Hard Core Predicates: How to encrypt?

Debdeep Mukhopadhyay
IIT Kharagpur

Recap

- *A encryption scheme is secured if for every probabilistic adversary A carrying out some specified kind of attack and for every polynomial p(.), there exists an integer N s.t. the probability that A succeeds in this attack is less than 1/p(n) for every n>N*

Hard Core Predicates

If $f:\{0,1\}^n \to \{0,1\}^n$, and bijective, a poly(n) computable $B:\{0,1\}^n \to \{0,1\}$ is $(t,\varepsilon)-hp$ for f if for every A with running time $\leq$ t(n),

$$\Pr_X [A(f(X))=B(X)] \leq \frac{1}{2} + \in (n)$$

One-way functions and Trapdoors

- They are class of functions which are easy to compute in one direction (poynomial time), but hard to invert (cannot be inverted in polynomial time)
- But can be easily inverted with a secret information, called the trap-door information.

Example with RSA

- $y=x^e \bmod(pq)$ [Easy to compute]
- Given y, e and N=pq, we do not know efficient techniques to compute x.
- But if we have a trap-door

$$d=e^{-1} \bmod (p-1)(q-1)$$

it becomes easy to compute x and hence invert the function.

Hard Core Predicate of trapdoor permutations

(G,F,I) is a family of trapdoor permutations,
G chooses (k, t_k)
$F(.,k)$ is bijective
$I(.,t_k,k)$ is inverse of $F(.,k)$
$st,$ G,F,I can be done in poly(n) time and inverting F without t_k is hard.

HP for trap-door permutations

If (G, F, I) is a family of trapdoor permutations, then polynomial time one bit output $B(X, k)$ is a hard-core predicate if for every A running in time $\leq$ t(n),

$$\Pr_{(k,t_k)\in G(n)}[\mathrm{A(F(X,k),k)=B(X,k)}] \leq \frac{1}{2} + \varepsilon(\mathrm{n})$$

Goldreich-Levin Theorem

- If there is a family of trapdoor permutations, then there is a family with a hard core predicate.

Encrypting a bit b

- **Given (G,F,I),t_k and a hardcore predicate B**
- **Key Generation:** G
  - Return (k,t_k)
- **Encryption:** $E(b,t_k)$
  - Pick random X Є $\{0,1\}^n$
  - Return $F(X,k), b \oplus B(X,k)$
- **Decryption:** $D((z,c),k,t_k)$
  - $X=I(z,t_k)$
  - Return $c \oplus B(X,k)$

The Encryption is MI secure

$$\Pr_{\substack{(k,t_k)\in G(n)\\ X\in\{0,1\}^n\\ b\in\{0,1\}}}[A(F(X,k), b\oplus B(X,k), k)=b] \le \frac{1}{2}+\varepsilon(n)$$

Proof:

Suppose this encryption is not $(t,\varepsilon)-MI$ secure.

$$\Pr_{\substack{(k,t_k)\in G(n)\\ X\in\{0,1\}^n\\ b\in\{0,1\}}}[A(F(X,k), b\oplus B(X,k), k)=b] > \frac{1}{2}+\varepsilon(n)$$

Consider algorithm A'(y,k)

Pick random $c \in \{0,1\}$

Return $c \oplus A(y,c,k)$

Thus,

$$\Pr_{X\in\{0,1\}^n}[A'(F(X,k),k)=B(X,k)]$$

$$=\Pr_{\substack{(k,t_k)\in G(n)\\ X\in\{0,1\}^n\\ c\in\{0,1\}}}[A(F(X,k),c,k)=B(X,k)\oplus c]$$

$$=\Pr_{\substack{(k,t_k)\in G(n)\\ X\in\{0,1\}^n\\ b\in\{0,1\}}}[A(F(X,k), b\oplus B(X,k), k)=b] > \frac{1}{2}+\varepsilon(n)$$

Example for RSA

- B(X,(N,e))=X mod 2 is a hp for RSA
  - that is given (N,e), X^e mod N it is hard to guess X mod 2 with a non-negligibly large probability than ½
- Encrypt bЄ{0,1} with RSA
  - Pick X Є$\{0,1\}^n$
  - Compute, X^e mod N, XOR(b,Xmod 2)

How to encrypt longer strings?

- **Given (G,F,I),t_k and a hardcore predicate B**
- **Key Generation:** G
  - Return (k,t_k)
- **Encryption:** E_{GM}(m,k), mЄ{0,1}

  for i=1 to n
  - Pick random X Є $\{0,1\}^n$
  - Return $F(X,k), m[i] \oplus B(X,k)$
- **Decryption:** D((z,c),k,t_k)
  - X=I(z,t_k)
  - for i=1 to n

    Return $d[i] \oplus B(X,k)$

Proof of MI secured

For every m, m' for every A running in time $\leq$ t(n)

$\Pr[A(E_{GM}(m,k),k)=1]-\Pr[A(E_{GM}(m',k),k)=1]\leq 2\varepsilon$

If we contradict this supposition, we have

$\exists A, m, m'$ s.t.

$\Pr[A(E_{GM}(m,k),k)=1]-\Pr[A(E_{GM}(m',k),k)=1]>2\varepsilon$

Contd.

Consider the following hybrid construction:

Pr[A(E(m[1])E(m[2])...E(m[n]))=1]=p_0

Pr[A(E(m'[1])E(m[2])...E(m[n]))=1]=p_1

...

Pr[A(E(m'[1])E(m'[2])...m'[i]m[i+1]...E(m[n]))=1]=p_i

Pr[A(E(m'[1])E(m'[2])...m'[i]m'[i+1]...E(m[n]))=1]=p_{i+1}

...

Pr[A(E(m'[1])E(m[2])...E(m'[n-1])E(m[n]))=1]=p_{n-1}

Pr[A(E(m[1])E(m[2])...E(m'[n-1]E(m'[n]))=1]=p_n

Contd.

So, from our contradiction we have:

$$p_0 - p_n > 2\varepsilon$$

$$or, \sum_{i=0}^{n-1} (p_i - p_{i+1}) > 2\varepsilon$$

$$or, \exists i : (p_i - p_{i+1}) > \frac{2\varepsilon}{n}$$

i, e

Pr[A(E(m'[1])E(m'[2])...m'[i]m[i+1]...E(m[n]))=1]

-Pr[A(E(m'[1])E(m'[2])...m'[i]m'[i+1]...E(m[n]))=1] $> \frac{2\varepsilon}{n}$

Contd.

Consider algorithm A'(c,k)

Compute, $c_1 = E(m'[0])$

$$\ldots$$

$$c_i = E(m'[i])$$

$$c_{i+2} = E(m[i+2])$$

$$\ldots$$

$$c_n = E(m[n])$$

Return $A(c_1, \ldots, c_i, c, c_{i+2}, \ldots, c_n)$

contd.

$$\Pr[A'(c,k)=1]-\Pr[A'(\neg c,k)=1]$$
$$=\Pr[A(c_1,\ldots,c_i,c,c_{i+1},\ldots,c_n)]-\Pr[A(c_1,\ldots,c_i,\neg c,c_{i+1},\ldots,c_n)]$$
$$>\frac{2\varepsilon}{n}$$

This contradicts the fact that one bit encryption was MI secure.

A Hard Core Predicate for any one-way function

Let (G,F,I) be a family of trap-door permutations.
Consider (G,F',I') , which is also a family of trap-door permutations.

$I'((z,r),t_k)=I(z,t_k),r$ and

$F'((x,r),k)=\langle F(x,k),r\rangle$

Then $B(x,r)=\sum_i x_i.r_i \mod 2$

is a hard core predicate for (G',F',I').

Proof

- Let us drop the variables k and t_k for simplicity. The proof is unchanged with them.
- Assume that there is a polynomial time algorithm A, that always correctly computes B(x,r) given F'(x)=(F(x),r)
  - we shall show that easy to compute x from f(x). This contradicts our assumption that F is one-way.

Details

- Let A be a PPT algorithm which computes the value of B(x,r) from F'(x,r)=F(x),r

$$\Pr_{x,r \leftarrow \{0,1\}^n}[A(F(x),r) = B(x,r)] = 1$$

- Now we shall frame an experiment A', which invokes A for i=1,2,…,n.
- The arguments being passed to A are x and e_i
  - e_i denotes a string with the i^{th} bit 1 and rest 0.
  - Since, A computes the term $B(x,e_i)=x_i$ with probability 1, the entire x is retrieved by A' by executing A n number of times.
  - Note that the run time of A' is also polynomial in n and also has a probability of 1.

But that is not all!

- The G-L Theorem says that the probability of computing B(x,r) from F'(X,r)=(F(x),r) should be greater than ½ by a negligible quantity
  - So, assuming a probability of 1 is a weak case.
  - Slightly more involved case (and more closer to the proof) will be if the probability is significantly greater than ¾.

Why the previous proofs does not work?

- It may be that A never succeeds in computing B(x,r) correctly when $r=e_i$
- The algorithm A' has no means of understanding that A has succeeded or not?
  - So, what does A' do in this case to increase his chance?
    - (repeat the experiment of A)

Two important observations

$$B(x,r) \oplus B(x, r \oplus e_i) = B(x, e_i) = x_i$$

- note that A is invoked with random inputs.
- There is no way to understand when A gives a correct answer. So, run A multiple times and take the majority.
- A preliminary step would be to prove that for many x's, the probability that A answers both the predicate queries correctly is very high.

Claim 1

If, $\Pr_{x,r \leftarrow \{0,1\}^n}[A(F(x),r) = B(x,r)] \geq \frac{3}{4} + \varepsilon(n)$.

Then there exists a set $S_n \subseteq \{0,1\}^n$ of size at least $\frac{\varepsilon(n)}{2} 2^n$, where for every $x \in S_n$:

$\Pr_{r \leftarrow \{0,1\}^n}[A(F(x),r) = B(x,r)] \geq \frac{3}{4} + \frac{\varepsilon(n)}{2}$

Proof

Define, $s(x) = \Pr_{r \leftarrow \{0,1\}}[A(F(x), r) = B(x, r)]$

We have to show that $|S_n| \geq \frac{\varepsilon(n)}{2} 2^n$

$\Pr_{x,r}[A(F(x), r) = B(x, r)]$

$= \Pr_{x,r}[A(F(x), r) = B(x, r) \mid x \in S_n] \Pr_x[x \in S_n]$

$+ \Pr_{x,r}[A(F(x), r) = B(x, r) \mid x \notin S_n] \Pr_x[x \notin S_n]$

$\leq \Pr_x[x \in S_n] + \Pr_{x,r}[A(F(x), r) = B(x, r) \mid x \notin S_n]$

$\therefore \Pr_x[x \in S_n] \geq \Pr_{x,r}[A(F(x), r) = B(x, r)]$

$- \Pr_{x,r}[A(F(x), r) = B(x, r) \mid x \notin S_n]$

i.e. $\Pr_x[x \in S_n] \geq \frac{3}{4} + \in(\mathrm{n}) - (\frac{3}{4} + \frac{\in(\mathrm{n})}{2}) = \frac{\in(\mathrm{n})}{2}$

Thus, $\mathrm{S_n}$ must be of size at least $\frac{\in(\mathrm{n})}{2} 2^{\mathrm{n}}$ (because x is uniformly distributed in $\{0,1\}^{\mathrm{n}}$)

Claim 2

If, $\Pr_{x,r \leftarrow \{0,1\}^n}[A(F(x), r) = B(x, r)] \geq \frac{3}{4} + \varepsilon(n)$.

Then there exists a set $\mathrm{S_n} \subseteq \{0,1\}^n$ of size at least $\frac{\varepsilon(n)}{2} 2^n$, where for every $x \in S_n$ and every i it holds that:

$\Pr_{r \leftarrow \{0,1\}^n}[A(F(x), r) = B(x, r) \wedge A(F(x), r \oplus e_i) = B(x, r \oplus e_i)]$

$\geq \frac{1}{2} + \varepsilon(n)$

Proof

We know for $x \in S_n$:

$$\Pr_{r \leftarrow \{0,1\}^n}[A(F(x),r) \neq B(x,r)] < \frac{1}{4} - \frac{\varepsilon(n)}{2}$$

Fixing any i, if r is uniformly distributed so,
is $r \oplus e_i$. So,

$$\Pr_{r \leftarrow \{0,1\}^n}[A(F(x),r) \neq B(x,r \oplus e_i)] < \frac{1}{4} - \frac{\varepsilon(n)}{2}$$

We wish to upper-bound the probability that at
least one of the two predicates are wrongly computed.
From the theory of probability, this is atmost:

$$\left(\frac{1}{4} - \frac{\varepsilon(n)}{2}\right) + \left(\frac{1}{4} - \frac{\varepsilon(n)}{2}\right) = \frac{1}{2} - \varepsilon(n)$$

So, A is correct on both the queries with probability
at least $\frac{1}{2} + \varepsilon(n)$.

The strategy of A'

For i=1,...,n

1. Choose a random $r \leftarrow \{0,1\}^n$ and guess that the value $x_i = A(y,r) \oplus A(y, r \oplus e_i)$.
2. Repeat this procedure for a large number of cases, (only the number of trials has to be polynomial in n) and return the majority as the correct guess.

Can this proof be extended to the general case?

- Since it involves two computations of B(), the error probability is doubled.
- for the actual proof (and even when the error probability is exactly ¼ this will not help in inverting F with a significant prob)
- Instead, we guess one B and compute the other.
- m=poly(n) and set $l=\log_2(m+1)$

Can this proof be extended to the general case?

- Choose *l* strings uniformly and independently in $\{0,1\}^n$ and denote them by $s_1,\ldots,s_l$.
- Then guess $B(x,s_1),\ldots,B(x,s_l)$ and call them $\sigma_1,\ldots,\sigma_l$.
- Probability that all of them are correct is $1/2^l=1/\text{poly}(n)$
- Fix J as a subset of $\{1,\ldots,l\}$ and define $r^J = \oplus_{j\in J} s^j$

  It may be shown that the r^J's are pairwise independent and uniformly distributed in $\{0,1\}^n$

Can this proof be extended to the general case?

- Note that:

$$B(x,r^J) = B(x, \oplus_{j \in J} s^j) = \oplus_{j \in J} B(x, s^j)$$

- So, our guess for B(x,r^J) is $\rho^J = \oplus_{j \in J} \sigma^j$

The Actual Proof

A's experiment:

1. Generate and independently set $s^1,\ldots,s^l \in \{0,1\}^n$ and $\sigma^1,\ldots,\sigma^l \in \{0,1\}$
2. For every non-empty subset of J, $J \subseteq \{1,\ldots,l\}$ computes a string, $r^J = \oplus_{j \in J} s^j$ and a bit $\rho^J = \oplus_{j \in J} \sigma^j$
3. For every $i \in \{1,..,n\}$ and every non-empty subset of J, $J \subseteq \{1,\ldots,l\}$ computes,
   $z_i^J = \rho^J \oplus A(y, r^J + e_i)$
4. For every $i \in \{1,..,n\}$ it sets z_i to be the majority of the z_i^J values.
5. It outputs $z = z_1 \ldots z_n$

Analysis

- Next, we show that if for all j∈{1,…,l}, σ^j's are equal to $B(x,s^j)$, then:

$$z_i^J = B(x,r^J) \oplus A(F(x), r^J \oplus e^i)$$

has a majority equal to x_i for all i∈{1,…,n}

Claim

For every $x \in S_n$ and every $1 \le \mathrm{i} \le \mathrm{n}$,

$$\Pr[|\{J : B(x,r^J) \oplus A(F(X), r^J \oplus e^i) = x_i\}| > \frac{1}{2}(2^l - 1)]$$

$$> 1 - \frac{1}{2\mathrm{n}}$$

Proof

For every J define a 0-1 r.v M^J which equals 1,

iff $B(x, r^J) \oplus A(F(X), r^J \oplus e^i) = B(x, e^i) = x_i$

$\Rightarrow M^J = 1$ iff $A(F(X), r^J \oplus e^i) = B(x, r^J \oplus e^i)$

Thus, $\mathrm{M}^J = 1$ with probability at least

$\frac{1}{2} + \frac{\varepsilon(n)}{2}$, as $x \in S_n$.

Note that $B(x, r^J) \oplus A(F(X), r^J \oplus e^i) = x_i$

iff $\mathrm{M}^J = 1$ for majority of j's, $\mathrm{j} \in \mathrm{J}$.

Thus, $\Pr[\sum_{\mathrm{J}} \mathrm{M}^J \le \frac{m}{2}] = ?$

Chebyshev's Inequality

Let X be a r.v and $\delta > 0$

$$\Rightarrow \Pr[|X - E(X)| \ge \delta] \le \frac{Var(X)}{\delta^2}$$

$$\Pr[\sum_J M^J \le \frac{m}{2}] \le \Pr[|\sum_J M^J - (\frac{1}{2}+\frac{\varepsilon(n)}{2})m| \ge \frac{\varepsilon(n)}{2} m]$$

Note, $E(\sum_J M^J)=(\frac{1}{2}+\frac{\varepsilon(n)}{2})m$

$$Var(\sum_J M^J) = m(\frac{1}{2}+\frac{\varepsilon(n)}{2})(\frac{1}{2}-\frac{\varepsilon(n)}{2}) < \frac{m}{4}$$

$$\Pr[\sum_J M^J \le \frac{m}{2}] \le \Pr[|\sum_J M^J - (\frac{1}{2}+\frac{\varepsilon(n)}{2})m| \ge \frac{\varepsilon(n)}{2} m]$$

$$\le \frac{m/4}{(\varepsilon(n)/2)^2 m^2} = \frac{1}{\varepsilon(n)^2 m}$$

Let, $m = \frac{2n}{\varepsilon(n)^2}$, we have:

$$\Pr[\sum_J M^J \le \frac{m}{2}] \le \frac{1}{2n}$$

$$\therefore \quad \Pr[\sum_J M^J > \frac{m}{2}] \ge 1 - \frac{1}{2n}$$

This completes the proof of the claim.

Thus the probability that A' is wrong for a

particular value of i is at most $\frac{1}{2n}$

(it occurs when $\sum_J M_J \le \frac{1}{2}m$).

Thus, the probability that A' returns a wrong result

for at least one value of i is atmost $\frac{1}{2n}n = \frac{1}{2}$.

Thus the probability that it is correct for all the i values is

at least $\frac{1}{2}$.

Reminder, this was under the assumption that the l guesses were

all correct probability of which is 2^{-l}.

Hence if $x \in S_n$, A' inverts F(x) with a probability of

$$\frac{1}{2}.2^{-l} = \frac{1}{2}\frac{1}{m+1} = \frac{1}{2}\frac{1}{\frac{2n}{\varepsilon(n)^2}+1}$$

Also, we know $\Pr_x[x \in S_n] = \frac{\varepsilon(n)}{2}$

Thus, the probabilty that A' is able to invert F(x)

is at least $\frac{1}{2}\frac{1}{\frac{2n}{\varepsilon(n)^2}+1}\frac{\varepsilon(n)}{2} = \frac{1}{4}\frac{1}{2np(n)^3 + p(n)}$

which is a contradiction to the assumption that F(x)

is a one-way function.