Previously on COS 433...
Perfect Security for Multiple Messages

Definition: A stateless scheme \((\text{Enc,Dec})\) has perfect secrecy for \(n\) messages if, for any two sequences of messages \((m_0^{(i)})_{i \in [d]}, (m_1^{(i)})_{i \in [d]} \in M^d\)

\[
(\text{Enc}(K, m_0^{(i)}))_{i \in [d]} \overset{d}{=} (\text{Enc}(K, m_1^{(i)}))_{i \in [d]}
\]

Notation: \(( f(i) )_{i \in [d]} = ( f(1), f(2), \ldots, f(n) )\)
Randomized Encryption

Syntax:
• Key space \( K \) (usually \( \{0,1\}^\lambda \))
• Message space \( M \) (usually \( \{0,1\}^n \))
• Ciphertext space \( C \) (usually \( \{0,1\}^m \))
• \( \text{Enc}: K \times M \rightarrow C \) (potentially probabilistic)
• \( \text{Dec}: K \times C \rightarrow M \) (usually deterministic)

Correctness:
• For all \( k \in K, m \in M, \)
  \[ \Pr[ \text{Dec}(k, \text{Enc}(k,m)) = m ] = 1 \]
Theorem: No stateless *randomized* encryption scheme can have perfect security for multiple messages
Today: Relaxing Perfect Secrecy
What do we do now?

Tolerate tiny probability of distinguishing

• If \( \Pr[c^{(0)} = c^{(1)}] = 2^{-128} \), in reality never going to happen
How Small Is Ok?

Practice:

• Something unlikely to happen in lifetime of data/person/civilization/universe

• Typically something like $2^{-80}$, $2^{-128}$, or maybe $2^{-258}$
  • Being struck by lightning twice: $2^{-23}$
  • Winning the lottery: $2^{-26}$
  • World-ending asteroid while on this slide: $2^{-46}$
How Small Is Ok?

Theory:
• Maybe things will change as technology improves
• Want a more conceptual answer
• Absolute constants unsatisfactory
• Instead, use ``negligible” functions
Negligible functions

Def: A function $f$ is polynomial if $f(n) = O(n^c)$ for some constant $c$.

Def: A function $g$ is super-polynomial if, for every polynomial $f$, $f(n) = O(g(n))$.

Def: A function $p$ is inverse polynomial if $1/p(n)$ is polynomial.

Def: A function $\varepsilon$ is negligible if, for every inverse polynomial $p$, $\varepsilon(n) = O(p(n))$ (equivalently, $1/\varepsilon$ is super-polynomial).
Examples

\[ 2^n \quad \text{super-polynomial} \]
\[ n^{-n/7} \quad \text{negligible} \]
\[ 3^{-5 \log n} \quad \text{inverse polynomial} \]
\[ 1.5^{-\frac{3}{n}} \quad \text{negligible} \]
\[ 8^{\log^3 n} \quad \text{super-polynomial} \]
\[ (\log n)/n \quad \text{inverse polynomial} \]
Security Parameter $\lambda$

Additional input to system, dictates “security level”

Key, message, ciphertext size all *polynomial* in $\lambda$

Probability of adversary success is *negligible* in $\lambda$
Defining Encryption Again

Syntax:
- Key space $K_\lambda$
- Message space $M_\lambda$
- Ciphertext space $C_\lambda$
- $\textbf{Enc}$: $K_\lambda \times M_\lambda \rightarrow C_\lambda$ (potentially randomized)
- $\textbf{Dec}$: $K_\lambda \times C_\lambda \rightarrow M_\lambda$

Correctness:
- $\log|K_\lambda|, \log|M_\lambda|, \log|C_\lambda|$ polynomial in $\lambda$
- For all $\lambda, k \in K_\lambda, m \in M_\lambda$,
  \[ \Pr\left[ \Pr[\text{Dec}(k, \text{Enc}(k,m)) = m] = 1 \right] = 1 \]
Statistical Distance

Given two distributions \( D_1, D_2 \) over a set \( X \), define

\[
\Delta(D_1,D_2) = \frac{1}{2} \sum_x | \Pr[D_1=x] - \Pr[D_2=x] |
\]

Observations:

\[
0 \leq \Delta(D_1,D_2) \leq 1
\]

\[
\Delta(D_1,D_2) = 0 \iff D_1 \overset{d}{=} D_2
\]

\[
\Delta(D_1,D_2) \leq \Delta(D_1,D_3) + \Delta(D_3,D_2)
\]

(\( \Delta \) is a metric)
Another View of Statistical Distance

Theorem: \( \Delta(D_1, D_2) \geq \varepsilon \) iff \( \exists (\text{potentially randomized}) \ A \) s.t.
\[
\left| \Pr[A(D_1) = 1] - \Pr[A(D_2) = 1] \right| \geq \varepsilon
\]

Terminology: for any \( A \),
\[
\left| \Pr[A(D_1) = 1] - \Pr[A(D_2) = 1] \right|
\]
is called the “advantage” of \( A \) in distinguishing \( D_1 \) and \( D_2 \)
Another View of Statistical Distance

Theorem: $\Delta(D_1, D_2) \geq \epsilon$ iff $\exists$ (potentially randomized) $A$ s.t.

$$\left| \Pr[A(D_1) = 1] - \Pr[A(D_2) = 1] \right| \geq \epsilon$$

To lower bound $\Delta$, just need to show adversary $A$ with that advantage
Examples

\( D_1 = \) Uniform distribution over \{0,1\}^n
\( \cdot \Pr[D_1=x] = 2^{-n} \)

\( D_2 = \) Uniform subject to even parity
\( \cdot \Pr[D_2=x] = 2^{-(n-1)} \) if \( x \) has even parity, 0 otherwise

\[ \Delta(D_1,D_2) = \frac{1}{2} \sum_{\text{even} x} |2^{-n} - 2^{-(n-1)}| + \frac{1}{2} \sum_{\text{odd} x} |2^{-n} - 0| \]

\[ = \frac{1}{2} \sum_{\text{even} x} 2^{-n} + \frac{1}{2} \sum_{\text{odd} x} 2^{-n} \]

\[ = \frac{1}{2} \]
Examples

$D_1 = $ Uniform over $\{1,\ldots,n\}$

$D_2 = $ Uniform over $\{1,\ldots,n+1\}$

$\Delta(D_1,D_2) = \frac{1}{2} \sum_{x=1}^{n} |1/n - 1/(n+1)|$

$\quad + \frac{1}{2} |0 - 1/(n+1)|$

$= \frac{1}{2} \sum_{x=1}^{n} 1/n(n+1) + \frac{1}{2} 1/(n+1)$

$= \frac{1}{2} 1/(n+1) + \frac{1}{2} 1/(n+1) = 1/(n+1)$
Definition: A scheme \((\text{Enc}, \text{Dec})\) has \(\varepsilon\)-statistical secrecy for \(d\) messages if \(\forall\) two sequences of messages \((m_0^{(i)})_{i \in [d]}, (m_1^{(i)})_{i \in [d]} \in M^d\)

\[
\Delta\left[ (\text{Enc}(K, m_0^{(i)}))_{i \in [d]}, (\text{Enc}(K, m_1^{(i)}))_{i \in [d]} \right] < \varepsilon
\]

We will call such a scheme \((d, \varepsilon)\) statistically secure.
Statistical Security (Asymptotic)

**Definition:** A scheme \((\text{Enc}, \text{Dec})\) has statistical secrecy for \(d\) messages if \(\exists\) negligible \(\varepsilon\) such that \(\forall\) two sequences \((m_0^{(i)})_{i \in [d]}, (m_1^{(i)})_{i \in [d]} \in M_\lambda^d\),

\[
\Delta \left[ \left( \text{Enc}(K_\lambda, m_0^{(i)}) \right)_{i \in [d]}, \left( \text{Enc}(K_\lambda, m_1^{(i)}) \right)_{i \in [d]} \right] < \varepsilon(\lambda)
\]

We will call such a scheme \(d\)-time statistically secure.
Stateless Encryption with Multiple Messages

Ex:

\[ M = C = \mathbb{Z}_p \quad (p \text{ a prime of size } 2^\lambda, \lambda=128) \]
\[ K = \mathbb{Z}_p^* \times \mathbb{Z}_p \]
\[ \text{Enc}( (a,b), m) = (am + b) \mod p \]
\[ \text{Dec}( (a,b), c) = (c-b)/a \mod p \]

Q: Is this statistically secure for two messages?
Example

Ex:

\[ M = \mathbb{Z}_p \quad (p \text{ a prime of size } 2^\lambda, \lambda=128) \]
\[ C = \mathbb{Z}_p^2 \]
\[ K = \mathbb{Z}_p^2 \]
\[ \text{Enc}( (a,b), m) = (r, (ar+b) + m) \]
\[ \text{Dec}( (a,b), (r,c) ) = c - (ar+b) \]

Q: Is this statistically secure for two messages?
Proof of Example

Let $D_b$ be distribution of $(\text{Enc}(k,m_b(i)))_{i \in \{1,2\}}$

Let $D_b'$ be $D_b$, but conditioned on $r_0 \neq r_1$

Fix $r_0 \neq r_1, m_0, m_1, c_0, c_1$

$\Pr[ar_0 + b + m_0 = c_0, ar_1 + b + m_1 = c_1] = 1/p^2$

$(a,b)$

So $D_0' \overset{d}{=} D_1'$ ( $\Delta(D_0', D_1') = 0$ )
Lemma: $\Delta(D_1, D_2) \leq \frac{1}{2}\Pr[\text{bad}|D_1] + \frac{1}{2}\Pr[\text{bad}|D_2] + \Delta(D_1', D_2')$

Where:

- "bad" is some event
- $\Pr[\text{bad}|D_b]$ is probability "bad" when sampling from $D_b$
- $D_b'$ is $D_b$, but conditioned on not "bad"
Proof of Lemma

\[ \Delta(D_1, D_2) = \frac{1}{2} \sum_x | \Pr[D_1 = x] - \Pr[D_2 = x] | \]

\[ = \frac{1}{2} \sum_{x: \text{bad}} | \Pr[D_1 = x] - \Pr[D_2 = x] | + \frac{1}{2} \sum_{x: \text{good}} | \Pr[D_1 = x] - \Pr[D_2 = x] | \]

\[ \leq \frac{1}{2} \sum_{x: \text{bad}} | \Pr[D_1 = x] | + \frac{1}{2} \sum_{x: \text{bad}} | \Pr[D_2 = x] | + \frac{1}{2} \sum_{x: \text{good}} | \Pr[D_1 = x] - \Pr[D_2 = x] | \]

\[ \leq \frac{1}{2} \Pr[\text{bad} | D_1] + \frac{1}{2} \Pr[\text{bad} | D_2] + \Delta(D_1', D_2') \]
Proof of Example

Let $D_b$ be distribution of $(Enc(k, m_{b(i)})_{i \in \{1, 2\}}$

Let $bad$ be when $r_0 = r_1$

Let $D'_b$ be $D_b$, but conditioned on not $bad$

$Pr[bad|D_b] = 1/p$

$\Delta(D'_0, D'_1) = 0$

Therefore, $\Delta(D_0, D_1) \leq 1/p \approx 2^{-\lambda}$
Summary so Far

Stateless encryption for multiple messages ✓

But, key length grows with number of messages ✗

And, key length grows with length of message ✗
Limits of Statistical Security

**Theorem:** Suppose \((\text{Enc, Dec})\) has plaintext space \(M = \{0, 1\}^n\) and key space \(K = \{0, 1\}^t\). Moreover, assume it is \((d, 0.4999)\)-secure. Then:

\[ t \geq d \cdot n \]

In other words, the key must be at least as long as the total length of all messages encrypted.
Proof Idea: Compression

Use an encryption protocol to build a compression protocol

\[ m \xrightarrow{\text{Comp}(m)} m' \xleftarrow{\text{Decomp}(m')} m \]

Goal: \(|m'| < |m|\)
For Now: Easier Goal

\[ m' \leftarrow \text{Comp}(s, m) \]

\[ s \leftarrow \text{Setup}() \]

\[ m \leftarrow \text{Decomp}(s, m') \]

Goal: \( |m'| < |m| \)
The Protocol

Let $m_0$ be some message in $M$

**Setup():**
- Choose random $k_0 \leftarrow K$
- Let $c_1 \leftarrow \text{Enc}(k_0, m_0)$, ..., $c_d \leftarrow \text{Enc}(k_0, m_0)$
- Output $(c_1, ..., c_d)$

**Comp( (c_1, ..., c_d), (m_1, ..., m_d) ):**
- Find $k, r_1, ..., r_d$ such that $c_i = \text{Enc}(k, m_i; r_i) \quad \forall i$
- If no such values exist, abort
- Output $k$
The Protocol

Let \( m_0 \) be some message in \( M \)

\[
\text{Comp}( (c_1, \ldots, c_d), (m_1, \ldots, m_d) ):
\]
• Find \( k, r_1, \ldots, r_d \) such that \( c_i = \text{Enc}(k, m_i; r_i) \) \( \forall i \)
• If no such values exist, abort
• Output \( k \)

\[
\text{Decomp}( (c_1, \ldots, c_d), k ):
\]
• Compute \( m_i = \text{Dec}(k, c_i) \)
• Output \( (m_1, \ldots, m_d) \)
Analysis of Protocol

If \textbf{Comp} succeeds, \textbf{Decomp} must succeed by correctness

• Since \(c_i = \text{Enc}(k, m_i; r_i)\), \(\text{Dec}(k, c_i)\) must give \(m_i\)

Therefore, must figure out when \textbf{Comp} succeeds

\textbf{Claim}: For any sequence of messages \(m_1, \ldots, m_d\), \textbf{Comp} succeeds with probability at least \(1-\varepsilon\)

(Probability over the randomness used by \textbf{Setup}())
Claim: For any sequence of messages $m_1, \ldots, m_d$, Comp succeeds with probability at least $1-\varepsilon$

Proof:
• Suppose Comp succeeds with probability $1-p$ for messages $m_1, \ldots, m_d$
• Let $A(c_1, \ldots, c_d)$ be the algorithm that runs Comp($((c_1, \ldots, c_d), (m_1, \ldots, m_d))$) and outputs 1 if Comp succeeds

• If $c_i = \text{Enc}(k_0, m_i)$, then $\Pr[A(c_1, \ldots, c_d) = 1] = 1$
• If $c_i = \text{Enc}(k_0, m_0)$, then $\Pr[A(c_1, \ldots, c_d) = 1] = 1-p$
• By $(d, \varepsilon)$-statistical security of Enc, $p$ must be $\leq \varepsilon$
Claim: For any sequence of messages $m_1,...,m_d$, Comp succeeds with probability at least $1-\epsilon$.

Claim: For a random sequence of messages $m_1,...,m_d$, Comp succeeds with probability at least $1-\epsilon$.

(Probability over the randomness used by Setup() and the random choices of $m_1,...,m_d$.)
Next step: Removing Setup

We know:

$$\Pr\left[ \text{Comp succeeds: } (c_1, \ldots, c_d) \leftarrow \text{Setup}(), \quad m_i \leftarrow M \right] \geq 1 - \varepsilon$$

Therefore, there must exist some $$(c_1^*, \ldots, c_d^*)$$ such that

$$\Pr[\text{Comp succeeds: } m_i \leftarrow M] \geq 1 - \varepsilon$$

Define: $$M' = \{(m_1, \ldots, m_d): \text{Comp succeeds}\}$$

• Note that $$|M'| \geq (1 - \varepsilon) |M|^d$$
The Protocol

Find $k, r_1, ..., r_d$ such that $c_i^* = \text{Enc}(k, m_i; r_i) \quad \forall i$

For each $i$, let $m_i \leftarrow \text{Dec}(k, c_i^*)$
Output $(m_1, ..., m_d)$

By previous analysis,
- Alice always successfully compresses
- Bob always successfully decompresses
Final Touches

Can compress messages in $M'$ into keys in $K$

Therefore, it must be that $|M'| \leq |K|$

Meaning $t = \log |K|$

\[ \geq \log |M'| \]
\[ \geq \log \left[ (1-\varepsilon) |M|^d \right] \]
\[ = d \log |M| + \log [1-\varepsilon] \]
\[ = dn + \log [1-\varepsilon] \]
\[ \geq dn \ (\text{as long as } \varepsilon<1/2) \]
Takeaway

If you don’t want to physically exchange keys frequently, you cannot obtain statistical security

So, now what?
Running Time

Timeline/Cipher sophistication

Cryptanalysis

Encrypt/decrypt
Computational Security

We are ok if adversary takes a really long time

Only considered attack for adversaries that don’t take too long
How Long Is Ok?

Practice:

• Lifetime of data/person/civilization/universe

• Typically something like $2^{80}$, $2^{128}$, or maybe $2^{258}$
  • Lifetime of universe in nanoseconds: $2^{58}$
  • Number of atoms in known universe: $2^{265}$
How Long Is Ok?

Theory:
• Maybe things will change as technology improves

• Want a more conceptual answer

• Absolute constants unsatisfactory

• Instead, consider an attack if time bounded by polynomial function
Brute Force Attacks

Simply try every key until find right one

If keys have length $\lambda$, $2^\lambda$ is upper bound on attack

Not always applicable. When?
Holiwudd Criptoe!

[TRANSLTR]’s three million processors would all work in parallel ... trying every new permutation as they went
Holiwudd Criptoe!

“What’s the longest you’ve ever seen TRANSLTR take to break a code?”

“About an hour, but it had a ridiculously long key—ten thousand bits”
Reminders

HW1 Due Feb 20th

Project 1 to be released hopefully this afternoon