PKE

The RSA PKE has some drawbacks:

1) RSA assumption vs Factoring
   they're not equivalent

2) Provable security
   we can prove sec of RSA only for small messages

3) Only CPA security
   ideally we'd like security against CCA

All these limitations can be overcome!

1) We can get TDPs from factoring

2 & 3) We can get CPA/CCA from DDH

TDPs FROM FACTORING

Look at the following function

special case of modular exponentiation

$$f(x) = x^2 \bmod n \qquad n = p \cdot q$$

Problem: squaring is not a permutation, as the Image is a subset of $\mathbb{Z}_n^*$.

Anyways, for some parameters it is a permutation!

crt

By CRT, $x \rightarrow (x_p, x_q)$. Let's understand squaring mod p

Since $\mathbb{Z}_p^*$ is a cyclic group:

$$\mathbb{Z}_p^* = \{g^0, g^1, g^2, \dots, g^{\frac{(p-1)}{2}-1}, g^{\frac{(p-1)}{2}}, \dots, g^{p-2}\}$$

$$QR_p = \{g^0, g^2, g^4, \dots, g^{p-3}, g^0\}$$

↳ quadratic residues of p

Since $g^{(p-1)/2}$ is mapped to $g^0 = 1$,

$$\Rightarrow g^{(p-1)/2} = -1 \bmod p$$

$$\Rightarrow \# QRP = (p-1)/2$$

CLAIM: It is a permutation when

$p = 3 \bmod 4$ i.e. $p = 4t+3$ for some $t \in \mathbb{N}$

This can be inverted.

Because for

$$2t+2 = \frac{p-3}{2} + 2 = \frac{p+1}{2} = \frac{p-1}{2} + 1$$

$$(y^{t+1})^2 = y^{2t+2} = y^{\frac{p-1}{2}+1} = (x^2)^{\frac{p-1}{2}+1} = x^2$$

↳ this cancels out

$$\Rightarrow x = \pm y^{t+1} \bmod p$$

Obs: $-1 = g^{(p-1)/2} \notin QR_p$

Because $\frac{p-1}{2} = 2t+1$ which is <u>odd</u>, so $g^{\frac{p-1}{2}}$ is not a square

$$\Rightarrow y \in QR_p \iff -y \notin QR_p \text{ for } p = 3 \bmod 4$$

Consider again $f(x) = x^2 \bmod n$

$n = p \cdot q \quad p, q = 3 \bmod 4$ RABIN TDP

By CRT $x = (x_p, x_q) \mapsto (x_p^2, x_q^2)$

$$QR_n = \{ y \in \mathbb{Z}_n^2 : y = x^2 \bmod n \}$$

$$f^{-1}(y) \leftarrow \{ (x_p, x_q), (x_p, -x_q), (-x_p, x_q), (-x_p, -x_q) \}$$

It is easy to show that $y \in QR_n \iff y_p \in QR_p$
$y_q \in QR_q$

$\Rightarrow$ Only one square root is a quadratic residue, because only one of $(x_p, -x_p)$ or $(x_q, -x_q)$ is a QR.

$$\Rightarrow \# QR_n = \# \mathbb{Z}_n^* / 4 = \frac{\varphi(n)}{4}$$

<u>LEMMA</u>: Given x, z s.t. $x^2 \equiv z^2 \equiv y \bmod n$ with $x \neq \pm z$, then we can factor $n = p \cdot q$

<u>Proof</u>: By the fact that x, z are distinct

$$x + z \in \{ (0, 2x_q), (2x_p, 0) \}$$

WLOG $x + z = (2x_p, 0)$

$x + z \equiv 0 \bmod q$

$x + z \not\equiv 0 \bmod p$

$\Rightarrow \gcd(x+z, n) = q$ $\square$

<u>THM</u> Factoring $\Rightarrow$ Rabin's Function is a TDP

dato n ma non p e q non puoi invertire

assumiamo A sia un attaccante capace di invertire la permutazione senza conosce p e 4

Proof: Assume not: $\exists$ PPT $\mathcal{A}$ s.t.

$$\mathcal{A}(1^\lambda) \qquad\qquad\qquad \mathcal{C}(1^\lambda)$$

$$\xleftarrow{\quad n, y \quad} \qquad n = p \cdot q$$

$$p, q = 3 \bmod 4$$

$$\xrightarrow[\text{w.p.} \geq 1/\text{poly}(\lambda)]{\quad \pm x \vee \pm z \quad} \qquad y = x^2 \bmod n$$

$$x \text{ random}$$

Then $\exists$ PPT $\mathcal{A}'$ breaking factoring

$$\mathcal{A} \qquad\qquad \mathcal{A}' \qquad\qquad \mathcal{C}'$$

$$\xleftarrow{\quad n, y \quad} \qquad \xleftarrow{\quad n \quad} \qquad n = p \cdot q$$

$$x \leftarrow \$ \, \mathbb{Z}_n^*$$

$$y = x^2$$

$$\xrightarrow{\quad z \quad}$$

If $z \neq \pm x \Rightarrow \mathcal{A}'$ can factor $n = p \cdot q$

$$\Pr[\mathcal{A}' \text{ wins}] \geq \Pr[\mathcal{A} \text{ wins}] \cdot \Pr[\pm x \neq z]$$

$$\geq \frac{1}{2} \cdot \frac{1}{\text{poly}(\lambda)} = \frac{1}{\text{poly}(\lambda)} \quad \square$$

ELGAMAL PKE

How to do CPA PKE from DDH

$$\Pi = (\text{KGen}, \text{Enc}, \text{Dec})$$

$$KGen(1^\lambda) = (\mathbb{G}, g, q) \leftarrow \$ \, GroupGen(1^\lambda)$$

$$x \leftarrow \mathbb{Z}_q, \; h = g^x$$

Example: $\mathbb{G} = \mathbb{QR}_p$, $p = 2q+1$, $h = g^x \bmod q$

$PK = (params, h)$ OBSERVE: $h^r = g^{xr}$

$sk = x$ so (g^x, g^r, h^r) is DDH

$$Enc(pk, m \in \mathbb{G}) \qquad r \leftarrow \$ \, \mathbb{Z}_q$$

$$c = (g^r, h^r \cdot m)$$

$$= (c_1, c_2) \leftarrow \mathbb{G}^2$$

h^r è uil pad per mascherare il messaggio

$$Dec(sk, (c_1, c_2)) = \frac{c_2}{c_1^x} = \frac{h^r \cdot m}{(g^r)^x} = \frac{\cancel{h^r} \cdot m}{\cancel{(g^x)^r}} = m$$

$\hookrightarrow h$

THM: Above PKE is CPA secure assuming DDH

Proof: Make a GAME HOP

$$GAME_{\Pi, \mathcal{A}}(\lambda, b)$$

$\mathcal{A}(1^\lambda)$ $\qquad\qquad$ $\mathcal{C}(1^\lambda)$

$\xleftarrow{pk}$ $\qquad$ $pk = h, params$

$sk = x \quad h = g^x$

$\xrightarrow{m_0^*, m_1^*}$

$\xleftarrow{c^*}$ $\qquad$ $c^* = (g^r, h^r \cdot m_b^*)$

$$\xrightarrow{b'} \quad r \leftarrow \$ \mathbb{Z}_q$$

$HYB_{\Pi,A}(\lambda, b)$

$A(1^\lambda)$ $\qquad\qquad$ $C(1^\lambda)$

$$\xleftarrow{pk} \quad pk = h, \text{params}$$

$sk = x \quad h = g^x$

$$\xrightarrow{m_0^*, m_1^*}$$

$$\xleftarrow{c^*} \quad c^* = (g^r, g^z \cdot m_b^*)$$

$$\xrightarrow{b'} \quad z \leftarrow \$ \mathbb{Z}_p$$

LEMMA $\forall b$, $GAME(\lambda, b) \approx_c HYB(\lambda, b)$

Proof by reduction to DDH

LEMMA $HYB(\lambda, 0) \equiv HYB(\lambda, 1)$

By inspection, because

$$(g^x = h, g^r, g^z \cdot m_b^*)$$

$$\equiv (g^x, g^r, m \leftarrow \$ \mathbb{G}) \quad \square$$

Inheresting properties:

- Elgamal is HOMOMORPHIC:

Given pk and $(c_1, c_2) \to m$

$(c_1', c_2') \to m'$

$$\Rightarrow (c_1 \cdot c_1', c_2 \cdot c_2') \to m \cdot m'$$

$$(g^{r+r'}, h^{r+r'} \cdot m \cdot m') \equiv Enc(pk, m \cdot m')$$

- Re-randomizability

$$\text{Given } (pk, c_1, c_2) \to m$$
$$r \leftarrow \$ \mathbb{Z}_q$$
$$(c_1 \cdot g^{r'}, c_2 \cdot h^{r'}) =$$
$$= (g^{r+r'}, h^{r+r'} \cdot m) \equiv Enc(pk, m)$$

However, Elgamal is not CCA secure!
Holy grail in PKE: FULLY HOMOMORPHIC ENCRYPTION (FHE) which means it has to be homomorphic for every efficient function.

$$c \leftarrow \$ Enc(pk, m)$$
$$Eval(pk, f, c) \to c'$$
$$\text{s.t. } Dec(sk, c') = f(m)$$

Alice (not very comp. powerful) — Cloud

$$pk, sk \xrightarrow{\quad pk \quad} Eval\ c \to c'$$

$$x, c \leftarrow Enc(pk, x) \xleftarrow{\quad c' \quad}$$

$$f(x) = Dec(sk, c')$$

This is non trivial (complexity of Dec indep. of f)

FHE has been a problem for some ~35 years

2010 GENTRY (based on non-standard asumption)

Now: how to get CCA security?

CRAMER - SHOUP ENCRYPTION

Main idea: Start with Elgamal and augment it to achieve CCA security

$\Rightarrow CTX = (c, \pi)$

↳ think of it as a short "proof" that you know the message being encrypted

The verifier can check π given sk. If "wrong" outputs $\perp$.

Proof π reveals nothing on msg, and can't produce π without knowing msg

$\Pi = (\text{kgen}, \text{Enc}, \text{Dec})$ THIS IS C-S LITE

kgen(1^λ): $x_1, y_1, x_2, y_2 \leftarrow \$ \mathbb{Z}_q$

$(\mathbb{G}, g_1, g_2, q)$ g_1, g_2 are generators

$h_1 = g_1^{x_1} g_2^{y_1}$

$h_2 = g_1^{x_2} g_2^{y_2}$

$pk = (\text{params}, h_1, h_2)$

$sk = (x_1, y_1, x_2, y_2)$

$$\mathrm{Enc}(pk, m): \; r \leftarrow \$ \mathbb{Z}_q$$

$$c = (c_1, c_2, c_3, c_4)$$

$$c_3 = h_1^r \cdot m$$

$$c_1 = g_1^r$$

$$c_2 = g_2^r$$

$c_4 = h_2^r$ proof che conosciamo r oppure che il cypher e ben formato

$$\mathrm{Dec}(sk, c): \text{ IF } c_4 = c_1^{x_2} \cdot c_2^{y_2}$$

$$\text{OUTPUT} = \frac{c_3}{c_1^{x_1} \cdot c_2^{y_1}}$$

Else

OUTPUT $\perp$

CORRECTNESS : Holds because:

$$c_4 = h_2^r$$

$$c_1^{x_2} \cdot c_2^{y_2} = (g_1^r)^{x_2} \cdot (g_2^r)^{y_2}$$

$$= (g_1^{x_2} \cdot g_2^{y_2})^r = h_2^r \quad \checkmark$$

$$\frac{c_3}{c_1^{x_1} c_2^{y_1}} = \frac{h_1^r \cdot m}{(g_1^r)^{x_1} (g_2^r)^{y_1}}$$

$$= \frac{h_1^r \cdot m}{\underbrace{(g^{x_1} \cdot g^{y_1})^r}_{h_1^r}} = m$$

CCA-1 : Decryption queries are allowed before c^* is generated.

Not known if Elgamal is CCA-1.

<u>THM</u> : CS LITE is CCA-1 secure under DDH

Proof: First define GAME, then HYB

GAME params: $(\mathbb{G}, g_1, g_2, q)$

$$\begin{array}{lcl}
\mathcal{A}(1^\lambda) & & \mathcal{C}(1^\lambda) \\
 & \xleftarrow{\quad pk = (params, h_1, h_2) \quad} & x_1, y_1, x_2, y_2 \\
 & & h_1 = g_1^{x_1} g_2^{y_1} \\
\#poly & & h_2 = g_1^{x_2} g_2^{y_2} \\
 & \xrightarrow{\quad c = (c_1, \dots, c_4) \quad} & \text{IF } c_4 = c_1^{x_2} c_2^{y_2} \\
 & \xleftarrow{\quad m \quad} & \quad \text{output } m = c_3 / c_1^{x_1} c_2^{y_1} \\
 & & \text{ELSE } m = \perp \\
 & \xrightarrow{\quad m_0^*, m_1^* \quad} & r \leftarrow \$ \mathbb{Z}_q \\
 & & c_1^* = g_1^r \\
 & & c_2^* = g_2^r \\
 & & c_3^* = h_1^r m_b^* \\
 & \xleftarrow{\quad c^* = (c_1^*, \dots, c_4^*) \quad} & c_4^* = h_2^r \\
 & \xrightarrow{\quad b \quad} &
\end{array}$$

HYB params: $(\mathbb{G}, g_1, g_2, q)$

$\mathcal{A}(1^\lambda)$ $\qquad\qquad\qquad\qquad$ $\mathcal{C}(1^\lambda)$

$$\xleftarrow{\quad pk = (\text{params}, h_1, h_2) \quad}$$

x_1, y_1, x_2, y_2

$h_1 = g_1^{x_1} g_2^{y_1}$

$h_2 = g_1^{x_2} g_2^{y_2}$

#poly

$$\xrightarrow{\quad c = (c_1, \dots, c_4) \quad}$$

$$\xleftarrow{\quad m \quad}$$

$$\begin{cases} \text{IF } c_4 = c_1^{x_2} c_2^{y_2} \\ \text{output } m = c_3 / c_1^{x_1} c_2^{y_1} \\ \text{ELSE } m = \perp \end{cases}$$

$$\xrightarrow{\quad m_0^*, m_1^* \quad}$$

$r, r' \leftarrow\$ \mathbb{Z}_q \quad r \neq r'$

$c_1^* = g_1^r$

$c_2^* = g_2^{r'}$

$c_3^* = (g_1^r)^{x_1} \cdot (g_2^{r'})^{y_1} \cdot m_b^*$

$c_4^* = (g_1^r)^{x_2} \cdot (g_2^{r'})^{y_2}$

$$\xleftarrow{\quad c^* = (c_1^*, \dots, c_4^*) \quad}$$

$$\xrightarrow{\quad b \quad}$$

<u>LEMMA</u>: $\text{GAME}(\lambda, b) \approx_c \text{HYB}(\lambda, b) \quad \forall b \in \{0,1\}$

Proof: Reduction to DDH.

$$(\mathbb{G}, g, q) \Rightarrow (g^x, g^y, g^z) \approx_c (g^x, g^y, g^{xy})$$

$$\Rightarrow (\mathbb{G}, g_1, g_2, q)$$

$$g = g_1 \quad g_2 = g_1^\alpha \quad \alpha \in \mathbb{Z}_q$$

$g_3 = g_1^r$ 　　g4=g1^(alpha*r)=g2^r

DDH: (g_1, g_2, g_1^r, g_2^r)

So $\log_{g_1} g_3 = \log_{g_2} g_4$

NON-DDH g_1, g_2, g_3, g_4 　　come in hybrid

$= (g_1, g_2, g_1^r, g_2^{r'})$ 　　$r \neq r'$

$\Rightarrow$ This enables reduction to DDH

$\mathcal{A}$ 　　　　　　　　$\mathcal{A}_{DDH}$ 　　　　　　　　$\mathcal{C}_{DDH}$

$\mathcal{A} \xleftarrow{pk} \mathcal{A}_{DDH} \xleftarrow{g_1, g_2, g_3, g_4} \mathcal{C}_{DDH}$

DDH 　　NON-DDH

x_1, x_2

y_1, y_2

h_1, h_2

$\xrightarrow{c}$

$\xleftarrow{m, \perp}$

$\mathcal{A}_{DDH}$ knows sk

$\xrightarrow{m_0^*, m_1^*}$ use g_3, g_4

$\xleftarrow{c^*}$ 　x_1, x_2

To be continued...