REU 2015: Complexity Across Disciplines

Introduction to Cryptography

Symmetric Key Cryptosystems

Hash Functions based on Block Ciphers

Iterated Block Ciphers

Definition

Let $KS : \mathcal{K} \to \mathcal{K}^s$ be a function that produces a set of subkeys $k_i \in \mathcal{K}$, $1 \leq i \leq s$ from any key $k \in \mathcal{K}$. A block cipher Π for which the encryption functions ϵ_k are of the form

$$T[k_s] \circ T[k_{s-1}] \circ T[k_{s-2}] \circ \cdots \circ T[k_1]$$

is called an *iterated block cipher* and $T[k_i]$ is called *ith*-round function.

Hash Functions

Definition

A **hash function** is a function $H : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is a set of strings of arbitrary length, $\mathcal{Y}$ is a finite set of strings of a fixed length and $|\mathcal{X}| > |\mathcal{Y}|$.

[1] *Computationally Infeasible* means solving the underlying problem is not possible within polynomial time

Hash Functions

Definition

A **hash function** is a function $H : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is a set of strings of arbitrary length, $\mathcal{Y}$ is a finite set of strings of a fixed length and $|\mathcal{X}| > |\mathcal{Y}|$.

A hash function H is *one-way hash function* for any $y \in \mathcal{Y}$ it is computationally infeasible[1] to find $x \in \mathcal{X}$ such that $H(x) = y$.

[1] *Computationally Infeasible* means solving the underlying problem is not possible within polynomial time

Hash Functions

Definition

A **hash function** is a function $H : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is a set of strings of arbitrary length, $\mathcal{Y}$ is a finite set of strings of a fixed length and $|\mathcal{X}| > |\mathcal{Y}|$.

A hash function H is *one-way hash function* for any $y \in \mathcal{Y}$ it is computationally infeasible[1] to find $x \in \mathcal{X}$ such that $H(x) = y$.

A hash function H is *second-preimage resistant* or *weakly-collision free* if for a given $x \in \mathcal{X}$ it is computationally infeasible to find $x' \neq x$ such that $H(x) = H(x')$.

[1] *Computationally Infeasible* means solving the underlying problem is not possible within polynomial time

Hash Functions

Definition

A **hash function** is a function $H : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is a set of strings of arbitrary length, $\mathcal{Y}$ is a finite set of strings of a fixed length and $|\mathcal{X}| > |\mathcal{Y}|$.

A hash function H is *one-way hash function* for any $y \in \mathcal{Y}$ it is computationally infeasible[1] to find $x \in \mathcal{X}$ such that $H(x) = y$.

A hash function H is *second-preimage resistant* or *weakly-collision free* if for a given $x \in \mathcal{X}$ it is computationally infeasible to find $x' \neq x$ such that $H(x) = H(x')$.

A hash function H is *first-preimage resistant* or *strongly-collision free* if it is computationally infeasible to find $x, x' \in \mathcal{X}$ such that $x' \neq x$ and $H(x) = H(x')$.

[1] *Computationally Infeasible* means solving the underlying problem is not possible within polynomial time

Groups Generated by Encryption Functions

Let $\mathcal{T}_\Pi = \{\epsilon_k : k \in \mathcal{K}\}$ be the set of all possible encryption transformations. In a cryptosystem the mapping ϵ is a permutation of $\mathcal{M}$.

$$\mathcal{T}_\Pi \subseteq \mathcal{S}_\mathcal{M}$$

Groups Generated by Encryption Functions

Let $\mathcal{T}_\Pi = \{\epsilon_k : k \in \mathcal{K}\}$ be the set of all possible encryption transformations. In a cryptosystem the mapping ϵ is a permutation of $\mathcal{M}$.

$$\mathcal{T}_\Pi \subseteq \mathcal{S}_\mathcal{M}$$

Definition

The group $\mathcal{G} = \langle \mathcal{T}_\Pi \rangle$ is called the *group generated by the cipher*.

Groups Generated by Encryption Functions

Let $\mathcal{T}_\Pi = \{\epsilon_k : k \in \mathcal{K}\}$ be the set of all possible encryption transformations. In a cryptosystem the mapping ϵ is a permutation of $\mathcal{M}$.

$$\mathcal{T}_\Pi \subseteq \mathcal{S}_\mathcal{M}$$

Definition
The group $\mathcal{G} = \langle \mathcal{T}_\Pi \rangle$ is called the *group generated by the cipher*.

If $\mathcal{T}_\Pi = \mathcal{G}$ then the set of permutations $\mathcal{T}_\Pi$ forms a group (the cipher is a group).

Groups Generated by Encryption Functions

Let $\mathcal{T}_\Pi = \{\epsilon_k : k \in \mathcal{K}\}$ be the set of all possible encryption transformations. In a cryptosystem the mapping ϵ is a permutation of $\mathcal{M}$.

$$\mathcal{T}_\Pi \subseteq \mathcal{S}_\mathcal{M}$$

Definition
The group $\mathcal{G} = \langle \mathcal{T}_\Pi \rangle$ is called the *group generated by the cipher*.

If $\mathcal{T}_\Pi = \mathcal{G}$ then the set of permutations $\mathcal{T}_\Pi$ forms a group (the cipher is a group). For such a cipher, multiple encryption doesn't offer better security than single encryption.

Groups Generated by Encryption functions

Group generated by $T[k]$:

$$\mathcal{G}_\tau = \langle T[k] | k \in \mathcal{K} \rangle$$

Groups Generated by Encryption functions

Group generated by $T[k]$:

$$\mathcal{G}_\tau = \langle T[k] | k \in \mathcal{K} \rangle$$

Group generated by an arbitrary composition of s-round functions with independent keys $k_1, k_2, \cdots, k_s \in \mathcal{K}$:

Groups Generated by Encryption functions

Group generated by $T[k]$:

$$\mathcal{G}_\tau = \langle T[k] | k \in \mathcal{K} \rangle$$

Group generated by an arbitrary composition of s-round functions with independent keys $k_1, k_2, \cdots, k_s \in \mathcal{K}$:

$$\mathcal{G}_\tau^s = \langle T[k_s]T[k_{s-1}]\cdots T[k_1] | k_i \in \mathcal{K} \rangle$$

Group generated by any composition of s-round functions permitted by the key schedule $KS : \mathcal{K} \to \mathcal{K}^s$ (group generated by the cipher):

Groups Generated by Encryption functions

Group generated by $T[k]$:

$$\mathcal{G}_\tau = \langle T[k] | k \in \mathcal{K} \rangle$$

Group generated by an arbitrary composition of s-round functions with independent keys $k_1, k_2, \cdots, k_s \in \mathcal{K}$:

$$\mathcal{G}_\tau^s = \langle T[k_s]T[k_{s-1}]\cdots T[k_1] | k_i \in \mathcal{K} \rangle$$

Group generated by any composition of s-round functions permitted by the key schedule $KS : \mathcal{K} \to \mathcal{K}^s$ (group generated by the cipher):

$$\mathcal{G} = \langle T[k_s]T[k_{s-1}]\cdots T[k_1] | KS(k) = (k_1, k_2, \cdots, k_s) \rangle$$

Groups Generated by Encryption functions

Group generated by $T[k]$:

$$\mathcal{G}_\tau = \langle T[k] | k \in \mathcal{K} \rangle$$

Group generated by an arbitrary composition of s-round functions with independent keys $k_1, k_2, \cdots, k_s \in \mathcal{K}$:

$$\mathcal{G}_\tau^s = \langle T[k_s]T[k_{s-1}]\cdots T[k_1] | k_i \in \mathcal{K} \rangle$$

Group generated by any composition of s-round functions permitted by the key schedule $KS : \mathcal{K} \to \mathcal{K}^s$ (group generated by the cipher):

$$\mathcal{G} = \langle T[k_s]T[k_{s-1}]\cdots T[k_1] | KS(k) = (k_1, k_2, \cdots, k_s) \rangle$$

Lemma

For every $s \in \mathbb{N}$, G_τ^s is a normal subgroup of G_τ.

Example: Merkle-Damgård Hash Function

Let $W(k, m)$ denote a block cipher that encrypts given plaintext m using a key k.

Example: Merkle-Damgård Hash Function

Let $W(k, m)$ denote a block cipher that encrypts given plaintext m using a key k. The hash functions based on the Merkle-Damgård scheme use a block cipher as a compression function.

Example: Merkle-Damgård Hash Function

Let $W(k, m)$ denote a block cipher that encrypts given plaintext m using a key k. The hash functions based on the Merkle-Damgård scheme use a block cipher as a compression function. Given a message m consisting of blocks $m_1, m_2, m_3, \cdots, m_t$, the hash function is defined as

$$H_i = W(H_{i-1}, m_i) \oplus m_i \oplus H_{i-1}, 0 \leq i \leq t \tag{1}$$

where H_0 is some initial value.

Algebraic Properties of a Cipher

Small cardinality.

If the cardinality of the group $\mathcal{G}$ is smaller than the cardinality of the key space, then the time for the key search can reduced to $O(\mathcal{K})$.

Intransitivity.

Let l, n denote natural numbers such that $0 < l \leq n$. A group $G \leq \mathcal{S}_n$ is called *l-transitive* if, for any pair $(a_1, a_2, \ldots, a_l)$ and $(b_1, b_2, \ldots, b_l)$ with $a_i \neq a_j$, $b_i \neq b_j$ for $i \neq j$, there is a permutation $g \in G$ with $g(a_i) = b_i$ for all $i \in \{1, 2, \ldots, l\}$. A 1-transitive permutation group is called *transitive*. The group that is not transitive is called *intransitive*.

Algebraic Properties of a Cipher

Imprimitivity.

A subset $B \subseteq X$ is called a *block* of G if for each $g \in G$ either $g(B) = B$ or $g(B) \cap B = \emptyset$. A block B is said to be *trivial* if $B \in \{\emptyset, X\}$ or $B = \{x\}$ where $x \in X$. The group $G \leq \mathcal{S}_n$ is called *imprimitive* if there is a non-trivial block $B \subseteq X$ of G; otherwise G is called *primitive*.