3 – Private Key Encryption

Eklavya Sharma

Contents

1 Computational Security

1.1 Definition of relaxations

Unlike perfect security, we make 2 additional assumptions to make secure encryption practical:

- Adversaries are **efficient** and only run for a feasible amount of time.
- Adversaries have a **negligible** probability of success.

All encryption schemes are parametrized by a security parameter n. n is usually the key length. The terms 'efficient' and 'negligible' are defined in terms of n.

Definition 1. *An efficient adversary is a probabilistic polynomial-time (PPT) algorithm, where the input is at least as large as the security parameter.*

Definition 2. *Denote the set of all negligible functions of n as* $\mathrm{negl}(n)$*, where*

$$\mathrm{negl}(n) = \bigcap_{k\in\mathbb{N}} o(n^{-k})$$

Theorem 1.

$$f \in \mathrm{negl}(n) \iff \forall p(x) \in \mathbb{R}[x], \exists N \in \mathbb{N}, \forall n \geq N, f(n) < \frac{1}{p(n)}$$

Theorem 2.

$$f \in \mathrm{negl}(n) \implies (\forall p(x) \in \mathbb{R}[x], p(n)f(n) \in \mathrm{negl}(n))$$

1.2 [Draft] Necessity of the relaxations

(TODO: Needs rigor)

- Powerful adversary can brute force the set of keys to break scheme with very high probability.
- Normal adversary can guess key and break scheme with slightly higher probability than pure guess.

2 Defining Computationally Secure Encryption

- The key-generation algorithm Gen takes input 1^n and returns key k. We assume (why?) that $|k| \geq n$.
- The encryption algorithm Enc takes the key and message as input and outputs a ciphertext.
- The decryption algorithm Dec takes the key and ciphertext as input and outputs a message.

Definition 3. *The adversarial indistinguishability experiment* $\mathrm{PrivK}^{eav}_{A,\Pi}(n)$*:*

1. *A is given input 1^n. It outputs 2 messages m_0 and m_1 with $|m_0| = |m_1|$.*
2. *$k \in \mathcal{K}$ is generated by running $\mathsf{Gen}(1^n)$. b is chosen uniformly randomly from $\{0, 1\}$. $c = e_k(m)$, called the challenge ciphertext, is computed and given to A.*
3. *A outputs a bit b'.*
4. $\mathrm{PrivK}^{eav}_{A,\Pi}(n) = \begin{cases} 1 & \text{if } b' = b \\ 0 & \text{if } b' \neq b \end{cases}$.

Messages output by adversary are required to be of the same length otherwise adversary can use ciphertext length to determine which message was encrypted.

Definition 4. *Scheme Π is EAV-secure iff for every PPT adversary A,*

$$\Pr\left[\mathrm{PrivK}^{eav}_{A,\Pi}(n) = 1\right] - \frac{1}{2} \in \mathrm{negl}(n)$$

Definition 5. *Let $\mathrm{PrivK}^{eav}_{A,\Pi}(n, b)$ be the experiment where the message chosen by the challenger is fixed to be m_b (instead of choosing uniformly randomly from $\{m_0, m_1\}$), but the adversary doesn't know this. Let $\mathrm{PrivKOut}^{eav}_{A,\Pi}(n, b)$ be the output b' of the adversary.*

Theorem 3. *Π is secure iff for all PPT adversaries A,*

$$\left|\Pr\left[\mathrm{PrivKOut}^{eav}_{A,\Pi}(n, 1) = 1\right] - \Pr\left[\mathrm{PrivKOut}^{eav}_{A,\Pi}(n, 0) = 1\right]\right| \in \mathrm{negl}(n)$$

3 Pseudorandom Generators

Definition 6. *Let $l(n)$ be a polynomially-bounded function. Let $G : \{0, 1\}^n \mapsto \{0, 1\}^{l(n)}$ be a deterministic polynomial-time algorithm. G is a pseudorandom generator (aka PRG) iff both these conditions hold:*

- *Expansion: $\forall n > 1, l(n) > n$.*
- *Pseudorandomness: For any PPT algorithm D,*

$$\left|\Pr_{s \in_R \{0,1\}^n}[D(G(s)) = 1] - \Pr_{r \in_R \{0,1\}^{l(n)}}[D(r) = 1]\right| \in \mathrm{negl}(n)$$

A pseudorandom generator G can be used to construct an encryption scheme Π_G:

- Gen: $k \in_R \mathcal{K}$.
- $e_k(m) = m \oplus G(k)$.
- $d_k(c) = c \oplus G(k)$.

Theorem 4. *G is a PRG $\implies$ Π_G is EAV-secure.*

4 Stronger notions of security

4.1 Multiple messages

The multiple-message eavesdropping experiment $\mathrm{PrivK}^{\mathrm{mult}}_{A,\Pi}(n)$:

1. The adversary A is given input 1^n and outputs $M_0 = [m_{0,i}]_{i=1}^t$ and $M_1 = [m_{1,i}]_{i=1}^t$ where $\forall i, |m_{0,i}| = |m_{1,i}|$.
2. $k = \mathsf{Gen}(1^n)$, $b \in_R \{0, 1\}$.
3. $\forall i, c_i = e_k(m_{b,i})$. $C = [c_i]_{i=1}^n$.

4. A is given C and it outputs a bit b'.

5. $\mathrm{PrivK}^{\mathrm{mult}}_{A,\Pi}(n) = \begin{cases} 1 & \text{if } b' = b \\ 0 & \text{if } b' \neq b \end{cases}.$

Definition 7. Π *has indistinguishable multiple encryptions iff*

$$\Pr\left[\mathrm{PrivK}^{mult}_{A,\Pi}(n) = 1\right] - \frac{1}{2} \in \mathrm{negl}(n)$$

Theorem 5. *Any stateless and deterministic encryption scheme has distinguishable multiple encryptions.*

Proof. The adversary chooses message lists $M_0 = (m, m)$ and $M_1 = (m, m')$. Then given $C = (c_1, c_2)$, it outputs $b' = (c_1 \neq c_2)$. The adversary succeeds with probability 1. □

5 Chosen-Plaintext Attack

The chosen-plaintext attack experiment $\mathrm{PrivK}^{\mathrm{cpa}}_{A,\Pi}(n)$:

1. $k = \mathsf{Gen}(1^n)$.
2. A is given input 1^n and oracle access to e_k and outputs (m_0, m_1) where $|m_0| = |m_1|$.
3. $b \in_R \{0, 1\}$. $c = e_k(m_b)$ is given to A.
4. A, which continues to have oracle access to e_k, outputs a bit b'.
5. $\mathrm{PrivK}^{\mathrm{cpa}}_{A,\Pi}(n) = \begin{cases} 1 & \text{if } b' = b \\ 0 & \text{if } b' \neq b \end{cases}.$

Definition 8. Π *is indistinguishable under chosen-plaintext attack iff*

$$\Pr\left[\mathrm{PrivK}^{cpa}_{A,\Pi}(n)\right] - \frac{1}{2} \in \mathrm{negl}(n)$$

Definition 9. *The LR-oracle* $\mathrm{LR}_{k,b}$ *is a function where* $\mathrm{LR}_{k,b}(m_0, m_1) = e_k(m_b)$.

The LR-oracle experiment $\mathrm{PrivK}^{\mathrm{lr\text{-}cpa}}_{A,\Pi}(n)$:

1. $k = \mathsf{Gen}(1^n)$. $b \in_R \{0, 1\}$.
2. A is given input 1^n and oracle access to $\mathrm{LR}_{k,b}$.
3. A outputs a bit b'.
4. $\mathrm{PrivK}^{\mathrm{lr\text{-}cpa}}_{A,\Pi}(n) = \begin{cases} 1 & \text{if } b' = b \\ 0 & \text{if } b' \neq b \end{cases}.$

Definition 10. Π *has indistinguishable multiple encryptions under chosen-plaintext attack if*

$$\Pr\left[\mathrm{PrivK}^{lr\text{-}cpa}_{A,\Pi}(n)\right] - \frac{1}{2} \in \mathrm{negl}(n)$$

Theorem 6. Π *is CPA-secure iff it is multi-CPA-secure.*

Proof. (Proof will appear in a later chapter) □

6 Pseudorandom Functions

Definition 11. *Let $|\mathcal{M}|, |\mathcal{K}|, |\mathcal{C}| \in \text{poly}(n)$. Let $\mathsf{Func}_{\mathcal{M},\mathcal{C}}$ be the family of all functions from $\mathcal{M}$ to $\mathcal{C}$. Let $F = \{F_k : k \in \mathcal{K}\} \subseteq \mathsf{Func}_{\mathcal{M},\mathcal{C}}$ be a function family. F is pseudorandom iff for every PPT distinguisher D,*

$$\left| \Pr_{k \in_R \mathcal{K}} \left[D^{F_k}(1^n) = 1 \right] - \Pr_{f \in_R \mathsf{Func}_{\mathcal{M},\mathcal{C}}} \left[D^f(1^n) = 1 \right] \right| \in \text{negl}(n)$$

Example 1. *$F_k(x) = x \oplus k$ is not pseudorandom.*

Definition 12. *A function family F is efficient iff $\forall f \in F$, f can be computed in polynomial time.*

Definition 13. *Let $|\mathcal{M}|, |\mathcal{K}| \in \text{poly}(n)$. Let $\mathsf{Perm}_{\mathcal{M}}$ be the family of all permutations of $\mathcal{M}$. Let $F = \{F_k : k \in \mathcal{K}\} \subseteq \mathsf{Perm}_{\mathcal{M}}$ be a permutation family. F is pseudorandom iff for every PPT distinguisher D,*

$$\left| \Pr_{k \in_R \mathcal{K}} \left[D^{F_k}(1^n) = 1 \right] - \Pr_{f \in_R \mathsf{Perm}_{\mathcal{M}}} \left[D^f(1^n) = 1 \right] \right| \in \text{negl}(n)$$

F is strongly pseudorandom iff for every PPT distinguisher D,

$$\left| \Pr_{k \in_R \mathcal{K}} \left[D^{F_k, F_k^{-1}}(1^n) = 1 \right] - \Pr_{f \in_R \mathsf{Perm}_{\mathcal{M}}} \left[D^{f, f^{-1}}(1^n) = 1 \right] \right| \in \text{negl}(n)$$

A pseudorandom permutation is also called a block cipher.

Definition 14. *A permutation family F is efficient iff $\forall f \in F$, f and f^{-1} can be computed in polynomial time.*

Theorem 7. *If $F = \{F_k : k \in \mathcal{K}\} \subseteq \mathsf{Perm}_{\mathcal{M}}$ is a pseudorandom permutation and $|\mathcal{M}| \geq |\mathcal{K}|$, then F is also a pseudorandom function.*

Proof. (TODO: Add proof) □

Theorem 8. *Let F be a pseudorandom function. Let $G(s) = \text{concat}_{i=1}^{l} F_s(i)$. Then G is a pseudorandom generator.*

Theorem 9. *A pseudorandom generator with expansion $l(n)$ can be used to construct a pseudorandom function with input and output size $O(\log n)$.*

Theorem 10. *Let F be a pseudorandom function family. Let $\Pi(n)$ be this scheme:*

- Gen : $k \in_R \{0,1\}^n$.
- $e_k(m) = (r, F_k(r) \oplus m)$, *where* $r \in_R \{0,1\}^n$.
- $d_k((r, c)) = F_k(r) \oplus c$.

Then $\Pi(n)$ is LR-CPA-secure.

7 CTR mode of operation

'Mode of operation' is a way of encrypting messages of variable lengths using a fixed-length block cipher.

Specification of CTR mode of operation, which uses a pseudorandom function family F:

- Gen: $k \in_R \{0,1\}^n$.
- $e_k([m_i]_{i=1}^l) = [\mathrm{IV}] + [F_k(c+i) \oplus m_i]_{i=1}^l$, where $\mathrm{IV} \in_R \{0,1\}^n$ is called the initialization vector.
- $d_k([\mathrm{IV}] + [m_i]_{i=1}^l) = [F_k(c+i) \oplus c_i]_{i=1}^l$

Theorem 11. *The CTR mode of operation is LR-CPA-secure.*