Defining Encryption (ctd.)

Lecture 3
SIM & IND security
Beyond One-Time: **CPA** security
Computational Indistinguishability
Onetime Encryption

Perfect Secrecy

**Perfect secrecy:** \( \forall m, m' \in \mathcal{M} \)

\[
\{\text{Enc}(m,K)\}_{K \leftarrow \text{KeyGen}} = \{\text{Enc}(m',K)\}_{K \leftarrow \text{KeyGen}}
\]

Distribution of the ciphertext is defined by the randomness in the key

In addition, require **correctness**

\( \forall m, K, \ Dec(\text{Enc}(m,K), K) = m \)

E.g. One-time pad: \( \mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^n \) and

\[
\text{Enc}(m,K) = m \oplus K, \ Dec(c,K) = c \oplus K
\]

More generally \( \mathcal{M} = \mathcal{K} = \mathcal{C} = \mathcal{G} \) (a finite group) and

\[
\text{Enc}(m,K) = m + K, \ Dec(c,K) = c - K
\]

A (2,2)-secret-sharing scheme: \( K \) and \( \text{Enc}(m,K) \) are shares of \( m \)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>x</td>
<td>y</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>b</td>
<td>y</td>
<td>x</td>
<td>z</td>
<td>y</td>
</tr>
</tbody>
</table>

Assuming \( K \) uniformly drawn from \( \mathcal{K} \)

- \( \Pr[\text{Enc}(a,K)=x] = \frac{1}{4} \)
- \( \Pr[\text{Enc}(a,K)=y] = \frac{1}{2} \)
- \( \Pr[\text{Enc}(a,K)=z] = \frac{1}{4} \)

Same for \( \text{Enc}(b,K) \).
Onetime Encryption

IND-Onetime Security

- IND-Onetime Experiment
  - Experiment picks a random bit $b$. It also runs KeyGen to get a key $K$.
  - Adversary sends two messages $m_0, m_1$ to the experiment.
  - Experiment replies with $\text{Enc}(m_b, K)$.
  - Adversary returns a guess $b'$.
  - Experiments outputs 1 iff $b' = b$.

IND-Onetime secure if for every adversary, $\Pr[b' = b] = 1/2$.
SIM-Onetime Encryption

SIM-Onetime Security

Class of environments which send only one message.

IDEAL = REAL

Equivalent to perfect secrecy + correctness

Recall

SIM-Onetime secure if:

∀ ∃ s.t.

IDEAL = REAL
Security of Encryption

- Perfect secrecy is too strong for multiple messages (though, as we shall see later, too weak in some other respects)
  - Requires keys as long as the messages
  - Relax the requirement by restricting to \textit{computationally bounded adversaries} (and environments)
  - Coming up: Formalizing notions of “computational” security (as opposed to perfect/statistical security)
  - Then, security definitions used for encryption of multiple messages
Symmetric-Key Encryption

The Syntax

- **Shared-key (Private-key) Encryption**
- **Key Generation**: Randomized
  
  \[ K \leftarrow \mathcal{K}, \text{ uniformly randomly drawn from the key-space} \]
  
  (or according to a key-distribution)

- **Encryption**: Randomized

  \[ \text{Enc: } \mathcal{M} \times \mathcal{K} \times \mathcal{R} \rightarrow \mathcal{C}. \text{ During encryption a fresh random string will be chosen uniformly at random from } \mathcal{R} \]

- **Decryption**: Deterministic

  \[ \text{Dec: } \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M} \]
Symmetric-Key Encryption

Security Definitions

<table>
<thead>
<tr>
<th>Security of Encryption</th>
<th>Information theoretic</th>
<th>Game-based</th>
<th>Simulation-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-time</td>
<td>Perfect secrecy &amp; Perfect correctness</td>
<td>IND-Onetime &amp; Perfect correctness</td>
<td>SIM-Onetime</td>
</tr>
<tr>
<td>Multi-msg</td>
<td></td>
<td>IND-CPA &amp; correctness</td>
<td>SIM-CPA</td>
</tr>
<tr>
<td>Active/multi-msg</td>
<td></td>
<td>IND-CCA &amp; correctness</td>
<td>SIM-CCA</td>
</tr>
</tbody>
</table>

**CPA: Chosen Plaintext Attack**
- The adversary can influence/choose the messages being encrypted
- Note: One-time security also allowed this, but for only one message
Symmetric-Key Encryption

SIM-CPA Security

Same as SIM-onetime security, but not restricted to environments which send only one message. Also, now all entities “efficient.”
Symmetric-Key Encryption

IND-CPA Security

- Experiment picks a random bit $b$. It also runs KeyGen to get a key $K$.
- For as long as Adversary wants:
  - Adv sends two messages $m_0, m_1$ to the experiment.
  - Expt returns $Enc(m_b, K)$ to the adversary.
- Adversary returns a guess $b'$.
- Experiment outputs 1 iff $b' = b$.
- **IND-CPA secure** if for all “efficient” adversaries $Pr[b' = b] \approx 1/2$.
Almost Perfect

For multi-message schemes we relaxed the “perfect” simulation requirement to IDEAL $\approx$ REAL

In particular, we settle for “almost perfect” correctness

Recall perfect correctness

$\forall m, \Pr_{K \leftarrow \text{KeyGen}, Enc} [ \text{Dec}(\text{Enc}(m,K), K) = m ] = 1$

Almost perfect correctness: a.k.a. Statistical correctness

$\forall m, \Pr_{K \leftarrow \text{KeyGen}, Enc} [ \text{Dec}(\text{Enc}(m,K), K) = m ] \approx 1$

But what is $\approx$ ?
Feasible Computation

In analyzing complexity of algorithms: Rate at which computational complexity grows with input size

- e.g. Can do sorting in $O(n \log n)$

Only the rough rate considered

- Exact time depends on the technology

- Real question: Do we scale well? How much more computation will be needed as the instances of the problem get larger.

- "Polynomial time" ($O(n)$, $O(n^2)$, $O(n^3)$, ...) considered feasible
Infeasible Computation

“Super-Polynomial time” considered infeasible
- e.g. $2^n$, $2^{\sqrt{n}}$, $n^{\log(n)}$
- i.e., as $n$ grows, quickly becomes “infeasibly large”

Can we make breaking security infeasible for Eve?

What is $n$ (that can grow)?

Message size?
- We need security even if sending only one bit!
Security Parameter

- A parameter that is part of the encryption scheme
- Not related to message size
- A knob that can be used to set the security level
- Will denote by $k$
- Security guarantees are given asymptotically as a function of the security parameter
Feasible and Negligible

We want to tolerate Eves who have a running time bounded by some polynomial in $k$

Eve could toss coins: **Probabilistic Polynomial-Time (PPT)**

It is better that we allow Eve high polynomial times too (we’ll typically tolerate some super-polynomial time for Eve)

But algorithms for Alice/Bob better be very efficient

Eve could be **non-uniform**: a different strategy for each $k$

Such an Eve should have only a “negligible” advantage (or, should cause at most a “negligible” difference in the behaviour of the environment in the SIM definition)

What is negligible?
Negligibly Small

A negligible quantity: As we turn the knob the quantity should “decrease extremely fast”

Negligible: decreases as $1/$superpoly$(k)$

i.e., faster than $1/$poly$(k)$ for every polynomial

e.g.: $2^{-k}$, $2^{-\sqrt{k}}$, $k^{-(\log k)}$.

Formally: $T$ negligible if $\forall c>0 \ \exists k_0 \ \forall k>k_0 \ \ T(k) < 1/k^c$

So that $\text{negl}(k) \times \text{poly}(k) = \text{negl}'(k)$

Needed, because Eve can often increase advantage polynomially by spending that much more time/by seeing that many more messages
Interpreting Asymptotics

If adversary runs for less than this long

Would like this to be super-polynomial

and this to be negligible

Then its advantage is no more than this

Time steps

Security parameter

Time to tolerate

set k here

Admissible advantage

Advantage
Symmetric-Key Encryption

SIM-CPA Security

SIM-CPA secure if:
\[ \forall \text{PPT} \exists \text{PPT} \text{s.t.} \forall \text{PPT} \text{Key}/\text{Enc} = \text{Key}/\text{Dec} \]

IDEAL \(\approx\) REAL

\[ | \Pr[\text{IDEAL}=0] - \Pr[\text{REAL}=0] | \text{ is negligible} \]
**Symmetric-Key Encryption**

**IND-CPA Security**

*Experiment picks a random bit $b$. It also runs KeyGen to get a key $K$*

*For as long as Adversary wants*

*Adv sends two messages $m_0$, $m_1$ to the experiment*

*Expt returns $\text{Enc}(m_b, K)$ to the adversary*

*Adversary returns a guess $b'$*

*Experiment outputs 1 iff $b' = b$*

**IND-CPA secure if for all “efficient” adversaries** $\Pr[b' = b] \approx 1/2$  

$\Pr[b' = b] - 1/2$ is negligible
Indistinguishability

Security definitions often refer to indistinguishability of two distributions: e.g., REAL vs. IDEAL, or Enc(m_0) vs. Enc(m_1).

- By a distinguisher who outputs a single bit
- 3 levels of indistinguishability
  - **Perfect**: the two distributions are identical
  - **Computational**: for all PPT distinguishers, probability of the output bit being 1 is only negligibly different in the two cases
  - **Statistical**: the two distributions are “statistically close”
- Hard to distinguish, irrespective of the computational power of the distinguisher
Statistical Indistinguishability

Given two distributions $A$ and $B$ over the same sample space, how well can a (computationally unbounded) test $T$ distinguish between them?

$T$ is given a single sample drawn from $A$ or $B$

How differently does it behave in the two cases?

$$\Delta(A,B) := \max_T | \Pr_{x \leftarrow A}[T(x)=1] - \Pr_{x \leftarrow B}[T(x)=1] |$$

Two distribution ensembles $\{A_k\}_k$, $\{B_k\}_k$ are \textbf{statistically indistinguishable} from each other if $\Delta(A_k,B_k)$ is negligible in $k$
Next

- Constructing (CPA-secure) SKE schemes
  - Pseudorandomness Generator (PRG)
  - One-Way Functions (& OW Permutations)
  - $\text{OWP} \rightarrow \text{PRG} \rightarrow \text{(CPA-secure) SKE}$