functions standardised by NIST.

First, the Davies-Meyer construction is used to obtain a fixed-length collision-resistant hash function from a block cipher.

The block ciphers were specifically designed for this purpose.

- SHACAL-1 - with block length equal to 160 - for SHA1.
- SHACAL-2 - with block length equal to 256 - for SHA2.
- Keys are 512-bit strings in both block ciphers.

Second, they are extended to handle arbitrary-length inputs using the Merkle-Damgård transform.

SHA-1

SHA-1 was introduced in 1995. Its use is not recommended (in 2017, a collision was obtained after $\approx 2^{63}$ SHA-1 evaluations).

It has output length equal to 160.

SHA-1

SHA-1 was introduced in 1995. Its use is not recommended (in 2017, a collision was obtained after $\approx 2^{63}$ SHA-1 evaluations).

It has output length equal to 160. Given an input x:

- the same padding of MD5 is executed;
- after the padding, each 512-bit block is expanded into eighty 32-bit chunks W_j;
- for $17 \leq j \leq 80$, $W_j := W_{j-3} \oplus W_{j-8} \oplus W_{j-14} \oplus W_{j-16}$.

SHA-1

For each block, 80 operations are executed. They are grouped into 4 rounds, each of 20 operations.

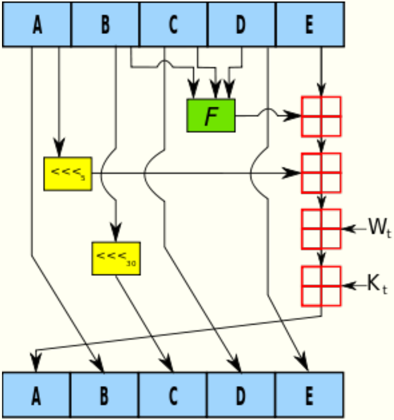
At the very beginning, five 32-bit strings, A, B, C, D and E, are initialised to constant values.

- The t-th operation sends A in B, $\lll_{30}(B)$ in C, C in D and D in E. The new value of A is obtained as

$$\lll_5 (A) + E + F(B, C, D) + K[t] + W_t;$$

- the non-linear function $F(\cdot)$ varies depending on the round;
- $K[t]$ denotes a 32-bit constant, different for each round.
- Addition is done modulo 2^{32}.

SHA-1



One SHA-1 operation (from wikipedia)

A blogpost on the functioning of SHA-1: http://www.metamorphosite.com/one-way-hash-encryption-sha1-data-software

SHA-1

Round 1

- $\mathrm{F}(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$.

Round 2

- $\mathrm{F}(B, C, D) = B \oplus C \oplus D$.

Round 3

- $\mathrm{F}(B, C, D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$.

Round 4

- $\mathrm{F}(B, C, D) = B \oplus C \oplus D$.

SHA-1

<u>Round 1</u>

- $\mathrm{F}(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$.

<u>Round 2</u>

- $\mathrm{F}(B, C, D) = B \oplus C \oplus D$.

<u>Round 3</u>

- $\mathrm{F}(B, C, D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$.

<u>Round 4</u>

- $\mathrm{F}(B, C, D) = B \oplus C \oplus D$.

The string obtained after processing a block is added to the value of $(A||B||C||D||E)$ at the beginning of the four rounds, obtaining a new initial state for the next block.

SHA-2

SHA-2 is a set of hash functions designed by the NSA and published in 2001.

The output length is equal to 256.

SHA-2

SHA-2 is a set of hash functions designed by the NSA and published in 2001.

The output length is equal to 256. Given an input x:

- the same padding of MD5 and SHA-1 is executed;
- after the padding, each 512-bit block is expanded into sixty-four 32-bit chunks W_j;
- for $17 \leq j \leq 80$:

$$
\begin{aligned}
s_0 &:= (\ggg_7 (W_{j-15}) \oplus (\ggg_{18} (W_{j-15}) \oplus (\ggg_3 (W_{j-15}) \\
s_1 &:= (\ggg_{17} (W_{j-2}) \oplus (\ggg_{19} (W_{j-2}) \oplus (\ggg_{10} (W_{j-2}) \\
W_j &:= W_{j-16} + s_0 + W_{j-7} + s_1
\end{aligned}
$$

SHA-2

For each block, 64 operations are executed.

At the very beginning, eight 32-bit strings, A, B, C, D, E, F, G and H are initialised to constant values.

- The t-th operation sends A in B, B in C, C in D, E in F, F in G and G in H. The new value of A is obtained as

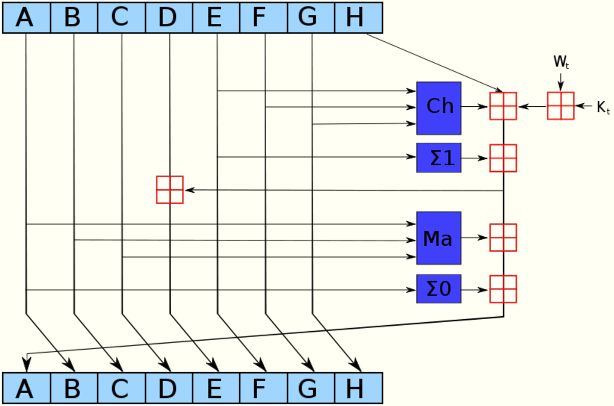
  $$W_t + K[t] + H + Ch(E, F, G) + \Sigma_1(E) + Ma(A, B, C) + \Sigma_0(A).$$

  The new value of E is obtained as

  $$W_t + K[t] + H + Ch(E, F, G) + \Sigma_1(E) + D.$$

- $K[t]$ denotes a 32-bit constant, different for each round.
- Addition is done modulo 2^{32}.

SHA-2



One SHA-2 operation (from wikipedia)

A detailed description of SHA-2:
http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf

SHA-256

The logical functions are as follows:

- $Ch(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$
- $Ma(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
- $\Sigma_0(A) = (\ggg_2 (A)) \oplus (\ggg_{13} (A)) \oplus (\ggg_{22} (A))$
- $\Sigma_1(E) = (\ggg_6 (E)) \oplus (\ggg_{11} (E)) \oplus (\ggg_{25} (E))$

SHA-256

The logical functions are as follows:

- $Ch(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$
- $Ma(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
- $\Sigma_0(A) = (\ggg_2 (A)) \oplus (\ggg_{13} (A)) \oplus (\ggg_{22} (A))$
- $\Sigma_1(E) = (\ggg_6 (E)) \oplus (\ggg_{11} (E)) \oplus (\ggg_{25} (E))$

The constant words, $K[1], \cdots, K[64]$ are the first 32 bits of the fractional parts of the cube roots of the first sixty-four primes.

SHA-256

The logical functions are as follows:

- $Ch(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$
- $Ma(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
- $\Sigma_0(A) = (\ggg_2 (A)) \oplus (\ggg_{13} (A)) \oplus (\ggg_{22} (A))$
- $\Sigma_1(E) = (\ggg_6 (E)) \oplus (\ggg_{11} (E)) \oplus (\ggg_{25} (E))$

The constant words, $K[1], \cdots, K[64]$ are the first 32 bits of the fractional parts of the cube roots of the first sixty-four primes.

The string obtained after processing a block is added to the value of $(A||B||C||D||E||F||G||H)$ before the 64 operations, obtaining a new initial state for the next block.

SHA-3 (Keccak)

In 2012, Keccak was announced as the winner of the NIST competition (called SHA-3) to design a new (family of) hash function(s).

It uses an unkeyed permutation f with block length equal to 1600.

Keccak uses a new construction, named sponge construction for the domain extension.

SHA-3 (Keccak)

The permutation f operates on blocks of length 1600. The output length d lies in $\{224, 256, 384, 512\}$.

A rate r is fixed. The capacity is $c = 1600 - r$. The bigger c, the bigger the number of bits of security and the slower the execution.

The digest is obtained after two phases: the absorbing phase, and the squeezing phase.

SHA-3 - Absorbing

The initial input is padded:

- the bit 1 is appended;
- the bit 0 is appended until the length of the string is congruent to $r-1$ modulo r;
- the bit 1 is appended.

The string is divided into n blocks $M_0, \ldots, M_{n-1}$ of length r. For each block, an evaluation of f is performed.

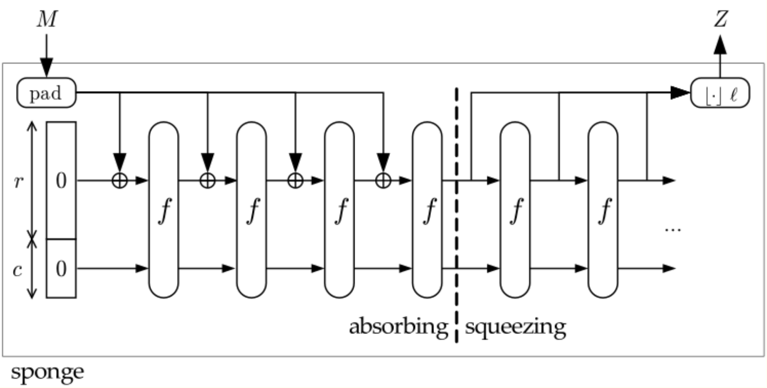
The state S is initialised as a string of 1600 zeros.

- M_i is extended appending $c = 1600 - r$ zeros, obtaining M'_i.
- The new state S is $f(S \oplus M'_i)$.

SHA-3 - Squeezing

- Z is initialised as the empty string.
- While the length of Z is less than d, the first r bits of S are appended to Z and S is updated applying f.
- Z is truncated to d bits.

Keccak- Sponge Function



Complete description: http://sponge.noekeon.org/CSF-0.1.pdf

SHA-3 (Keccak)

The permutation f works on a state S represented by a 5x5 matrix of 64-bit words.

The image is computed by repeating 24 times a round composed by 5 steps: θ, ρ, π, χ and ι.

Only χ is not linear.

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.