Formal Notions of Encryption

Debdeep Mukhopadhyay
IIT Kharagpur

Notion of Security

- "A Good disguise should not reveal the person's height"
  - Shafi Goldwasser and Silvio Micali, 1982

Design of Encryption Algorithms

- Encryption algorithms are used for privacy of data.
  - which means they do not leak any information about the plaintext
- The question is when are we satisfied that the cipher really does not leak?
  - For this we need to know the power of the adversary.

Notations of Encryption

Enc takes as input a key, $k \in K$, a message, $m \in M$, and outputs a ciphertext $c \in C$.

The encryption algorithm is a probabilistic algorithm, which means that the same, message may yield a different ciphertext, if run multiple times.
Thus, $c \leftarrow Enc_k(m)$.

Decryption must give the message

$$\forall k \in K \text{ and } m \in M,\ c \leftarrow Enc_k(m),$$
$$m = Dec_k(c),$$
with probability 1.

The distribution of *K* and *M* are independent..

Pr[M=m] is the probability that the message is m.

Given the encryption scheme, the distribution over *C* is fully determined by the distributions over *K* and *M.*

Notion of Perfect Secrecy

- The adversary likely knows the probability distribution over *M*.
- The adversary observes the ciphertext being generated.
  - Ideally, however this ciphertext should not leak any information. to the adversary.
  - For any message, m, the a posteriori probability that m was sent, should be same as the a priori probability.

Formalization

DEFINITION 1

An encryption scheme (Gen,Enc,Dec) over a message space M is perfectly secret if for every probability distribution over M, every message $m \in M$, and every ciphertext $c \in C$ for which $\Pr[C = c] > 0$:

$$\Pr[M=m|C=c]=\Pr[M=m]$$

Shannon formalized this concept, and called it perfect secrecy.

An equivalent statement

DEFINITION 2

An encryption scheme (Gen,Enc,Dec) over a message space M is perfectly secret if and only if for every probability distribution over M, every message $m \in M$, and every ciphertext $c \in C$:

$$\Pr[C=c|M=m]=\Pr[C=c]$$

Perfect Indistinguishability

- A useful formulation.
- It is impossible to distinguish an encryption of m_0 from an encryption of m_1.
- Thus the ciphertext distribution contains no information of the plaintext.

DEFINITION 3

An encryption scheme (Gen,Enc,Dec) over a message space M is perfectly secret if and only if for every probability distribution over M, every $m_0, m_1 \in M$, and every ciphertext $c \in C$:

$$\Pr[C=c|M=m_0]=\Pr[C=c|M=m_1]$$

Proof

Assume perfect secrecy:

$$\Pr[C=c \mid M=m_0]=\Pr[C=c]=\Pr[C=c \mid M=m_1]$$

Assume next that for every distribution over M, every $m_0, m_1 \in M$, every $c \in C$, it holds that:

$$\Pr[C=c|M=m_0]=\Pr[C=c|M=m_1].$$

Define, $p=\Pr[C=c \mid M=m_0]$.

$$\begin{aligned}\Pr[C=c] &= \sum_{m\in M} \Pr[C=c \mid M=m].\Pr[M=m] \\ &= \sum_{m\in M} p.\Pr[M=m] \\ &= p\sum_{m\in M} \Pr[M=m]=p \\ &= \Pr[C=c \mid M=m_0]\end{aligned}$$

What Shannon said?

- Shannon said in his classical work that using a one-time pad, the cipher achieved "perfect secrecy"
  - no attacker, even with infinite power of computation can obtain any information about the plain-text.
  - But the one-time pad is impractical.

Adversary's Experiment

- The definition of perfect secrecy is based on an experiment A.
- This experiment is essentially a game between an adversary, A, who is trying to break a cryptographic algorithm and an imaginary tester who wishes to see if the adversary succeeds.
- The definition tries to formalize the inability of A to distinguish the encryption of one plaintext from the encryption of another plaintext.

The Experiment

Define experiment PrivK^{eav} : Private-key encryption setting.
The experiment is defined for any encryption scheme:
Π =(Gen,Enc,Dec) over message space M and for any adversary A.
The steps are defined as follows:
1. The adversary A outputs a pair of messages $m_0, m_1 \in M$.
2. Imaginary entity generates a random key k by running Gen, and a random bit $b \xleftarrow{R} \{0,1\}$ is chosen.
Computes, $c \leftarrow Enc_k(m_b)$ and gives it to A.
3. A outputs a bit b'.
4. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

We write $\text{PrivK}^{eav}_{A,\Pi} = 1$ if the output is 1, and in this case we say that A succeeded.

Adversarial definition of perfect secrecy

DEFINITION 4

An encryption scheme (Gen,Enc,Dec) over a message space M is perfectly secret if for every adversary A,

$$\Pr[\text{Priv}K^{eav}_{A,\Pi} = 1] = \frac{1}{2}$$

Proofs (Definition 1 => Definition 4)

The scheme is also perfectly secret for the message space
$M = \{m_0, m_1\}$.
Thus, from message indistinguishability,
we have $\Pr[c \in C_0 \mid m = m_0] = \Pr[c \in C_0 \mid m = m_1]$.

$\therefore Adv_{A,\Pi} = \Pr[\text{Priv}K^{eav}_{A,\Pi} = 1] = \Pr[b = b']$

$= \Pr[b=0]\Pr[\text{Priv}K^{eav}_{A,\Pi} = 1 \mid b=0] + \Pr[b=1]\Pr[\text{Priv}K^{eav}_{A,\Pi} = 1 \mid b=1]$

$= \frac{1}{2}(\Pr[A \text{ outputs } 0|\text{b=0}] + \Pr[A \text{ outputs } 1|\text{b=1})$

Definition 1 => Definition 4

Let A outputs 0 if $c \in C_0$, and outputs 1 if $c \in C_1$. Also, $C = C_0 \cup C_1$.
Thus, we have

$$
\begin{aligned}
Adv_{A,\Pi} &= \frac{1}{2}\Big(\sum_{C_0} \Pr[c \in C_0 \mid m = m_0] + \sum_{C_1} \Pr[c \in C_1 \mid m = m_1]\Big) \\
&= \frac{1}{2}\Big(\sum_{C_0} \Pr[c \in C_0 \mid m = m_1] + \sum_{C_1} \Pr[c \in C_1 \mid m = m_1]\Big) \\
&= \frac{1}{2}\Big(\sum_{C_0 \cup C_1} \Pr[c \in C \mid m = m_1]\Big) = \frac{1}{2}.
\end{aligned}
$$

Exercise

- Definition 4 => Definition 1.

Proof by contradiction

$\neg$Defn 1 $\Rightarrow$ $\neg$Defn 4

Assume that Π is not perfectly secret.

$\Rightarrow \exists m_0, m_1 \in M$ and a ciphertext $\bar{c} \in C$ st.

$\Pr[\mathrm{C}=\bar{c} \mid M = m_0] = \Pr[\mathrm{C}=\bar{c} \mid M = m_1]$

Define an A, st. $\mathrm{A}(\mathrm{C}=\bar{c}) = 0,$

$\mathrm{A}(\mathrm{C} \neq \bar{c}) = b(\text{random guess})$

Advantage of A

$$\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1]=$$
$$\frac{1}{2}(\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid M=m_0]+\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid M=m_1])$$
$$\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid M=m_0]$$
$$=\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \wedge C=\bar{c} \mid M=m_0]+\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \wedge C\neq\bar{c} \mid M=m_0]$$
$$=\Pr[C=\bar{c} \mid M=m_0]\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid C=\bar{c}, M=m_0]$$
$$+\Pr[C\neq\bar{c} \mid M=m_0]\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid C\neq\bar{c}, M=m_0]$$
$$=\Pr[C=\bar{c} \mid M=m_0]+\frac{1}{2}\Pr[C\neq\bar{c} \mid M=m_0]$$

Advantage of A

Likewise,

$$\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid M=m_1]$$
$$=\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \wedge C=\bar{c} \mid M=m_1]+\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \wedge C\neq\bar{c} \mid M=m_1]$$
$$=0+\Pr[C\neq\bar{c} \mid M=m_1]\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1 \mid C\neq\bar{c}, M=m_1]$$
$$=\frac{1}{2}\Pr[C\neq\bar{c} \mid M=m_1]$$

The Contradiction

$$\Pr[\mathrm{PrivK}^{eav}_{A,\Pi}=1]=\frac{1}{2}(\Pr[C=\bar{c}\mid M=m_0]+\frac{1}{2}\Pr[C\neq\bar{c}\mid M=m_0])+$$
$$\frac{1}{2}\frac{1}{2}\Pr[C\neq\bar{c}\mid M=m_1]$$
$$=\frac{1}{2}(\Pr[C=\bar{c}\mid M=m_0]+\frac{1}{2}(1-\Pr[C=\bar{c}\mid M=m_0]))$$
$$+\frac{1}{4}\Pr[C\neq\bar{c}\mid M=m_1]$$
$$=\frac{1}{4}+\frac{1}{4}(\Pr[C=\bar{c}\mid M=m_0]+\Pr[C\neq\bar{c}\mid M=m_1])$$
$$\neq\frac{1}{4}+\frac{1}{4}(\Pr[C=\bar{c}\mid M=m_1]+\Pr[C\neq\bar{c}\mid M=m_1])$$
$$=\frac{1}{2}$$

One Time Pad

Let $a\oplus b$ denote the bit-wise XORof two binary strings, a and b, $a=a_1...a_l, b=b_1...b_l$ and $a\oplus b=a_1\oplus b_1...a_l\oplus b_l$

1. Fix an integer $l>0$. Then the message space M, key space K, and ciphertext space C are all equal to $\{0,1\}^l$.
2. The key generation algorithm Gen works by choosing a string from $K=\{0,1\}^l$ uniformly.
3. Encryption Enc works as follows: given a key $k\in\{0,1\}^l$, output $c=k\oplus m$.
4. Decryption Dec works as follows: given a key $k\in\{0,1\}^l$, output $m=k\oplus c$.

Proof of Perfect Secrecy

$$\Pr[C = c \mid M = m] = \Pr[M \oplus K = c \mid M = m]$$
$$= \Pr[m \oplus K = c] = \Pr[K = m \oplus c] = \frac{1}{2^l}$$

This holds true for any message belonging to M.

Large key space

Let (Gen, Enc, Dec) be a perfectly secret encryption scheme over a message space M, and let K be the keyspace as determined by Gen. Then $|K| \geq |M|$.

Proof

Assume $|K| < |M|$.

Let $M(c)$ be the set of all possible messages which are possible decryptions of the ciphertext c.

$\therefore M(c) = \{m \mid m = Dec_k(c) \text{ for some } k \in K\}$

Clearly, $|M(c)| \leq |K|$, but $|K| < |M|$ by assumption.

Thus, $\exists m' \in M$, but $\notin M(c)$.

$\Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$.

This violates definition 1.

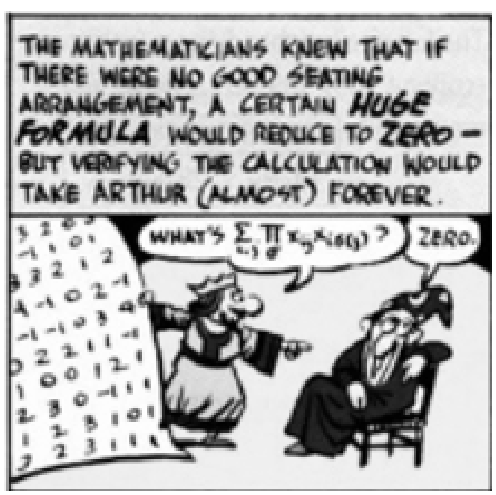
Computational Security

- The previous schemes which are secured against the unbounded adversary are called information theoretic secured.
- However they are not practical.
- In the practical world, we try to develop computationally secured ciphers.
- These definitions are weaker than that of perfect secrecy.
- But the proof techniques have to be still formally stated, with assumptions etc.

What is computationally secured?

- A cipher must be practically, if not mathematically, indecipherable.
- Goal is to design a cipher which cannot be broken in “reasonable time” with a “reasonable probability of success”.

Mathematicians and Time



Relaxations from notion of perfect secrecy

- Security is only preserved against efficient adversaries that run in a feasible amount of time.
- Adversaries can succeed with a very small probability of success.

Two approaches

- Concrete approach: quantifies security of a crypto scheme by explicitly bounding the maximum success probability of any adversary running for at most specified amount of time.

A scheme is $(t, \varepsilon)-$secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ε.

Asymptotic Approach

- This approach origins from complexity theory.
- It views the running time of the adversary as well as its success probability as functions of some parameter (not concrete numbers).
- The cryptographic scheme has a security parameter, which is denoted by n.
- The honest party initializes the scheme G, by choosing n.
- This value is known to the adversary.
- Running time of the honest parties and the adversary are all functions of n.
- The adversaries success probability is also a function of n.

PPT

- Probabilistic Algorithms or randomized algorithms, A, may toss a coin a finite number of times during its computation.
- The output y, and the next step may depend on the results of the preceding coin tosses.
- The coin is in general fair.
- Examples: Primality test algorithms, factoring algorithms etc.

Efficiency

- By efficient, we mean that for some constants a, c, the algorithm runs in time $a.n^c$, for the security parameter n.
  - Honest parties are efficient.
  - Adversaries with a run time which is superpolynomial can be considered "impractical"

Negligible Function

A function f is negligible if for every polynomial $p(\ )$ there exists an N such that for all integers $n > N$ it holds that $f(n) < \dfrac{1}{p(n)}$

Closure of negligible functions

- The function $negl_3$ defined by $negl_3(n)=negl_1(n)+negl_2(n)$ is negligible
- For any positive polynomial p, the function $negl_4$ defined by $p(n).negl_1(n)$ is negligible.

Negligible Probability

- Inverse polynomial: n^{-c}, for a constant c.
- A function that grows slower than any inverse polynomial.
- This means that for every constant c, if the success probability of the adversary is smaller than n^{-c}, then the probability is said to be negligible.

Informal Definition

- A scheme is secured if every **PPT adversary** succeeds in breaking the scheme with only **negligible probability**.
  - useful for large values of n.

    Consider a scheme where “an adversary running for n^3 minutes can succeed in breaking the scheme with probability $2^{40}2^{-n}$.

    Need n around 500 for the adversary to run for more than 200 years to break with a probability of 2^{-500}.

Increase of Security Parameter

- Consider a cryptographic scheme where honest parties are required to run for 10^6n^2 cycles.
- An adversary running for 10^8n^4 cycles can break the scheme with probability $2^{20}2^{-n}$.
- Consider a 1 GHz computer and n=50.
  - Run time (honest parties)=2.5 sec, adversary run time 1 week, prob of succ=2^{-30}.
- Condsider a 16 GHz processor, n=100
  - Run time (honest parties)=0.625 sec, adversary run time 16 weeks, prob of succ=2^{-80}.

In general increase in n, will increase the security of the scheme.

Proofs by Reduction

- Central to provable cryptography
- Assumption: Some problem X cannot be solved by any polynomial time algorithm except with negligible probability.
- We want to prove that some cryptographic construction (Pi) is secured, say in computational sense.

General Proof method

1. Fix some efficient adversary A attacking Π.
Denote this adversary's success probability by $\varepsilon(n)$.
2. Construct an efficient algorithm A' that attempts to solve problem X using adversary A as a subroutine.

General Proof method

Note that A' knows nothing about how A works. It only knows that A attempts to break Π. So, given an instance x of X, the algorithm A' will simulate for A an instance of Π st:

i) The view of A, when it is run as a sub-routine of A' should be distributed identically to the view of A, when it is run directly with Π itself.

ii) If A succeeds in breaking the instance of Π, that is being simulated by A', this will enable A' to solve the instance x of X with a non-negligible probability (greater than an inverse probability 1/p(n)

General Proof method

This implies we have an efficient algorithm A' which solves problem X with a probability greater than $\varepsilon(n) / p(n)$.
This contradicts the initial assumption.

Thus given the assumption regarding X, no efficient adversary A can succeed in breaking Π with probability that is not negligible.

Formalizing Computational Security

Refining Definition 4 for Computational Security

- We consider only adversaries running in polynomial time.
- The adversary might determine the encrypted message with probability negligibly better than ½ .

Eavesdropping Indistinguishability Experiment

Thus given the assumption regarding X, no efficient adversary A can succeed in breaking Π with probability that is not negligible.

1. The adversary A is given input 1^n, and outputs a pair of messages m_0, m_1 of the same length.

2. The key k is generated by running $\mathrm{Gen}(1^n)$, and a random bit $b \leftarrow \{0,1\}$ is chosen.

Eavesdropping Indistinguishability Experiment

A ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to A. We call c the challenge ciphertext.

3. A outputs a bit b'.

4. The output of the experiment is defined to be 1 if b'=b, and 0 otherwise.

If $\mathrm{PrivK}^{eav}_{A,\Pi}(n) = 1$, we say that A succeeded.

Formal Definition

A private key encryption scheme Π=(Gen,Enc,Dec) has indistinguishable encryptions in the presence of an eavesdropper if for all probabilistic polynomial adversaries *A* there exists a negligible function *negl* such that:

$$\Pr[\mathrm{Priv}^{eav}_{A,\Pi}(n)=1] \le \frac{1}{2} + negl(n)$$

where the probability is taken over the random coins used by *A*, as well as the random coins used in the expe -riment (for choosing the key, the random bit b, and any random coins used in the encryption process).

Definition of Semantic Security (SS)

For every distribution X over $\{0,1\}^n$ and
For every partial information $h: \{0,1\}^n \to \{0,1\}^n$
For every interesting information f: $\{0,1\}^n \to \{0,1\}^*$
For every attacking algorithm A running in time
t' ≤ t(n) [t(n) is a polynomial in n], there exists a
simulating algorithm S such that:

$$\Pr_{\substack{m\leftarrow X \\ k\leftarrow G(n)}}[A(E_k(m),h(m)) = f(m)] \le \Pr_{m\leftarrow X}[S(h(m)) = f(m)] + \varepsilon(n)$$

- Here ε(n) is a negligible quantity.
- Notion tries to attempt ideal security.
- That is the eavesdropper is disconnected from the communication.
- In spite of observing the ciphertext, he obtains no extra interesting observation than the case when he has not seen the ciphertext.

Message Indistinguishability (MI)

For every two messages $m_0, m_1 \in \{0,1\}^n$

For every attacking algorithm A that runs in time $\leq$ t(n)

$$\Pr_{\substack{i\in\{0,1\}\\ k\leftarrow G}}[A(E_k(m_i)) = i] \leq \frac{1}{2} + \varepsilon(n)$$

- SS and MI are equivalent

Proofs : SS => MI

If $X = \{m_0, m_1\}$, $f : f(m_0) = 0,\ f(m_1) = 1,$ h(): empty output string

From SS, for every adversary A there is a simulator S, st.

$$\Pr_{\substack{m\leftarrow X\\ k\leftarrow G}}[A(E(m)) = i] \leq \Pr_{m\leftarrow X}[S() = i] + \varepsilon(n)$$

Now, since the simulator receives no information:

$\Pr[S() = i] = 1/2$, regardless of S.

Thus, $\Pr_{\substack{i\in\{0,1\}\\ k\leftarrow G}}[A(E(m_i)) = i] \leq \dfrac{1}{2} + \varepsilon(n)$

MI => SS

For every $m_0, m_1 \in \{0,1\}^n$, for every algorithm A that runs in time $\leq t(n)$, for every $a \in \{0,1\}^*$,

$\Pr_{k \in G}[A(E_k(m_1)) = a] - \Pr_{k \in G}[A(E_k(m_0)) = a] \leq 2 \in (n)$

(*)

$(t, \in) - MI \Rightarrow * \equiv \neg(*) \Rightarrow \neg(t, \in) - MI$

MI => *

Define, $A'(c) = \begin{cases} 1, & \text{if } A(c) = a \\ 0, & \text{otherwise} \end{cases}$

$$\therefore \Pr_{\substack{i \in \{0,1\} \\ k \leftarrow G}}[A'(E_k(m_i)) = i]$$

$$= \frac{1}{2}\Pr_{k \leftarrow G}[A'(E_k(m_0)) = 0] + \frac{1}{2}\Pr_{k \leftarrow G}[A'(E_k(m_1)) = 1]$$

$$= \frac{1}{2}(1 - \Pr_{k \leftarrow G}[A(E_k(m_0)) = a]) + \frac{1}{2}\Pr_{k \leftarrow G}[A(E_k(m_1)) = a]$$

$$= \frac{1}{2} + \frac{1}{2}(\Pr_{k \leftarrow G}[A(E_k(m_1)) = a] - \Pr_{k \leftarrow G}[A(E_k(m_0)) = a])$$

$$> \frac{1}{2} + \in(n) \Rightarrow (t, \in) - MI \text{ is violated.}$$

(t,ε)-MI=>(t',2ε)-SS

- Thus ┐ (t',2ε)-SS =>┐ (t,ε)-MI

define $S(z)$, where z is some information on m

Pick k $\leftarrow G$ at random

Return $A(E_k(m_0), z)$

/* Note that the run time of S is running time of A+poly(n) */

(t,ε)-MI=>(t',2ε)-SS

$\neg$(t',2ε)-SS $\Rightarrow$

$$\Pr_{\substack{m\leftarrow X \\ k\leftarrow G}}[A(E(m),h(m)) = f(m)] > \Pr_{m\leftarrow X}[S(h(m)) = f(m)] + 2\varepsilon(n)$$

$$or,\ \Pr_{\substack{m\leftarrow X \\ k\leftarrow G}}[A(E(m),h(m)) = f(m)]$$

$$> \Pr_{\substack{m\leftarrow X \\ k\leftarrow G}}[A(E(0),h(m)) = f(m)] + 2\varepsilon(n)$$

$$or,\ \sum_{m} \Pr[X = m](\Pr_{k\leftarrow G}[A(E(X),h(X)) = f(X)]$$

$$- \Pr_{k\leftarrow G}[A(E(0),h(X)) = f(X)]) > 2\varepsilon(n)$$

$$\Rightarrow \exists m' \in X,\ \text{st. } \Pr_{k\leftarrow G}[A(E(m'),h(m')) = f(m')]$$

$$- \Pr_{k\leftarrow G}[A(E(0),h(m')) = f(m')]) > 2\varepsilon(n)$$

$\Rightarrow$ as there exists a pair of messages for which (*) does not hold

$\Rightarrow (t,\in) - MI$ does not hold.