Public Key Encryption Algorithms and the Random Oracle

SUSHANT MOHAN DEWAL

17th April 2009

Some schemes such as El-Gamal are both provable secure and efficient at the same time ,for these schemes we require a construction known as Random Oracle.

El-Gamal

$c = < g^y, h^y, m >, h = g^x$,y is random, m is the message,x is the secret
Enc : $< r^e modN, m \oplus G(r) >$
we run an IND-CPA exp on it
Pr[M']=Pr[$m.g^z = m'$]=Pr[$m = g^{-z}.m'$]
Case 1 :when function is random, $[g^y, g^z.m_b]$
Case 2 :when z is replaced by xy, $[g^y, g^{xy}.m_b]$
for Case 2(assuming $\epsilon(n)$ non-negligible :
$\mathrm{Pr}_{A,\pi}$ [Enc($p_k, m_i) = i$)] $= 1/2 + \epsilon(n)$
hence $\mathrm{Pr}_{A,\pi'}$ [Enc($p_k, m_1) = 1$)] - $\mathrm{Pr}_{A,\pi}$ [Enc($p_k, m_1) = 1$)] $\leq \beta$
but as per Decisional Diffie-Hellman assumption β is negligible.
$|1/2 - 1/2 - \epsilon(n)| \leq \beta$
so contradiction, Hence El-Gamal is secure

Random Oracle For convenience, a random oracle R is a map from $\{0,1\}^*$ to $\{0,1\}^\infty$ chosen by selecting each bit of R(x) uniformly and independently, for every x.

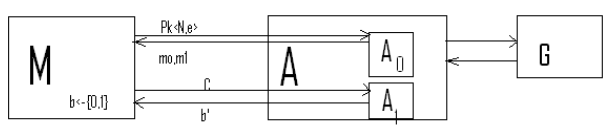
Here by infinity we mean that it is "sufficiently large".
A random oracle can be seen as a large book of random numbers, on any input x to the oracle the oracle returns a random number written on that page of the book,for same x the same number is returned but for different x random numbers are returned.

Random Oracle Model : A popular methodology for designing a cryptographic protocol consists of the following two steps :

1. Design an ideal system in which all parties (including the adversary) have oracle access to a truly random function, and proves the security of this ideal system.

2. one replaces the random oracle by a "good cryptographic hashing function" (such as MD5 or SHA),providing all parties (including the adversary)with succinct description of this function.

here instantiating the oracle with h(hash function) is only heuristic whose success we trust from experience
thus, one obtains an implementation of the ideal system in a world where random oracles do not exist.



Concrete Scheme

Consider a RSA based scheme,

- public key : [N,e]
- secret key : d

Enc : $<$[r$^e modN, m\oplus$H(r)]$>$
where m$\in \{0,1\}^{l(n)}$
it is to be proved that this encryption scheme is secure under IND-CPA;

Proof Technique

We will show that if Adversary A is able to break the Scheme using Random Oracle, than it can be used to break the std. cryptographic assumption of trapdoor function. For this we create a reduction that may choose values for the output of Random Oracle and return it to A- (programmability)
also this reduction sees all the queries that A makes to the Random Oracle.
Note: here PRG/PRP cannot be used $\because$ r has to be random, in case of RSA if information about r say LSB leaks then it is no more random and hence PRG/PRP cannot be used.

Construction

Assumptions :

1. RSA is hard to invert.
2. H is modeled as Random oracle.
3. all queries made to oracle are distinct.

Let A be PPT, ϵ(n)=Pr[Pub$^{eav}_{A,\pi}$(n)=1]

The Experiment Pub$^{eav}_{A,\pi}$(n) is defined as:

1. A random function H is chosen.
2. Generate $<$ N,e,d $>$.
3. A(adversary) is given p$_k$ $=< N, e >$ and may query H(.). A outputs m$_0, m_1 \leftarrow \{0,1\}^{l(n)}$

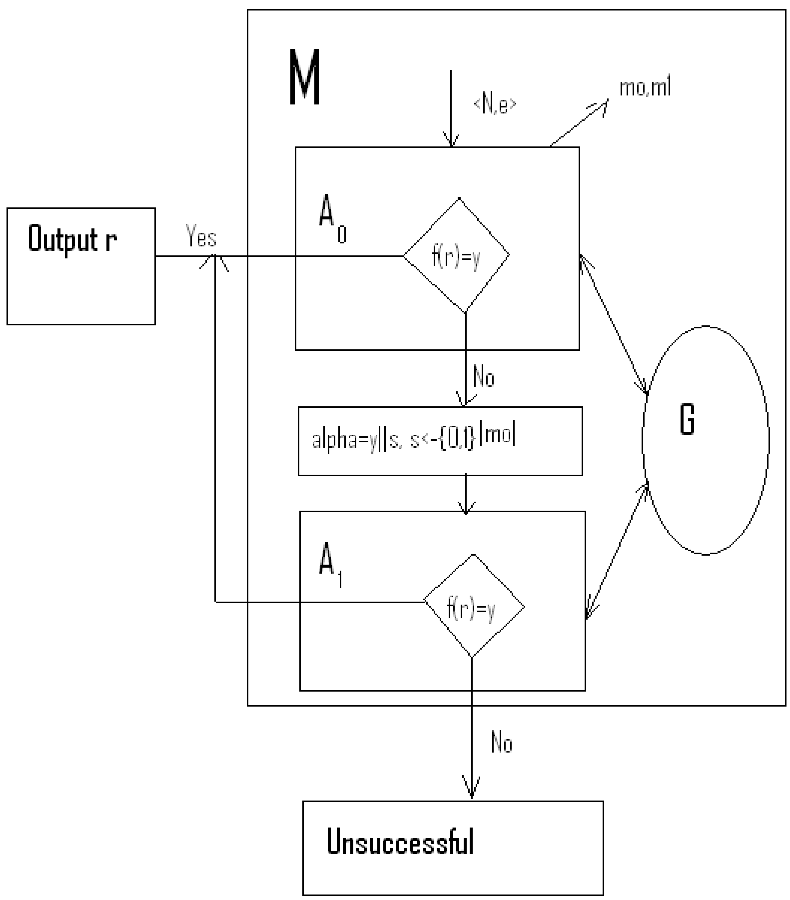
4. A random bit b$\leftarrow$ {0,1}and a random r$\leftarrow$ Z_n^* are chosen. A is given the cipher text <[r$^e modN, m_b$$\oplus$H(r)]>. the adversary can still query H(.).

5. Finally, A outputs b'. Pub$^{eav}_{A,\pi}$ returns 1, if $b = b'$, else 0 is returned

Proof for Encryption

the proof is by contradiction
suppose we have an adversary $A = (A_0, A_1)$ which is successful against our encryption scheme .Now we create a master algorithm M(f,d,y) such that $(f, f^{-1}, d) \leftarrow G(1^k); r \leftarrow d(1^k); y \leftarrow f(r)$, M is successful against our scheme
so,

$$E(x) = \{y \leftarrow f(r)\}||\{(f, f^{-1}, d) \leftarrow G(1^k)\}\oplus : r \leftarrow d(1^k)$$



1. A_0 simulates the oracle in natural way and samples $(m_0, m_1) \leftarrow A_0^G(E)$ if ever A_0 asks G an r such that $f(r) = y$, then M outputs r and halts, otherwise A_0 terminates after some polynomial number of queries and and M chooses $\alpha \leftarrow y||s$ for $s \leftarrow \{0,1\}^{|m_0|}$.

2. Then M simulates $A_1^G(E, m_0, m_1, \alpha)$, watching the oracle queries that A_1 makes to see if there is any oracle query r for which $f(r) = y$(.i.e. instead of feeding A_1 cipher text, it is asked $f(r)||s$ where $s \leftarrow \{0,1\}^{|m_0|}$). If there is M outputs r.

So, A_0 outputs m_0, m_1 and A_1 distinguishes between m_0, m_1
now define query as an event that at any point $A = (A_0, A_1)$ queries r to the RO (where r is the value used to generate the challenge, c).
$\therefore$

$$\begin{aligned}\Pr[\text{success}] &= \Pr[\text{success } \wedge \overline{Query}] + \Pr[\text{success} \wedge \text{Query}] \\ &< \Pr[\text{success } |\overline{Query}] + \Pr[\text{Query}]\end{aligned}$$

As $G(r)$ is random, if A does not query for r then

$$\Pr[\text{success}|\overline{Query}] \leq \frac{1}{2}$$

construct a reduction D, which takes as input $c_1 = r^e \bmod N$ and has to output r
This D randomly generates $c_2 \in \{0,1\}^{l(n)}$ and sends to A. A makes some queries to $G(.)$. D observes the queries and check if r_i^e mod$n$ $=c_1$. If a match occur then RSA is broken and our assumption becomes invalid
$\therefore$ Pr[Query] must be negligible.

Now define A_k as an event that A_1 asks query $r = f^{-1}(y)$

$$\frac{1}{2} + \epsilon(n) = \Pr[A \text{ succeeds}|A_k].\Pr[A_k] + \Pr[A \text{ succeeds}|\overline{A_k}].\Pr[\overline{A_k}]$$
$$\frac{1}{2} + \epsilon(n) \leq \Pr[A_k] + \Pr[A \text{ succeeds}|\overline{A_k}]$$
$$\frac{1}{2} + \epsilon(n) \leq \Pr[A_k] + \frac{1}{2}$$

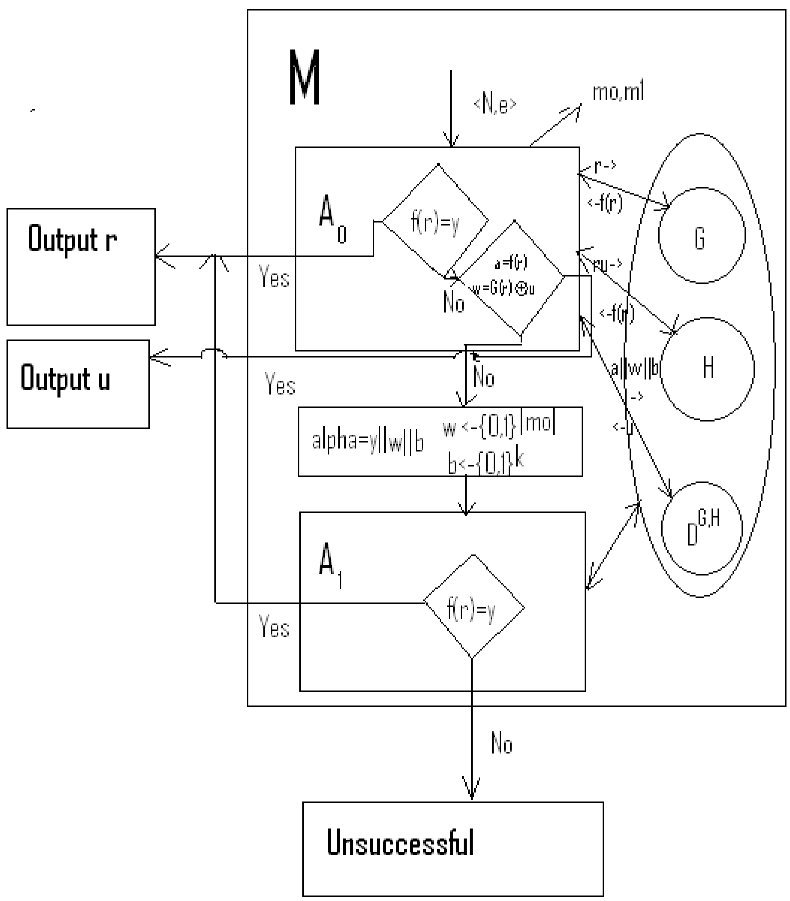
$\therefore$ $Pr[\text{A}_k]$ must be non-negligible, and M succeeds non-negligibly often in inverting f, which is not possible as per the concept of Trapdoor function. so, we arrive at a contradiction.
Hence, $E(x) = f(r)||G(x) \oplus x$ is a polynomially secure scheme against CPA

$E(x) = f(r)||G(x) \oplus x||H(rx)$ **is Secure against CCA**

suppose we have an adversary $A = (A_0, A_1)$ which is successful against our encryption scheme .Now we create a master algorithm M(f,d,y) such that $(f, f^{-1}, d) \leftarrow G(1^k); r \leftarrow d(1^k); y \leftarrow f(r)$, M is successful against our scheme

so,

$$E(x) = \{y \leftarrow f(r)\}||\{(f, f^{-1}, d) \leftarrow G(1^k)\}\oplus : r \leftarrow d(1^k)||H(rx)$$



1. A_0 simulates the 3 oracle namely $G, H, D^{g,h}$ in natural way and samples $(m_0, m_1) \leftarrow A_0^G(E)$ if ever A_0 asks G an r such that $f(r) = y$, then M outputs r and halts,otherwise A_0 returns a random string of the appropriate length, if ever A_0 asks H an rx such that $f(r) = y$, then M outputs r and halts,otherwise A_0 returns a random string of the appropriate length, if ever A_0 asks $D^{G,H}$ a $a||w||b$ such as $a = f(r), w = G(r) \oplus u$ (i.e it asks $f(r)||G(r) \oplus u||b$)when A_0 communicated with G, H for some query of r, ru then M outputs u ,otherwise M returns Invalid

2. Then M simulates $A_1^G(E, m_0, m_1, \alpha)$,where $\alpha = y||w||b$ for $w \leftarrow \{0,1\}^{|m_0|}, b \leftarrow \{0,1\}^k$, watching the oracle queries that A_1 makes to see if there is any oracle query r for which $f(r) = y$(.i.e. instead of feeding A_1 cipher text, it is asked $f(r)||b$ where $b \leftarrow \{0,1\}^{|m_0|}$). If there is M outputs r.

So, A_0 outputs m_0, m_1 and A_1 distinguishes between m_0, m_1

Proof for Encryption

the proof is by contradiction
Consider a successful adversary $A = (A_0, A_1)$ with the Pr[Success] $> 1/2 + \epsilon$
Define A_k : Event that A makes an oracle call at $G(r)$ or $H(ru)$
Define L_k : Event that $D^{G,H}$ is asked query for $a||w||b$, where

$$b = H(f^{-1}(a)||w \oplus G(f^{-1}(a))$$

note : decryption algo. is never asked query at the cipher text
$\therefore$

$$1/2+\epsilon < \Pr[A \text{ succeeds}|L_k].\Pr[L_k]+\Pr[A \text{ succeeds}|\neg L_k \wedge A_k].\Pr[\neg L_k \wedge A_k] + \Pr[A \text{ succeeds}|\neg L_k \wedge \neg A_k].\Pr[\neg L_k \wedge \neg A_k]$$

it is obvious that $\Pr[A \text{ succeeds } |\overline{L_k} \wedge \overline{A_k}] = 1/2$
if L_k is the total no. of queries then $\Pr[L_k] \leq n(k).2^{-k}$
$\therefore$
$1/2 + \epsilon < \Pr[L_k] + \Pr[A_k] + 1/2$
$\epsilon < n(k).2^{-k} + \Pr[A_k]$
$\therefore \Pr[A_k] > \epsilon - n(k).2^{-k}$
hence contradiction and the scheme is secure..