Cryptographic hash functions III

Formal security notions

Tanja Lange

Eindhoven University of Technology

2MMC10 – Cryptology

Some terms from complexity theory

- Polynomial time algorithm:
  An algorithm A that on input of length n takes time $t_A \leq \text{poly}(n)$ to complete. Here $\text{poly}(n)$ means

Some terms from complexity theory

- Polynomial time algorithm:
  An algorithm A that on input of length n takes time $t_A \leq \text{poly}(n)$ to complete. Here poly(n) means

  $$\exists d : t_A \in O(n^d)$$

  There exists a d such that time t_A is in $O(n^d)$.

Some terms from complexity theory

- Polynomial time algorithm:
  An algorithm A that on input of length n takes time $t_A \leq \text{poly}(n)$ to complete. Here poly(n) means
  $$\exists d : t_A \in O(n^d)$$
  There exists a d such that time t_A is in $O(n^d)$.
- Probabilistic polynomial time (PPT) algorithm:
  Randomized algorithm taking polynomial time whose answer is correct with some probability.

Some terms from complexity theory

- Polynomial time algorithm:
  An algorithm A that on input of length n takes time $t_A \leq \text{poly}(n)$ to complete. Here poly(n) means
  $$\exists d : t_A \in O(n^d)$$
  There exists a d such that time t_A is in $O(n^d)$.
- Probabilistic polynomial time (PPT) algorithm:
  Randomized algorithm taking polynomial time whose answer is correct with some probability.
- Negligible: very, very small
  A function $f(n)$ is negligible in n if
  $$\exists n_c \geq 0 : \forall n > n_c : f(n) < 1/\text{poly}(n).$$
  There exists an $n_c \geq 0$ such that for all $n > n_c$ it holds that $f(n) < 1/\text{poly}(n)$.

These are asymptotic statements, like O, so describe behavior as parameter n grows.

Cryptographic hash functions - practical definition

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \to \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.

y is fixed and known to be the image of some $x \in \{0,1\}^*$. Typically there are many such x, but it should be computationally hard to find any.

Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

$x \in \{0,1\}^*$ fixes $H(x) = y$. Typically there are many other $x' \neq x$ with the same image, but it should be computationally hard to find any.

Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

This property leaves full flexibility to choose any target y. Nevertheless it should be computationally hard to find any $x \neq x'$ with the same image.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \to \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.

Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \to \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $y \in H(\{0,1\}^*)$ finds x with $H(x) = y$.
Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \to \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $y \in H(\{0,1\}^*)$ finds x with $H(x) = y$.
Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $x \in \{0,1\}^*$ finds $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$.
Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \to \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $y \in H(\{0,1\}^*)$ finds x with $H(x) = y$.
Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $x \in \{0,1\}^*$ finds $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$.
Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.
Collisions exist.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \rightarrow \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $y \in H(\{0,1\}^*)$ finds x with $H(x) = y$.
Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $x \in \{0,1\}^*$ finds $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$.
Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Collisions exist. There *exists* an attack that outputs a collision, even if we do not know how to find it.

Towards a formal treatment of hash functions

A cryptographic hash function H maps bit strings of arbitrary length to bit strings of length n.

$$H : \{0,1\}^* \rightarrow \{0,1\}^n$$

The input space might be further restricted.

A secure hash function satisfies the following 3 properties:

Preimage resistance: Given $y \in H(\{0,1\}^*)$ finding $x \in \{0,1\}^*$ with $H(x) = y$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $y \in H(\{0,1\}^*)$ finds x with $H(x) = y$.
Second preimage resistance: Given $x \in \{0,1\}^*$ finding $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.
Formally: there does not exist an attack faster than $O(2^n)$ that given $x \in \{0,1\}^*$ finds $x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$.
Collision resistance: Finding $x, x' \in \{0,1\}^*$ with $x \neq x'$ and $H(x') = H(x)$ is hard.

Collisions exist. There *exists* an attack that outputs a collision, even if we do not know how to find it. Formalize ignorance?

Formal treatment of hash functions I

Make statements about families of hash functions or keyed hash functions. Note the "key" is public and not under the control of the attacker.

Formal treatment of hash functions I

Make statements about families of hash functions or keyed hash functions. Note the "key" is public and not under the control of the attacker.

A keyed cryptographic hash function H maps a key of length n and a bit string of length $\ell(n)$ to a bit string of length n.

$$H : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^n$$

Preimage resistance: For any PPT algorithm A

$$\Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^{\ell(n)}, y \leftarrow H(k,x), x' \leftarrow A(k,y) : H(k,x') = y]$$

is negligible in n.

For any PPT algorithm A the probability that given randomly chosen $k \in \{0,1\}^n$ and given $y = H(k,x)$ for some randomly chosen $x \in \{0,1\}^{\ell(n)}$ the algorithm A outputs $x' \in \{0,1\}^{\ell(n)}$ with $H(k,x') = y$ is negligible in n.

This property is often denoted PRE.

Formal treatment of hash functions II

Second preimage resistance: For any PPT algorithm A

$$\Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^{\ell(n)}, x' \leftarrow A(k,x) : H(k,x') = H(k,x) \wedge x' \neq x]$$

is negligible in n.

For any PPT algorithm A the probability that given randomly chosen $k \in \{0,1\}^n$ and $x \in \{0,1\}^{\ell(n)}$ the algorithm outputs $x' \in \{0,1\}^{\ell(n)}$ with $H(k,x') = H(k,x)$ and $x' \neq x$ is negligible in n.

This property is often denoted SPR.

Formal treatment of hash functions II

Second preimage resistance: For any PPT algorithm A

$$\Pr[k \leftarrow_R \{0,1\}^n, x \leftarrow_R \{0,1\}^{\ell(n)}, x' \leftarrow A(k,x) : H(k,x') = H(k,x) \wedge x' \neq x]$$

is negligible in n.

For any PPT algorithm A the probability that given randomly chosen $k \in \{0,1\}^n$ and $x \in \{0,1\}^{\ell(n)}$ the algorithm outputs $x' \in \{0,1\}^{\ell(n)}$ with $H(k,x') = H(k,x)$ and $x' \neq x$ is negligible in n.

This property is often denoted SPR.

Collision resistance: For any PPT algorithm A

$$\Pr[k \leftarrow_R \{0,1\}^n, (x,x') \leftarrow A(k) : H(k,x') = H(k,x) \text{ and } x' \neq x]$$

is negligible in n.

For any PPT algorithm A the probability that given randomly chosen $k \in \{0,1\}^n$ the algorithm outputs $x, x' \in \{0,1\}^{\ell(n)}$ with $H(k,x') = H(k,x)$ and $x' \neq x$ is negligible in n.

This property is often denoted CR.