

Notes from CS 6260 (Applied Cryptography)

Georgia Tech, Fall 2012

Christopher Martin
chris.martin@gatech.edu

1 Symmetric cryptography scheme

Key space	\mathcal{K}
Message space	\mathcal{M}
Cypher space	\mathcal{C}
Key generator	$\text{Gen} : \phi \rightarrow \mathcal{K}$
Encryption function	$\text{Enc} : \{\mathcal{K} \times \mathcal{M}\} \rightarrow \mathcal{C}$
Decryption function	$\text{Dec} : \{\mathcal{K} \times \mathcal{C}\} \rightarrow \mathcal{M}$

2 Information theoretic security

Information theoretic security repels even resource-unbounded attackers. Shannon secrecy and perfect secrecy are equivalent definitions of information theoretic security for symmetric cryptography schemes.

Shannon secrecy A scheme is Shannon-secret with respect to the distribution D over \mathcal{M} iff the ciphertext reveals no additional information about the message.

$$\forall M \in \mathcal{M}, C \in \mathcal{C} : \Pr_{\substack{k \leftarrow \text{Gen} \\ m \in D}} [m = M \mid \text{Enc}_k(m) = C] = \Pr_{m \in D} [m = M]$$

Perfect secrecy A scheme is perfectly secret iff the distributions of ciphertexts for any two messages are identical.

$$\forall M_1, M_2 \in \mathcal{M}, C \in \mathcal{C} : \Pr_{K_1 \leftarrow \text{Gen}} [\text{Enc}_{K_1}(M_1) = C] = \Pr_{K_2 \leftarrow \text{Gen}} [\text{Enc}_{K_2}(M_2) = C]$$

This model considers only a single message and ciphertext, so although a one-time pad is perfectly secret, a “two-time pad” is not.

Theorem 1. Perfect secrecy $\Rightarrow |\mathcal{K}| \geq |\mathcal{M}|$.

Proof. If not, \exists 2 messages with different probabilities of encrypting to the same ciphertext. \square

3 Pseudo-random functions

Uniformly random function U is a random variable chosen uniformly from the set of all functions $\{0, 1\}^m \rightarrow \{0, 1\}^n$.

Pseudo-random function A PRF belongs to a family of functions $F : \{0, 1\}^\ell \times \{0, 1\}^m \rightarrow \{0, 1\}^n$. Write $F_k(\cdot)$ to denote $F(k, \cdot)$.

Distinguishing advantage Consider an adversary \mathcal{A} who knows F , having oracle access to F_k where k was chosen uniformly at random, trying to distinguish the oracle's responses from a random function. The distinguishing advantage of \mathcal{A} against F is

$$\text{Adv}_F^{\text{prf}}(\mathcal{A}) \equiv \Pr_{k \in \{0,1\}^\ell} [\mathcal{A}^{F_k} \text{ accepts}] - \Pr_U [\mathcal{A}^U \text{ accepts}] .$$

In time $O(t)$, we can brute-force t keys to get advantage $t/2^\ell$.

(t, q) -bounded adversary $\begin{array}{l|l} t & \text{Running time} \\ q & \text{Number of queries} \end{array}$

(t, q, ε) -secure PRF F is (t, q, ε) -secure iff $\forall (t, q)$ -bounded \mathcal{A} ,

$$\text{Adv}_F^{\text{prf}}(\mathcal{A}) \leq \varepsilon .$$

Examples of reasonable constants $\begin{array}{l|l} t & 2^{128} \\ q & 2^{64} \text{ or } 2^{32} \\ \varepsilon & 2^{-128} \end{array}$

Existence The existence of secure PRFs has not been proven, but there are some functions that have never been broken and are widely assumed to be PRFs.

4 Reduction

Karp (many-to-one) reduction Reduction from A to B transforms an instance of A to an instance of B .

Cook (Turing) reduction Reduction from A to B solves A using a subroutine that solves B .

Key recovery security F is (t, q, ε) -kr-secure iff $\forall (t, q)$ -bounded \mathcal{A} ,

$$\text{Adv}_F^{\text{kr}} \equiv \Pr_{k \in \{0,1\}^\ell} [\mathcal{A}^{F_k(\cdot)} \text{ outputs } k] \leq \varepsilon .$$

Theorem 2. If F is a (t, q, ε) -secure PRF for $q < 2^m$, then F is (t', q', ε') -kr-secure for $t' \approx t$, $q' = q - 1$, $\varepsilon' = \varepsilon + 2^{-n}$.

Proof. Cook reduction. For any kr-adversary \mathcal{A}' running in time t' and making $q' < 2^m$ queries, let \mathcal{A} be the PRF adversary:

$k' \leftarrow \mathcal{A}'(\mathcal{O})$
 $x \leftarrow$ a value that \mathcal{A}' did not query with
 $y \leftarrow \mathcal{O}(x)$
 Accept iff $y = F_{k'}(x)$

\mathcal{A} runs in time $t \approx t'$ and makes $q = q' + 1$ queries.

$$\text{Adv}_F^{\text{prf}}(\mathcal{A}) \geq \text{Adv}_F^{\text{kr}}(\mathcal{A}') - 2^{-n} . \quad \square$$

Example PRF construction For PRF $F : \{0, 1\}^\ell \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, $F'_k(x) \equiv F_k(F_k(x)) \parallel F_k(\overline{F_k(x)})$

Theorem 3. F' is $(t, \frac{q}{3}, \varepsilon + \frac{q^2}{2^n})$ -secure.

Proof. Let \mathcal{A}' be an attacker on F' . Define \mathcal{A} as:

$\mathcal{O}' \equiv \mathcal{O}(\mathcal{O}(x)) \parallel \mathcal{O}(\overline{\mathcal{O}(x)})$ (done with 3 queries to \mathcal{O})
 Accept iff $\mathcal{A}'^{\mathcal{O}'}$ accepts

$\mathcal{O}'(x)$ simulates F' perfectly, so $\Pr_k [\mathcal{A}^{F_k}] = \Pr_k [\mathcal{A}'^{F'_k}]$.

\mathcal{O}' does not simulate U perfectly, but it is close. We have independence as long as all of the $\mathcal{O}(x)$, $\overline{\mathcal{O}(x)}$ are distinct. Using union bound, this probability $\leq \frac{q^2}{2^n}$ \square

5 Pseudo-random permutations

In a permutation family $F : \{0, 1\}^\ell \times \{0, 1\}^n \rightarrow \{0, 1\}^n$, every F_k is bijective.

A secure PRP is computationally indistinguishable from a uniformly random permutation.

Some well-known PRPs	DES	$\ell = 56$	$n = 64$
	AES ₁₂₈	$\ell = 128$	$n = 128$
	AES ₁₉₂	$\ell = 192$	$n = 128$

Strong PRP / block cipher Attackers with oracle access to both F and F^{-1} have small advantage.

$$\text{Adv}_F^{\text{sprp}} \equiv \Pr_k [\mathcal{A}^{F_k, F_k^{-1}} \text{ accepts}] - \Pr_P [\mathcal{A}^{P, P^{-1}} \text{ accepts}] \leq \varepsilon$$

PRF/PRP switching lemma If G is a (t, q, ε) -secure PRP (not necessarily strong), then F is a $(t, q, \varepsilon + \frac{q^2}{2^{n+1}})$ -secure PRF.

6 Secure symmetric encryption

Perfect secrecy is impossible where $m > \ell$, but computational security is possible with pseudorandom objects.

Electronic code block (ECB) Suppose F is a secure PRP $\{0, 1\}^\ell \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ with F and F^{-1} efficiently computable.

Gen	$k \leftarrow \{0, 1\}^\ell$
Enc	$M' \leftarrow \text{Pad message } M \text{ with 1 and some 0s to a multiple of } n.$ Break M' into n -bit blocks m_0, m_1, \dots Apply F_k to each of the $\{m\}$
Dec	Apply F'_k to each of the $\{m\}$

Repeated blocks give repeated ciphertext. Never use ECB.

Security model Adversary, seeing all ciphertexts and having oracle access to Enc_k , learns nothing about plaintexts (except message length, which is unavoidable).

$SE = (\text{Gen}, \text{Enc}, \text{Dec})$ is (t, σ, ε) -IND-CPA secure (“indistinguishable under chosen-plaintext attack”) iff $\forall (t, \sigma)$ -bounded \mathcal{A} ,

$$\text{Adv}_{SE}^{\text{indcpa}}(\mathcal{A}) \equiv \Pr_{k \leftarrow \text{Gen}} [\mathcal{A}^{L_k} \text{ accepts}] - \Pr_{k \leftarrow \text{Gen}} [\mathcal{A}^{R_k} \text{ accepts}] ,$$

$$L_k(m, m') \equiv \text{Enc}_k(m) \text{ if } |m| = |m'| \text{ else } \perp ,$$

$$R_k(m, m') \equiv \text{Enc}_k(m') \text{ if } |m| = |m'| \text{ else } \perp ,$$

t is the running time, and σ total length of all message queries.

Equivalent definition: Enc_k is computationally indistinguishable from a zero-encrypting oracle $Z_k \equiv \text{Enc}_k(0^m)$.

Query repetition Enc in an IND-CPA-secure scheme should not always return the same ciphertext for multiple encryptions of the same message. This attack has advantage 1 against any deterministic and stateless scheme:

$$\begin{array}{|l} c \leftarrow \mathcal{O}(\langle 0 \rangle, \langle 0 \rangle) \\ c' \leftarrow \mathcal{O}(\langle 0 \rangle, \langle 1 \rangle) \\ \text{Accept iff } c = c' \end{array}$$

7 Block cipher modes

Stateful counter mode (CTRS) Let F be a PRF with $m = n$.

$$\begin{array}{|l} \text{Gen} \quad k \leftarrow \{0, 1\}^\ell, \text{ counter} \leftarrow 0 \\ \text{Enc} \quad \text{echo counter} \\ \quad \text{for each message block } m: \\ \quad \quad \text{echo } F_k(\text{counter}) \oplus m_i \\ \quad \quad \text{increment counter} \end{array}$$

CTRS is not used much, because preserving *counter* is difficult.

Theorem 4. If F is a (t, q, ε) -secure PRF, then $\text{CTRS}(F)$ is $(t' \approx t, qn, 2\varepsilon)$ -IND-CPA secure.

Proof. We will show using a hybrid argument that $\forall (t', \sigma)$ -bounded \mathcal{A}' against $\text{CTRS}(F)$ where $\sigma \leq n 2^m$, there is a $(t \approx t', q = \sigma/n)$ -bounded attacker \mathcal{A} attacking F such that $\text{Adv}_{\text{CTRS}(F)}^{\text{indcpa}}(\mathcal{A}') \leq 2 \text{Adv}_F^{\text{prf}}(\mathcal{A})$.

Given \mathcal{O} that is either F or U :

$$\mathcal{A} \equiv \mathcal{A}_L \equiv \left| \begin{array}{l} \text{counter} \leftarrow 0 \\ \mathcal{O}'(m, m') \equiv \left| \begin{array}{l} \text{If } |m| = |m'|, \text{ return } \perp \\ \text{Split } m \text{ into blocks } m_0, m_1, \dots, m_{t-1} \\ y_i \leftarrow \mathcal{O}(\text{counter} + i) \ \forall i \in [0, t) \\ \text{Return } \text{counter} \parallel \text{join}_i(m_i \oplus y_i) \\ \text{counter} \leftarrow \text{counter} + t \end{array} \right. \\ \text{Accept iff } \mathcal{A}'^{\mathcal{O}'} \text{ accepts} \end{array} \right.$$

Also define \mathcal{A}_R similarly using m' instead of m .

$\mathcal{A}_L^{F_k}$ perfectly simulates L_k to \mathcal{A}' .

\mathcal{A}_L^U does not simulate R_k , but it does simulate an oracle $\$$:

$$\$(m, m') \equiv \left| \begin{array}{l} \text{If } |m| = |m'|, \text{ return } \perp \\ \text{Return } \text{counter} \parallel [\text{random bits}] \\ \text{counter} \leftarrow \text{counter} + \text{number of blocks} \end{array} \right.$$

$$P_\ell = \Pr_k[\mathcal{A}_L^{F_k}] = \Pr_k[\mathcal{A}'^{L_k}]$$

$$P_r = \Pr_k[\mathcal{A}_R^{F_k}] = \Pr_k[\mathcal{A}'^{R_k}]$$

$$P_\$ = \Pr_k[\mathcal{A}_L^U] = \Pr_k[\mathcal{A}_R^U] = \Pr_k[\mathcal{A}'^\$]$$

$$\begin{aligned} \text{Adv}_{\text{CTRS}(F)}^{\text{indcpa}}(\mathcal{A}') &= |P_\ell - P_r| \\ &\leq |(P_\ell - P_\$) + (P_r - P_\$)| \quad (\text{triangle inequality}) \\ &\leq \varepsilon + \varepsilon = 2\varepsilon \end{aligned}$$

□

Counter modes

CTRS	One global counter	$\text{adv}^{\text{indcpa}} \leq 2 \text{adv}^{\text{prf}}$
CTR\$	Random IV for each message	$\text{adv}^{\text{indcpa}} \leq 2 \text{adv}^{\text{prf}} + q^2/2^n$
CTR\$\$	Random IV for each block	

Cipher block chaining (CBC) $C_0 = \text{IV}$, $C_i = F_k(C_{i-1} \oplus m_i)$

Dec requires being able to calculate F^{-1} .

If F is a (t, q, ε) -secure PRF, then $\text{CBC}[F]$ is $(\approx t, \sigma = qn, 2\varepsilon + q^2/2^n)$ -ind-cpa-secure. The proof requires showing that for U , all inputs to U are distinct (minus a birthday term).

8 Message authentication code (MAC)

Alice sends message m and $t \leftarrow \text{Tag}_k(m)$. Eve intercepts (m, t) and delivers (m', t') to Bob. Bob runs $\text{Ver}_k(m', t')$.

Ver_k returns $\begin{cases} m & \text{if } t' \text{ is a valid tag (Ver}_k \text{ “accepts”)} \\ \perp & \text{otherwise (Ver}_k \text{ “rejects”)} \end{cases}$

Eve has access to a Tag_k oracle and can make many attempts on Ver . Eve “wins” if Ver accepts on an m' not previously queried to Tag_k .

Concerns ignored by this model

Dropped messages

Replay attacks (“freshness” of messages)

Message sequence

Unforgeability under chosen message attack

$$\text{Adv}_{\text{MAC}}^{\text{ufmca}}(\mathcal{A}) \equiv \Pr_{k \leftarrow \text{Gen}} \left[\mathcal{A}^{\text{Tag}_k, \text{Ver}_k} \text{ “wins”} \right]$$

MAC is $(t, q_t, q_v, \varepsilon)$ -uf-cma-secure iff advantage of an attacker bounded by time t , number of Tag queries q_t , and number of Ver queries q_v is less than ε .

Examples of reasonable constants	t	2^{80} or 2^{128}
	q_v, q_t	2^{40} or 2^{56}
	ε	2^{-40} or 2^{-56}

Brute-force MAC attacks

Key search: Get a few oracle tags, and guess k . $\text{Adv} = t/2^\ell$.

Tag search: $\text{Adv} = t/2^s$ where s is the tag length.

PRF-based MAC $\text{Tag}_k \equiv F_k$

$\forall (t, q_t, q_v)$ -bounded \mathcal{B} , $\exists (\approx t, q_t + q_v)$ -bounded \mathcal{A} such that

$$\text{Adv}_{\text{PRF MAC}[F]}^{\text{ufcma}}(\mathcal{B}) \leq \text{Adv}_F^{\text{prf}}(\mathcal{A}) + q_v/2^n .$$

CBC-MAC For a fixed t , and $F : \{0, 1\}^{nt} \rightarrow \{0, 1\}^n$, $\text{CBC-MAC}[F]$ is secure, losing $(qt)^2/2^n$ advantage from that of F .

Cipher-based MAC (CMAC) Adds an extra step to the end of CBC-MAC to make it secure for arbitrary-length messages.

Precompute $k_1, k_2 \in \{0, 1\}^n$ using $F_k(0^m)$.

$$m'_t \leftarrow \begin{cases} m'_t \oplus k_1 & : |m'_t| = n \\ m' \| 000 \dots \oplus k_2 & : |m'_t| < n \end{cases}$$

Run $m_1 \| \dots \| m_t$ through CBC-MAC.

9 Combining authenticity and privacy

Integrity of ciphertexts (INT-CTXT) $\text{Dec}_k(c)$: returns decryption of c , or \perp if c is invalid.

$SE = (\text{Gen}, \text{Enc}, \text{Dec})$ is INT-CTXT secure iff \forall bounded \mathcal{A} ,

$$\text{Adv}_{SE}^{\text{int-ctxt}}(\mathcal{A}) \equiv \Pr_{k \leftarrow \text{Gen}} \left[\mathcal{A}^{\text{Enc}_k, \text{Dec}_k} \text{ wins} \right] < \varepsilon .$$

UF-CMA-security does not necessarily give INT-CTXT security. For example: If the output of **Tag** has a spurious bit that is ignored by **Ver**. So we require a stronger condition:

Strong unforgeability (SUF-CMA) Winning is redefined as: Ver_k accepts (m', t') that was not previously a query/answer pair to **Tag** $_k$.

Bad idea: Encrypt-and-tag $\text{AEnc} \equiv \text{EEnc}_{k_e}(m) \| \text{Tag}_{k_m}(m)$. The tag could reveal information about m .

Bad idea: Tag-then-encrypt $\text{AEnc} \equiv \text{EEnc}_{k_e}(m \| \text{Tag}_{k_m}(m))$. The ciphertext might be forgeable (for example, if EEnc appends a spurious bit).

Good idea: Encrypt-then-tag $\text{AEnc} \equiv \text{EEnc}_{k_e}(m) \| \text{Tag}_{k_m}(\text{EEnc}_{k_e}(m))$.

Indistinguishability under chosen ciphertext attack (IND-CCA)
 $\text{IND-CPA} \wedge \text{INT-CTXT} \Rightarrow \text{IND-CCA}$.

10 Hashing

Hash function $h : D \rightarrow \{0, 1\}^n, D > 2^n$

Collision $x, x' \in D : h(x) = h(x') \wedge x \neq x'$

Hash family $H : \{0, 1\}^\ell \times D \rightarrow \{0, 1\}^n$

Collision resistance (CR) H is (t, ε) -collision resistant if $\text{Adv}_H^{\text{cr}}(\mathcal{A}) \leq \varepsilon$
 \forall t -bounded \mathcal{A} .

$$\text{Adv}_H^{\text{cr}}(\mathcal{A}) = \Pr_{k \leftarrow \{0,1\}^\ell} [\mathcal{A}(k) \text{ outputs a collision in } H_k]$$

	MD4	$n = 128$	Broken
	MD5	$n = 128$	Broken
Real-world hash functions	SHA-1	$n = 160$	Maybe broken
	SHA-256	$n = 256$	Good
	SHA-3		Good

Hash output lengths need to be longer than encryption key lengths because brute-force attacks can test $\approx q^2$ pairs in q hashes (“birthday attack”).

Second preimage resistance / target collision resistance (TCR)

Given x , attacker must find x' such that x, x' is a collision.

$$\text{Adv}_H^{\text{tcr}}(\mathcal{A}) = \Pr_{\substack{k \leftarrow \{0,1\}^\ell, \\ x \leftarrow D}} [\mathcal{A}(k, x) \text{ outputs a collision}]$$

where D is some distribution over the message space.

Brute force attack does not have a birthday advantage in this game.

CR \Rightarrow TCR.

One-wayness (OW)

$$\text{Adv}_H^{\text{ow}}(\mathcal{A}) = \Pr_{\substack{k \leftarrow \{0,1\}^\ell, \\ x \leftarrow D}} [\mathcal{A}(k, H_k(x)) \text{ outputs } x' : H_k(x') = H_k(x)]$$

TCR \Rightarrow OW for “high-entropy” D (so that $H_k(x)$ reveals very little about x).

Merkle-Damgård (MD) transform $\text{MD}[h]_k(M \in \{0, 1\}^*)$

Uses a compression function $h_k : \{0, 1\}^\ell \times \{0, 1\}^{b+n} \rightarrow \{0, 1\}^n$

Break M into M_1, \dots, M_t s.t. $\|M_i\| = b$

$$\begin{aligned}
y_1 &\leftarrow h_k(M_1 \parallel \langle 0 \rangle) \\
y_2 &\leftarrow h_k(M_2 \parallel y_1) \\
&\vdots \\
y_i &\leftarrow h_k(M_i \parallel y_{i-1}) \\
&\vdots \\
y_t &\leftarrow h_k(M_t \parallel y_{t-1}) \\
y &\leftarrow h_k(\langle t \rangle \parallel y_t)
\end{aligned}$$

If h is CR, then MD[h] is CR Let \mathcal{B} attack MD[h] in CR game. We can build \mathcal{A} attacking h in CR game. $\mathcal{A}(k \in \{0, 1\}^\ell)$ will use a collision x, x' in MD[h_k] to find a collision in h_k .

If x, x' have different numbers of blocks: Then $\langle t \rangle \parallel M_t, \langle t' \rangle \parallel M'_t$ is a collision in h_k .

Otherwise: Walk backward through the MD process to find a step where $M_i \parallel y_{i-1} \neq M'_i \parallel y'_{i-1}$.

HMAC Secure MAC based on hash function.

$\text{HMAC}_k(m) = H((k \oplus \text{opad}) \parallel H((k \oplus \text{ipad}) \parallel m))$ where k is the MAC key padded to the length of the compression function, ipad and opad are fixed strings of the same length.

Only TCR security of H is required for the security of HMAC.