Contents

Chapter 1

Introduction

Cryptography is the art and science of encrypting and decrypting a message.

1.1 Symmetric cipher

A symmetric cipher scheme Π can be viewed as a triplet $(\text{Gen}, \text{Enc}, \text{Dec})$ of algorithms. Suppose $\mathcal{M}$ be the set of all possible messages and $\mathcal{K}$ be the set of all keys. Gen chooses a key $k \in \mathcal{K}$ and then $\text{Enc} : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ encrypts the message m with key k and returns the cipher c. Lastly, $\text{Dec} : \mathcal{C} \times \mathcal{K} \to \mathcal{M} \cup \perp$ decrypts the cipher c with key k and returns either a message or an error, denoted as $\perp$. Without loss of generality we can assume that Gen picks k uniformly from $\mathcal{K}$. Futhermore, Enc can be randomized, however Dec is deterministic and for every message m and key k we must have

$$\text{Dec}_k(\text{Enc}_k(m)) = m$$

1.2 Kerckhoff's principle

Kerckhoff's principle assumes the following for every encryption scheme

1. The encryption and decryption is known to everyone.
2. The security of the scheme is only dependent on the key.

1.3 Attacks

Some possible attacks include (in increasing power)

Cihpertext only Attacker only knows the ciphertexts.

Known-plaintext Attacker knows one or more plaintext/ciphertext generated by the key.

Chosen-plaintext Attacker can obtain encryption of plaintexts of his choice.

Chosen-ciphertext Attacker can obtain decryption of ciphertexts of his choise.

Chapter 2

Perfectly Secret Encryption

2.1 perfectly secure encryption

Let K and M be two random variables, where K is the result of Gen and M is the message. We can assume that they are independent. Furthermore, $C = \text{Enc}_K(M)$ is also a random varible. By the Kerckhoff's principle, we assume that the distribution on M and Enc is known and only K is unknown.

Definition (Perfectly secure encrption): An encryption scheme is perfectly secure if for all $c \in \mathcal{C}$ with $\mathbb{P}(C = c) > 0$:

$$\forall m \in \mathcal{M}, \quad \mathbb{P}(M = m \mid C = c) = \mathbb{P}(M = m) \tag{2.1}$$

Proposition 2.1. *An encryption scheme* Π *is perfectly secure if and only if*

$$\forall m, m' \in \mathcal{M}, \quad \mathbb{P}(\text{Enc}_K(m) = c) = \mathbb{P}(\text{Enc}_K(m') = c) \tag{2.2}$$

Proof. Suppose Π is perfectly secure then (assuming that $\mathbb{P}(M = m) > 0$)

$$\begin{aligned}\mathbb{P}(\text{Enc}_K(m) = c) = \mathbb{P}(C = c \mid M = m) &= \frac{\mathbb{P}(M = m \mid C = c)\mathbb{P}(C = c)}{\mathbb{P}(M = m)} \\ &= \frac{\mathbb{P}(M = m)\mathbb{P}(C = c)}{\mathbb{P}(M = m)} = \mathbb{P}(C = c)\end{aligned}$$

Now if the equation holds for Π then (again assuming that $\mathbb{P}(M = m) > 0$)

$$\begin{aligned}\mathbb{P}(M = m \mid C = c) &= \frac{\mathbb{P}(C = c \mid M = m)\mathbb{P}(M = m)}{\mathbb{P}(C = c)} \\ &= \frac{\text{Enc}_K(m)\mathbb{P}(M = m)}{\sum_{m^*} \mathbb{P}(C = c \mid M = m^*)\mathbb{P}(M = m^*)} \\ &= \frac{\mathbb{P}(M = m)}{\sum_{m^*} \mathbb{P}(M = m^*)} = \mathbb{P}(M = m)\end{aligned}$$

2.2 Prefect adversarial indistinguishability

An encryption scheme is **perfectly indistinguishable** if no adversary $\mathcal{A}$ can succeed with probability better than $\frac{1}{2}$. Formally, we run the following experiment $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}$

1. $\mathcal{A}$ outputs a pair $m_0, m_1 \in \mathcal{M}$.
2. $k = \mathrm{Gen}$ and b - chosen from $\{0,1\}$ uniformly - then the **challenge ciphertext** $c = \mathrm{Enc}_k(m_b)$ is given to $\mathcal{A}$.
3. $\mathcal{A}$ tries to determine the which message was encrypted and then outputs b'.
4. $$\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi} \begin{cases} 1 & b' = b \text{ then } \mathcal{A} \text{ succeeds} \\ 0 & b' \neq b \text{ then } \mathcal{A} \text{ fails} \end{cases}$$

Since $\mathcal{A}$ can guess randomly $\mathbb{P}\big(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi} = 1\big) \geq \frac{1}{2}$ and thus a scheme is perfectly indistinguishable if

$$\mathbb{P}\big(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi} = 1\big) = \frac{1}{2}, \quad \forall \mathcal{A}$$

Proposition 2.2. Π *is perfectly secret if and only if it is perfectly indistinguishable.*

Proof.

$$\mathbb{P}\big(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi} = 1\big) = \mathbb{P}(M = m \mid C = c)$$

2.3 One-time pad

Let $l \in \mathbb{N}^*$ and $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^l$ then *one-time pad* scheme is describe as follows

- Gen is uniform.
- $\mathrm{Enc}_k(m) = k \oplus m$.
- $\mathrm{Dec}_k(c) = k \oplus c$.

Theorem 2.3. *One-time pad is perfectly secure.*

Proof.

$$\begin{aligned} \mathbb{P}(M = m \mid C = c) &= \frac{\mathbb{P}(C = c \mid M = m)\mathbb{P}(M = m)}{\sum_{m^*} \mathbb{P}(C = c \mid M = m^*)\mathbb{P}(M = m^*)} \\ &= \frac{\mathbb{P}(K = c \oplus m)}{\sum_{m^*} \mathbb{P}(K = c \oplus m^*)\mathbb{P}(M = m^*)} \mathbb{P}(M = m) \\ &= \mathbb{P}(M = m) \end{aligned}$$

Proposition 2.4. *If* Π *is perfectly secure then we must have* $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. Suppose $|\mathcal{K}| < |\mathcal{M}|$ and let $c \in \mathcal{C}$ be a ciphertext and define $\mathcal{M}(c)$ to the

$$\mathcal{M}(c) = \{m \mid m = \mathrm{Dec}_k(c) \text{ for some } k \in \mathcal{K}\}$$

Then $|\mathcal{M}(c)| \leq |\mathcal{K}| < |\mathcal{M}|$ and therefore there exists $m \in \mathcal{M}$ such that $m \notin \mathcal{M}(c)$ hence

$$\mathbb{P}(M = m \mid C = c) = 0 \neq \mathbb{P}(M = m)$$

Note that we assumed the distribution over $\mathcal{M}$ is uniform. ■

Theorem 2.5 (Shannon's Theorem). Π *with* $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$ *is perfectly secure if and only if*

1. Gen *is uniform.*
2. $\forall m \in \mathcal{M}$ *and* $c \in \mathcal{C}$, $\exists! k \in \mathcal{K}$ *such that* $\mathrm{Enc}_k(m) = c$.

Proof.

Chapter 3

Private-Key

3.1 Asymptotic security

Definition: A scheme is (t, ϵ)-secure if any adversary running for time at most t succeeds in breaking the scheme with probability at ϵ at most.

Consider the following definitions

Definition: An **efficient algorithm** (it might be probabilistic) runs in polynomial time, that is, there is p such that for all $x \in \{0,1\}^*$, the alogrithm $A(x)$ terminates in at most $p(|X|)$ steps.

Definition: $f : \mathbb{N} \to \mathbb{R}^+$ is **negligible** if

$$\forall p \geq 0,\ \exists N \text{ s.t. } n \geq N \implies f(n) < \frac{1}{p(n)}$$

Proposition 3.1. *Suppose f, g are negligible then*

1. *$h = f + g$ is negligible.*
2. *for $p \geq 0$, $h = pf$ is negligible.*

We can now define **asymptotic security** as

Definition: A scheme is secure if for every probabilistic polynomial time adversary $\mathcal{A}$ carrying out at attack from some formally specified type, the probability of $\mathcal{A}$ succeeding is negligible.

Note that being negligible is asymptotic by definition.

3.2 Computational security

Let $\mathcal{M} = \{0,1\}^*$ and $\text{Dec}_k(c)$ returns an error $\perp$ if c is invalid. If $\forall k \leftarrow \text{Gen}(1^n)$, Enc is only defined for $m \in \{0,1\}^{l(n)}$, Π is a fixed-length private key encryption scheme for messages of length $l(n)$. Furthermore, unless specified, we assume that Enc and Dec are *stateless*. That is, each call is independent of previous calls. We revise the definition of $\text{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$ so that $|m_0| = |m_1|$. That is

1. $\mathcal{A}$ outputs a pair $m_0, m_1 \in \mathcal{M}$ such that $|m_0| = |m_1|$.
2. $k = \text{Gen}$ and b - chosen from $\{0,1\}$ uniformly - then the **challenge ciphertext** $c = \text{Enc}_k(m_b)$ is given to $\mathcal{A}$.
3. $\mathcal{A}$ tries to determine the which message was encrypted and then outputs b'.
4. $$\text{PrivK}^{eav}_{\mathcal{A},\Pi}(n) \begin{cases} 1 & b' = b \text{ then } \mathcal{A} \text{ succeeds} \\ 0 & b' \neq b \text{ then } \mathcal{A} \text{ fails} \end{cases}$$

Then the definition of indistinguishability becomes

Definition: Π has *indistinguishable encryptions in presence of eavesdropper* or it is *EAV-secure* if for any PPT $\mathcal{A}$ there is a negligible function such that

$$\mathbb{P}\big(\text{PrivK}^{eav}_{\mathcal{A},\Pi}(n) = 1\big) = \frac{1}{2} + negl(n)$$

Proposition 3.2. *If b is fixed in the aforementioned EAV-security is equivalent to*

$$\big|\mathbb{P}\big(\text{PrivK}^{eav}_{\mathcal{A},\Pi}(n, b=0) = 1\big) - \mathbb{P}\big(\text{PrivK}^{eav}_{\mathcal{A},\Pi}(n, b=1) = 1\big)\big| \leq negl(n)$$

Theorem 3.3. *Let Π be an EAV-secure fixed-length encryption scheme. Then for all PPT $\mathcal{A}$ and any bit m^i, $i \in \{1, \ldots, l\}$ there is a negligible function such that*

$$\mathbb{P}\big(\mathcal{A}(1^n, \text{Enc}_k(m)) = m^i\big) \leq \frac{1}{2} + negl(n)$$

That is, $\mathcal{A}$ can not determine any bit any better than guessing it.

Proof. proof by reduction ■

Theorem 3.4. *Let Π be defined as above. Then for any PPT $\mathcal{A}$, there is a PPT algorithm $\mathcal{A}'$ such that for any $S \subset \{0,1\}^l$ and any function $f : \{0,1\}^l \to \{0,1\}$, there is a negligible function such that*

$$|\mathbb{P}(\mathcal{A}(1^n, \text{Enc}_k(m)) = f(m)) - \mathbb{P}(\mathcal{A}'(1^n) = f(m))| \leq negl(n)$$

That is, no $\mathcal{A}$ can do any better finding a function of the message if they had the ciphertext than when they do not.

Lastly, we must take into account any external information $h(m)$ about the plaintext that may be leaked.

Definition: Π is sementically secure if $\forall \mathcal{A}$, $\exists \mathcal{A}'$ (both are PPT) such that for any PPT algorithm *Samp* and polynomial time computable functions f and h,

$$|\mathbb{P}(\mathcal{A}(1^n, \text{Enc}_k(m), h(m)) = f(m)) - \mathbb{P}(\mathcal{A}'(1^n, |m|, h(m)) = f(m))|$$

is negligible. That is, given the additionally leaked information, no $\mathcal{A}$ can do better finding a function of the message if they have the ciphertext than when when they don not have the ciphertext but know about the length of the message.

Theorem 3.5. Π *is sementically secure if and only if it is EAV-secure.*

3.3 Pseudorandom generator

A **pseudorandom generator** G is an efficient deterministic algorithm for transforming a short uniform string called seed, into a longer uniform-looking, pseudorandom, output string.

Definition: Let a l be polynomial and G be a deterministic polynomial time algorithm such that $\forall n, s \in \{0,1\}^n$, $G(s)$ returns a string of length $l(n)$. Then G is a pseudorandom random generator if

1. $\forall n,\ l(n) \geq n$.

2. For any PPT algorithm D, there is a negligible function such that

$$|\mathbb{P}(D(G(s)) = 1) - \mathbb{P}(D(r) = 1)| \leq negl(n)$$

where is r is taken uniformly from $\{0,1\}^{l(n)}$

A **stream cipher** is pair of deterministic alogrithm $(Init, GetBits)$ where

Definition:

Init takes as input a seed s and optional initialization vector IV, and outputs an initial state st_0.

GetBits takes as input state st_i and return a bit y and updated state st_{i+1}.

3.4 Proof by reduction

Assume X can not be solved by any polynomial time algorithm with negligible probability. Then to prove Π is secure we must show

1. Fix some efficient adversary $\mathcal{A}$ attacking Π with success probability $\epsilon(n)$.

2. Construct $\mathcal{A}'$ that attempts to solve X using $\mathcal{A}$ as a subroutine. Given an instance x of X, $\mathcal{A}'$ simulates Π for $\mathcal{A}$ such that

   (a) As far as $\mathcal{A}$ can tell, it is interacting with Π.

   (b) If $\mathcal{A}$ breaks Π, this should allow $\mathcal{A}'$ to solve x at least with probability $\frac{1}{p(n)}$ for some polynomial.

3. Taken together, they imply that $\mathcal{A}'$ can solve X with probability $\frac{\epsilon(n)}{p(n)}$. If ϵ is not negligible then $\frac{\epsilon}{p}$ is not negligible neither. Moreover, if $\mathcal{A}$ is efficient we obtain an efficient algorithm $\mathcal{A}'$ solving X with non-negligible probability, contradicting our assumptions.

4. Given our assumption about X there is no efficient adversary $\mathcal{A}$ that can succeed in breaking with non-negligible probability. Meaning that Π is computationally secure.

3.5 Security for multiple encryptions

Define $\text{PrivK}^{mult}_{\mathcal{A},\Pi}(n)$:

1. $\mathcal{A}$ is given input 1^n and output $M_0 = (m_{0,1}, \dots, m_{0,n})$ and $M_1 = (m_{1,1}, \dots, m_{1,n})$ with $|m_{0,i}| = |m_{1,i}|$, $\forall i$.
2. $k \leftarrow \text{Gen}(1^n)$ and $b \in \{0,1\}$ are chosen uniformly. $C = (c_1, \dots, c_n)$ is constructed with $c_i \leftarrow \text{Enc}_k(m_{b,i})$
3. $\mathcal{A}$ outputs its guess b'.

Then Π is mult-secure if for all PPT $\mathcal{A}$ there is a negligible function such that

$$\mathbb{P}\big(\text{PrivK}^{mult}_{\mathcal{A},\Pi}(n)\big) \leq \frac{1}{2} + negl(n)$$

Proposition 3.6. $\text{PrivK}^{mult}_{\mathcal{A},\Pi}(n) \implies \text{PrivK}^{eav}_{\mathcal{A},\Pi}(n)$, *however, the converse is not true.*

Proposition 3.7. *If* Π *is stateless and* Enc *is deterministic then* Π *can not be mult-secure.*

3.6 CPA and CPA-security

Let $\text{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$

1. $k \leftarrow \text{Gen}(1^n)$ unknown to $\mathcal{A}$.
2. $\mathcal{A}$ is given 1^n and $\text{Enc}_k(\cdot)$ and output messages m_0, m_1 with $|m_0| = |m_1|$.
3. $b \in \{0,1\}$ is chosen uniformly and $c \leftarrow \text{Enc}_k(m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ ouptut b'.

Π is indistinguishable against CPA attacks, CPA-secure, if $\forall \mathcal{A}$ running in PPT there exists a negligible function such that

$$\mathbb{P}\big(\text{PrivK}^{cpa}_{\mathcal{A},\Pi}(n) = 1\big) \leq \frac{1}{2} + negl(n)$$

which then can be extended to multiple messages by $\text{PrivK}^{LR-cpa}_{\mathcal{A},\Pi}(n)$. Instead of outputting lists of messages in this scheme attacker can sequentially query Π.

1. $k \leftarrow \text{Gen}(1^n)$ and $b \in \{0,1\}$ is uniformly chosen, both unknown to $\mathcal{A}$.
2. $\mathcal{A}$ is given 1^n and $LR_{k,b}(\cdot,\cdot)$.
3. $\mathcal{A}$ ouptut b'.

where $LR_{k,b}(m_0, m_1) = \text{Enc}_k(m_b)$ with $|m_0| = |m_1|$.

Π has indistinguishable multiple encryptions under CPA, CPA-secure for multiplie messages, if for any PPT $\mathcal{A}$ there exists a negligible function such that

$$\mathbb{P}\Big(\text{PrivK}^{LR-cpa}_{\mathcal{A},\Pi}(n) = 1\Big) \leq \frac{1}{2} + negl(n)$$

Theorem 3.8. *CPA-secure is equivalent to CPA-secure for multiple messages.*

Corollary 3.9. *CPA-security for fixed-length messages can be extended to arbitrary length.*

3.7 Pseudorandom function

$F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a key function $F(k,x)$. We assume that $k \in \{0,1\}^{l_{key}(n)}$, $x \in \{0,1\}^{l_{in}(n)}$, and $F_k(x) \in \{0,1\}^{l_{out}(n)}$ with $l_{key}(n) = l_{in}(n) = l_{out}(n) = n$. F is a psuedorandom if the function for any uniformly chose k, F_k, is indistinguishable from a function chosen uniformly from the set of all functions having the same domain and range. That is, F is a pseudorandom function if for all polynomial-time distinguisher D, there is a negligible function such that

$$\left|\mathbb{P}\big(D^{F_k(\cdot)}(1^n) = 1\big) - \mathbb{P}\big(D^{f(\cdot)}(1^n) = 1\big)\right| \leq negl(n)$$

where the first probability is taken over uniform $k \in \{0,1\}^n$ and D and the second probability is taken over uniform f and D. Note that D only evaluate F_k or f polynomially many times. F is a keyed permutation if $l_{in} = l_{out}$ and $\forall k \in l_{key}(n)$, F_k is a permutation. F is efficient if there is polynomial-time algorithm to compute $F_k(x),\ \forall k, x$ and $F_k^{-1}(y),\ \forall k, y$.

Proposition 3.10. *If F is a permutation pseudorandom and $l_{in}(n) \geq n$ then F is also a pseudorandom function.*

Definition: F is strong pseudorandom permutation if for all PPT distinguisher D, there is a negligible function such that

$$\left|\mathbb{P}\Big(D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n) = 1\Big) - \mathbb{P}\Big(D^{f(\cdot),f^{-1}(\cdot)}(1^n) = 1\Big)\right| \leq negl(n)$$

Definition: Block ciphers are secure instances of strong pseudorandom permutations with some fixed key length and block length.

–Construction of pseudorandom generators and stream ciphers from pseudorandom functions.

Definition: CPA encryption with pseudorandom function can be achieved with

- Gen takes 1^n and outputs $k \in \{0,1\}^n$ uniformly.
- Enc chooses uniform $r \in \{0,1\}^n$ and outputs $c = \langle r, F_k(r) \oplus m\rangle$, where F is a pseudorandom function.
- Dec takes k and $c = \langle r, s\rangle$ and outputs

$$m = F_k(r) \oplus s$$

Theorem 3.11. *If F is a pseudorandom function, then the above Construction is a CPA-secure for messages of length n.*

3.8 Modes of operation

3.8.1 Stream cipher

Synchronized mode

It is good for single communication and messages that are send and recieved in order without being loss. – insert shape Let $G_\infty(s,l)$ where s is the seed and l is the desired length, then

$$c = G_\infty\big(k, 1^{|m|}\big) \oplus m, \quad m = c \oplus G_\infty\big(k, 1^{|m|}\big)$$

Note that the message m can be composed of $m_1||m_2||\dots$ and then we can calculate c by XORing the corresponding portion of G_∞ with m_i.

Proposition 3.12. *If stream cipher is indistinguishable this is EAV-secure.*

3.8.2 Unsynchronized mode

We let G_∞ to take an initialization vector $IV \in \{0,1\}^n$ as well. Note that IV and k are uniformly chosen.

$$c = \langle IV, G_\infty\big(k, IV, 1^{|m|}\big) \oplus m\rangle$$

Proposition 3.13. *If there exists a weak pseudorandom function such that*

$$F_K(IV) = G_\infty\big(k, IV, 1^l\big)$$

then the unsynchronized mode is CPA-secure.

3.8.3 Block-cipher

Let F be a block cipher with length n. We assume that $|m|$ is a multiple of n. That is $m = m_1||\dots||m_l$ with $m_i \in \{0,1\}^n$_ if not we can append 1 followed by sufficiently many zeros.

Electronic Code Block mode (EBC)

$$\begin{aligned} c &= \langle F_K(m_1), \dots, F_K(m_l)\rangle \\ m &= \big\langle F_K^{-1}(c_1), \dots, F_K^{-1}(c_l)\big\rangle \end{aligned}$$

it is not even EAV-secure, because of repeated portions.

Cipher Block Chaining (CBC)

$$\begin{aligned} c &= \langle c_0 = IV \in \{0,1\}^n, F_K(m_1 \oplus c_0), \dots, F_K(m_l \oplus c_{l-1})\rangle \\ m &= \big\langle F_K^{-1}(c_1) \oplus c_0, \dots, F_K^{-1}(c_l) \oplus c_{l-1}\big\rangle \end{aligned}$$

Proposition 3.14. *If F is a pseudorandom permutation then this is CPA secure.*

Then chained (stateful) CBC variant uses that last c_l for encryption of new message m' instead of a fresh new IV. However, chained CBC is vulenrable to CPA. Consider the following scheme

1. Attacker knows $m_1 \in \{m_1^0, m_1^1\}$.
2. Attacker observes $\langle IV, c_1, c_2, c_3\rangle$.
3. Attacker chooses $m_4 = IV \oplus m_1^0 \oplus c_3$ and observes c_4.
4. $m_1 = m_0^1 \iff c_1 = c_4$.

Output feedback mode (OFB)

$$c = \left\langle IV, m_1 \oplus F_K(IV), m_2 \oplus F_K(F_K(IV)), \ldots, m_l \oplus F_K^{(l)}(IV) \right\rangle$$

Proposition 3.15. *OFB is CPA-secure if F is a pseudorandom function. To improve the efficiency we can calculate $F_K^{(n)}(IV)$ ahead of time. Stateful variant is secure as well.*

Counter mode (CTR)

Let $ctr \in \{0,1\}^n$ be a uniformly chosen number then

$$c = \langle ctr, m_1 \oplus F_K(ctr+1), \ldots, m_l \oplus F_K(ctr+l) \rangle$$

Proposition 3.16. *CTR is CPA-secure if F is a pseudorandom function.*

Stateful version is secure as well. It can be parallelized.

Chapter 4

Symmetric Key Primitive

4.1 Stream cipher

4.1.1 LFSR

A *linear feedback shift register* is a collection of n registers with some feedback rule. The value of the register at any given time is – insert diagram

$$s_i^{(t+1)} = s_i^{(t)} \quad i = 0, \ldots, n-2$$
$$s_{n-1}^{(t+1)} = \bigoplus_{i=0}^{n-1} c_i s_i^{(t)}$$

Note that the feedback goes into the last register s_{n-1}. The outputs are given by

$$y_i = s_{i-1}^0 \quad i = 1, \ldots n$$
$$y_i = \bigoplus_{j=0}^{n-1} c_j y_{i-n+j-1}$$

Proposition 4.1. *Solving LFSR coefficients takes* $2n$ *bits.*

To make LFSRs better and more secure we may consider a non-linear alternative where

$$s_i^{(t+1)} = s_i^{(t)} \quad i = 0, \ldots, n-2$$
$$s_{n-1}^{(t+1)} = g\left(s_0^{(t)}, \ldots, s_{n-2}^{(t)}\right)$$

where g is a non-linear balanced function, that is, $\mathbb{P}(g(s_0, \ldots, s_{n-1})) \simeq \frac{1}{2}$. Instead of using non-linear functions, we can combine multiple LFSR – usually of different length/degree –.

4.1.2 Trivium

It is a combination of 3 LFSR with an intertwined feedback rule. –insert diagram
The INIT algorithm

1. load 80 bits of seed into leftmost bits of A - 0 for the remaining bits.
2. load 80 bits of IV into leftmost bits of B - 0 for the remaining bits.
3. Set all bits C to 1.
4. Run for 4×288 times.

4.1.3 RC4

Algorithm 1: INIT algorithm

input : 16 byte key k
output: Initial state (S, i, j)
/* S contains a permutation of $\{0, 1, \ldots, 255\}$. */
/* Key k is expanded to holds 256 byte. */
for $i = 0 \rightarrow 255$ **do**
    $S[i] = i$
    $k[i] = k[i \mod 16]$
end
$j = 0$
for $i = 0 \rightarrow 255$ **do**
    $i = j + S[i] + k[i]$
    Swap $S[i]$ and $S[j]$
end
$i = 0, j = 0$
return (S, i, j)

Algorithm 2: GetBit algorithm

input : Current state (S, i, j)
output: Byte y and next state (S, i, j)
/* S contains a permutation of $\{0, 1, \ldots, 255\}$ */
$i = i + 1$
$j = j + S[i]$
Swap $S[i]$ and $S[j]$
$t = S[i] + S[j]$ $y = S[t]$ **return** $(y, (S, i, j))$ /* Every member gets swapped at least once every 256 iteration, ensuring a good mixing of the permutation of S. */

To incorporate iV we add it to the key k– used in Wired Equivalent Principle. In unsynchronized mode IV is sent to reciever which leads to attack. There is a way of attacking RC4. Let S_t be the array S after t generation of GetBit and consider S_0 to be uniform. Therefore,

$$\mathbb{P}(S_0[2] = 0 \wedge S_0[1] \neq 2) = \frac{1}{256} - \frac{1}{256 \times 255} \simeq \frac{1}{256}$$

Let $X = S_0[1]$ then, in the first generation, $j = X$ and $S[1]$ gets swapped with $S[X]$ therefore

$$S_1[X] = S_0[1] = X \qquad S_1[1] = S_0[X]$$

In the second generation, $j = X + S_1[2] = X$ and $S[2]$ gets swapped with $S[X]$ therfore

$$S_2[X] = S_1[2] = 0 \qquad S_2[2] = S_1[X] = X$$

and

$$y_2 = S_2[S_2[X] + S_2[2]] = S_2[X] = 0$$

Note that when $S_0[2] \neq 0$ the second byte y_2 is uniform, hence

$$\begin{aligned}\mathbb{P}(y_2 = 0) &= \mathbb{P}(S_0[2] = 0 \wedge S_0[1] \neq 2) + \frac{1}{256}(1 - \mathbb{P}(S_0[2] = 0 \wedge S_0[1] \neq 2)) \\ &= \frac{1}{256} - \frac{1}{256 \times 255} + \frac{1}{256} - \frac{1}{(256)^2 255} \simeq \frac{2}{256}\end{aligned}$$

which is twice the uniform case, and we are able to extract information.

Chapter 5

Message Authenticatiion

5.1 Message authentication codes (MAC)

Definition: Message authentication code $\Pi(\text{Gen}, \text{Mac}, \text{Vrfy})$ works as follows

1. $k \leftarrow \text{Gen}(1^n)$ with $|k| \geq n$.
2. $t \leftarrow \text{Mac}_k(m)$.
3. $$\text{Vrfy}_k(m,t) = \begin{cases} 1 & \text{valid} \\ 0 & \text{invalid} \end{cases}$$

Furthermore, for all k and m we must have

$$\text{Vrfy}_k(m, \text{Mac}_k(m)) = 1$$

Consider MAC existentially unforgeable (can not forge a tag) under an adaptive chosen message attack(knows the tag for arbitrary messages). Mac-forge$_{\mathcal{A},\Pi}(n)$:

1. $k \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given 1^n and given access to $\text{Mac}_k(\cdot)$. Adversary outputs (m,t). Let $\mathcal{Q}$ be the set of queried messages.
3. $\mathcal{A}$ succeeds if $\text{Vrfy}_k(m,t) = 1$ and $m \notin \mathcal{Q}$.

Definition: $\Pi(\text{Gen}, \text{Mac}, \text{Vrfy})$ is *existentially unforgeable under an adaptive chosen message attack* or simply *secure* if for all PPT adversaries $\mathcal{A}$ there exists a negligible function

$$\mathbb{P}\big(\text{Mac-forge}_{\mathcal{A},\Pi}(n) = 1\big) \leq \text{negl}(n)$$

A stronger version MAC-security, Mac-sforge$_{\mathcal{A},\Pi}(n)$:

1. $k \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given 1^n and given access to $\text{Mac}_k(\cdot)$. Adversary outputs (m',t'). Let $\mathcal{Q}$ be the set of queried messages and their tag (m,t).
3. $\mathcal{A}$ succeeds if $\text{Vrfy}_k(m',t') = 1$ and $(m',t') \notin \mathcal{Q}$.

Definition: $\Pi(\text{Gen}, \text{Mac}, \text{Vrfy})$ is *strongly secure* or *strong* MAC, if for all PPT adversaries $\mathcal{A}$ there exists a negligible function

$$\mathbb{P}\big(\text{Mac-sforge}_{\mathcal{A},\Pi}(n) = 1\big) \leq \text{negl}(n)$$

In canonical verification, Vrfy checks if $t = \text{Mac}_k(m)$. Clearly, Mac must be deterministic.

Proposition 5.1. *If* Π *is a secure MAC that uses canonical verification then* Π *is a strong MAC. Furthermore, a strong MAC remains strongly secure with* Vrfy *oracle.*

5.2 Construction of secure MAC

Let F be a PRF and consider the following construction.

- $\text{Mac}_k(m) = F_k(m)$. If $|m| \neq |k|$ it outputs nothing.
- $\text{Vrfy}_k(m, t) = 1$ if $t = F_k(m)$ and 0 otherwise.

Theorem 5.2. *If* F *is a PRF, then the construction above is a secure fixed length MAC.*

5.3 Domain extension for MAC

To extend the length of MAC we will consider the following ideas. Let $m = m_1||\ldots||m_d$ and Π' be a secure fixed length MAC.

- $t_i = \text{Mac}'_k(m_i)$ is susceptible to re-ordering attacks.
- $t_i = \text{Mac}'_k(i||m_i)$ is susceptible to truncation attacks. The length of m_i is decreased.
- $t_i = \text{Mac}'_k(l||i||m_i)$ is susceptible to max-and-match attacks. The length of m_i is decreased.

Now consider the following construction.

1. $k \leftarrow \text{Gen}(1^n)$
2. Mac : takes $k \in \{0,1\}^n$ and $m \in \{0,1\}^*$ with length less than $2^{n/4}$. $m = m_1||\ldots||m_d$ with each $|m_i| = n/4$, the last block might need to be padded with zero. $r \in \{0,1\}^{n/4}$ chosen uniformly. $t_i \leftarrow \text{Mac}_k(r||l||i||m_i)$. $t = \langle r, t_1, \ldots, t_d\rangle$.
3. Vrfy : takes $k \in \{0,1\}^n$, $m \in \{0,1\}^*$ with length less than $2^{n/4}$. $m = m_1||\ldots||m_d$ with each $|m_i| = n/4$, the last block might need to be padded with zero, and $t = \langle r, t_1, \ldots, t'_d\rangle$.

$$\text{Vrfy}_k(m, t) = \begin{cases} 1 & d = d' \wedge \text{Vrfy}_k(r||l||i||m_i, t_i) = 1 \forall i \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5.3. *If* Π' *is secure MAC, then the construction above is secure MAC.*

5.4 CBC-MAC

Let F be a PRF with $l(n) >$). Then basic CBC is as follows:

1. Mac : takes $k \in \{0,1\}^n$ and $|m| = nl(n)$, $m = m_1||\dots||m_l$ with $|m_i| = n$. $t_0 = 0^n$ and $t_i = F_k(t_{i-1} \oplus m_i)$ for $i = 1$ to l. Outputs $t = t_l$.
2. Vrfy : takes $k \in \{0,1\}^n$ and m, t.

$$\text{Vrfy}_k(m,t) = \begin{cases} 1 & \text{if } |m| = nl(n) \ \wedge \ t = \text{Mac}_k(n) \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5.4. *Let l be a polynomial and F be a PRF, then the construction is a secure MAC for messages of length $nl(n)$.*

5.5 Authenticated Messages

Let Π be a private key encryption. The unforgeable encryption experiment Enc-forge$_{\mathcal{A},\Pi}(n)$.

1. $k \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given 1^n and $\text{Enc}_k(\cdot)$. Adversary outputs ciphertext c.
3. Let $m = \text{Dec}_k(c)$ and let $\mathcal{Q}$ be the set of all queried messages. $\mathcal{A}$ succeeds if

$$\text{Enc-forge}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & m \neq \perp, m \notin \mathcal{Q} \\ 0 & \text{otherwise} \end{cases}$$

Definition: A private key scheme Π is unforgeable if for all PPT $\mathcal{A}$ there is a negligible function such that

$$\mathbb{P}\big(\text{Enc-forge}_{\mathcal{A},\Pi}(n) = 1\big) \leq \text{negl}(n)$$

Definition: A private key encryption scheme is an authenticated encryption scheme if it is CCA-secure and unforgeable.

Using different schemes for authenticated encryption: Let Π_E be CPA-secure and Π_M be a secure MAC.

1. Encrypt and Authenticate

$$c \leftarrow \text{Enc}_{k_E}(m) \quad t \leftarrow \text{Mac}_{k_M}(m)$$

   and send $\langle c, t \rangle$. Secure-MAC does not guarantee secrecy therefore this scheme leaks information abot m. Using deterministic MAC implies no CPA-security.

2. Authenticate then encrypt

$$t \leftarrow \text{Mac}_{k_M}(m) \quad c \leftarrow \text{Enc}_{k_E}(m||t)$$

   and send c. If Π is CBC-mode-with-padding, then we can have two error, "bad padding" and "MAC tag" which means attacker can apply some CCA.

3. Encrypt then Authenticate

$$c \leftarrow \mathrm{Enc}_{k_E}(m) \quad t \leftarrow \mathrm{Mac}_{k_M}(c)$$

send $\langle c, t\rangle$. This approach is ok as long as MAC is strongly secure. CCA-security reduces to CPA-security of Π_E since sMAC-security implies it can not generate valid ciphertext.

Theorem 5.5. *Let Π_E be a CPA-secure private key encryption scheme and Π_M be a strongly secure MAC. Then encrypt-then-authenticate method is an authenticated encryption scheme.*

5.6 Secure communication session

Let Π be an authenticated encryption scheme: The following attacks are possible.

1. Re-ordering attacks. use counters.

2. Reply attacks. use counters.

3. Reflection attacks. use directionaly bit

hence we can improve these by $c \leftarrow \mathrm{Enc}(b_{A,B}||ctr_{A,B}||m)$.

Remark 1. CCA-security for encryption only is different than CCA-security for authenticated encryption. But in real world most construction that satisfy the former, also satisfy the latter.

5.7 Information theoretic MAC

The one time message authentication experiment $\text{Mac-forge}^{1-time}_{\mathcal{A},\Pi}(n)$

1. $k \leftarrow \mathrm{Gen}(1^n)$.

2. $\mathcal{A}$ outputs m' and in return is give a tag $t' \leftarrow \mathrm{Mac}_k(m')$.

3. $\mathcal{A}$ outputs m, t.

4. $\mathcal{A}$ succeeds if

$$\text{Mac-forge}^{1-time}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & \mathrm{Vrfy}_k(m,t) = 1 \wedge m \neq m' \\ 0 & \text{otherwise} \end{cases}$$

Definition: Π is one-time ϵ-secure if for all adversaries $\mathcal{A}$

$$\mathbb{P}\big(\text{Mac-forge}^{1-time}_{\mathcal{A},\Pi}(n) = 1\big) \leq \epsilon$$

5.7.1 Construction of IT-MAC

Definition: $h : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is strongly universal if for all distinct $m, m' \in \mathcal{M}$ and all $t, t' \in \mathcal{T}$

$$\mathbb{P}(h_k(m) = t \wedge h_k(m') = t') = \frac{1}{|\mathcal{T}|^2}$$

The probability is taken over uniform choice of k.

Let $h : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ be strongly universal function.

1. $k \leftarrow \mathrm{Gen}(1^n)$.
2. $\mathrm{Mac}_k(m) = h_k(m)$.
3. $$\mathrm{Vrfy}(m, t) = \begin{cases} 1 & t = h_k(m) \wedge m \in \mathcal{M} \\ 0 & \text{otherwise} \end{cases}$$

Theorem 5.6. *Construction above is $\frac{1}{|\mathcal{T}|}$-secure MAC for messages in $\mathcal{M}$.*

Let $\mathcal{M}, \mathcal{T} = \mathbb{Z}_p, \mathcal{K} = \mathbb{Z}_p \times \mathbb{Z}_p$ and

$$h_{a,b}(m) = am + b \mod p$$

Theorem 5.7. *For any prime p the function h is strongly universal.*

5.7.2 Bounds on IT-MAC

Theorem 5.8. Π *be 2^{-n} secure MAC where all keys outputed by* Gen *are the same length. Then the keys of Gen have length at least $2n$.*

Chapter 6

Hash functions

6.1 Collision Resistance

Definition: A hash function with output (digest) length l is a pair of probabilistic polynomial-time algorithms (Gen, H) satisfying

1. $s \leftarrow \text{Gen}(1^n)$.
2. $x \in \{0,1\}^*, H^s(x) \in \{0,1\}^{l(n)}$

If H^s is only defined for $x \in \{0,1\}^{l'(n)}$ and $l'(n) > l(n)$ then (Gen, H) is a fixed-length function for input length l'. H is also called a compression function.

The collision finding experiment Hash-coll$_{\mathcal{A},\Pi}(n)$:

1. $s \leftarrow \text{Gen}(1^n)$.
2. $\mathcal{A}$ is given s and outputs x, x'.
3. The result of the experiment is 1 if $H^s(x) = H^s(x')$ and $x \neq x'$ and 0 otherwise.

Definition: A hash function $\Pi = (\text{Gen}, H)$ is collision resistant if for all PPT adversary $\mathcal{A}$ there is a negligible function such that

$$\mathbb{P}(\text{Hash-coll}_{\mathcal{A},\Pi}(n) = 1) \neq \text{negl}(n)$$

Definition: Weaker notions of security for a hash function

- second-preimage or target collision resistance assumes that given s and uniform x it is infeasible for a PPT adversary $\mathcal{A}$ to find $x' \neq x$ such that $H^s(x') = H^s(x)$.
- preimage resistance assumes that given s and uniform y it is infeasible for a PPT adversary to find x such that $H^s(x) = y$ (one-wayness).

collision resistance implies second preimage implies preimage.

6.2 Domain extension

Merkle-Damgard construction Let (Gen, h) be a fixed-length hash function with input $2n$ and output n. Construct (Gen, H) as follow

- Gen remains unchange.
- Give s and $x \in \{0,1\}^*$ of length $L < 2^n$.
  1. Set $B = \lceil \frac{L}{n} \rceil$ and pad x with zeros so it is a multiple of n. Let $x = x_1 \dots x_B$ and $x_{B+1} = L$ as n-bit binary number.
  2. Set $z_0 = 0^n$ (IV can be some other constant)
  3. For $i = 1, \dots, B+1$ computer $z_i = h^s(z_{i-1}||x_i)$.
  4. Output z_{B+1}.

Theorem 6.1. *If* (Gen, h) *is a collision resistant hash function then* (Gen, H) *its Merkle-Damgard transform is collision resistant as well.*

6.3 Hash and message authentication

6.3.1 Hash and MAC

Construction: Let $\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$ be MAC for messages of length $l(n)$ and let $\Pi_H = (\text{Gen}_H, H)$ be a hash function with output length $l(n)$. Construct $\Pi = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ for arbitrary length messages as follows

Gen' $k \leftarrow \text{Gen}(1^n)$ and $s \leftarrow \text{Gen}_H(1^n)$ then return $k' = \langle k, s \rangle$.

Mac' takes k' and $m \in \{0,1\}^*$ then $t \leftarrow \text{Mac}_k(H^s(m))$.

Vrfy' on input $k', m \in \{0,1\}^*, t$

$$\text{Vrfy}'_{k'}(m,t) = \text{Vrfy}_k(H^s(m), t)$$

Theorem 6.2. *If* Π_M *is a secure MAC and* Π_H *is a collision resistant hash function then, the above construction is a secure MAC.*

6.3.2 HMAC

Construction: Let (Gen_H, H) be a hash function constructed by applying Merkle-Damgard transform to a compression function Gen_H, h taking input of length $n + n'$. Let *opad* and *ipad* be fixed constants of length n'.

Gen Let $s \leftarrow \text{Gen}_H(1^n)$ and k is chosen uniformly from $\{0,1\}^{n'}$ then, output $\langle k, s \rangle$.

Mac on input $\langle k, s \rangle$ and $m \in \{0,1\}^*$ output

$$t = H^s((k \oplus opad)||H^s((k \oplus ipad)||m))$$

Vrfy is the canonical verify on $\langle k, s \rangle, m, t$.

Let

$$G^s(k) = h^s(IV||(k \oplus opad)||h^s(IV||(k \oplus ipad))) = k_{out}||k_{in}$$

If G^s is a pseudorandom generator for s then, k_{out} and k_{in} can be treated as independent and uniform keys when k is uniform.

Theorem 6.3. *Assume G^s is a pseudorandom generator for any s,*

$$\mathrm{Mac}_k(y) = h^s(k||\hat{y})$$

is a secure fixed-length MAC, and (Gen_H, H) is weakly collision resistant. Then, HMAC is a secure MAC for arbitrary-length messages.

6.4 Generic Attacks

6.4.1 Birthday attacks for find collisions

Let $H : \{0,1\}^* \to \{0,1\}^l$ be a hash function. $y_1, \ldots, y_q$ are chosen uniformly from $\{1, \ldots, N\}$ then, $\mathbb{P}(collision) = \frac{1}{2}$ when $q = \Theta\left(\sqrt{N}\right)$. small space birthday attack pg 168,169 and finding meaningful collisions.

6.5 Random-oracle model

Instead of having keyed hash function, we use a random oracle as a substitute. This is purely theoretical. In random-oracle model we assume that

1. H is chosen randomly "on the fly".
2. Nobody knows what queries have already been made except those made by themselves.

This model has the following useful properties

1. If x has not been queried to H, then $H(x)$ is uniform.
2. If $\mathcal{A}$ queries x to H, the reduction can see this query and learn x, "extractablility".
3. The reduction can set the value of $H(x)$ to a value of its choice as long as this value is correctly distribute. "programmablility"

6.5.1 Simple illustration of attacks

Suppose oracle gets l_{in}-bit input and gives l_{out}-bit output, where $l_{in}, l_{out} < n$.

- For $l_{out} > l_{in}$ random oracle can be used as a pseudorandom generator.
- For $l_{out} < l_{in}$ random oracle is a collision resistant hash function. read Appendix A4

Construction of PRF from random oracle: Suppose $l_{in} = 2n$ and $l_{out} = n$. Then, $F_k(x) = H(k||X)$ is a PRF. Above claims work even when the adversary is computationally unbounded but only makes polynomial calls to oracle.

6.6 Addition Application of hash functions

6.6.1 Fingerprinting and duplication

6.6.2 Merkle Trees

Theorem 6.4. *Let* Gen_H, H *be collision resistent. Then,* $(\mathrm{Gen}_H, \mathcal{MT}_t)$ *is also collision resistant for fixed* t*.*

Chapter 7

Number Theory

7.1 Preliminaries

- Divisiblilty and Euclidean algorithm (EEA).
- Modular arithmetic. b is invertible modulu N if there exists c such that $bc \equiv 1 \mod N$. This implies that, $\gcd(b, N) = 1$.

7.2 Group

$(G, \circ)$ is a group if

1. G is closed under $\circ$.
2. There exists $e \in G$ such that $\forall g \in G,\ g \circ e = e \circ g = g$.
3. For all $g \in G$ there exists $h \in G$ such that $h \circ g = g \circ h = e$.
4. $\forall g_1, g_2, g_3 \in G$ we have $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.

Also if $\circ$ is commutative, that is $h \circ g = g \circ h$ for any two elements $g, h \in G$, then the group is abelian group. If G is finite, $|G|$ denotes the order of the group. $H \subset G$ forms a subgroup of G if $(H, \circ)$ is a group. G and $\{e\}$ are both subgroups of G. H is a strict subgroup of G if $H \neq G$.

Lemma 7.1. *Let $a, b, c \in G$. If $a \circ c = b \circ c$ then $a = b$.*

Proof. Let $d \in G$ be the inverse of c. Then,

$$(a \circ c) \circ d = a \circ (c \circ d) = a \circ e = a$$
$$(b \circ c) \circ d = b \circ (c \circ d) = b \circ e = b$$

Thus

$$a \circ c = b \circ c \implies (a \circ c) \circ d = (b \circ c) \circ d \implies a = b$$

Suppose m is a positive integer then $\forall g \in G$

$$g^m = \underbrace{g \circ \cdots \circ g}_{m}$$

and if m is a negative integer and h is the inverse of g then

$$g^m = \underbrace{g \circ \cdots \circ g}_{m}$$

we also define $g^0 = e$. In this way, for any two integers m, n we have $g^{m+n} = g^m \circ g^n$.

Let H be a subgroup of G. Left cosets of H are the sets obtained by multiplying each element of H by a fixed element $g \in G$.

$$gH = \{g \circ h \,|\, h \in H\}$$

Similarly, right cosets of H are the sets obtained by multiplying g by each element of H

$$Hg = \{h \circ g \,|\, h \in H\}$$

Proposition 7.2. *For two $g_1, g_2 \in G$ either $g_1H = g_2H$ or $g_1H \cap g_2H = \emptyset$.*

Proposition 7.3. *Suppose $a \in g_1H, g_2H$. Then, there are $h_1, h_2 \in H$ such that $a = g_1 \circ h_1 = g_2 \circ h_2$. It follows that $g_2 = g_1 \circ h_1 \circ h_2^{-1} = g_1 \circ h_3$ for some $h_3 \in H$. Thus, any element of g_2H is in g_1H and similarly we can show that $g_1H \subset g_2H$. Therefore, $g_1H = g_2H$ if they have a common element.*

Proposition 7.4. *Every left/right costs of H have the same cardinality as H.*

Proof. Let $a, b \in gH$ henc there are $h_1, h_2 \in H$ such that $a = g \circ h$ and $b = g \circ h_2$. It follows that $h_1 = g^{-1} \circ a$ and $h_2 = g^{-1} \circ b$. By 7.1 and its converse $a = b \implies h_1 = h_2$ and hence $|gH| = |H|$. ■

The number of left/right cosets of H is known as the index of H in G, denoted by $[G : H]$.

Theorem 7.5 (Largrange's theorem). *If H is subgroup of G then*

$$|G| = [G : H]|H|$$

Proof. Note that since $e \in H$ then for all $g \in G$ there exists a left coset of H that contains g, namely $g \in gH$. Since the number of distinct cosets of H are disjoint and they contain all the element of G we must have

$$|G| = [G : H]|H|$$

Equivalently, cosets of H define an equivalence class on G. ■

For any $g \in G$ we can form a subgroup of all its integer power $\langle g \rangle = \left\{g^k \,\middle|\, k \in \mathbb{Z}\right\}$. Note that,

- $\langle g \rangle$ is closed under $\circ$ since $g^m \circ g^n = g^{m+n} \in \langle g \rangle$.
- $g^0 = e \in \langle g \rangle$.
- For any m, g^{-m} is the inverse of g^m.

- Associativity holds as it holds for G.

Proposition 7.6. *If G has order of m then for any $g \in G$, $g^m = e$.*

Proof. By Largrange's theorem

$$|G| = [G : \langle g \rangle]|\langle g \rangle|$$

Let $\operatorname{ord} g$ be the smallest positive integer n such that $g^n = e$. Then $\operatorname{ord} g = |\langle g \rangle|$. Therefore, $\operatorname{ord} g \mid |G| \implies g^m = g^{nk} = e^k = e$. ■

Corollary 7.7. *Let G be a finite group with $|G| > 1$. Let $n > 0$ be an integer and define $f_n : G \to G$ by $f_n(g) = g^n$. If $\gcd(n, m) = 1$ then, f_n is bijective. Moreover, if $k \equiv n^{-1} \mod m$ then f_k is the inverse of f_n.*

Proof. f_n is bijective if and only if f_n has an inverse. This implies that second part implies the first part. Then, note that

$$f_k(f_n(g)) = f_k(g^n) = g^{kn} = g^1 = g \qquad f_n(f_k(g)) = f_n\big(g^k\big) = g^{nk} = g^1 = g$$

$\mathbb{Z}_N^*$ is the multiplicative group of $\mathbb{Z}_N$. In fact, $\mathbb{Z}_N^* = \{b \mid b \in \mathbb{Z}_n, \gcd(b, N) = 1\}$. Euler toutient function ϕ is defined as $\phi(N) = |\mathbb{Z}_N^*|$.

Proposition 7.8. *If $N = \prod_{i=1}^n p_i^{\alpha_i}$ then $\phi(N) = \prod_{i=1}^n p_i^{\alpha_i - 1}(p_i - 1)$.*

Proposition 7.9. *This is implied by the 7.11 and the fact that $\phi(p^\alpha) = p^{\alpha-1}(p-1)$.*

Corollary 7.10. *Let $N > 1$ and $n > 0$ and $f_n : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ defined by $f_n(x) = x^n \mod N$. If $\gcd(n, phi(N)) = 1$ then, f_n is a permutation. Moreover if $m = n^{-1} \mod \phi(N)$ then f_m is the inverse of f_n.*

Two groups G, H are isomorphic, denoted by $G \simeq H$, if there exists a bijective map $f : G \to H$ such that

$$f(g_1 \circ_G g_2) = f(g_1) \circ_H f(g_2)$$

Theorem 7.11 (Chinese remainder theorem). *Let $c = ab$ where a, b are relatively prime then*

$$\mathbb{Z}_c \simeq \mathbb{Z}_a \times \mathbb{Z}_b \quad \textit{and} \quad \mathbb{Z}_c^* \simeq \mathbb{Z}_a^* \times \mathbb{Z}_b^*$$

7.3 Primes and RSA

To see this algorithm is probabilistic polynomial time in n we need to know

1. prime distribution. to know the probability that an n-bit uniform integer is prime.
2. Efficient primality test.

7.3.1 Prime distribution

Theorem 7.12 (Bertrand's Postulate). *For any $n > 1$, the fraction of n-bit integers that are primes is at least $\frac{1}{3n}$.*

Therefore, by setting $t = 3n^2$ probability of failing is

$$\left(1 - \frac{1}{3n}\right)^t = \left(\left(1 - \frac{1}{3n}\right)^{3n}\right)^n \leq e^{-n}$$

which is negligible.

Algorithm 3: Generating random primes

input : n, t
output: A uniform n-bit prime
for $i = 1 \to t$ **do**
  $p' \leftarrow \{0,1\}^{n-1}$
  $p = 1||p'$ **if** p *is prime* **then**
    return p
  end
end
return $\perp$

7.3.2 Primality test

Deterministic poly-time algorithm for primality test exist but they are slower than probabilistic ones.

Algorithm 4: Miller-Robin primality test

input : p, t
output: "prime" or "composite"
if p *is even* **then**
  return *"composite"*
end
if p *is perfect power* **then**
  return *"composite"*
end
Compute r, u with $N - 1 = 2^r u$ where $r \geq 1$ and u is odd
for $i = 1 \to t$ **do**
  $a \leftarrow \{1, \ldots, p-1\}$
  if a *is a strong witness that* p *is composite* **then**
    return *"composite"*
  end
end
return *"prime"*

Theorem 7.13. *If p is prime, then Miller-Robin test always outputs "prime" and if p is composite, the algorithm outputs "composite" except with probability at most 2^{-t}*

First consider the following two lemmas

Lemma 7.14. *If $H \subset G$, H is non-empty, and for all $a, b \in H$, $a \circ b \in H$. Then, H is a subgroup of G.*

Lemma 7.15. *If H is a strict subgroup of G. Then $|H| \leq \frac{|G|}{2}$.*

We say that a is a witness that N is composite if $a \in \mathbb{Z}_N^*$ but $a^{N-1} \not\equiv 1 \mod N$.

Theorem 7.16. *Suppose there is a witness that N is composite. Then, at least half of the element of $\mathbb{Z}_N^*$ are witness that N is composite.*

Carmicheal numbers are composite numbers which do not have any witness. So let $N-1 = 2^r u$ where $r \geq 1$ and u is odd. Consider the sequence that $a^u, a^{2^1 u}, \ldots, a^{2^r u}$ all modulu N. Then, $a \in \mathbb{Z}_N^*$ is a strong witness that N is composite if

1. $a^u \not\equiv \pm 1 \mod N$.
2. $a^{2^i u} \not\equiv -1 \mod N$.

If a is not a strong witness then it is not a witness neither. Thus, if a is a witness it is also a strong witness. If N is a prime then N does not have any strong witness.

Theorem 7.17. *Let N be an odd number that is not a prime power. Then, at least half of the elements of $\mathbb{Z}_N^*$ are strong witnesses that N is composite.*

Putting all these theorems and lemmas together gives us a proof for 7.13.