

Proofs - Introduction to Cryptography

Sven Hluchy & Moritz Hölting

February 4, 2025

Contents

1	Week 1 – Perfect Security	3
1.1	Theorem 1.5	3
1.2	Theorem 1.6	3
1.3	Theorem 1.7	3
1.4	Theorem 1.8	4
1.5	Theorem 1.9	4
1.6	Theorem 1.10	4
1.7	Theorem 1.11	5
2	Week 2 – Computational Ciphers	5
2.1	Theorem 2.9	5
2.2	Theorem 2.12	6
2.3	Theorem 2.14	7
3	Week 3 – Block Ciphers	8
3.1	Theorem 2.19 – From Predictability to Block Adversary	8
3.2	Corollary 2.20	10
4	Week 4 – Probabilistic Ciphers	10
4.1	Example 3.3	10
5	Week 5 – Stream Ciphers and PRGs	11
5.1	Examples of Non-PRGs	11
5.2	Theorem 4.7	12
5.3	Theorem 4.11	14
6	Week 6 – MACs and PRFs	15
6.1	Theorem 5.6	15
7	Week 7 – MACs and Hashing	17
7.1	Theorem 5.11	17
7.2	Remarks on Definition of Collision Resistant Hash Function . . .	18
7.3	Proof of Theorem 5.15	19

7.3.1	a)	19
7.3.2	b)	19
7.3.3	c)	20
7.3.4	d)	20
8	Proof of Lemma 5.16	20
9	Week 8 – Public Key Encryption and RSA	21
9.1	Extended Euclidean Algorithm	21
9.2	Theorem 6.6	22
9.3	Lemma 6.9	22
9.4	Lemma 6.11	23
9.5	Theorem 6.16	23
10	Week 9 – Digital Signatures and RSA	24
10.1	Theorem 6.20	24
10.2	Theorem 6.21	24
10.3	Theorem 6.22	24
11	Week 10 – Key Exchange Protocols and the Discrete Logarithm Problem	25
11.1	Diffie-Hellman Key Exchange – Number Theory	25
11.2	Theorem 7.10	25
11.3	Theorem 7.13	26
11.4	Theorem 7.14	26
11.5	Theorem 7.16	27
11.6	Lemma 7.17	27
12	Week 11 – DLog Hashing and Secure Public-Key Encryption under the DLog Problem (CPA and CCA)	27
12.1	Theorem 8.4	27
12.2	Theorem 8.6	27
12.3	Theorem 8.11	28
12.4	Theorem 8.15	29
13	Week 12 – DLog Hashing and Secure Public-Key Encryption under the DLog Problem (Signatures)	29
13.1	Correctness – Schnorr ID Scheme	29
13.2	Theorem 10.2	29
13.3	Theorem 10.4	31
13.4	Security of Schnorr ID Scheme	31
13.5	Correctness – Schnorr Signatures	31
13.6	Theorem 10.11	31

1 Week 1 – Perfect Security

1.1 Theorem 1.5

$m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$

$b \in \{0, 1\}$

$$\begin{aligned} & Pr[E(\underline{k}, m_b) = c] \\ &= Pr[m_b \oplus \underline{k} = c] \\ &= Pr[\underline{k} = m_b \oplus c] \\ &= \frac{1}{2^l} \end{aligned}$$

1.2 Theorem 1.6

$m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$

$c = m_0$

$$\begin{aligned} & Pr[E(\underline{k}, m_0) = m_0] \\ &= Pr[\underline{k} = 0^l] = 0 \\ & Pr[E(\underline{k}, m_1) = m_0] \\ &= Pr[\underline{k} = m_1 \oplus m_0] \\ &= \frac{1}{|\mathcal{K}|} \text{ if } m_0 \neq m_1 \end{aligned}$$

Because probabilities are not the same, the scheme is not perfectly secure.

1.3 Theorem 1.7

Let $L = 2, a, b \in \Sigma, a \neq b$

Set $m_0 = (a, a), m_1 = (a, b), c = (a, a)$

$$Pr[E(\underline{k}, m_0) = c] = \frac{1}{|\mathcal{K}|}$$

$$Pr[E(\underline{k}, m_1) = c] = 0$$

because $k(a) = a, k(b) = a$ cannot be correct, because it is a permutation

1.4 Theorem 1.8

For all $m, c \in \Sigma$

$$Pr[E(\underline{k}, m) = c] = p$$

Hence the substitution cipher is perfectly secure

1.5 Theorem 1.9

(E, D) perfectly secure.

$m_0, m_1 \in \mathcal{M}$, $c \in \mathcal{C}$ arbitrary

$$S := \{c \in \mathcal{C} \mid \phi(c)\}$$

Then:

$$\begin{aligned} & Pr[\phi(E(\underline{k}, m_0))] \\ &= \sum_{c \in S} Pr[E(\underline{k}, m_0) = c] \\ &= \sum_{c \in S} Pr[E(\underline{k}, m_1) = c] \\ &= Pr[\phi(E(\underline{k}, m_1))] \end{aligned}$$

1.6 Theorem 1.10

$$Pr[\underline{m} = m \mid \underline{c} = c] = Pr[\underline{m} = m]$$

equivalent to

$$Pr[\underline{c} = c \wedge \underline{m} = m] = Pr[\underline{c} = c] \cdot Pr[\underline{m} = m]$$

by definition of conditional probabilities.

$$\begin{aligned} & Pr[\underline{c} = c \wedge \underline{m} = m] \\ &= Pr[E(\underline{k}, \underline{m}) = c \wedge \underline{m} = m] \\ &= Pr[E(\underline{k}, m) = c \wedge \underline{m} = m] \\ &= Pr[E(\underline{k}, m) = c] \cdot Pr[\underline{m} = m] \end{aligned}$$

need to show:

$$Pr[E(\underline{k}, m) = c] = Pr[\underline{c} = c]$$

$$\begin{aligned}
Pr[\underline{c} = c] &= Pr[E(\underline{k}, \underline{m}) = c] \\
&= \sum_{m' \in \mathcal{M}} Pr[E(\underline{k}, m') = c \wedge \underline{m} = m'] \\
&= \sum_{m' \in \mathcal{M}} \underbrace{Pr[E(\underline{k}, m') = c]}_{\text{Same for all messages (Perfect Security)}} \cdot Pr[\underline{m} = m'] \\
&= \sum_{m' \in \mathcal{M}} Pr[E(\underline{k}, m) = c] \cdot Pr[\underline{m} = m'] \\
&= Pr[E(\underline{k}, m) = c] \cdot \sum_{m' \in \mathcal{M}} Pr[\underline{m} = m'] \\
&= Pr[E(\underline{k}, m) = c]
\end{aligned}$$

because $\sum_{m' \in \mathcal{M}} Pr[\underline{m} = m'] = 1$

1.7 Theorem 1.11

Proof by contraposition.

Assume $|\mathcal{K}| < |\mathcal{M}|$, show $\mathcal{E} = (E, D)$ is not perfectly secure.

more precisely: $\exists m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$:

$Pr[E(\underline{k}, m_0) = c] > 0$ (a)

and $Pr[E(\underline{k}, m_1) = c] = 0$ (b)

pick $m_0 \in \mathcal{M}$ arbitrary, pick $k \in \mathcal{K}$ arbitrary

$c := E(k, m) \Rightarrow$ (a) holds

$S := \{D(k', c) : k' \in \mathcal{K}\}$

$|S| \leq |\mathcal{K}| < |\mathcal{M}|$ hence exists $m_1 \in \mathcal{M} \setminus S$

then $Pr[E(\underline{k}, m_1) = c] = 0 \Rightarrow$ (b)

2 Week 2 – Computational Ciphers

2.1 Theorem 2.9

$\mathcal{A}$ fixed, arbitrary in message recovery game

$p := Pr[W]$, $\mathcal{A}$'s winning probability

need:

$MRadv(\mathcal{A}, \mathcal{E}) = \left| p - \frac{1}{|\mathcal{M}|} \right|$ negligible

From $\mathcal{A}$ construct $\mathcal{B}$ such that:

- $\mathcal{B}$ efficient, if $\mathcal{A}$ efficient
- $MRadv(\mathcal{A}, \mathcal{E}) \leq SSadv(\mathcal{B}, \mathcal{E})$

$SSadv(\mathcal{B}, \mathcal{E})$ negligible, since $\mathcal{E}$ is semantically secure

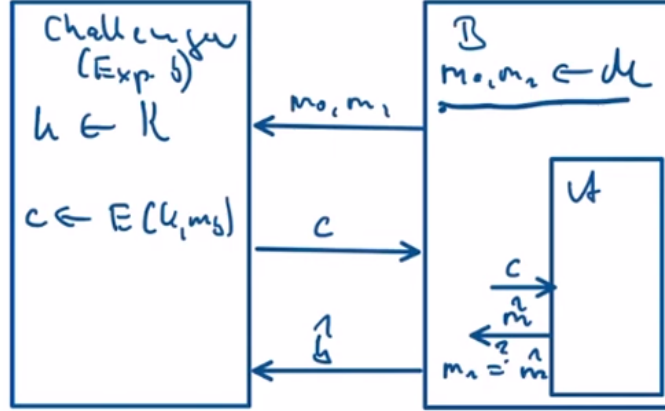


Figure 1: $\mathcal{B}$ from $\mathcal{A}$

- $\mathcal{B}$ efficient, if $\mathcal{A}$ efficient ✓
- $MRadv(\mathcal{B}, \mathcal{E})$: $b \in \{0, 1\}$: p_b : probability $\mathcal{B}$ outputs 1 in experiment b
hence $SSadv(\mathcal{B}, \mathcal{E}) = |p_0 - p_1|$
 $b = 1$: $c = E(k, m_1) \Rightarrow p_1 = p$
 $b = 0$: $p_0 = \frac{1}{|\mathcal{M}|}$, since m_0 & c are stochastically independent from m_1

$$|p_0 - p_1| = \left| \frac{1}{|\mathcal{M}|} - p \right| = MRadv(\mathcal{A}, \mathcal{E})$$

2.2 Theorem 2.12

$\mathcal{A}$ arbitrary, fixed adversary against $\mathcal{E}$ in parity prediction

$p := Pr[W]$, $\mathcal{A}$'s win probability

set $p = \frac{1}{2} + \epsilon$, then $PPadv(\mathcal{A}, \mathcal{E}) = |\epsilon|$

From $\mathcal{A}$ construct $\mathcal{B}$ such that

- $\mathcal{B}$ efficient, if $\mathcal{A}$ efficient
- $PPadv(\mathcal{A}, \mathcal{E}) = \frac{1}{2} SSadv(\mathcal{B}, \mathcal{E})$

$SSadv(\mathcal{B}, \mathcal{E})$ negligible, since $\mathcal{E}$ semantically secure

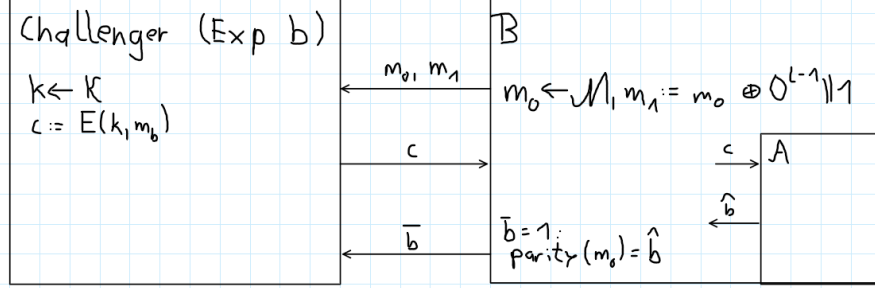


Figure 2: $\mathcal{B}$ from $\mathcal{A}$

$$\text{parity}(m_0) \neq \text{parity}(m_1)$$

$p_b := \Pr[W_b]$ = probability $\mathcal{B}$ outputs 1 in experiment b

$$SSadv(\mathcal{B}, \mathcal{E}) = |p_0 - p_1|$$

$b = 0$:

$\mathcal{A}$ receives encryption of m_0 . $\mathcal{B}$ outputs 1, if $\mathcal{A}$ outputs $\hat{b} = \text{parity}(m_0) \Rightarrow$ happens with probability $p_0 = \frac{1}{2} + \epsilon = p$

$b = 1$:

$\mathcal{A}$ receives the encryption of m_1 . $\mathcal{A}$ outputs $\hat{b} = \text{parity}(m_1)$ with probability $p = \frac{1}{2} + \epsilon$. $\mathcal{A}$ outputs $\text{parity}(m_0) = \neg \text{parity}(m_1)$ with $\frac{1}{2} - \epsilon = p_1$

$$SSadv(\mathcal{B}, \mathcal{E}) = |p_0 - p_1| = |\frac{1}{2} + \epsilon - (\frac{1}{2} - \epsilon)| = 2\epsilon = 2 \cdot PPadv(\mathcal{A}, \mathcal{E})$$

2.3 Theorem 2.14

Let $\mathcal{A}$ be arbitrary:

Show:

$$SSadv^*(\mathcal{A}, \mathcal{E}) = \frac{1}{2} SSadv(\mathcal{A}, \mathcal{E})$$

$W_b := \mathcal{A}$ outputs 1 in Experiment b of semantic security

$$p_b := \Pr[W_b]$$

Consider bit guessing game with $\mathcal{A}$

Conditioned on $b = 0$: inputs for $\mathcal{A}$ are as in Experiment 0

Conditioned on $b = 1$: inputs for $\mathcal{A}$ are as in Experiment 1

$$Pr[\hat{b} = 1|b = 0] = p_0$$

$$Pr[\hat{b} = 1|b = 1] = p_1$$

$$\begin{aligned}
Pr[W] &= Pr[\hat{b} = b] \\
&= Pr[\hat{b} = b \wedge b = 0] + Pr[\hat{b} = b \wedge b = 1] \\
&= Pr[\hat{b} = b|b = 0] \cdot Pr[b = 0] + Pr[\hat{b} = b|b = 1] \cdot Pr[b = 1] \\
&= Pr[\hat{b} = 0|b = 0] \cdot Pr[b = 0] + Pr[\hat{b} = 1|b = 1] \cdot Pr[b = 1] \\
&= \frac{1}{2} \cdot Pr[\hat{b} = 0|b = 0] + \frac{1}{2} \cdot Pr[\hat{b} = 1|b = 1] \\
&= \frac{1}{2} \left(1 - Pr[\hat{b} = 1|b = 0]\right) + \frac{1}{2} Pr[\hat{b} = 1|b = 1] \\
&= \frac{1}{2}(1 - p_0 + p_1) \\
&\Rightarrow SSadv^*(\mathcal{A}, \mathcal{E}) \\
&= |Pr[\hat{b} = b] - \frac{1}{2}| \\
&= \left| \frac{1}{2}(1 - p_0 + p_1) - \frac{1}{2} \right| \\
&= \left| \frac{1}{2} + \frac{1}{2}(-p_0 + p_1) - \frac{1}{2} \right| \\
&= \left| -\frac{1}{2}(p_0 - p_1) \right| \\
&= \frac{1}{2}|p_0 - p_1| \\
&= \frac{1}{2} \cdot SSadv(\mathcal{A}, \mathcal{E})
\end{aligned}$$

3 Week 3 – Block Ciphers

3.1 Theorem 2.19 – From Predictability to Block Adversary

Show:

- (1) B effective if A effective
- (2) $PDadv(\mathcal{A}, \mathcal{E}) = BCadv(\mathcal{B}, \mathcal{E})$

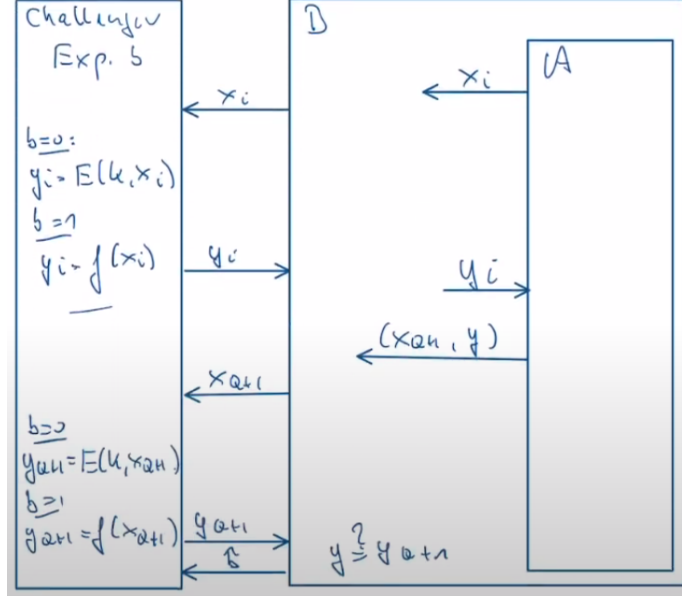


Figure 3: From Predictability to Block Adversary

Because (1) is trivial, we only show (2):

$\mathcal{A}$ is our prediction adversary, $w := \mathcal{A}$ wins, $p := \Pr[w]$

$$BCadv(\mathcal{B}, \mathcal{E}) = |\Pr[w_0] - \Pr[w_1]|$$

Remember that: $w_b := B$ outputs 1 in experiment b .

$b = 0$:

B running experiment 0, $\mathcal{A}$ plays prediction against $\mathcal{E}$, $\Pr[w_0] = \Pr[w] = p$.

$b = 1$: $f \leftarrow \text{Perm}(X)$

Fact: Let $X_1, \dots, X_Q, X_{Q+1}$ be pairwise distinct and let y_1, y_Q, y be pairwise distinct, then

$$\begin{aligned} & \Pr_{f \leftarrow \text{Perm}[X]} [f(X_{Q+1}) = y \mid f(x_i) = y_i, i = 1, \dots, Q] \\ &= \frac{1}{|X| - Q} \end{aligned}$$

From this we can conclude $\Pr[w_1] = \frac{1}{|X| - Q}$, then

$$\begin{aligned} BCadv(\mathcal{B}, \mathcal{E}) &= \left| p - \frac{1}{|X| - Q} \right| \\ &= PDadv(\mathcal{A}, \mathcal{E}) \quad \square \end{aligned}$$

3.2 Corollary 2.20

Let A be an arbitrary efficient adversary, let Q be the number of queries of A . Since A is efficient $\Rightarrow Q$ is poly bounded (of a reasonable size).

$$\begin{aligned} \Pr[w] &= \left| \Pr[w] - \frac{1}{|X| - Q} + \frac{1}{|X| - Q} \right| \\ &\leq \underbrace{\left| \Pr[w] - \frac{1}{|X| - Q} \right|}_{\text{negligible after Theorem 2.19}} + \frac{1}{|X| - Q} \end{aligned} \quad (*)$$

We can now argue that

$$\frac{1}{|X| - Q} \leq \frac{Q}{|X|}$$

provided $|X| \geq Q \geq 2$

$|X|$ is super-poly $\Rightarrow \frac{1}{|X|}$ is negligible $\Rightarrow \frac{Q}{|X|}$ is also negligible, then the $\frac{1}{|X| - Q}$ expression in (*) is also negligible, which makes (*) a sum of two negligible terms, which makes itself negligible as well. $\square$

4 Week 4 – Probabilistic Ciphers

4.1 Example 3.3

Let $\mathcal{E} = (E, D)$ be a deterministic encryption scheme. Define adversary $\mathcal{A}$ as follows:

- $\mathcal{A}$ chooses $(m_{1,0}, m_{1,1})$ arbitrarily with $|m_{1,0}| = |m_{1,1}|$ and $m_{1,0} \neq m_{1,1}$.
- $\mathcal{A}$ chooses $(m_{2,0}, m_{2,1})$ such that $m_{2,0} = m_{1,0}$ but $m_{2,1} \neq m_{1,1}$.
- $\mathcal{A}$ receives two ciphertexts from the challenger, namely c_1, c_2 , where $c_1 = E(k, m_{1,1})$ and $c_2 = E(k, m_{2,1})$ in Experiment b .
- $\mathcal{A}$ outputs 1, if $c_1 = c_2$.
- $\Pr[w_0] = 1$.
- $\Pr[w_1] = 0$.

From this, we get a $(\mathcal{A}, \mathcal{E}) = 1$, which is non-negligible. $\square$

5 Week 5 – Stream Ciphers and PRGs

5.1 Examples of Non-PRGs

Example 1

- $S := \{0, 1\}^l$, $l \in \mathbb{N}$
- $R := \{0, 1\}^{l+1}$, $L = l + 1$

$$G_1 : \{0, 1\}^l \rightarrow \{0, 1\}^{l+1}$$

$$s \mapsto s \parallel \text{xor}(s),$$

where $s = s_1, \dots, s_l$, $s_i \in \{0, 1\}$, and $\text{xor} := s_1 \oplus s_2 \oplus \dots \oplus s_l$.

Claim: G_1 is not a secure PRG.

Proof: Construct efficient $\mathcal{A}$ such that $\text{PRGAdv}(\mathcal{A}, G_1)$ is not negligible.

- $\mathcal{A}$ on input $r \in \{0, 1\}^{l+1}$:
 - $r = r_1, \dots, r_{l+1}$
 - Compute $\hat{b} := \text{xor}(r_1, \dots, r_l)$
 - Output 1 if $\hat{b} = r_{l+1}$, otherwise output 0.

$\Pr[w_0]$: If $r = G_1(s)$, $s \in \{0, 1\}^l$ then $r_{l+1} = \text{xor}(r_1, \dots, r_l)$

$\Pr[w_1]$: $r_{l+1} \leftarrow \{0, 1\}$, hence

$$\Pr[r_{l+1} = \text{xor}(r_1, \dots, r_l)] = \frac{1}{2}, \quad \Pr[w_1] = \frac{1}{2}.$$

From that, we get

$$\text{PRGAdv}(\mathcal{A}, G_1) = \left| 1 - \frac{1}{2} \right| = \frac{1}{2},$$

which is non-negligible.

$\Rightarrow G_1$ is not a secure PRG. $\square$

Example 2

- $S := \{0, 1\}^l$, $l \in \mathbb{N}$
- $R := \{0, 1\}^L$, $L \in \mathbb{N}$, $L > l$

Fix $A \in \{0, 1\}^{L \times l}$.

$$G_2 : \{0,1\}^l \rightarrow \{0,1\}^L$$

$$s \mapsto A \cdot s \pmod{2}.$$

Claim: G_2 is not a secure PRG.

Proof: Construct $\mathcal{A}$ as follows:

- $\mathcal{A}$ on input $r \in \{0,1\}^L$:
 - Using linear algebra, check whether an $x \in \{0,1\}^l$ exists such that $A \cdot x = r \pmod{2}$.
 - If such an x exists, output 1, otherwise output 0.

$\Pr[w_0]$:

For experiment 0, an x with $A \cdot x = r \pmod{2}$ always exists, hence $\Pr[w_0] = 1$.

$\Pr[w_1]$:

$$Y := \{r \in \{0,1\}^L \mid \exists x \in \{0,1\}^l \text{ with } A \cdot x = r \pmod{2}\}$$

Given $|Y| \leq 2^l$

$$\Pr[w_1] = \frac{|Y|}{2^L} \leq \frac{2^l}{2^L} = 2^{l-L} \leq \frac{1}{2}, \text{ since } L > l.$$

$\Rightarrow G_2$ is not a secure PRG. $\square$

5.2 Theorem 4.7

Proof: $\mathcal{A}$ be fixed, but arbitrary and efficient.

Show:

$$\text{SSAdv}^*(\mathcal{A}, \mathcal{E}) = \text{PRGAdv}(\mathcal{B}, G)$$

(Then $\forall \mathcal{A} : \text{SSAdv}^*(\mathcal{A}, \mathcal{E}) = \frac{1}{2} \cdot \text{SSAdv}(\mathcal{A}, \mathcal{E})$)

Bit guessing game:

- $\mathcal{A}$ chooses m_0, m_1 .
- Challenger $s \leftarrow \{0,1\}^l$, $b \leftarrow \{0,1\}$, encrypts m'_b using key s and sends ciphertext to $\mathcal{A}$.
- $\mathcal{A}$ outputs $\hat{b} \in \{0,1\}$.

Call this game 0.

- $w_0 :=$ event that $\hat{b} = b$ in game 0.

- $\text{SSAdv}^*(\mathcal{A}, \mathcal{E}) = |\Pr[w_0] - \frac{1}{2}|$

Define another game, called game 1. Here the challenger does the following:

- Upon receiving $m_0, m_1, b \leftarrow \{0, 1\}, r \leftarrow \{0, 1\}^L, c := r[0, \dots, v-1] \oplus m_b$, sends c to $\mathcal{A}$.

w_1 : event $b = \hat{b}$ in game 1.

From the analysis of the one-time pad, we obtain: $\Pr[w_1] = \frac{1}{2}$ (Example 2.6).

Construct B from $\mathcal{A}$, set $\delta(x, y) = \begin{cases} 1 & x = y, \\ 0 & x \neq y \end{cases}$ (Kronecker δ).

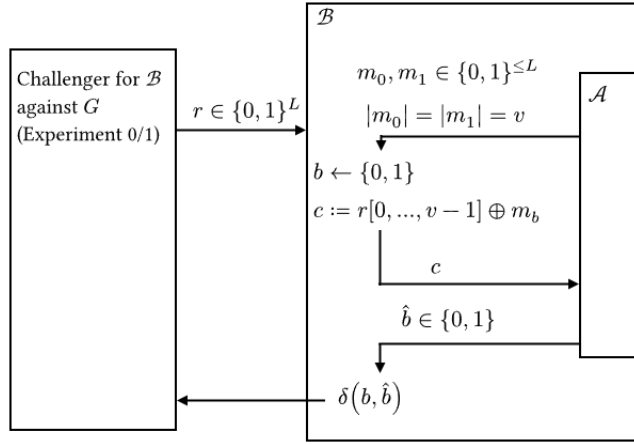


Figure 4: Proof diagram of Theorem 4.7

$\hat{w}_b$: event that $\mathcal{B}$ outputs 1 in experiment b .

- $\Pr[\hat{w}_0] = p_0, \Pr[\hat{w}_1] = p_1$
- $\text{PRGAdv}(\mathcal{B}, G) = |p_0 - p_1|$
- $p_0 = \Pr[w_0], p_1 = \Pr[w_1]$, hence

$$\begin{aligned} \text{SSAdv}^*(\mathcal{A}, \mathcal{E}) &= |\Pr[w_0] - \frac{1}{2}| \\ &= |\Pr[w_0] - \Pr[w_1]| \\ &= |p_0 - p_1| \\ &= \text{PRGAdv}(\mathcal{B}, G) \end{aligned}$$

- Furthermore, $\mathcal{B}$ is efficient if $\mathcal{A}$ is efficient.

5.3 Theorem 4.11

Show: For all efficient $\mathcal{A}$, there exists efficient $\mathcal{B}$

$$\text{PDAAdv}(\mathcal{A}, G) = \text{PRGAdv}(\mathcal{B}, G)$$

$\mathcal{A}$ is arbitrary but fixed and efficient. Assume $\mathcal{A}$ wins the game with a probability of $\frac{1}{2} + \epsilon$.

$$\Rightarrow \text{PDAAdv}(\mathcal{A}, G) = |\epsilon|$$

Construct $\mathcal{B}$ from $\mathcal{A}$:

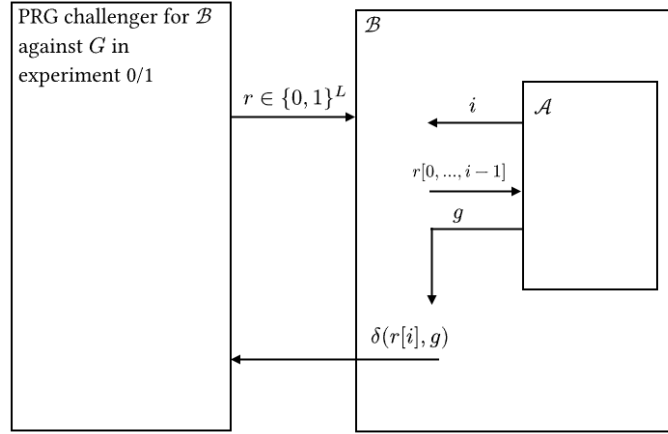


Figure 5: Proof diagram of Theorem 4.11

- $\mathcal{B}$ is efficient if $\mathcal{A}$ is efficient.

For $b \in \{0, 1\}$, w_b : event that $\mathcal{B}$ outputs 1 in experiment b .

Experiment 0

- r : pseudo-random output from G .
- w_0 occurs if and only if $g = r[i]$.

$$\Rightarrow \Pr[w_0] = \frac{1}{2} + \epsilon$$

Experiment 1

- r is chosen uniformly at random.
 - $r[i]$ is independent from $r[0, \dots, i-1]$ for all i .
- $$\Rightarrow g \text{ is independent from } r[i].$$

$$\Rightarrow \Pr[w_1] = \Pr[r[i] = g] = \frac{1}{2}.$$

Overall:

$$\begin{aligned} \text{PRGAdv}(\mathcal{B}, G) &= |\Pr[w_0] - \Pr[w_1]| \\ &= \left| \frac{1}{2} + \epsilon - \frac{1}{2} \right| \\ &= |\epsilon| \\ &= \text{PDAdv}(\mathcal{A}, G) \quad \square \end{aligned}$$

6 Week 6 – MACs and PRFs

6.1 Theorem 5.6

$\mathcal{A}$ is a MAC adversary in the forgery game with verification queries:

- $Q_s := \#$ of signature queries of $\mathcal{A}$
- $Q_u := \#$ of verification queries

Using $\mathcal{A}$, construct adversary $\mathcal{B}$ in the forgery game without verification queries:

- $\mathcal{B}$ uses $\mathcal{A}$
- Plays as the challenger for $\mathcal{A}$
- Needs to answer $\mathcal{A}$'s queries:
 - For signature queries: $\mathcal{B}$ uses its own challenger
 - For verification queries: $\mathcal{B}$ answers with reject

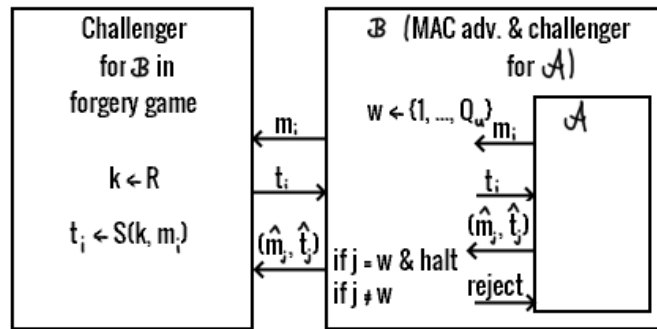


Figure 6: $\mathcal{B}$ from $\mathcal{A}$

Define 3 games for $\mathcal{A}$, which differ in the way the challenger answers $\mathcal{A}$'s verification queries.

Games: Forgery game with verification queries

Challenger for $\mathcal{A}$ in game 0:

- $k \leftarrow \mathcal{K}$
- Upon receiving a signature query m_i from $\mathcal{A}$:
 - $t_i \leftarrow S(k, m_i)$
 - Sends t_i to $\mathcal{A}$
- Upon receiving a verification query $(\hat{m}_j, \hat{t}_j)$ from $\mathcal{A}$:
 - $r_j \leftarrow V(k, \hat{m}_j, \hat{t}_j)$
 - Send r_j to $\mathcal{A}$ (*)

This is the way the challenger works in the original forgery game with verification queries.

- $w_0 := \text{event } r_j = \text{accept for some } j \text{ in game 0.}$
- $\Pr[w_0] = \text{MACAdv}(\mathcal{A}, I)$

Game 1: Modify the challenger for $\mathcal{A}$; instead of (*), send reject to $\mathcal{A}$.

- $w_1 := \text{event in game 1, } r_j = \text{accept for some } j.$

Game 2: Modify the challenger for $\mathcal{A}$; instead of (*), send reject.

- At the beginning of the game, the challenger chooses $w \leftarrow \{1, \dots, Q_v\}$.
- Otherwise, the challenger behaves as in game 1.
- $w_2 := \text{event that } r_w = \text{accept in game 2.}$

$$\Pr[w_2] = \text{MACAdv}(\mathcal{B}, I)$$

To analyze $\Pr[w_1]$, note that until $\mathcal{A}$ makes a verification query with the answer accept, game 1 and game 0 are identical.

$$\Pr[w_0] = \Pr[w_1]$$

More precisely:

$$\begin{aligned}
 \Pr[w_1] &= \Pr \left[\bigvee_{j=1}^{Q_v} r_j = \text{accept} \right] \\
 &= \Pr [\exists j : r_j = \text{accept and } j \text{ is the smallest such index in game 1}] \\
 &= \Pr [\exists j : r_j = \text{accept and } j \text{ is the smallest such index in game 0}] \\
 &= \Pr[w_0]
 \end{aligned}$$

$$\begin{aligned}
\Pr[w_2] &= \Pr[w_2 \wedge w_1] \\
&= \underbrace{\Pr[w_2 \mid w_1]}_{\geq 1/Q_v} \cdot \Pr[w_1] \\
&\geq \frac{1}{Q_v} \cdot \Pr[w_1]
\end{aligned}$$

$$\Pr[w_2 \mid w_1] \geq \frac{1}{Q_v}$$

Let $A = \{j : r_j = \text{accept}\}$.

Conditional on $w_1 : A \neq \emptyset \Rightarrow |A| \geq 1$.

$$\Rightarrow \Pr[w_2 \mid w_1] \geq \frac{1}{Q_v}$$

To conclude:

$$\begin{aligned}
MACAdv(\mathcal{B}, I) &= \Pr[w_2] \\
&\geq \frac{1}{Q_v} \cdot \Pr[w_1] \\
&= \frac{1}{Q_v} \cdot \Pr[w_0] \\
&= \frac{1}{Q_v} \cdot MACAdv^{\text{eq}}(\mathcal{A}, I) \quad \square
\end{aligned}$$

7 Week 7 – MACs and Hashing

7.1 Theorem 5.11

Remarks on Theorem 5.11

- block cipher: indistinguishable from random permutations
- pseudo-random functions: indistinguishable from random functions
- $|X| = N$ poly-bounded, then
- a random permutation can be distinguished from a random function
- otherwise not

Experiment b :

Challenger chooses f as follows:

$$\underline{b=0}: f \leftarrow \text{Perm}[X]$$

$$\underline{b=1}: f \leftarrow \text{Funs}[X, X]$$

Adversaries distinguishing between experiment 0 & experiment 1, can query the challenger with $x_i \in X$ and gets back $f(x_i), i = 1, \dots, Q$. Eventually, $\mathcal{A}$ outputs $\hat{b} \in \{0, 1\}$.

Then call w_b the event that $\mathcal{A}$ outputs

- $\hat{b} = 1$ in exp. b

Random permutations can be distinguished from random functions, if there is an efficient adversary $\mathcal{A}$ such that

$$|Pr[w_0] - Pr[w_1]|$$

is not negligible.

Claim: If $|X|$ is poly-bounded then such an adversary exists.

Proof:

$\mathcal{A}$:

- for all $x \in X$ queries challenger for $f(x)$
- if there exist x, x' with $x \neq x'$ and $f(x) = f(x')$ output 1, otherwise output 0. (*)
- $Pr[w_0] = 0$
- $Pr[w_1] = \frac{|Funs(X, X)| - |Perm(X)|}{|Funs(X, X)|}$

(Step (*) is only efficiently possible if X is poly-bounded.)

$|X| = N$:

- $|Perm(X)| = N!$
- $|Funs(X, X)| = N^N$
- $Pr[w_1] = \frac{N^N - N!}{N^N} = 1 - \frac{N!}{N^N} \geq \frac{1}{2}$

7.2 Remarks on Definition of Collision Resistant Hash Function

$H : \mathcal{M} \rightarrow \mathcal{T}, |\mathcal{M}| > |\mathcal{T}|$

$\Rightarrow$ collisions must exist

$H : \mathcal{M} \rightarrow \mathcal{T}$ fixed, collision (m, m') can be "hardwired" into adversary $\mathcal{A}$, i.e. $\mathcal{A}$ always outputs (m, m') .

(*hardwired* means that the output is fixed no matter what)

7.3 Proof of Theorem 5.15

Let $\mathcal{A}$ be arbitrary but fixed adversary against I' . $m_1, m_2, \dots$ are $\mathcal{A}$'s signature queries. (m, t) is $\mathcal{A}$'s output.

3 events:

- X : $\mathcal{A}$ wins forgery game with respect to I'
- Y : $\exists i : m \neq m_i \wedge H(m) = H(m_i)$
- Z : X occurs, but Y does not

Show:

- a) $MACadv(\mathcal{A}, I) \leq \Pr[Z] + \Pr[Y]$
- b) $\exists B_H : \Pr[Y] = CRadv(B_H, H)$
- c) $\exists B_I : \Pr[Z] = MACadv(B_I, I)$
- d) $\mathcal{A}$ efficient $\Rightarrow B_H, B_I$ efficient

7.3.1 a)

$$\begin{aligned}
 MACadv(\mathcal{A}, I') &= \Pr[X] \\
 &= \Pr[X \wedge Y] + \Pr[X \wedge \neg Y] \\
 &= \Pr[X|Y] \cdot \Pr[Y] + \Pr[X \wedge \neg Y] \\
 &\leq \Pr[Y] + \Pr[Z]
 \end{aligned}$$

7.3.2 b)

From $\mathcal{A}$ construct B_H as follows:

B_H playing challenger for $\mathcal{A}$ in forgery game with respect to I' :

- $k \leftarrow \mathcal{K}$
- upon receiving signing query $m_i \in \mathcal{M}$ from $\mathcal{A}$
 - $t_i \leftarrow S(k, H(m_i))$
 - send t_i to $\mathcal{A}$
- upon receiving message/tag pair (m, t) from $\mathcal{A}$
 - If $H(m) = H(m_i)$ and $m \neq m_i$ for some i , then output (m, m_i)

Hence: Y happens with the same probability as in forgery game with B_H playing challenger

- Y leads to success for B_H in collision resistance game
- $\Rightarrow \Pr[Y] = CRadv(B_H, H)$

7.3.3 c)

Construct B_I from $\mathcal{A}$ as follows:

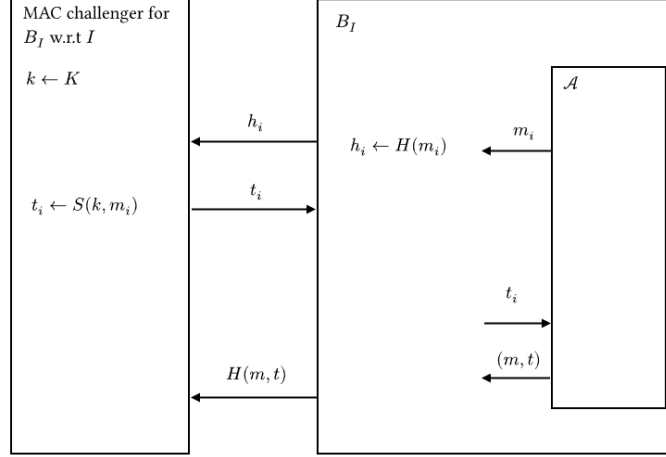


Figure 7: Construction of B_I from $\mathcal{A}$

B_I answers queries as in forgery game against I' for adversary $\mathcal{A}$.

$\Rightarrow Z$ happens with the same probability as in original forgery game

$Z \Leftrightarrow B_I$ outputs successful forgery in forgery game against I .

$\Rightarrow \Pr[Z] = \text{MACadv}(B_I, I)$.

7.3.4 d)

$\mathcal{A}$ efficient $\Rightarrow B_H, B_I$ efficient

8 Proof of Lemma 5.16

BA := event birthday attack succeeds

Fact: $\Pr[BA]$ is minimized if for m uniform from $\mathcal{M}$, $H(m)$ is uniform in $\mathcal{T}$.

Assume that $H(m)$ uniform in $\mathcal{T}$ for m uniform in $\mathcal{M}$.

CM := event that in birthday attack, there exists $i \neq j$, with $m_i = m_j$.

$$\Pr[BA] = 1 - \Pr[\neg BA]$$

$$\begin{aligned}
\Pr[\neg BA] &= \Pr[\neg BA \wedge CM] + \Pr[\neg BA \wedge \neg CM] \\
&= \Pr[\neg BA|CM] \cdot \Pr[CM] + \Pr[\neg BA|\neg CM] \cdot \Pr[\neg CM] \\
&\leq \Pr[CM] + \Pr[\neg BA|\neg CM]
\end{aligned}$$

Fact: $\Pr[CM] \leq 1/5$

Claim: $\Pr[\neg BA|\neg CM] \leq 1/4$

Hence: $\Pr[BA] = 1 - \Pr[\neg BA] \geq 1 - 1/5 - 1/4 > 1/2$

Proof of Claim:

use *forall* $x : 1 - x \leq e^{-x} \quad (*)$

$$\begin{aligned}
\Pr[\neg BA|\neg CM] &= \prod_{i=2}^s \frac{N - (i-1)}{N} \\
&= \prod_{i=1}^{s-1} \frac{N - i}{N} \\
&= \prod_{i=1}^{s-1} \left(1 - \frac{1}{N}\right) \\
&\stackrel{(*)}{\leq} \prod_{i=1}^{s-1} e^{-i/N} \\
&= e^{-\sum_{i=1}^{s-1} i/N} \\
&= e^{-\frac{s(s-1)}{2N}} \\
&\leq 1/4 \quad \checkmark
\end{aligned}$$

The security of hash functions degrades at least with the square root of the size of the text space.

9 Week 8 – Public Key Encryption and RSA

9.1 Extended Euclidean Algorithm

- (a) at the end of the first while loop: $e_j = 0, \gamma_k = \alpha = \gcd(e, \gamma)$
- (b) invariant for second while loop: $\alpha = \delta_j \cdot \gamma_j + d_j \cdot e_j$

$$\begin{aligned}
j = 0, \alpha &= \delta_0 \cdot \gamma_0 + d_0 \cdot e_0 \\
&= \delta_0 \cdot \gamma + d_0 \cdot e
\end{aligned}$$

(a) follows from: $\forall a, b \in \mathbb{Z}$:

$$\begin{aligned} \gcd(a, b) &= \gcd(b, a \bmod b) \\ \gcd(a, 0) &= a \end{aligned}$$

(b) from (a) before first iteration of second while loop: $e_j = 0, \gamma_j = \alpha, \delta_j = 1, d_j = 0$

$$\Rightarrow \delta_j \cdot \gamma_j + d_j \cdot e_j = \alpha$$

assume that invariant holds before an iteration, then from first while loop:

$$\gamma_{j-1} = q_{j-1} \cdot e_{j-1} + e_j$$

$$\Leftrightarrow e_j = \gamma_{j-1} - q_{j-1} \cdot e_{j-1}$$

$$\begin{aligned} \alpha &= \delta_j \cdot \gamma_j + d_j \cdot e_j \\ &= \delta_j \cdot \gamma_j + d_j (\gamma_{j-1} - q_{j-1} \cdot e_{j-1}) \\ &= \delta_j \cdot e_{j-1} + d_j \cdot \gamma_{j-1} - d_j \cdot q_{j-1} \cdot e_{j-1} \\ &= d_j \cdot \gamma_{j-1} + (\delta_j - d_j \cdot q_{j-1}) \cdot e_{j-1} \\ &= \delta_{j-1} \cdot \gamma_{j-1} + d_{j-1} \cdot e_{j-1} \\ &\Rightarrow \text{Invariant true after iteration} \end{aligned}$$

9.2 Theorem 6.6

$$Enc((N, e), m) = m^e \bmod N$$

$$Dec((N, d), c) = c^d \bmod N$$

$$Dec((N, d), Enc((N, e), m)) = (m^e)^d \bmod N$$

9.3 Lemma 6.9

$$m \in \mathbb{Z}_N^*$$

$$e \cdot d = 1 \bmod \varphi(N)$$

$$\Rightarrow \exists k \in \mathbb{Z} : e \cdot d = 1 + k \cdot \varphi(N)$$

need to show that:

$$(m^e)^d = m \bmod N$$

$$\begin{aligned} (m^e)^d &= m \\ &= m^{1+k \cdot \varphi(N)} \\ &= m \cdot m^{k \cdot \varphi(N)} \\ &= m \cdot \left(m^{\varphi(N)}\right)^k \\ &= m \cdot 1^k \\ &= m \bmod N \end{aligned}$$

9.4 Lemma 6.11

$$m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$$

$$m = 0 \pmod{P} \text{ or } m = 0 \pmod{Q}$$

P, Q primes: $m \neq 0 \pmod{P}$ then $m \in \mathbb{Z}_p^*$ and similarly for Q

$$m^{e \cdot d} = m \pmod{P}$$

$$m^{e \cdot d} = m \pmod{Q}$$

If $m = 0 \pmod{P}$ ✓

Of $m \neq 0 \pmod{P}$ then by Euler's Theorem $m^p = m$

$$\begin{aligned} m^{e \cdot d} &= m^{1+k \cdot \varphi(N)} \\ &= m \cdot m^{k \cdot \varphi(N)} \\ &= m \cdot (m^{P-1})^{k \cdot (Q-1)} \\ &= m \pmod{P} \end{aligned}$$

similarly for Q

$$m^{ed} = m \pmod{P}$$

$$m^{ed} = m \pmod{Q}$$

Hence

$$\begin{aligned} &(m \pmod{P}, m \pmod{Q}) \\ &= (m^{ed} \pmod{P}, m^{ed} \pmod{Q}) \end{aligned}$$

9.5 Theorem 6.16

$\mathcal{A}$ upon receiving (pk, c) , $pk = (N, e)$:

1. asks for the decryption of $2^e \cdot c \pmod{N}$

2. upon receiving answer $\tilde{m}$, $\mathcal{A}$ sets $\hat{m} := 2^{-1} \cdot \tilde{m} \pmod{N}$

2^{-1} exists, since N is odd

$$\begin{aligned} \hat{m} &= 2^{-1} \cdot (2^e \cdot c)^d \\ &= 2^{-1} \cdot 2^{e \cdot d} \cdot c^d \\ &= 2^{-1} \cdot 2 \cdot m \\ &= m \pmod{N} \end{aligned}$$

10 Week 9 – Digital Signatures and RSA

10.1 Theorem 6.20

(similar to proof of Theorem 6.16)

$\mathcal{A}$ on input $1^\lambda, pk, (pk = (N, e))$:

1. Pick $m \in \mathbb{Z}_N$ arbitrary with $m \neq 0$
2. Ask for signature $\bar{\sigma}$ for message $\bar{m} := 2^e \cdot m \bmod N$
3. After receiving $\bar{\sigma} = \bar{m}^d \bmod N$ compute $\sigma = 2^{-1} \cdot \bar{\sigma} \bmod N$ & output (m, σ)

Show: $\mathcal{A}$ always succeeds

$$\begin{aligned}\sigma^e &= (2^{-1} \cdot \bar{\sigma})^e \\ &= 2^{-e} \cdot (\bar{\sigma})^e \\ &= 2^{-e} \cdot 2^e \cdot m \\ &= m \bmod N\end{aligned}$$

10.2 Theorem 6.21

if there exist $i, j \in \{1, 2, 3\}$, $i \neq j$, $\gcd(N_i, N_j) \neq 1$, then one can factor N_i, N_j efficiently.

assume $\gcd(N_i, N_j) = 1$ for $i, j \in \{1, 2, 3\}, i \neq j$.

$$m < N_i \quad \forall i \in \{1, 2, 3\}$$

use Chinese Remainder Theorem to compute $c \in \mathbb{Z}_N$, $N = N_1 \cdot N_2 \cdot N_3$

$$c_i = c \bmod N_i, \quad i = 1, 2, 3$$

$$\Rightarrow c = m^3 \bmod N_i, \quad i = 1, 2, 3$$

$$c < N, \quad m^3 < N \Rightarrow c = m^3 \text{ in } \mathbb{Z}$$

can compute m from c by taking 3rd roots in $\mathbb{Z}$

10.3 Theorem 6.22

$$\gcd(e_1, e_2) = 1 \Rightarrow \exists s_1, s_2 \in \mathbb{Z} : s_1 \cdot e_1 + s_2 \cdot e_2 = 1$$

compute:

$$\begin{aligned}c_1^{s_1} \cdot c_2^{s_2} &= (m^{e_1})^{s_1} \cdot (m^{e_2})^{s_2} \\ &= m^{e_1 \cdot s_1} \cdot m^{e_2 \cdot s_2} \\ &= m^{e_1 \cdot s_1 + e_2 \cdot s_2} \\ &= m \bmod N\end{aligned}$$

11 Week 10 – Key Exchange Protocols and the Discrete Logarithm Problem

11.1 Diffie-Hellman Key Exchange – Number Theory

$Ord_p(g) = q \Rightarrow u^a = u^b \pmod p$ for $u \in \mathbb{G}$, $a = b \pmod q$

Suffices to prove this for $u = g$

$a = b \pmod q \Rightarrow \exists k : b = k \cdot q + a$

$$\begin{aligned} g^b &= g^{k \cdot q + a} = g^{k \cdot q} \cdot g^a \\ &= (g^q)^k \cdot g^a \\ &= 1^k \cdot g^a \\ &= g^a \pmod p \end{aligned}$$

11.2 Theorem 7.10

(by contraposition)

Assume DL assumption does not hold for $\mathbb{G}$, i.e. exists efficient $\mathcal{A}$ with

$$DLadv(\mathcal{A}, \mathbb{G}) = p$$

is not negligible.

Construct adversary $\mathcal{B}$ as follows:

- Upon receiving $u = g^\alpha, v = g^\beta$ for $\alpha, \beta \leftarrow \mathbb{Z}_q$, use $\mathcal{A}$ to compute $\hat{\alpha}, \hat{\beta} \in \mathbb{Z}_q$
- Compute & output $\hat{w} = g^{\hat{\alpha} \cdot \hat{\beta}}$

If $\mathcal{A}$ is efficient, $\mathcal{B}$ is efficient.

$$\begin{aligned} CDHadv(\mathcal{B}, \mathbb{G}) &\geq Pr[\alpha = \hat{\alpha} \wedge \beta = \hat{\beta}] \\ &= Pr[\alpha = \hat{\alpha}] \cdot Pr[\beta = \hat{\beta}] \text{ (because } \alpha, \beta \leftarrow \mathbb{Z}_q \text{ independant)} \\ &= DLadv(\mathcal{A}, \mathbb{G}) \cdot DLadv(\mathcal{A}, \mathbb{G}) \\ &= p^2 \end{aligned}$$

p is not negligible, then p^2 is also not negligible

11.3 Theorem 7.13

Show that for every $\mathcal{A}$ in CDH game, there is a $\mathcal{B}$ in the DDH game, such that

$$DDHadv(\mathcal{B}, \mathbb{G}) + \frac{1}{q} \geq CDHadv(\mathcal{A}, \mathbb{G})$$

Define $\mathcal{B}$ as follows:

- Upon receiving the tripe (u, v, w) $\mathcal{B}$ runs $\mathcal{A}$ with input (u, v) to obtain $\hat{w}$
- if $\hat{w} = w$, then $\mathcal{B}$ outputs 1 otherwise 0

If $\mathcal{A}$ is efficient, then $\mathcal{B}$ is efficient

$$\begin{aligned} Pr[\mathcal{A} \text{ wins CDH game}] &= \epsilon \\ Pr[W_0] &= \epsilon \\ Pr[W_1] &= \frac{1}{q} \text{ as } u, v \text{ are independant of } \gamma \end{aligned}$$

$$\begin{aligned} DDadv(\mathcal{B}, \mathbb{G}) &= |Pr[w_0] - Pr[W_1]| \\ &\geq \epsilon - \frac{1}{q} \\ &= CDHadv(\mathcal{A}, \mathbb{G}) - \frac{1}{q} \end{aligned}$$

11.4 Theorem 7.14

$\mathcal{A}$ as in Theorem. Construct algorithm $\mathcal{B}$ as follows:

$\mathcal{B}$ on input g, u :

- Choose $\sigma \leftarrow \mathbb{Z}_q$ & set $u_1 := u \cdot g^\sigma$
- Run $\mathcal{A}$ on input g, u_1 & let α_1 be $\mathcal{A}$'s output
- If $u_1 \neq g^{\alpha_1}$, output *fail*, otherwise output $\alpha := \alpha_1 - \sigma$

Remarks:

- $g^{\alpha_1} = u_1 \Leftrightarrow u = g^\alpha = g^{\alpha_1 - \sigma} = g^{\alpha_1} \cdot g^{-\sigma}$
- σ uniform $\Rightarrow \alpha + \sigma$ uniform

$\Rightarrow$ distribution of u_1 is uniform in group $\mathbb{G}$

$\Rightarrow Pr[\mathcal{A} \text{ on input } u_1 \text{ outputs } \alpha_1 = Dlog_g(u_1)] = \epsilon$

$\Rightarrow Pr[\mathcal{B} \text{ outputs } Dlog_g(u)] = \epsilon$

11.5 Theorem 7.16

$\mathcal{B}$ from $\mathcal{A}$: On input (g, h) :

- If $h = 1$, output 0
- else run $\mathcal{A}$ with input H_{dl} (based on g, h)
- if $\mathcal{A}$ finds a collision consisting of pairs $(\alpha, \beta), (\alpha', \beta')$, i.e. $g^\alpha \cdot h^\beta = g^{\alpha'} \cdot h^{\beta'}$
output $(\alpha - \alpha') \cdot (\beta' - \beta)^{-1} \mod q$

$\mathcal{A}$ is efficient implies $\mathcal{B}$ is efficient

$h = 1$: $\mathcal{B}$ succeeds

$h \neq 1$: $\mathcal{B}$ succeeds if $\mathcal{A}$ succeeds (Lemma 7.17)

hence: $DLadv(\mathcal{B}, \mathbb{G}) \geq CRadv(\mathcal{A}, H_{dl})$

11.6 Lemma 7.17

let $u = g^\alpha \cdot h^\beta = g^{\alpha'} \cdot h^{\beta'}$

then $g^{\alpha - \alpha'} = h^{\beta' - \beta}$

$\Leftrightarrow h = g^{(\alpha - \alpha') \cdot (\beta' - \beta)^{-1}} \mod q$

$(\beta' - \beta)^{-1} \mod q$ exists, since $(\alpha, \beta) \neq (\alpha', \beta')$, hence $\beta \neq \beta'$

therefore $(\beta' - \beta)^{-1} \neq 0 \mod q$ and the inverse exists

12 Week 11 – DLog Hashing and Secure Public-Key Encryption under the DLog Problem (CPA and CCA)

12.1 Theorem 8.4

$c = (c_0, c_1) = (g^r, y^r \cdot m)$, $y = g^x = pk$, $x = sk$

$y^r \cdot (g^r)^{-x} \cdot m = g^{x \cdot r} \cdot g^{-r \cdot x} \cdot m = m$

12.2 Theorem 8.6

$\mathcal{A}$ arbitrary, but fixed, against OW-CPA

$\mathcal{B}$ on input (g, u, v)

$(u = g^\alpha, \alpha \leftarrow \mathbb{Z}_q, v = g^\beta, \beta \leftarrow \mathbb{Z}_q)$

- choose $c_1 \leftarrow \mathbb{G}$ uniformly, run $\mathcal{A}$ with input (g^α, v, c_1)
- upon receiving $\hat{m}$ from $\mathcal{A}$, compute and output $\hat{w} = c_1 \cdot \hat{m}^{-1}$ in $\mathbb{G}$

$$pk = g^\alpha, c = (c_0, c_1), c_0 = u, c_1$$

$\mathcal{A}$ efficient $\Rightarrow \mathcal{B}$ efficient

$$c = (c_0, c_1) = (u, c_1) \text{ corresponds to encryption } m = c_1 \cdot g^{-\alpha \cdot \beta}$$

$$\mathcal{A} \text{ wins } (\hat{m} = m) \Rightarrow \hat{w} = w = g^{\alpha \cdot \beta}$$

$$m \leftarrow \mathcal{M} = \mathbb{G} \Leftrightarrow m \cdot y^r = c_1 \leftarrow \mathbb{G}$$

$$\Rightarrow \text{CDHadv}(\mathcal{B}, \mathbb{G}) \geq \text{DW-CPAadv}(\mathcal{A}, \mathcal{E})$$

12.3 Theorem 8.11

$\mathcal{A}$ arbitrary in IND-CPA game against $\mathcal{E}$

Construct $\mathcal{B}$ in DDH experiment as follows

$\mathcal{B}$ upon receiving (u, v, w) :

- run $\mathcal{A}$ with input $pk = u$ to obtain m_0, m_1
- pick $b' \leftarrow \{0, 1\}$, set $c = (v, w \cdot m_{b'})$ and run $\mathcal{A}$ with input c
- upon receiving $\hat{b}$ from $\mathcal{A}$, if $b' = \hat{b}$ output 1, otherwise output 0

$W := \mathcal{A} \text{ wins, i.e. } \hat{b} = b', W_0, W_1 \text{ as before}$

Experiment 0: $u = g^\alpha, v = g^\beta, w = g^{\alpha \cdot \beta}$

$$\Rightarrow pk = u, c = (c_0, c_1), c_0 = g^\beta, c_1 = m_{b'} \cdot g^{\alpha \cdot \beta}$$

$\Rightarrow c$ is an encryption of $m_{b'}$ under $pk = u$

$\Rightarrow$ input to $\mathcal{A}$ is as in the bit guessing indistinguishability CPA game

$$\Rightarrow \Pr[W_0] = \Pr[W]$$

Experiment 1: $w = g^\gamma, \gamma \leftarrow \mathbb{Z}_q$

$\Rightarrow w$ is independent of u, v

$$\Rightarrow (u, c) \leftarrow \mathbb{G}^2 \text{ } ((u, v) \text{ independent from } b')$$

$$\Rightarrow \Pr[\hat{b} = b'] = \frac{1}{2}$$

Overall

$$\begin{aligned} \text{DDHadv}(\mathcal{B}, \mathbb{G}) &= |\Pr[W_0] - \Pr[W_1]| \\ &= |\Pr[W] - \frac{1}{2}| \\ &= \text{IND-CPAadv}^*(\mathcal{A}, \mathcal{E}) \end{aligned}$$

$\mathcal{A}$ efficient $\Rightarrow \mathcal{B}$ efficient

12.4 Theorem 8.15

$\mathcal{A}$ defined as follows:

- picks m_0, m_1 so that $m_0 \neq m_1$
- upon receiving $c = (c_0, c_1)$ it computes $\hat{c} = (c_0, 2 \cdot c_1)$ and asks for decryption of $\hat{c}$
- upon receiving message $\hat{m}$, $\mathcal{A}$ computes $m' = 2^{-1} \cdot \hat{m}$ (in $\mathbb{G}$) and outputs 1 if $m' = m_0$, otherwise it outputs 0

$\mathcal{A}$ efficient

Experiment 0:

$$c = (c_0, c_1) = (g^r, g^{r \cdot x} \cdot m_0), x = sk, y = g^x = pk$$

$$\hat{c} = (c_0, 2 \cdot c_1) = (g^r, g^{r \cdot x} \cdot 2m_0) \Rightarrow \hat{c} \text{ encryption of } 2 \cdot m_0$$

$$\Rightarrow \hat{m} = 2 \cdot m_0 \Rightarrow m' = m_0$$

$$\Rightarrow Pr[W_0] = 1$$

Experiment 1: $\hat{m} = 2 \cdot m_1, m' = m_1$

$$\Rightarrow Pr[W_0] = 0$$

13 Week 12 – DLog Hashing and Secure Public-Key Encryption under the DLog Problem (Signatures)

13.1 Correctness – Schnorr ID Scheme

$$g^r = g^{c \cdot x + \alpha} = (g^x)^c \cdot g^\alpha = h^c \cdot u$$

13.2 Theorem 10.2

Proof idea: Use P to compute two triples (u, c, r) , (u, c', r') that are different and lead to acceptance by V .

- $(u, c, r) \neq (u, c', r') \Rightarrow c \neq c'$
- $g^r = h^c \cdot u, g^{r'} = h^{c'} \cdot u$
 $\Rightarrow g^r \cdot h^{-c} = u = g^{r'} \cdot h^{-c'}$
 $\Rightarrow g^{r-r'} = h^{c-c'}$
 $\Rightarrow g^{(r-r')(c-c')^{-1} \bmod q} = h = DLog_g(h) = x$

E on input (g, h) and interacting with P :

- Run P to obtain $u = g^\alpha$
- Choose $c \leftarrow \mathbb{Z}_q$ and simulate P with input (u, c) to obtain r

- If (u, c, r) is accepted, go to next step, otherwise output “fail”
- Choose $c' \leftarrow \mathbb{Z}_q$ and simulate P with input (u, c') to obtain r'
- If $(u, c', r') \neq (u, c, r)$ and is accepting, then output $(r - r') \cdot (c - c')^{-1} \bmod q$

P is efficient $\Rightarrow$ E efficient

Analysis $Pr[E \text{ outputs } x] = \epsilon'$:

$\epsilon := Pr[P \text{ makes V accept}]$

Assume that P after picking u is deterministic.

$\forall u : C_u := \{c \mid (u, c, r) \text{ is accepting}\}$

- u good: $|C_u| \geq \epsilon/2 \cdot q$
- u bad: otherwise

$$\begin{aligned}
\epsilon' &= Pr[E \text{ outputs } x] \\
&\geq Pr[P \text{ chooses good } u \wedge c, c' \in C_u \wedge c \neq c'] \\
&= Pr[c, c' \in C_u, c \neq c' \mid u \text{ good}] \cdot Pr[u \text{ good}] \\
&= \underbrace{Pr[c' \in C_u \wedge c \neq c' \mid c \in C_u \wedge u \text{ good}]}_{(1)} \cdot \underbrace{Pr[c \in C_u \mid u \text{ good}]}_{(2)} \cdot \underbrace{Pr[u \text{ good}]}_{(3)}
\end{aligned}$$

$$(1) \geq \epsilon/2 - 1/q$$

$$(2) \geq \epsilon/2$$

$$(3) \geq \epsilon/2$$

$$\text{hence: } \epsilon' \geq \epsilon^2/4 \cdot (\epsilon/2 - 1/q)$$

$$\Rightarrow \epsilon' \geq \epsilon^3/2 - 1/q$$

$$\epsilon \text{ non-negligible} \Rightarrow \epsilon' \text{ non-negligible}$$

Remains to prove (1) & (3):

$$\begin{aligned}
\epsilon &= Pr[P \text{ makes V accept}] \\
&= Pr[P \text{ makes V accept} \wedge u \text{ good}] + Pr[P \text{ makes V accept} \wedge u \text{ bad}] \\
&= Pr[P \text{ makes V accept} \mid u \text{ good}] \cdot Pr[u \text{ good}] + Pr[P \text{ makes V accept} \mid u \text{ bad}] \cdot Pr[u \text{ bad}] \\
&\leq Pr[u \text{ good}] + \epsilon/2
\end{aligned}$$

$$\Rightarrow \epsilon/2 \leq Pr[u \text{ good}]$$

13.3 Theorem 10.4

S on input (g, h) :

- $r \leftarrow \mathbb{Z}_q, c \leftarrow \mathbb{Z}_q$
- set $u := g^r \cdot h^{-c}$
- output triple (u, c, r)

in the protocol:

- $c \leftarrow \mathbb{Z}_q$
- $r \leftarrow \mathbb{Z}_q$
- u satisfies $u = g^r \cdot h^{-c}$

13.4 Security of Schnorr ID Scheme

Security of Schnorr ID Scheme because of order of which the values are picked/calculated

13.5 Correctness – Schnorr Signatures

$$m, \sigma = (c, r), u = g^r \cdot h^{-c} = g^\alpha$$

$$r = c \cdot x + \alpha \Rightarrow g^r \cdot h^{-c} = g^{c \cdot x + \alpha} \cdot h^{-c} = h^c \cdot g^\alpha \cdot h^{-c} = g^\alpha$$

13.6 Theorem 10.11

$$\begin{aligned} g^v \cdot h^w &= g^{H(m) \cdot r^{-1}} \cdot h^{u \cdot r^{-1}} \mod p \\ &= g^{H(m) \cdot r^{-1}} \cdot g^{x \cdot u \cdot r^{-1}} \mod p \\ &= g^{r^{-1}(H(m) + x \cdot u)} \mod p \\ &= g^\alpha \mod p \underset{\text{as by definition}}{=} (u \mod p) \mod q \end{aligned}$$