EE 720 Lecture Notes

Autumn 2023

Saravanan Vijayakumaran

December 13, 2024

Table of contents

Preface

These notes were created to support the course *EE720: An Introduction to Number Theory and Cryptography* at IIT Bombay. This course runs in the Electrical Engineering department and is offered to both undergraduate and postgraduate students.

I have taught EE720 three times (2018–2020) using the excellent textbook Introduction to Modern Cryptography by Jonathan Katz and Yehuda Lindell. Time constraints allowed me to cover only a small subset of the book's content. I used other sources to teach topics from abstract algebra in more detail.

These notes are meant to **supplement** the book by Katz & Lindell. They will contain the following.

- Proofs of some results stated without proof in the book.
- Expanded coverage of some material relevant to the course.

Saravanan Vijayakumaran
July 2023

⚠ Warning

These notes are a work in progress and may contain errors. Please email the author at sarva@ee.iitb.ac.in to report any errors.

About Me

My name is Saravanan Vijayakumaran. I am an Associate Professor in the Department of Electrical Engineering at IIT Bombay. I am currently (mid 2023) interested in cryptocurrency blockchains and applications of zero-knowledge proofs.

Chapter 1

Perfectly Secret Encryption

1.1 Proof of Lemma 2.7

Lemma 1.1. *Encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is perfectly secret if and only if it is perfectly indistinguishable.*

1.1.1 Forward Direction

- Suppose a scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly secret.
- Consider an adversary $\mathcal{A}$ who picks two messages m_0, m_1 and gives it to Π.
- Π picks a bit b randomly and a key k. It gives $c = \mathsf{Enc}_k(m_b)$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs b'.
- We want to prove that $\Pr[b = b'] = \frac{1}{2}$. Note abuse of notation; lower case letters for random variables.

1.1.1.1 Deterministic Adversary

- If $\mathcal{A}$ is deterministic, then there exists a partition C_0, C_1 of $\mathcal{C}$ such that $\mathcal{C}_0 \cap \mathcal{C}_1 = \emptyset$ and

$$\mathcal{A}(c) = \begin{cases} 0 & \text{if } c \in \mathcal{C}_0, \\ 1 & \text{if } c \in \mathcal{C}_1. \end{cases}$$

Then we have

$$\begin{aligned}\Pr[b = b'] &= \frac{\Pr[\mathcal{A}(C) = 0 \mid b = 0] + \Pr[\mathcal{A}(C) = 1 \mid b = 1]}{2} \\ &= \frac{\Pr[C \in \mathcal{C}_0 \mid b = 0] + \Pr[C \in \mathcal{C}_1 \mid b = 1]}{2}\end{aligned}$$

Consider the first term in the numerator

$$\begin{aligned}\Pr[C \in \mathcal{C}_0 \mid b = 0] &= \Pr[C \in \mathcal{C}_0 \mid M = m_0] \\ &= \sum_{c \in \mathcal{C}_0} \Pr[C = c \mid M = m_0] = \sum_{c \in \mathcal{C}_0} \Pr[C = c] \\ &= \Pr[C \in \mathcal{C}_0]\end{aligned}$$

Similarly, the second term in the numerator is equal to $\Pr[C \in \mathcal{C}_1]$. Plugging these back into the previous equation, we get

$$\Pr[b = b'] = \frac{\Pr[C \in \mathcal{C}_0] + \Pr[C \in \mathcal{C}_1]}{2} = \frac{1}{2}.$$

1.1.1.2 Probabilistic Adversary (not exclusive from previous case)

Suppose the adversary is probabilistic.

$$\Pr[b = b'] = \frac{\Pr[\mathcal{A}(C) = 0 \mid b = 0] + \Pr[\mathcal{A}(C) = 1 \mid b = 1]}{2}$$

Consider the first term in the numerator.

$$\begin{aligned}\Pr[\mathcal{A}(C) = 0 \mid b = 0] &= \sum_{c \in \mathcal{C}} \Pr[\mathcal{A}(C) = 0 \mid b = 0, C = c] \Pr[C = c \mid b = 0] \\ &= \sum_{c \in \mathcal{C}} \Pr[\mathcal{A}(C) = 0 \mid C = c] \Pr[C = c \mid b = 0] \\ &= \sum_{c \in \mathcal{C}} \Pr[\mathcal{A}(C) = 0 \mid C = c] \Pr[C = c] \\ &= \Pr[\mathcal{A}(C) = 0].\end{aligned}$$

where the second equality follows because the adversary's decision is independent of the message once we condition on the ciphertext, and the third equality follows from Lemma 2.5 and the assumption of perfect secrecy.

Similarly, we have $\Pr[\mathcal{A}(C) = 1 \mid b = 1] = \Pr[\mathcal{A}(C) = 1]$. Plugging these back into the previous equation, we get

$$\Pr[b = b'] = \frac{\Pr[\mathcal{A}(C) = 0] + \Pr[\mathcal{A}(C) = 1]}{2} = \frac{1}{2}.$$

1.1.2 Reverse Direction

Let A be perfect secrecy and B be perfect indistinguishability. We have already proved $A \implies B$.

To prove $B \implies A$, we can prove $A^c \implies B^c$.

Suppose Π is not perfectly secret. Then there exist $m, m' \in \mathcal{M}$ and $c_0 \in \mathcal{C}$ such that

$$\Pr[C = c_0 \mid M = m] \neq \Pr[C = c_0 \mid M = m'].$$

WLOG, assume

$$\Pr[C = c_0 \mid M = m] > \Pr[C = c_0 \mid M = m']$$

We need to construct an adversary who can exploit this fact. A natural choice for m and m' is $m_0 = m$ and $m_1 = m'$.

What should the strategy be? Consider the following strategy

$$b' = \mathcal{A}(c) = \begin{cases} 0 & \text{if } c = c_0, \\ \text{random bit} & \text{if } c \neq c_0. \end{cases}$$

We have

$$\Pr[b = b'] = \frac{\Pr[\mathcal{A}(C) = 0 \mid b = 0] + \Pr[\mathcal{A}(C) = 1 \mid b = 1]}{2}.$$

Consider the first term in the numerator.

$$\begin{aligned} \Pr[\mathcal{A}(C) = 0 \mid b = 0] =& \Pr[\mathcal{A}(C) = 0 \mid b = 0, C = c_0] \Pr[C = c_0 \mid b = 0] \\ &+ \Pr[\mathcal{A}(C) = 0 \mid b = 0, C \neq c_0] \Pr[C \neq c_0 \mid b = 0] \\ =& 1 \cdot \Pr[C = c_0 \mid b = 0] + \frac{1}{2} \Pr[C \neq c_0 \mid b = 0]. \end{aligned}$$

Consider the second term in the numerator.

$$\begin{aligned}\Pr[\mathcal{A}(C) = 1 \mid b = 1] =& \Pr[\mathcal{A}(C) = 1 \mid b = 1, C = c_0] \Pr[C = c_0 \mid b = 1] \\ &+ \Pr[\mathcal{A}(C) = 1 \mid b = 1, C \neq c_0] \Pr[C \neq c_0 \mid b = 1] \\ =& \frac{1}{2} \Pr[C \neq c_0 \mid b = 1].\end{aligned}$$

Combining these two equations, we get

$$\begin{aligned}\Pr[b = b'] &= \frac{\Pr[C = c_0 \mid b = 0] + \frac{1}{2} \Pr[C \neq c_0 \mid b = 0] + \frac{1}{2} \Pr[C \neq c_0 \mid b = 1]}{2} \\ &> \frac{\frac{1}{2} \Pr[C = c_0 \mid b = 0] + \frac{1}{2} \Pr[C = c_0 \mid b = 1] + \frac{1}{2} \Pr[C \neq c_0 \mid b = 0] + \frac{1}{2} \Pr[C \neq c_0 \mid b = 1]}{2} \\ &= \frac{\Pr[C = c_0] + \Pr[C \neq c_0]}{2} = \frac{1}{2}.\end{aligned}$$

Chapter 2

Private-Key Encryption

2.1 Proof that EAV-Security Implies Definition 3.9

Suppose that a private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is EAV-secure.

Let $\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,b)\right)$ denote the output b' of $\mathcal{A}$ when m_b is encrypted

We want to prove that for all PPT adversaries $\mathcal{A}$ there is a negligible function negl such that, for all n,

$$\left|\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right)=1\right]-\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right)=1\right]\right| \leq \mathsf{negl}(n). \tag{2.1}$$

Since Π is EAV-secure, we have $\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right] \leq \frac{1}{2}+\mathsf{negl}(n)$. It follows that

$$\begin{aligned}
&\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right]\\
&=\Pr[b'=b]\\
&=\frac{\Pr[b'=0 \mid b=0]+\Pr[b'=1 \mid b=1]}{2}\\
&=\frac{\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right)=0\right]+\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right)=1\right]}{2}\\
&=\frac{1-\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right)=1\right]+\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right)=1\right]}{2}\\
&\leq\frac{1}{2}+\mathsf{negl}(n),
\end{aligned}$$

The final inequality gives us

$$\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right)=1\right]-\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right)=1\right]\le\mathsf{negl}(n).$$

But the result in Equation 2.1 has an absolute value on the left hand side. So if we can also show that

$$\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right)=1\right]-\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right)=1\right]\le\mathsf{negl}(n),$$

then we will be done.

Claim: If $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is EAV-secure, then for any PPT adversary $\mathcal{A}$ the following must hold: $\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right]\ge\frac{1}{2}-\mathsf{negl}(n)$.

In other words, it is not possible for a PPT adversary to fail with probability that is too far away from $\frac{1}{2}$. Let us rearrange the terms in our claim.

$$\begin{aligned}\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right]&\ge\frac{1}{2}-\mathsf{negl}(n)\\ \iff\frac{1}{2}-\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right]&\le\mathsf{negl}(n)\end{aligned}$$

We will prove our claim by contradiction. Suppose that the function $\frac{1}{2}-\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)=1\right]$ is not negligible. Then there is an adversary $\mathcal{A}'$ and a positive polynomial $p(n)$ such that

$$\frac{1}{2}-\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A}',\Pi}(n)=1\right]>\frac{1}{p(n)}$$

for infinitely many n. This implies that

$$\Pr[b'=b]=\Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A}',\Pi}(n)=1\right]<\frac{1}{2}-\frac{1}{p(n)}$$

for infinitely many n, where b' is the output of $\mathcal{A}'$ and b is the bit chosen in the experiment.

Since $\mathcal{A}'$ is good at failing in the experiment, flipping the output of $\mathcal{A}'$ will give us an adversary that will succeed with a probability that is significantly better than $\frac{1}{2}$.

Consider an adversary $\mathcal{A}''$ who uses $\mathcal{A}'$ as a subroutine and outputs the opposite of whatever $\mathcal{A}'$ outputs.

$$b''=\mathcal{A}''(c)=1-\mathcal{A}'(c)=\begin{cases}0 & \text{if } b'=1,\\ 1 & \text{if } b'=0.\end{cases}$$

If $\mathcal{A}'$ is PPT, then $\mathcal{A}''$ is also PPT. Then we have

$$\begin{aligned}\Pr[b'' = b] &= \Pr[b' \neq b] = 1 - \Pr[b' = b] \\ &= 1 - \Pr[b' = b] > 1 - \left(\frac{1}{2} - \frac{1}{p(n)}\right) \\ &= \frac{1}{2} + \frac{1}{p(n)}\end{aligned}$$

for infinitely many n. This is a contradiction. Hence our claim must be true.

We then have

$$\begin{aligned} & \frac{1}{2} - \Pr\left[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathsf{negl}(n) \\ \iff & \frac{1}{2} - \Pr\left[b' = b\right] \leq \mathsf{negl}(n) \\ \iff & \frac{1 - \Pr[b' = 0 \mid b = 0] - \Pr[b' = 1 \mid b = 1]}{2} \leq \mathsf{negl}(n) \\ \iff & \frac{\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]}{2} \leq \mathsf{negl}(n) \\ \iff & \Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1] \leq 2\mathsf{negl}(n) \\ \iff & \Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right) = 1\right] - \Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right) = 1\right] \leq 2\mathsf{negl}(n). \end{aligned}$$

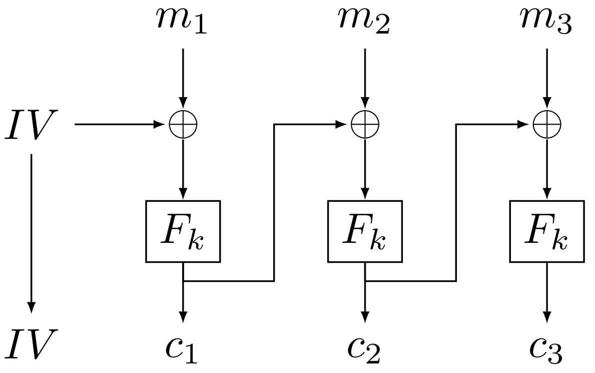
As $2\mathsf{negl}(n)$ is also negligible, we have the other inequality we need to prove that

$$\left|\Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,0)\right) = 1\right] - \Pr\left[\mathsf{out}_{\mathcal{A}}\left(\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n,1)\right) = 1\right]\right| \leq \mathsf{negl}(n).$$

Chapter 3

CCA-Security and Authenticated Encryption

3.1 CBC-Mode Encryption



Let $\vec{m} = \langle m_1, m_2, \ldots, m_l \rangle$ where $m_i \in \{0,1\}^n$. Let F be a block cipher with block length n. To encrypt $\vec{m}$, the following steps are followed.

1. A uniform initialization vector (IV) of length n is chosen
2. $c_0 = IV$. For $i = 1, \ldots, l$,
$$c_i := F_k(c_{i-1} \oplus m_i).$$

To decrypt the ciphertext, for $i = 1, 2, \ldots, l$,

$$m_i := F_k^{-1}(c_i) \oplus c_{i-1}.$$

3.2 Padding Scheme from PKCS #7 Standard

In CBC-mode encryption, the message length is a multiple of the block length. If not, message needs to be padded.

Let m denote the original message. Assume that $|m|$ is an integral number of bytes. Let L be the block length of F in bytes. We will assume that $L < 256$, so it can fit in a byte.

Let b denote the padding length in bytes. We will require that b is an integer from 1 to L, i.e. $b = 0$ is not allowed.

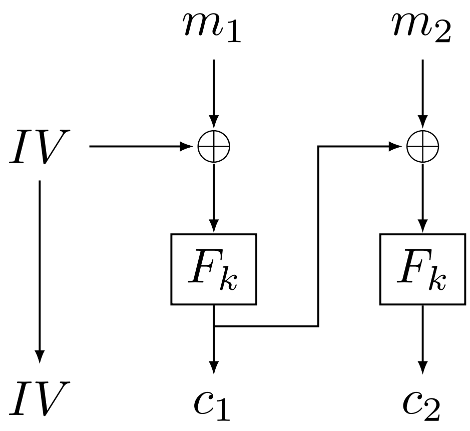
We append to the message the integer b (represented in 1 byte) repeated b times. For example, if 4 bytes are needed then 0x04040404 is appended. If $|m|$ is already a multiple of L, then L bytes are appended each of which is equal to L.

To remove the padding, the receiver first applies the decryption algorithm on the ciphertext corresponding to encoded data. The value b of the last byte of the encoded data is read. The final b bytes of the encoded data are checked to be equal to b.

3.3 Decryption Oracle from Padding Errors

If padding is incorrect, the standard procedure is to return a “bad padding” error. This error message is a *partial decryption oracle*. Using these error messages, an attacker *can completely recover* the original message.

3.4 Padding Oracle Attack Example



Consider a 3-block ciphertext from CBC-mode encryption. The second plaintext block is obtained as

$$m_2 = F_k^{-1}(c_2) \oplus c_1.$$

Note that m_2 ends in $\mathsf{0x}b\mathsf{0x}b\ldots\mathsf{0x}b$.

Suppose an attacker changes c_1 to c_1', but keeps IV and c_2 the same. Then m_2 is changed to

$$m_2' = F_k^{-1}(c_2) \oplus c_1'.$$

If the c_1 and c_1' differ only in the ith byte, then m_2 and m_2' also differ only in the ith byte.

3.4.1 Learning b

Suppose there is a server that receives ciphertexts and returns padding oracle error (if any). The attacker can learn b as follows.

- The attacker modifies the first byte of c_1 to get c_1', and sends (IV, c_1', c_2) to the server
  - If error is returned, then $b = L$
  - The server is reading all L bytes of m_2'
- Otherwise, the attacker modifies the second byte of c_1 and sends (IV, c_1', c_2). And so on.
- The first byte where decryption fails reveals b

3.4.2 Learning m_2

Suppose $b = L$. Then the message block m_2 consists of L bytes each equal to L. Only m_1 remains to be recovered. The method for recovering m_1 is described in the next section.

Consider the case when $b < L$. The attacker knows that the m_2 ends with

$$\mathsf{0x}M\mathsf{0x}b\ldots\mathsf{0x}b$$

where M is an unknown byte.

Define Δ_i as

$$\begin{aligned}
\Delta_i =& \mathsf{0x00}\cdots\mathsf{0x00} \qquad \mathsf{0x}i\; \overbrace{\mathsf{0x}(b+1)\cdots\mathsf{0x}(b+1)}^{b\text{ times}}\\
\oplus& \mathsf{0x00}\cdots\mathsf{0x00} \quad \mathsf{0x00}\; \overbrace{\mathsf{0x}b\cdots\mathsf{0x}b}^{b\text{ times}}
\end{aligned}$$

M is recovered as follows.

- Attacker sends $(IV, c_1 \oplus \Delta_i, c_2)$ to the server
- The final $b+1$ bytes of the resulting encoded data are

$$\mathsf{0x}(M \oplus i)\,\underbrace{\mathsf{0x}(b+1)\cdots\mathsf{0x}(b+1)}_{b \text{ times}}$$

- Decryption fails unless $\mathsf{0x}(M \oplus i) = \mathsf{0x}(b+1)$
- Attacker needs to try at most 256 values for i
- When decryption succeeds, attacker learns that

$$M = \mathsf{0x}(b+1) \oplus \mathsf{0x}i$$

If $b = L-1$, m_2 has now been fully recovered and the attacker can proceed to recover m_1. Otherwise, the attacker knows that the m_2 ends with

$$\mathsf{0x}M\mathsf{0x}a\mathsf{0x}b\ldots\mathsf{0x}b$$

for an unknown M and known a.

The attacker uses a technique similar to the one described above to recover the remaining bytes of m_2.

3.4.3 Learning m_1

In the normal scenario, message block m_1 is decrypted as

$$m_1 = F_k^{-1}(c_1) \oplus IV.$$

Suppose an attacker changes IV to IV', but keeps c_1 the same. Then m_1 is changed to

$$m_1' = F_k^{-1}(c_1) \oplus IV'.$$

If the IV and IV' differ only in the ith byte, then m_1 and m_1' also differ only in the ith byte.

For $0 \leq i \leq 255$, the attacker chooses Δ_i defined as

$$\Delta_i = \underbrace{\mathsf{0}\|\mathsf{0}\|\cdots\|\mathsf{0}}_{L-1 \text{ times}}\,\|\,\underbrace{\mathsf{i}}_{1 \text{ byte}}$$

where i is the byte representation of the integer i and $\mathsf{0}$ is the byte representation of the integer 0.

The attacker calculates $IV^{(i)} = IV \oplus \Delta_i$ and sends $(IV^{(i)}, c_1)$ to the padding oracle. Then m_1 is changed to

$$m_1^{(i)} = F_k^{-1}(c_1) \oplus IV^{(i)} = F_k^{-1}(c_1) \oplus IV \oplus \Delta_i = m_1 \oplus \Delta_i.$$

The **key observation** is that there exists at least one i for which the padding oracle returns ok. And there may be more than one value of such an i.

Let $m_{1,j}$ denote the jth byte of m_1 for $j \in \{1, 2, \ldots, L\}$.

- The value of i such that $m_{1,L} \oplus i = 1$ will result in an ok response from the padding oracle irrespective of the values of the other bytes $m_{1,1}, m_{1,2}, \cdots, m_{1,L-1}$. It corresponds to the case of 1 byte of padding.
- The value of i such that $m_{1,L} \oplus i = 2$ will also result in an ok response if $m_{1,L-1} = 2$.
- The value of i such that $m_{1,L} \oplus i = 3$ will also result in an ok response if $m_{1,L-2} = m_{1,L-1} = 3$.

To distinguish between these possible cases, the attacker first finds an i such that the padding oracle returns ok. At this point, the attacker knows that $(IV^{(i)}, c_1)$ corresponds to a message $m^{(i)}$ that ends with b bytes each equal to b, for some unknown b.

Then the attacker uses the technique in Section 3.4.1 to estimate the padding length b in the encoded data corresponding to $(IV^{(i)}, c_1)$. The details are left as an exercise.

Since $m_{1,L} \oplus i = b$ and i is known, he can recover $m_{1,L}$. This also reveals that the values of $b - 1$ bytes $m_{1,L-b+1}, \ldots, m_{1,L-2}, m_{1,L-1}$ are all equal to b.

If $b = L$, all the bytes of m_1 have been recovered. Once again, consider the case when $b < L$. The attacker knows that the $m_1^{(i)} = m_1 \oplus \Delta_i$ ends with

$$\mathsf{0x}M\mathsf{0x}b \ldots \mathsf{0x}b$$

where M is an unknown byte. The attacker uses the technique in Section 3.4.2 to recover all the remaining bytes in m_1. The details are left as an exercise.

Chapter 4

Number Theory

4.1 Lagrange's Theorem

Theorem 4.1 (Lagrange)**.** *If H is a subgroup of a finite group G, then $|H|$ divides $|G|$.*

Definition 4.1. Let H be a subgroup of a group G. For any $g \in G$, the set $H+g = \{h+g \mid h \in H\}$ is called a **right coset** of H.

To prove Lagrange's theorem, we need the following lemmas

Lemma 4.1. *Two right cosets of a subgroup are either equal or disjoint.*

Proof. Suppose two right cosets $H+g_1$ and $H+g_2$ are disjoint. Then there is nothing to prove.

Suppose that they are not disjoint, i.e. $H+g_1 \cap H+g_2 \neq \emptyset$.

Let $g \in H+g_1 \cap H+g_2$. Then

$$g = h_1 + g_1 = h_2 + g_2$$

for some $h_1, h_2 \in H$. This implies that

$$\begin{aligned} g_1 &= -h_1 + h_2 + g_2, \\ g_2 &= -h_2 + h_1 + g_1. \end{aligned}$$

To show that the two right cosets are equal, we show that they are subsets of each other.

Let $a \in H + g_1$. Then $a = h_3 + g_1$ for some $h_3 \in H$. Substituting for g_1 we get

$$a = h_3 - h_1 + h_2 + g_2 = h' + g_2.$$

for some $h' \in H$. This implies that $a \in H + g_2$. Hence $H + g_1 \subseteq H + g_2$.

Let $b \in H + g_2$. Then $b = h_4 + g_2$ for some $h_4 \in H$. Substituting for g_2 we get

$$b = h_4 - h_2 + h_1 + g_1 = h'' + g_1.$$

for some $h'' \in H$. This implies that $b \in H + g_1$. Hence $H + g_2 \subseteq H + g_1$.

So the two right cosets $H + g_1$ and $H + g_2$ are equal. □

Lemma 4.2. *If H is a finite subgroup, then all its right cosets have the same cardinality.*

Proof. We claim that the map $f_g : H \mapsto H + g$ given by $f_g(h) = h + g$ is a one-to-one map.

If $f_g(h_1) = f_g(h_2)$, the $h_1 + g = h_2 + g$. By the cancellation law, we get $h_1 = h_2$. So f_g is a one-to-one map.

This implies that all cosets have the same number of elements as H, completing the proof of the lemma. □

Theorem 4.1. Note that every $g \in G$ belongs to a right coset. This is because we can write g as $e + g$ where e is the identity of the group. As e also belongs to H, we have that $g \in H + g$.

When $\coprod$ denotes disjoint union, we have

$$\begin{aligned} G &= \bigcup_g (H + g) \\ &= \coprod_{i=1}^{k} (H + g_i) \end{aligned}$$

for some $g_1, g_2, \ldots, g_k$. Counting elements on both sides we get

$$\begin{aligned} |G| &= \sum_{i=1}^{k} |H + g_i| \\ &= \sum_{i=1}^{k} |H| \\ &= k|H|. \end{aligned}$$

This implies that $|H|$ divides $|G|$. □

4.2 Square-and-Multiply Algorithm for Group Exponentiation

Suppose G is a group with an operation expressed in mutliplicative notation. To compute g^m, it might seem that $m-1$ multiplications are required. One can reduce the number of multiplications required by using the square-and-multiply algorithm.

For example, to compute g^8, one can compute

- $g_1 = g^2$ using one multiplication
- $g_2 = g_1^2$ using one multiplication
- $g_3 = g_2^2$ using one multiplication

But $g_3 = g_2^2 = g_1^4 = g^8$. So g^8 can be calculated using 3 multiplications instead of 7 multiplications.

The above example uses the fact that $8 = 2^3$. But how can we handle the case when the exponent is not a power of 2?

Let the binary representation of m be $(b_{k-1}, b_{k-2}, \ldots, b_1, b_0)$ where b_0 is the least significant bit. Let e be the identity of the group. Then g^m can be calculated as follows.

```
$y \leftarrow e, i \leftarrow k-1$
while $i \geq 0$
  $y \leftarrow y^2$
  if $b_i = 0$ then $y \leftarrow g \cdot y$
  $i \leftarrow i - 1$
return $y$
```

The running time of the above algorithm is proportional to k, the number of bits in the exponent m