

Corso di Crittografia

Prova in Itinere del 21 Giugno 2013

1. Si definisca formalmente il concetto di indistinguibilità `ind-id-cpa` per cifrari basati sull'identità.
2. Si consideri il seguente problema computazionale, che chiameremo Diffie-Hellman randomizzato (RDH), definito su un gruppo ciclico G avente ordine primo q . Sia g un generatore di G , definiamo il seguente esperimento

```
EspRDHG,g(A)
  a, b ←R ℤq*; g1 ← ga; g2 ← gb;
  (h, y) ← A(g1, g2);
  If (y = hab ∧ h ∈ G) return 1 else return 0
    // Si noti che h è un elemento random in G
```

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}^{\text{RDH}}_{G,g} = \Pr [\mathbf{Esp}_{G,g}^{\text{RDH}}(\mathcal{A}) = 1]$$

Dimostrare che il problema SDH non può essere più difficile del problema del computazionale Diffie Hellman in G . In altre parole, si dimostri che, se esiste un avversario \mathcal{B} capace di risolvere CDH in G , tale avversario può essere sfruttato per risolvere RDH in G .

3. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
4. Si dimostri che il seguente schema di firma non è sicuro. L'algoritmo di generazione delle chiavi prende in input un parametro k e sceglie sceglie quattro primi p, q, q', p' tali che $|p| = |q| = k/2$, $q = 2q' + 1$ e $p = 2p' + 1$. Supponiamo, inoltre che H sia una funzione hash (resistente alle collisioni) che prende in input elementi di lunghezza arbitraria e restituisce in output numeri primi di 256 bit. Lo spazio dei messaggi è $\mathcal{M} = \{(m_1, m_2) : m_1, m_2 \geq 2\}$. L'algoritmo, restituisce $VK = (N, H, \mathcal{M})$ come chiave pubblica e $SK = (p, q)$ come chiave privata.

Algoritmo di firma

```

Sign( $SK, (m_1, m_2)$ )
  If  $(m_1, m_2) \notin \mathcal{M}$  return  $(\perp, \perp)$ 
   $e_1 \leftarrow H(m_1); e_2 \leftarrow H(m_2);$ 
  Usando l'algoritmo esteso di Euclide calcola  $d_1, d_2$  tali che
     $d_1 e_1 \equiv 1 \pmod{\phi(N)}$ 
     $d_2 e_2 \equiv 1 \pmod{\phi(N)}$ 
   $R \leftarrow_R \mathbb{Z}_N^*;$ 
   $\sigma \leftarrow R^{d_1 d_2} \pmod{N};$ 
  return  $(R, \sigma)$ 

```

Algoritmo di verifica

```

Verify( $VK, (m_1, m_2), (R, \sigma)$ )
  If  $\sigma = \perp$  return 0;
   $e_1 \leftarrow H(m_1); e_2 \leftarrow H(m_2);$ 
  If  $\sigma^{e_1 e_2} = R \pmod{N}$  return 1
  else return 0

```

5. Si consideri il seguente seguente cifrario asimmetrico. L'algoritmo di generazione delle chiavi prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si considerino, inoltre, due gruppi G e G_T per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine p). Sia g un generatore di G . L'algoritmo procede scegliendo $x \in \{1, \dots, p-1\}$ (a caso) e ponendo $h = g^x$. La chiave pubblica è quindi $PK = (h, g, p, e, G, G_T)$, mentre la chiave privata è $SK = x$

L'algoritmo di cifratura funziona nel seguente modo

```

Enc( $PK, m$ )
  if  $(m \notin \mathcal{M})$  return  $\perp$ 
   $r, s \leftarrow_R \{1, \dots, p-1\}$ 
   $C_1 \leftarrow g^r; C_2 \leftarrow g^s; C_3 \leftarrow e(g, h)^{rs} \cdot m;$ 
  Return  $C = (C_1, C_2, C_3);$ 

```

L'algoritmo di decifratura, invece, opera come segue

```

Dec( $SK, C = (C_1, C_2, C_3)$ )
   $D \leftarrow C_1^x \pmod{N}$ 
   $m \leftarrow C_3 \cdot e(D, C_2)^{-1} \pmod{N}$ 
  Return  $m.$ 

```

Dimostrare che tale cifrario non è sicuro in senso IND-CCA.

M°1

SIA IBE = (Setup, KeyDer, Enc, Dec) UNO SCHEMA DI CIFRATURA BASATO SULL'IDENTITÀ E SIANO:

- Setup: LA FUNZIONE CHE RESTITUISCE UNA MASTER SECRET KEY, FORNITA DALL'ORGANIZZAZIONE CHE SI OCCUPA DELLA GENERAZIONE DELLE CHIAVI, E DEI PARAMETRI PUBBLICI, PK.
- KeyDer: LA FUNZIONE CHE UTILIZZANDO msk E I PARAMETRI PUBBLICI GENERA UNA SK_{ID} A PARTIRE DA UNA IDENTITÀ ID.
- Enc: LA FUNZIONE DI CIFRATURA DI UN MESSAGGIO CHE UTILIZZA L'IDENTITÀ ID.
- Dec: CHE EFFETTUÀ UNA DECIFRATURA DI UN CRITTOTESTO UTILIZZANDO LA CHIAVE SEGRETA SK_{ID}.

SIA A UN AVVERSARIO CHE TENTA DI GENERARE UNA SUA IDENTITÀ ID* DIVERSA DA QUELLA CHE GIÀ DETIENE, PROVANDO A GENERARE UNA CHIAVE SEGRETA.

SI CONSIDERINO I SEGUENTI ESPERIMENTI:

$\text{Exp}^{\text{ind}-\text{id}-\text{cpa}-1}(A)$:

$$\begin{aligned} & (\text{pk}, \text{msk}) \leftarrow \text{Setup} \\ & b \leftarrow A^{\text{Enc}_{\text{pk}}(\text{id}^*, \text{LR}(\cdot, \cdot, 1))}, \text{KeyDer}_{\text{msk}}(\cdot) \end{aligned}$$

if A IMBROGLIA RETURN 0

ELSE RETURN b

$\text{Esp}^{\text{ind-id-CPA-0}}(A)$:

$(pk, msk) \leftarrow \text{Setup}$

$b \leftarrow A^{\text{Enc}(id^*, LRC(\cdot, \cdot, 0))}, \text{keyDer}_{msk}(\cdot)$

if A INBROGLIA RETURN 0

ELSE RETURN b

IN ENTRAMBI GLI ESPERIMENTI INIZIALMENTE VIENE

RICHIAMATA LA FUNZIONE Setup .

L'AVVERSARIO A HA ACCESSO A DUE ORACOLI: UNO DI CIFRATURA E UN ORACOLO keyDer .

L'ORACOLO O_{enc} PRENDE 3 PARAMETRI: ID , m_0 , m_1 ;
L'ORACOLO O_{keyDer} PRENDE IN INPUT UNA IDENTITÀ $ID \neq id^*$
E RESTITUISCE LA CHIAVE SEGRETA SK_{ID} .

L'AVVERSARIO PUÒ ESEGUIRE UN NUMERO ILLIMITATO DI RICHIESTE
ALL'ORACOLO keyDer , MENTRE PUÒ EFFETTUARE UNA
SOLO RICHIESTA ALL'ORACOLO DI CIFRATURA.

L'OBBIETTIVO DELL'AVVERSARIO È QUELLO DI
CAPIRE IN QUALE DEI DUE ESPERIMENTI SI TROVA.

IL SUO VANTAGGIO È DEFINITO COME SEGUENTE:
 $\text{Adv}^{\text{ind-id-CPA}}(A) = |\Pr[\text{Exp}^{\text{ind-id-CPA-1}}(A) = 1] +$
 $- \Pr[\text{Exp}^{\text{ind-id-CPA-0}}(A) = 1]|$

DICIAMO CHE IBE È INDISTINGUIBILE CONTRO ATTACCHI
A MESSAGGIO SCELTO NEL MODELLO IDENTITY BASED SE
IL VANTAGGIO DI OGNI AVVERSARIO POLINOMIALMENTE
ILLIMITATO È PROSSIMO A ZERO.

m°2

SIA G CICLICO , $q = |G| = \phi(G)$, $g \in G$ GENERATORE
DIMOSTRARE CHE RDH NON PUÒ ESSERE PIÙ DIFFICILE DI CDH.

SIA B UN AVVERSARIO IN GRADO DI RISOLVERE CDH IN G .
COSTRUIAMO UN AVVERSARIO A CHE SFRUTTA B PER
RISOLVERE RDH IN G .

$A(g_1, g_2)$:

$$X \leftarrow g_1;$$

$$Y \leftarrow g_2;$$

$$\gamma \leftarrow B(X, Y); // Z = g^{ab}$$

$$h \leftarrow g;$$

return (h, γ)

h È UN ELEMENTO RANDOM DI G , ESSENDO g UN
GENERATORE DI G , QUELLO CHE DEVE SUCCEDERE È CHE
 $g^{ab} = h^{ab}$ E QUINDI $g = h$.

$$\text{Adv}_{G,g}^{\text{RDH}} = \Pr[\text{ESPG}_{g,g}^{\text{RDH}}(A) = 1] = \text{Adv}_{G,g}^{\text{CDH}} \Rightarrow \text{TESI}$$

m°3

SIA DS = (KeyGen, Sign, VF) UNO SCHEMA DI FIRMA DIGITALE TALE CHE:

- KeyGen : SIA L'ALGORITMO RANDOMIZZATO DI GENERAZIONE DELLA CHIAVE CHE PRENDE NESSUN INPUT E RITORNA UNA COPPIA (vk, sk) DI CHIAVI, RISPECTIVAMENTE LA CHIAVE PUBBLICA (VERIFY KEY) E LA CORRISPONDENTE CHIAVE PRIVATA (Sign key).
- Sign: SIA L'ALGORITMO DI FIRMA CHE PRENDE IN INPUT UNA CHIAVE SEGRETA SK ED UN MESSAGGIO m E RITORNA UNA FIRMA $\{0,1\}^*$ $\cup \{\perp\}$.
- VF: SIA L'ALGORITMO DI VERIFICA CHE PRENDE IN INPUT LA CHIAVE PUBBLICA VK, UN MESSAGGIO m E UNA FIRMA CAN DI DATA o PER m E RITORNA UN bit $b = \{0, 1\}$.

SIA A UN AVVERSARIO IL COI OBIETTIVO E QUELLO DI PRODURRE UNA COPPIA (π, α) TALE CHE $\text{VF}(vk, \pi, \alpha)$ PRODUCA 1 E π NON STA MAI STATO DATO IN INPUT ALL'ORACOLO DI FIRMA, QUINDI A PRODUCE UN FALSO.

SI CONSIDERI IL SEGUENTE ESPERIMENTO :

$\text{Esp}_{\text{DS}}^{\text{uf-cma}}(A)$:

$$\begin{aligned} (vk, sk) &\leftarrow \text{KeyGen} \\ (\pi, \alpha) &\leftarrow A^{\text{Sign}_{sk}(\cdot)} \end{aligned}$$

if ACCADE CHE :

- $\text{VF}_{vk}(\pi, \alpha) = 1$
- π È MESSAGGIO(VK)
- π NON RICHIESTO A OSIGN

RETURN 1

ELSE :

RETURN 0

IL VANTAGGIO uf - CMA DELL'AVVERSARIO A È DEFINITO
COME SEQUE:

$$\text{Adv}_{\text{DS}}^{\text{uf-CMA}}(A) = \Pr[\text{Exp}_{\text{DS}}^{\text{uf-CMA}}(A) = 1]$$

DIREMO CHE DS È NON FALSIFICABILE CONTRO ATTACCHI
A MESSAGGIO SCELTO SE IL VANTAGGIO DI OGNI AVVERSARIO
POLINOMIALMENTE LIMITATO È PROSSIMO A ZERO.

m° 4

$$\text{KeyGen}(K) := (p, q, p', q') \quad |p| = |q| = K/2$$
$$q = 2q' + 1$$
$$p = 2p' + 1$$

SIA H CR2-UNA TALE CHE $H: \{0,1\}^* \rightarrow \{\text{PRIMI DA 256 bit}\}$

$$H = \{(m_1, m_2): m_1, m_2 \geq 2\}$$

VK = (N, H, H) COME CHIAVE PUBBLICA
SK = (p, q) COME CHIAVE PRIVATA

SUPPONIAMO CHE $m_1 = 2, m_2 = 3$ ALLORA:

$$\text{Sign}_{\text{SK}}(m_1, m_2) = (R, \alpha) = (R, R^{d_1 d_2}) \quad \text{TACE CHE:}$$

$$d_1 \cdot e_1 \equiv 1 \pmod{\phi(N)} \quad d_2 \cdot e_2 \equiv 1 \pmod{\phi(N)}$$

$$e_1 = H(m_1) \quad e_2 = H(m_2)$$

A QUESTO PUNTO POSSIAMO CONSIDERARE:

$$\text{Sign}_{\text{SK}}(m_2, m_1) = (R, \alpha) = (R, R^{d_2 d_1})$$

POSSIAMO OSSERVARE CHE $(m_1, m_2) \neq (m_2, m_1)$ MA LE FIRME SONO UGUALI.

FORMACCIAMO L'AVVERSARIO:

$A(VK):$

$$m_1 \leftarrow 2;$$

$$m_2 \leftarrow 3;$$

$$(R, \alpha) \leftarrow \text{Sign}_{\text{SK}}(m_1, m_2)$$

$$\text{VF}(VK, (m_2, m_1), (R, \alpha))$$

$$\text{Adv}_{\text{DS}}^{\text{uf-�Oma}}(A) = 1$$

m° 5

- KeyGen(κ) := $|P| = \kappa$

$G \in Gr$ $e: G \times G \rightarrow Gr$ $|G| = |Gr| = p$

$g \in G : g \in \text{GenSet}(G)$

$x \leftarrow \{1, \dots, p-1\}$ $h = g^x$

$PK = (h, g, p, e, G, Gr)$

$SK = x$

CONSIDERIAMO $m_0, m_1 \in \text{PlainText}(pn) = M$

$\text{Enc}(PK, m_0) = (g^r, g^s, e(g, h)^{rs} \cdot m_0) = (c_1, c_2, c_3)$

A QUESTA PONTE POSSO CONSIDERARE $C' = (c_2, c_1, c_3)$:

$\text{Dec}(SK, C') = e(g, h)^{rs} \cdot m_0 \cdot e(g^{sx}, g^r)^{-1} =$
 $= e(g, g)^{xs} \cdot e(g, g)^{rsx} \cdot m_0 = m_0$

FORMALIZZIAMO L'AVVERSARIO:

A(AE):

$m_0, m_1 \in M$;

$C \leftarrow O_{\text{ENC}_{PK}}(m_0, m_1); // C = (c_1, c_2, c_3)$

$C' \leftarrow (c_2, c_1, c_3);$

$m_3 \leftarrow O_{\text{DEC}_{SK}}(C');$

if ($m_3 == m_2$) return 1

else return 0

$\text{Adv}_{AE}^{\text{ind-cca}}(A) = 1 - 0 = 1$

Corso di Crittografia

Prova del 30 Gennaio 2018

1. Si definisca formalmente il concetto di indistinguibilità contro attacchi a messaggio scelto **ind-id-cpa** per cifrari basati sull'identità.
2. Si fornisca lo pseudo-codice e si spieghi il funzionamento dell'algoritmo Square and Multiply.
3. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si considerino, inoltre, due gruppi G e G_T per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine p). Sia g un generatore di G . L'algoritmo procede scegliendo $x_0, x_1, \dots, x_n \in \{1, \dots, p - 1\}$ (a caso) e ponendo $h_i = g^{x_i}$ per $i = 0, \dots, n$. La chiave pubblica è quindi $VK = (h_0, \dots, h_n, g, e, G, G_T)$, mentre la chiave privata è $SK = (x_0, \dots, x_n)$. Lo spazio dei messaggi è $\mathcal{M} = \{0, 1\}^n - 0^n$ (il messaggio nullo non è ammesso).

Algoritmo di firma

```
Sign( $SK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
  Sia  $m = m_1 \dots m_n$  // Gli  $m_i$  sono i bit di  $m$ 
   $y \leftarrow x_0(\sum_{i=1}^n x_i m_i) \bmod p$ ;  $\sigma = g^y$ 
  return  $(\sigma)$ 
```

Algoritmo di verifica

```
Verify( $VK, m, (\sigma)$ )
  If  $\sigma \notin G$  return 0
  If  $e(\sigma, g) = e(\prod_{i=0}^n h_i^{m_i}, h_0)$  return 1
  else return 0
```

4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si consideri il seguente problema computazionale, che chiameremo Cube-Bilinear-Diffie-Hellman (CBDH), definito su un gruppo ciclico G avente ordine primo q ,

per il quale è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (anche G_T è supposto avere ordine q). Sia g un generatore di G , definiamo il seguente esperimento

$$\begin{aligned} & \mathbf{Esp}_{G,g}^{\text{CBDH}}(\mathcal{B}) \\ & a \leftarrow_R \mathbb{Z}_q^*; g_1 \leftarrow g^a; \\ & y \leftarrow \mathcal{B}(g_1); \\ & \text{If } (y = e(g, g)^{a^3}) \text{ return 1 else return 0} \end{aligned}$$

Il vantaggio di \mathcal{B} è definito come

$$\mathbf{Adv}_{G,g}^{\text{CBDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{CBDH}}(\mathcal{B}) = 1]$$

Dimostrare che il problema (computazionale) Bilineare Diffie Hellman (BDH) non può essere più difficile del problema CBDH in G . In altre parole, si dimostri che, se esiste un avversario \mathcal{B} capace di risolvere CBDH in G , tale avversario può essere sfruttato per risolvere BDH.

Per completezza ricordiamo che il problema BDH è definito come segue

$$\begin{aligned} & \mathbf{Esp}_{G,g}^{\text{BDH}}(\mathcal{A}) \\ & a, b, c \leftarrow_R \mathbb{Z}_q^*; \\ & y \leftarrow \mathcal{A}(g^a, g^b, g^c); \\ & \text{If } (y = e(g, g)^{abc}) \text{ return 1 else return 0} \end{aligned}$$

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{BDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{CBDH}}(\mathcal{A}) = 1]$$

// Suggerimento: Si usi l'avversario B più volte, ricordando la formula $(a + b + c)^3 = (a + b)^3 + (a + c)^3 + (b + c)^3 - a^3 - b^3 - c^3 + 6abc$

m° 1

SIA $\text{IBE} = (\text{Setup}, \text{KeyDer}, \text{Enc}, \text{Dec})$ UNO SCHEMA DI CIFRATURA BASATO SULL'IDENTITÀ TALE CHE:

- Setup : SIA L'ALGORITMO DI GENERAZIONE DEI PARAMETRI PUBBLICI PK E RESTITUISCE MSK , LA MASTER SECRET KEY;
 - KeyDer : SIA L'ALGORITMO CHE PRENDE IN INPUT LA msh E I PARAMETRI PUBBLICI PK PER GENERARE UNA SKID A PARTIRE DA UNA IDENTITÀ ID .
 - Enc CHE EFFETTUÀ LA CIFRATURA DI UN MESSAGGIO UTILIZZANDO UNA IDENTITÀ ID .
 - Dec CHE EFFETTUÀ UNA DECIFRATURA DI UN CRITTOTESTO UTILIZZANDO LA CHIAVE SEGRETA SKID .
- SIA A UN AVVERSARIO CHE DETIENE UN CERTO NUMERO DI IDENTITÀ ID , IL SUO OBIETTIVO È QUELLO DI UTILIZZARE UNA IDENTITÀ ID^* DIVERSA DA QUELLA CHE GIÀ DETIENE, CON LA QUALE CIFRA UN MESSAGGIO UTILIZZANDO UN ORACOLO DI CIFRATURA $\text{Oenc}(\text{ID}^*, m_0, m_1)$ CHE RITORNA IL CRITTOTESTO DI UNO DEI DUE MESSAGGI. SUCCESSIVAMENTE, L'AVVERSARIO PUÒ UTILIZZARE L'ORACOLO Onyder CHE PRENDE IN INPUT $\text{ID} \neq \text{ID}^*$ E RESTITUISCE SKID .

SI CONSIDERINO I SEGUENTI ESPERIMENTI:

$\text{ESP}_{\text{IBE}}^{\text{ind-id-CPA-1}}(A)$:

$(pk, msu) \leftarrow \text{Setup}();$

$b \leftarrow A^{\text{Enc}_{pk}(ID^*, \text{LR}(\cdot, \cdot, 1)), \text{KeyDer}_{msu}(\cdot)}$;

if A IMBROGLIA RETURN 0

ELSE RETURN 1

$\text{ESP}_{\text{IBE}}^{\text{ind-id-CPA-0}}(A)$:

$(pk, msu) \leftarrow \text{Setup}();$

$b \leftarrow A^{\text{Enc}_{pk}(ID^*, \text{LR}(\cdot, \cdot, 0)), \text{KeyDer}_{msu}(\cdot)}$;

if A IMBROGLIA RETURN 0

ELSE RETURN 1

DIRETTO CHE A IMBROGLIA SE EFFETTUÀ UNA CHIAMATA
ALL'OPACOLO KeyDer_2 UTILIZZANDO D^* .

IL VANTAGGIO DELL'AVVERSARIO È DEFINITO COME:

$$\text{Adv}_{\text{IBE}}^{\text{ind-id-CPA}}(A) = |\Pr[\text{ESP}_{\text{IBE}}^{\text{ind-id-CPA-1}}(A) - \Pr[\text{ESP}_{\text{IBE}}^{\text{ind-id-CPA-0}}(A)]|$$

m^2

SQUARE-AND-MULTIPLY(x, e):

$k = \text{bitlen}(e)$

$d = 1$

for $i = k-1$ down to 0 do:

$d = d^2 \bmod N$

if ($e.\text{bit}[i] == 1$):

$d = (d * x) \bmod N$

return d

SIA $x \in \mathbb{Z}_N^*$ LO SCOPO È QUELLO DI CALCOLARE $x^e \bmod N$. SI CONSIDERI LA RAPPRESENTAZIONE BINARIA DI e :

$$e = e_k \cdot 2^k + e_{k-1} \cdot 2^{k-1} + \dots + e_0$$

QUINDI SI HA:

$$\begin{aligned} x^e \bmod N &= x^{e_k \cdot 2^k + e_{k-1} \cdot 2^{k-1} + \dots + e_0} \bmod N \\ &= (x^{2^k})^{e_k} \cdot (x^{2^{k-1}})^{e_{k-1}} \cdots (x)^{e_0} \bmod N \end{aligned}$$

L'ALGORITMO CONSISTE NEL CALCOLARE SINGOLARMENTE I FATTORI MOLTIPLICATIVI NEL SEGUENTE MODO:

• SI CONVERTE L'ESPOLENTE e IN BINARIO E SI ANALIZZA
NO ITERATIVAMENTE I BIT PARTENDO DAL MSB:

- SE È IL MSB ALLORA AGGIUNGIAMO x E BASTA;
- SE IL BIT È 0 SI EFFETTUÀ UN QUADRATO;
- SE IL BIT È 1 SI EFFETTUÀ UN QUADRATO E UNA
MOLTIPLICAZIONE PER x .

m°3

$$VK = (h_0, \dots, h_m, g, e, G, G_T)$$

$$SK = (x_0, \dots, x_n)$$

$$H = \{0, 1\}^m - \{0^m\}$$

CONSIDERIAMO m=2 ALLORA H = {01, 10, 11}

CONSIDERIAMO m=01

Sign(SK, 01) :

$$\begin{aligned} y &\leftarrow x_0(x_1 \cdot 0 + x_2 \cdot 1) = x_0 \cdot x_2 \\ o &\leftarrow g^{x_0 \cdot x_2} \end{aligned}$$

Verify(VK, 01, $g^{x_0 \cdot x_2}$):

$$e(g^{x_0 \cdot x_2}, g) = e(g^{x_2}, g^{x_0}) = e(g, g)^{x_0 \cdot x_2}$$

$$\text{Sign}(SK, 10) = g^{x_0 \cdot x_2} =$$

$$\text{Sign}(SK, 11) = g^{x_0 \cdot x_1 + x_0 \cdot x_2}$$

QUELLO CHE SI PUÒ NOTARE È CHE È POSSIBILE EFFETTUARE UNA MOLTIPLICAZIONE TRA LE FIRME DI 01 e 10 PER OTTENERE QUELLA DI 11.

FORMALIZZIAMO L'AVVERSARIO:

A(VK):

$$m_0 \leftarrow 01;$$

$$m_1 \leftarrow 10;$$

$$m_2 \leftarrow 11;$$

$$\sigma_0 \leftarrow \text{Sign}_{SK}(m_0); // \sigma_0 = g^{x_0 \cdot x_2}$$

$$\sigma_1 \leftarrow \text{Sign}_{SK}(m_1); // \sigma_1 = g^{x_0 x_1}$$

$$\sigma_3 \leftarrow \sigma_1 \cdot \sigma_2; // \sigma_3 = g^{x_0 x_1 + x_0 x_2}$$

$$VF(VK, m_3, \sigma_3)$$

m°4

SIA DS = (KeyGen, Sign, VF) UNO SCHEMA DI FIRMA DIGITALE
TALE CHE:

- KeyGen SIA L'ALGORITMO DI GENERAZIONE DELLE CHIAVI VK E SK;
- Sign SIA L'ALGORITMO DI FIRMA CHE PRENDE IN INPUT UNA CHIAVE SEGRETA SK ED UN MESSAGGIO m E RITORNA UNA FIRMA $\sigma \in \{0, 1\}^* \cup \{\perp\}$.
- VF È L'ALGORITMO DI VERIFICA CHE PRENDE IN INPUT LA CHIAVE PUBBLICA VK, UN MESSAGGIO m E UNA FIRMA σ E RITORNA UN BIT $b \in \{0, 1\}$.

SIA A UN AVVERSARIO CHE TENTA DI PRODURRE UN FALSO, OVVERO UNA COPPIA (M, σ) TALI CHE $VF(VK, M, \sigma) = 1$

SI CONSIDERI IL SEGUENTE ESPERIMENTO:

$\text{Esp}_{DS}^{\text{inf.-com}}(A)$:

$$(VK, SK) \leftarrow \text{KeyGen}$$

$$(M, \sigma) \leftarrow A^{\text{Sign}_{SK}(\cdot)}$$

if ACCADE CHE:

- $m \in \text{MESSAGES}(VK)$

- $VF(VK, m, \sigma) = 1$

\bar{m} non è stato chiesto a sign
 return_2

ELSE:
RETURN 0

L'AVVERSARIO A PUÒ EFFETTUARE DELLE QUERY ALL'ORACOLO DI FIRMA Osigns_n.

IL VANTAGGIO OF-CMA DI A È DEFINITO COME:

$$\text{Adv}_{\text{DS}}^{\text{nf-Cma}}(A) = \Pr_{\text{r}} [\text{ESP}_{\text{DS}}^{\text{nf-Cma}}(A) = 1]$$

DIREMO CHE DS È NON FALSIFICABILE CONTRO ATTACCHI A NESSAGGIO SCELTO SE IL VANTAGGIO DI OGNI AVVERSARIO POLINOMICAMENTE LIMITATO È PROSSIMO A ZERO.

m° 5

$$CBDH \quad G \quad \phi(G) = q \quad g \in G$$

B RISOLVERE CBDH

CREARE A CHE SFRUTTA B E RISOLVE BDH.

$$B(g^a) = e(g, g)^{a^3} = t^{a^3}$$

$$B(g^b) = e(g, g)^{b^3} = t^{b^3}$$

$$B(g^c) = e(g, g)^{c^3} = t^{c^3}$$

$$B(g^{a+b+c}) = e(g, g)^{(a+b+c)^3} = t^{(a+b+c)^3}$$

$$B(g^{a+b}) = e(g, g)^{(a+b)^3} = t^{(a+b)^3}$$

$$B(g^{a+c}) = e(g, g)^{(a+c)^3} = t^{(a+c)^3}$$

$$B(g^{b+c}) = e(g, g)^{(b+c)^3} = t^{(b+c)^3}$$

PER CUI SI HA:

$$\frac{(t^{(a+b)^3} + t^{(a+c)^3} + t^{(b+c)^3} - a^3 - b^3 - c^3 + 6abc \cdot t^{a^3} \cdot t^{b^3} \cdot t^{c^3})}{t^{(a+b)^3} \cdot t^{(a+c)^3} \cdot t^{(b+c)^3}} =$$

$$= t^{6abc} = e(g, g)^{6abc}$$

$$\sqrt[6]{t^{6abc}} = e(g, g)^{abc} \quad \checkmark$$

Corso di Crittografia

Prova in Itinere del 30 Gennaio 2017

1. Si definisca formalmente il concetto di indistinguibilità contro attacchi a messaggio scelto **ind-cpa** per cifrari asimmetrici.
2. Si fornisca lo pseudo-codice e si spieghi il funzionamento dell'algoritmo Baby-Step Giant-Step (algoritmo di Shanks).
3. Si consideri il seguente cifrario asimmetrico.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave è simile a quello della funzione RSA. Esso restituisce in output un modulo $N = pq$ ed un intero $e > 2^{60}$, tale che $\gcd(e, \phi(N)) = 1$; la chiave segreta è d tale che $ed = 1 \pmod{\phi(N)}$, lo spazio dei messaggi è \mathbb{Z}_N^* .

Algoritmo di cifratura. L'algoritmo riceve in input un messaggio m

```
Enc( $N, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \mathbb{Z}_N^*$ ;  $C_1 \leftarrow r^e \pmod{N}$ ;
   $C_2 \leftarrow r^{e+1}m \pmod{N}$ ;
  return  $(C_1, C_2)$ 
```

Algoritmo di decifratura. L'algoritmo di decifratura è il seguente.

```
Dec( $SK, C_1, C_2$ )
   $r \leftarrow C_1^d \pmod{N}$ ;
   $m \leftarrow C_2 / (r^{e+1}) \pmod{N}$ ;
  return  $m$ 
```

Dimostrare che tale cifrario non è sicuro in senso ind-cca.

4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si consideri il seguente problema computazionale, che chiameremo Diffie-Hellman Esteso (EDH), definito su un gruppo ciclico G avente ordine primo q , per il quale è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (anche G_T è supposto avere ordine q). Sia g un generatore di G , definiamo il seguente esperimento

```

EspG,gEDH( $\mathcal{A}$ )
   $a, b, c, d \leftarrow_R \mathbb{Z}_q^*$ ;  $g_1 \leftarrow g^a$ ;  $g_2 \leftarrow g^b$ ,  $g_3 \leftarrow g^c$ ;  $g_4 \leftarrow g^d$ ;
   $y \leftarrow \mathcal{A}(g_1, g_2, g_3, g_4)$ ;
  If ( $y = e(g, g)^{abcd}$ ) return 1 else return 0

```

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{EDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{EDH}}(\mathcal{A}) = 1]$$

Dimostrare che il problema EDH non può essere più difficile del problema computazionale Diffie Hellman in G . In altre parole, si dimostri che, se esiste un avversario \mathcal{B} capace di risolvere CDH in G , tale avversario può essere sfruttato per risolvere EDH.

M° 3

SIA $AE = (\text{KeyGen}, \text{Enc}, \text{Dec})$ UNO SCHEMA DI CIFRATURA ASIMMETRICA TALE CHE:

- KeyGen : DEFINISCE L'ALGORITMO DI GENERAZIONE DELLE CHIAVI PK e SK ;
- Enc : DEFINISCE L'ALGORITMO DI CIFRATURA $\text{Enc}(PK, m) = c_0 \perp$
- Dec : DEFINISCE L'ALGORITMO DI DECIFRATURA $\text{Dec}(SK, c) = m \perp$

SIA A UN AVVERSARIO CON ACCESSO A DUE ORACOLI:

- Lr-encryption ;
- DECIFRATURA.

SI CONSIDERINO I SEGUENTI ESPERIMENTI:

$\text{ESP}_{AE}^{\text{ind}-\text{CCA}}(A)$:

$(PK, SK) \leftarrow \text{KeyGen}$;
 $b \leftarrow A^{\text{Enc}_{PK}(\text{LRC}, \cdot, 1)), \text{Dec}_{SK}(\cdot)}$,
if A IMBROGLIA: return 0;
else return b;

$\text{ESP}_{AE}^{\text{ind}-\text{CCA}^0}(A)$:

$(PK, SK) \leftarrow \text{KeyGen}$;
 $b \leftarrow A^{\text{Enc}_{PK}(\text{LRC}, \cdot, 0)), \text{Dec}_{SK}(\cdot)}$,
if A IMBROGLIA: return 0;
else return b;

$\text{Adv}_{AE}^{\text{ind}-\text{CCA}}(A) = [\Pr[\text{ESP}_{AE}^{\text{ind}-\text{CCA}}(A) = 1] - \Pr[\text{ESP}_{AE}^{\text{ind}-\text{CCA}^0}(A) = 1]]$

DIREMO CHE AE GARANTISCE SICUREZZA RELATIVAMENTE AD ATTACCHI A CRIPTOTESTO SELEZIONATO SE $\forall A$ p.l. $\text{Adv}(A) \approx 0$.

m°2

SI CONSIDERI UN GRUPPO G, IL SUO ORDINE È $|G|=m$ E $g \in G$ È UN SUO GENERATORE. SI CONSIDERI $X \in G$ TALE CHE $X = g^x$ E CON TALE ALGORITMO SI VOGLIE DETERMINARE IL LOGARITMO DISCRETO DI X ovvero:

$$D\log_{G,g}(X) = D\log_{G,g}(g^x) = x$$

Lo PSEUDOCODICE È IL SEGUENTE:

BABY-STEP-GIANT-STEP(X):

$$m = \lceil \sqrt{m} \rceil;$$

for $b=0$ to n do: $B[Xg^{-b}] = b$;

for $a=0$ to m do :

$$Y = g^{ma};$$

if $B[Y]$ is not empty :

$$x_0 = B[Y]$$

$$x_1 = a;$$

return $m x_1 + x_0$;

L'ALGORITMO DETERMINA $m = \lceil \sqrt{m} \rceil$ E $0 \leq x_0, x_1 \leq m$ TACI CHE $x = m x_1 + x_0$. PER FARE CIÒ SI CONSIDERA UNA LISTA B E SAPENDO CHE $g^x = g^{m x_1 + x_0} \Rightarrow X g^{-x_0} = g^{m x_1}$ ALLORA:

- IN $B[Xg^{-b}]$ VENGONO SALVATI I VALORI b_i CON $i \in \{0, \dots, m\}$
- NEL SECONDO CICLO FOR SI CERCA UN VALORE g^{ma} DELLA LISTA CHE NON È VUOTA;
- INFINE SI OTTIENE $B[Xg^{-b}] = B[g^{ma}]$ PER CUI SI HA CHE $x_0 = b$ e $x_1 = a$.

m^3

KeyGen :

- $N = p \cdot q$
- $e > 2^{60}$
- $\gcd(e, \phi(N)) = 1$
- d CHIAVE SEGRETA

$$M = \mathbb{Z}_N^*$$

$$C_1 \leftarrow n^e$$

$$C_2 \leftarrow n^{e+1} \cdot m = n^e \cdot n \cdot m$$

$$C_2/C_1 = n \cdot m = C_3$$

Dec(C_1, C_3) :

$$n \leftarrow C_1^d$$

$$m \leftarrow \frac{n \cdot m}{n^e} = \frac{m}{n^e} \cdot C_1 = \underline{\underline{m}}$$

A(AE) :

$$m_0, m_1 \in M = \mathbb{Z}_N^*;$$

$$(C_1, C_2) = O_{\text{ENC}}(m_0, m_1);$$

$$m_3 = O_{\text{DEC}}(C_1, C_2/C_1);$$

if ($m_3 \cdot C_1 == m_1$) return 1;

else return 0;

$$\Pr[\text{ESPAE}^{\text{ind-CCA-1}}(A) = 1] = 1 \quad \Pr[\text{ESPAE}^{\text{ind-CCA-0}}(A) = 1] = 0$$

$$\text{Adv}_{AE}^{\text{ind-CCA}}(A) = 1$$

m°4

SIA DS = (KeyGen, Sign, VF) UNA SCHEMA DI FIRMA DIGITALE TALE CHE:

- KeyGen → VK e SK;
- Sign → (M, α) e $\alpha \in \{0,1\}^* \cup \{\perp\}$
- VF → {0,1}

SIA A UN AVVERSARIO CHE HA ACCESSO AD UN ORA COLO DI FIRMA E VOGLIE GENERARE UN FALSO.

SI CONSIDERI IL SEGUENTE ESPERIMENTO:

$\text{ESP}_{\text{DS}}^{\text{uf-cma}}(A)$:

$$(\text{VK}, \text{SK}) \leftarrow \text{KeyGen}(); \\ (\text{M}, \alpha) \leftarrow A^{\text{Sign}_{\text{SK}}(\cdot)}(\text{VK});$$

IF ACCADE CHE:

- $\pi \in \text{MESSAGGI};$
- $\text{VF}(\text{VK}, \pi, \alpha) = 1;$
- M NON CHIESO A O sign_{SK} :

return 1;

else return 0;

$$\text{Adv}_{\text{DS}}^{\text{uf-cma}}(A) = \Pr[\text{ESP}_{\text{DS}}^{\text{uf-cma}}(A) = 1]$$

m° 5

H_p : ESISTE B CHE RISOLVE EDH IN G

T_S : ESISTE A CHE RISOLVE EDH IN G

SUPPONIAMO PER ASSURDO CHE NON VAGA L'AFTESI.

$A(g^a, g^b, g^c, g^d)$:

$X \leftarrow B(g^a, g^b)$;

$Y \leftarrow B(g^c, g^d)$;

$Z \leftarrow e(X, Y) = e(g^{ab}, g^{cd}) = e(g, g)^{abcd}$

return Z

$\text{Adv}^{\text{EDH}}(A) = \text{Adv}^{\text{CDH}}(B)$

Corso di Crittografia

Prova in Itinere del 8 Febbraio 2007

1 Varianti di El Gamal

In classe abbiamo descritto il cifrario El Gamal nel seguente modo.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave prende in input un parametro k e sceglie due primi q, p tali che $|q| = k$ e q divide $p - 1$. Quindi procede come segue. Detto G un sottogruppo di \mathbb{Z}_p^* di ordine q , pone $\mathcal{M} = G$ come spazio dei messaggi. Quindi sceglie un generatore g di G , sceglie (a caso secondo la distribuzione uniforme) $x \in \{1, \dots, q\}$ e pone $h = g^x \bmod p$. Infine, restituisce $PK = (p, q, g, h, \mathcal{M})$ come chiave pubblica e $SK = x$ come chiave privata.

Algoritmo di cifratura

```
Enc( $PK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \{1, \dots, q\}; C_1 \leftarrow g^r \bmod p;$ 
   $C_2 \leftarrow h^r m \bmod p$ 
  return  $(C_1, C_2)$ .
```

Algoritmo di decifratura

```
Dec( $SK, C_1, C_2$ )
   $A \leftarrow C_1^x \bmod p;$ 
   $m \leftarrow C_2 / A \bmod p$ 
  return  $m$ .
```

In classe abbiamo dimostrato che tale cifrario è sicuro, relativamente ad attacchi di tipo CPA, nell'ipotesi che il problema Diffie-Hellman decisionale sia difficile.

Il problema di tale cifrario è che esso sembra imporre restrizioni fastidiose sullo spazio dei messaggi (i messaggi devono essere elementi in G). Si consideri allora la seguente variante, che potremmo chiamare El Gamal'. Il cifrario El Gamal' ha come unica differenza, rispetto a El Gamal, che $\mathcal{M} = \{0, \dots, p - 1\}$. E' El Gamal' un cifrario sicuro? Giustificare formalmente la risposta fornita.

2 Cifrario Paillier

Descrivere dettagliatamente il cifrario Paillier (nella versione semplificata presentata in classe). In particolare, descrivere l'algoritmo di generazione della chiave, l'algoritmo di cifratura e l'algoritmo di decifratura.

3 Firme Digitali I

Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.

4 Firme digitali II

Si consideri il seguente schema.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave prende in input un parametro k e sceglie quattro primi p, q, q', p' tali che $|p| = |q| = k/2$, $q = 2q' + 1$ e $p = 2p' + 1$. Quindi procede come segue. Sia S un quadrato residuo in \mathbb{Z}_N^* (si noti che un tale elemento avrà ordine $p'q'$ in \mathbb{Z}_N^*) e sia $\mathcal{M} = \{m : m \geq 2\}$. L'algoritmo, restituisce $VK = (N, S, \mathcal{M})$ come chiave pubblica e $SK = (p, q)$ come chiave privata.

Algoritmo di firma

```
Sign(SK, m)
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $\sigma \leftarrow \sqrt[m]{S} \bmod N;$ 
  return  $\sigma$ 
```

Algoritmo di verifica

```
Verify(VK, m,  $\sigma$ )
  If  $\sigma = \perp$  return 0;
  If  $\sigma^m = S \bmod N$  return 1
  else return 0
```

E' questo schema sicuro? Giustificare formalmente la risposta fornita.

m° 1

CONSIDERO $m_0 = 0$ e $m_1 = 1$:

$$\text{Enc}(\text{PK}, m_0) = (C_1, C_2) = (g^n, 0)$$

$$\text{Enc}(\text{PK}, m_1) = (C_1, C_2) = (g^n, h^n \cdot m_1)$$

POSSO CONTROLLARE SE $C_2 \neq 0$.

FORMALIZZIAMO L'AVVERSARIO:

A(AE):

$$m_0 = 0;$$

$$m_1 = 1;$$

$$(C_1, C_2) = \text{O}_{\text{ENC}}(m_0, m_1)$$

if ($C_2 \neq 0$) return 1

else return 0

$$\Pr [\text{ESP}_{\text{AE}}^{\text{ind-CPA-1}}(A) = 1] = 1$$

$$\Pr [\text{ESP}_{\text{PAE}}^{\text{ind-CPA-0}}(A) = 1] = 0$$

$$\text{Adv}_{\text{AE}}^{\text{ind-CPA}}(A) = 1$$

m° 2

PER POTER PARLARE DEL CIFRARIO PAILLIER È NECESSARIO INTRODURRE IN CONCETTO DI N-RESIDUOSITÀ.

SIA $G = \mathbb{Z}_{N^2}^{*}$ E SIA $y \in G$ ALLORA DIREMO CHE y È UN N-RESIDUO SE $y \equiv w^N \pmod{N^2}$.

CONSIDERIAMO I SEGUENTI FATTI:

1. SIA $T = \{(1 + xN) : x \in \mathbb{Z}_N\}$ ALLORA $\forall z \in T$ SI HA CHE $z^N \equiv 1 \pmod{N^2}$
2. $\phi(N^2) = (p^2 - p)(q^2 - q) = \phi(N) \cdot N$ QUESTO VUOL DIRE CHE OGNI N-RESIDUO HA ORDINE $\phi(N)$.
3. SIA $w \in \mathbb{Z}_{N^2}^{*}$ ALLORA $w = (1 + xN)y^N$ CON $x \in \mathbb{Z}_N$ E $y \in \mathbb{Z}_N^{*}$.

IL PROBLEMA DELLA N-RESIDUOSITÀ È RAPIRE SE UN ELEMENTO È UN N-RESIDUO E SI CONGETTURA CHE PER RISOLVERE TALE PROBLEMA È NECESSARIA LA CONOSCENZA DELLA FATTOORIZZAZIONE DI N. INFATTI:

$$w^{N\phi(N)} \pmod{N^2} = w \pmod{N^2}$$

NEL CIFRARIO PAILLIER L'ALGORITMO DI GENERAZIONE DELLE CHIAVI È SIMILE A QUELLO DI RSA.

KeyGen(K):

```
N, p, q ← K-rsa( $K$ );  
return(N, (N, p, q));
```

IN QUESTO CASO NON È NECESSARIO GENERARE $e \in \mathbb{Z}_{\phi(N)}^{*}$.

Enc(PK, m): // $m \in \mathbb{H} = \mathbb{Z}_N$

```
y ←  $\mathbb{Z}_{N^2}^{*}$ ;  
c ←  $(1 + mN)y^N \pmod{N^2}$ ;  
return c;
```

$\text{Dec}(\text{SK}, C)$:

$$C' \leftarrow C^{\phi(N)} = (1+mN)^{\phi(N)} \bmod N^2;$$

$$C'' \leftarrow C^{\phi(N) \cdot d} = (1+m \cdot N) \bmod N^2;$$

$$m \leftarrow \frac{C'' - 1}{N};$$

return m ;

m° 4

$$S \in QR(\mathbb{Z}_n^*)$$

$$|S| = p^1 q^1$$

$$M = \{m : m \geq 2\}$$

$$\sigma = \sqrt[m]{S} = S^{\frac{1}{m}}$$

SE $m = 2$:

$$\sigma = S^{\frac{1}{2}}$$

SE $m = 4$:

$$\sigma = \sqrt[4]{S} = S^{\frac{1}{4}}$$

$$\sigma^2 = S^{\frac{1}{2}}$$

A($\vee n$):

$$\sigma = \text{O}_{\text{Sign}_\text{SA}}(4);$$

$$\vee F(\vee n, 2, \sigma^2)$$

Corso di Crittografia

Prova in Itinere del 14 Febbraio 2008

1 Primitive asimmetriche

In classe abbiamo accennato al problema bilineare Diffie Hellman, definito su gruppi per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine primo q) come quella discussa a lezione. Tale problema sembra essere intrattabile. Più precisamente, sia g un generatore di G , definiamo il seguente esperimento

$$\begin{aligned} \mathbf{Esp}_{G,g}^{\text{BDH}}(\mathcal{A}) \\ a, b, c \leftarrow_R \mathbb{Z}_q^*; A \leftarrow g^a; B \leftarrow g^b; C \leftarrow g^c; \\ x \leftarrow \mathcal{A}(A, B, C); \\ \text{If } x = e(g, g)^{abc} \text{ return 1 else return 0} \end{aligned}$$

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{BDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{BDH}}(\mathcal{A}) = 1]$$

1. Definire in maniera analoga l'ipotesi di intrattabilità del problema Computazionale Diffie-Hellman (CDH) sul gruppo G .
2. Dimostrare che il problema BDH non può essere più difficile del problema computazionale Diffie Hellman in G . In altre parole, si dimostri che, se esiste un avversario A capace di risolvere il problema CDH in G , tale avversario può essere sfruttato per risolvere il problema BDH in G e G_T .

2 Indistinguibilità

Introdurre e definire formalmente il concetto di indistinguibilità `ind-id-cpa` per cifrari basati sull'identità.

3 Cifrari Asimmetrici

Si descriva formalmente il cifrario El Gamal e si dimostri che tale cifrario non è sicuro relativamente ad attacchi a crittotesto scelto.

4 Firme Digitali

Si consideri il seguente schema.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave prende in input un parametro k e sceglie due primi p, q tali che $|p| = |q| = k/2$, $p = q = 3 \bmod 4$ (questa ultima condizione serve solo a garantire la correttezza formale dello schema che verrà presentato). Quindi procede come segue. Sia $\mathcal{M} = \mathbb{Z}_N^*$ lo spazio dei messaggi. Si noti che $\forall m \in \mathbb{Z}_N^*$ esattamente uno tra m e $-m$ è quadrato residuo. L'algoritmo restituisce $VK = (N, \mathcal{M})$ come chiave pubblica e $SK = (p, q)$ come chiave privata.

Algoritmo di firma

```
Sign(SK, m)
  If  $m \notin \mathcal{M}$  return  $\perp$ 
  If  $m \notin \text{QR}_N$  sia  $m \leftarrow -m$ 
   $\sigma \leftarrow \sqrt{m} \bmod N;$ 
  return  $\sigma$ 
```

Algoritmo di verifica

```
Verify(VK, m,  $\sigma$ )
  If  $\sigma = \perp$  return 0;
  If  $m^2 = \sigma \bmod N$  or  $(-m)^2 = \sigma \bmod N$  return 1
  else return 0
```

E' questo schema sicuro? Giustificare formalmente la risposta fornita.

m° 1

H_P: ESISTE UN AVVERSARIO A CHE RISOLVE CDH

T_S: ESISTE B CHE RISOLVE BDH

B(g^a, g^b, g^c):

$$g^{ab} \leftarrow A(g^a, g^b);$$

$$g^{abc} \leftarrow A(g^{ab}, g^c)$$

$$z = e(g, g^{abc}) = e(g, g)^{abc}$$

return z

$$\text{Adv}^{\text{BDH}}(B) = \text{Adv}^{\text{CDH}}(A) \Rightarrow \text{TESI}$$

m°3

SI CONSIDERINO DUE PRIMI P & q TALE CHE $q \mid p-1$ E
SIA G UN SOTTOGRUPPO DI \mathbb{Z}_p^* DI ORDINE q. LO SPAZIO DEI
MESSAGGI $M = G = \mathbb{Z}_p^*$ ESIA gEG UN GENERATORE.

L'ALGORITMO DI GENERAZIONE DELLE CHIAVI È IL SEGUENTE:

KeyGen(G, p, q, g):

$$x \leftarrow \{1, \dots, q\};$$

$$h \leftarrow g^x;$$

$$SK = x;$$

return $((G, p, q, g, h), x)$;

L'ALGORITMO DI CIFRATURA È:

Enc(PK, m):

$$n \leftarrow \{1, \dots, q\};$$

$$C_1 \leftarrow g^n;$$

$$C_2 \leftarrow h^n \cdot m;$$

return (C_1, C_2) ;

L'ALGORITMO DI DECIFRATURA È:

Dec(SK, C_1, C_2):

$$A \leftarrow C_1^{-1};$$

$$m \leftarrow C_2 / A;$$

return m;

CONSIDERIAMO $m_0, m_1, m_2 \in \mathbb{N}$:

- $\text{Enc}(\text{LR}(m_0, m_1, b)) = (g^u, h^u \cdot m)$
- $C'_2 = C_2 \cdot m_2 = h^u \cdot m \cdot m_2$
- $\text{Dec}(C_2, C'_2) = m \cdot m_2$
- if $m \cdot m_2 / m_2 == m_2$ return 1 else return 0;

m° 4

$$p = q = 3 \bmod 4 \quad M = \mathbb{Z}_N^*$$

$$\text{VK} = (N, M) \quad \text{SK} = (p, q)$$

A(VK) :

$$m \xleftarrow{R} M;$$

$$\sigma \leftarrow \text{Sign}_{SK}(m);$$

$$\text{VF}(\text{VK}, -m, \sigma);$$

$$\text{Adv}_{DS}^{\text{uf-cma}}(A) = 1$$

CORRETTO PERCHE SE m E -m PRODUCONO LA STESSA FIRMA.

Corso di Crittografia

Prova in Itinere del 29 Gennaio 2009

1. Si dimostri che il cifrario Paillier non è sicuro in senso IND-CCA (cioé non garantisce indistinguibilità contro attacchi a crittotesto scelto)
2. Si presenti lo pseudocodice del cifrario Boneh-Franklin e si discuta il funzionamento degli algoritmi di setup, di Key Extraction, di cifratura e di decifratura.
3. Si consideri il seguente cifrario asimmetrico. L'algoritmo di generazione della chiave è simile all'algoritmo di generazione dei parametri di RSA. Su input un parametro di sicurezza k , produce un modulo $N = p \cdot q$, tale che $|p| = |q| = k$ dove $p = 2p' + 1$, $q = 2q' + 1$, con p' , q' entrambi primi. Come esponente pubblico si consideri $e = 2$. La chiave pubblica è (N, e) , la corrispondente chiave privata è il valore d tale che $e \cdot d = 1 \pmod{p'q'}$ (si noti che la specificità di p, q fa sì che l'ordine di ogni quadrato residuo in \mathbb{Z}_N^* sia un divisore di $p'q'$). Lo spazio dei messaggi è l'insieme $\mathcal{M} = \{1, 2\}$.

L'algoritmo di cifratura funziona nel seguente modo

```
Enc( $N, e, m$ )
  if ( $m \notin \mathcal{M}$ ) return  $\perp$ 
   $\rho \leftarrow_R \mathbb{Z}_N^*$ 
   $r \leftarrow \rho^2 \pmod{N}$ ; Si noti che r è un quadrato residuo in  $\mathbb{Z}_N^*$ 
   $C_1 \leftarrow r^2 \pmod{N}$ ;  $C_2 \leftarrow (r + 1)^2 m \pmod{N}$ ;
  Return  $C = (C_1, C_2)$ ;
```

L'algoritmo di decifratura, invece, opera come segue

```
Dec( $d, C = (C_1, C_2)$ )
   $r \leftarrow C_1^d \pmod{N}$ 
   $m \leftarrow C_2 \cdot ((r + 1)^e)^{-1} \pmod{N}$ 
  Return  $m$ .
```

Dimostrare che tale cifrario non è sicuro nemmeno in senso IND-CPA.

4. Si consideri il seguente schema di firma digitale. L'algoritmo di generazione della chiave funziona nel seguente modo. Su input un parametro di sicurezza k , produce un modulo $N = p \cdot q$, tale che $|p| = |q| = k$ e fissa due valori e_1, e_2 tale che

$\gcd(e_i\phi(N)) = 1$ ($i = 1, 2$). La chiave pubblica è (N, e_1, e_2) , la corrispondente chiave privata è costituita dal valore d tale che $e_1 \cdot d = 1 \pmod{\phi(N)}$.

Lo spazio dei messaggi è l'insieme \mathbb{Z}_N^* . Gli algoritmi di firma e di verifica sono definiti come segue:

```

Sign( $d, m$ )
    if ( $m \notin \mathbb{Z}_N^*$ ) return  $\perp$ 
    Sia  $r \leftarrow_R \mathbb{Z}_N^*$ 
     $\alpha \leftarrow r^{e_2} \pmod{N}$ 
     $\beta \leftarrow m \cdot \alpha^{-1} \pmod{N}$ ;
     $s \leftarrow \beta^d \pmod{N}$ 
    Return  $(r, s)$ .

```

```

Ver( $N, (r, s), m$ )
    if ( $m \notin \mathbb{Z}_N^*$ ) return  $\perp$ 
    if ( $m == s^{e_1}r^{e_2} \pmod{N}$ ) return 1
    else return 0

```

E' quello appena esposto uno schema di firma digitale sicuro? Giustificare formalmente la risposta fornita.

m°2

CONSIDERIAMO G_1, G_2, G_T DI ORDINE PRIMO q E CONSIDERIAMO LA SEGUENTE MAPPA BICINEARE:

$$P^1 \in G_2 \quad \varrho \in G_1$$

$$\varrho : G_2 \times G_2 \rightarrow G_T$$

NOTARE ESISTE UN ISOMORFISMO $\varphi : G_2 \rightarrow G_1$.

Setup(q):

$$msk \xleftarrow{R} \{1, \dots, q\};$$

$$PK \leftarrow (P^1)^{msk};$$

return (msk, PK);

KeyGen(id, msk):

$$Pid \leftarrow H_1(id);$$

$$msk \leftarrow Pid^{msk};$$

return msk ;

Ene(PK, m, id):

$$n \xleftarrow{R} \{1, \dots, q\};$$

$$Pid \leftarrow H_1(id);$$

$$K \leftarrow e(Pid, PK)^n;$$

$$C_1 \leftarrow (P^1)^n;$$

$$C_2 \leftarrow H_2(K) \oplus m;$$

return (C_1, C_2);

Dec(msk, C_1, C_2):

$$K \leftarrow e(msk, C_1);$$

$$m \leftarrow H_2(K) \oplus C_2;$$

return m ;

M°3

$$e = 2 \quad PK = (N, e) \quad Sk = d \quad M = \{1, 2\}$$

$$C_1 = n^2$$

$$C_2 = \underbrace{(n^2 + 2n + 1)}_{m} m$$

$$m = 2$$

$$C_1 = n^2$$

$$C_2 = n^2 + 2n + 1$$

$$C_2 - C_1 = \cancel{n^2} + 2n + 1 - \cancel{n^2} = C_3$$

$$C_3 - 1 = \frac{2n}{2} = C_4^2 = C_1 \text{ OK!}$$

A(PK):

$$m_1 = 1 ;$$

$$m_0 = 2 ;$$

$$(C_2, C_2) = O_{Enc}(m_0, m_1) ;$$

$$C_3 = (C_2 - C_1 - 1)/2 \bmod N ;$$

if ($C_3^2 = C_1$) return 1;

else return 0;

$$\Pr [\text{ESP}_{AE}^{\text{ind-CPA-1}}(A) = 1] = 1$$

$$\Pr [\text{ESP}_{AE}^{\text{ind-CPA-0}}(A) = 1] = 0$$

$$\text{Adv}_{AE}^{\text{ind-CPA}}(A) = |1 - 0| = 1$$

M°4

$$S = \beta^d = (m \cdot d^{-1})^d = (m \cdot r^{-\ell_2})^d = m^d \cdot r^{-\ell_2 \cdot d}$$

$$(m^{d+2} \cdot r^{-\ell_2 \cdot d})^{e_1} \cdot r^{\ell_2} = m^{d+2 \cdot e_1} \cdot r^{-\ell_2 \cdot d + \ell_2} = m^{d+2 \cdot e_1} \cdot r^{0}$$

Posso considerare il messaggio m^{1+e_1} e tag (r, s)

un'altra soluzione è la seguente:

- si consideri $s' = r$;
- si consideri $m' = r^{\ell_2} \cdot r^{\ell_2}$
- allora $r^{\ell_1} \cdot r^{\ell_2} = r^{\ell_1} \cdot r^{\ell_2}$

formalizziamo l'avversario:

A(λ):

$$r \leftarrow \mathbb{Z}_N^*;$$

$$m \leftarrow r^{\ell_2} \cdot r^{\ell_2};$$

$$s \leftarrow r;$$

$$\text{VF}(N, (r, s), m);$$

$$\text{Adv}_{\text{DS}}^{\text{uf-com}}(A) = 1$$

Corso di Crittografia

Prova in Itinere del 28 Gennaio 2010

1 Primitive asimmetriche

Si consideri il seguente problema computazionale (che potremmo chiamare Bilinear Square), definito su gruppi per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine primo q) come quella discussa a lezione. Sia g un generatore di G , definiamo il seguente esperimento

$$\begin{aligned} \mathbf{Esp}_{G,g}^{\text{BSquare}}(\mathcal{A}) \\ a \leftarrow_R \mathbb{Z}_q^*; A \leftarrow g^a; \\ x \leftarrow \mathcal{A}(A); \\ \text{If } x = e(g, g)^{a^2} \text{ return 1 else return 0} \end{aligned}$$

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{BSquare}} = \Pr [\mathbf{Esp}_{G,g}^{\text{BDH}}(\mathcal{A}) = 1]$$

Dimostrare che il problema BSquare non può essere più difficile del problema computazionale Diffie Hellman in G . In altre parole, si dimostri che, se esiste un avversario B capace di risolvere il problema CDH in G , tale avversario può essere sfruttato per risolvere il problema BSquare in G e G_T .

2 Indistinguibilità

Introdurre e definire formalmente il concetto di indistinguibilità `ind-id-cpa` per cifrari basati sull'identità.

3 Cifrari Asimmetrici

Dimostrare che il seguente cifrario non è sicuro in senso IND-CCA.

Algoritmo di Generazione delle Chiavi. Prende in input un parametro k e sceglie due primi q, p tali che $|q| = k$ e q divide $p - 1$. Detto G un sottogruppo di \mathbb{Z}_p^* di ordine q , pone $\mathcal{M} = G$ come spazio dei messaggi. Quindi sceglie un

generatore h di G , sceglie (a caso) $x_1, x_2 \in \{1, \dots, q\}$ e calcola g_1 e g_2 tali che $h = g_1^{x_1} \pmod p$ e $h = g_2^{x_2}$ (si noti che non è necessario poter estrarre logaritmi discreti per effettuare questa operazione efficientemente). Infine, restituisce $PK = (p, q, g_1, g_2, h, \mathcal{M})$ come chiave pubblica e $SK = (x_1, x_2)$ come chiave privata.

Algoritmi di cifratura e decifratura

```

Enc( $PK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r, s \leftarrow_R \{1, \dots, q\};$ 
   $C_1 \leftarrow g_1^r \pmod p; \quad C_2 \leftarrow g_2^s \pmod p; \quad C_3 \leftarrow h^{r+s}m \pmod p$ 
  return  $(C_1, C_2, C_3)$ .
Dec( $SK, C_1, C_2, C_3$ )
   $A \leftarrow C_1^{x_1} \pmod p; \quad B \leftarrow C_2^{x_2} \pmod p; \quad m \leftarrow C_3 / (A \cdot B) \pmod p$ 
  return  $m$ .

```

4 Firme Digitali

Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.

5 Firme Digitali - II

Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k sceglie due primi p, q tali che $|p| = |q| = k/2$ e un intero e tale che $\gcd(e, \phi(N)) = 1$. Inoltre sia A un elemento in \mathbb{Z}_N^* (di ordine sufficientemente grande). Infine, sia $\mathcal{M} = \mathbb{Z}_N^*$ lo spazio dei messaggi. L'algoritmo restituisce $VK = (N, \mathcal{M}, e, A)$ come chiave pubblica e $SK = (p, q)$ come chiave privata.

Algoritmo di firma

```

Sign( $SK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $\sigma \leftarrow \sqrt[e]{A^m} \pmod N;$ 
  return  $\sigma$ 

```

Algoritmo di verifica

```

Verify( $VK, m, \sigma$ )
  If  $\sigma = \perp$  return 0;
  If  $A^m = \sigma^e \pmod N$  return 1
  else return 0

```

m° 1

H_p: ESISTE B CHE RISOLVE CDH

T_S: ESISTE UN AVVERSARIO A CHE RISOLVE B^{Square}

A(g^a):

$X \leftarrow B(g^a, g^a)$; // $X = g^{a^2}$

returne(X, g);

$\text{Ad}_{\sqrt{G}, g}^{\text{BSquare}}(A) = \text{Ad}_{\sqrt{G}, g}^{\text{CDH}}(A) \Rightarrow \text{TESI}$

m^3

$$h \in G, \quad x_1, x_2 \in \{1, \dots, q\}$$

SI CALCOLA g_1 e g_2 TACI CHE $h = g_1^{x_1} \text{ mod } p$ e
 $h = g_2^{x_2} \text{ mod } p$

$$PK = (p, q, g_1, g_2, h, n)$$

$$SK = (x_1, x_2)$$

$$Enc(PK, m) = (g_1^r, g_2^s, h^{n+s} \cdot m)$$

$$\begin{aligned} Dec(SK, C_1, C_2, C_3) &= (h^{n+s} / (g_1^{x_2 \cdot r} \cdot g_2^{x_2 s})) \cdot m \text{ mod } p \\ &= m \end{aligned}$$

POSSIAMO EFFETTUARE LA SEGUENTE OSSERVAZIONE:

- CONSIDERIAMO C_3 E MOLTIPLICHIAMO UNO DEI DUE MESSAGGI m_0, m_1
- CONTROLLIAMO SE $Dec(SK, C_1, C_2, C_3 \cdot m_1) = m_1^2$,
SE VERO ALLORA TORNIAMO ALTRIMENTI 0.

A(PK):

$$m_0, m_1 \in \mathbb{N};$$

$$(C_1, C_2, C_3) \leftarrow O_{ENC_{PK}}(m_0, m_1);$$

$$m_2 \leftarrow O_{DEC_{SK}}(C_1, C_2, C_3 \cdot m_1);$$

$$\text{if } (m_2 == m_1^2) \text{ return } 1;$$

$$\text{else return } 0;$$

$$\Pr [ESP_{AE}^{\text{ind-cca-1}}(A) = 1] = 1$$

$$\Pr [ESP_{AE}^{\text{ind-cca-0}}(A) = 1] = 0$$

$$\text{Adv}_{AE}^{\text{ind-cca}}(A) = |1 - 0| = 1$$

m° 5

$A \in \mathbb{Z}_N^*$, $M \in \mathbb{Z}_N^*$, $VK = (N, M, e, A)$, $SK = (p, q)$

$$\sigma = \underbrace{A^{\frac{m}{e}}}_{\text{ }} = \sqrt[e]{A^m}$$

$$m^2, \sigma^m = A^{\frac{m}{e} \cdot m} = A^{\frac{m^2}{e}}$$

$$VF(VK, m^2, \sigma^m) = \sigma^{m \cdot e} = A^{\frac{m \cdot m \cdot e}{e}} = A^{m^2}$$

A (VK):

$$m \leftarrow \mathbb{Z}_N^*$$

$$\sigma \leftarrow \text{Sign}_SK(m);$$

$$VF(VK, m^2, \sigma^m);$$

$$\text{Adv}_{\text{mt-ema}}(A) = 1$$

Corso di Crittografia

Prova in Itinere del 17 Giugno 2011

1. Si consideri il seguente problema computazionale (che chiameremo Modified Bilinear Diffie-Hellman), definito su gruppi per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine primo q) come quella discussa a lezione. Sia g un generatore di G , definiamo il seguente esperimento

```
EspMBDH $G,g$ ( $\mathcal{A}$ )
   $a,b,c \leftarrow_R \mathbb{Z}_q^*$ ;  $A \leftarrow g^a$ ;  $B \leftarrow g^b$ ;  $C \leftarrow g^c$ 
   $x \leftarrow \mathcal{A}(A,B,C)$ ;
  If  $x = e(g,g)^{\frac{ab}{c}}$  return 1 else return 0
  // La quantità  $\frac{ab}{c}$  all'esponente è equivalente a  $c^{-1} \bmod q$ 
```

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{MBDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{MBDH}}(\mathcal{A}) = 1]$$

Dimostrare che il problema MBDH non può essere più difficile del problema del logaritmo discreto (DL) in G . In altre parole, si dimostri che, se esiste un avversario \mathcal{B} capace di risolvere DL in G , tale avversario può essere sfruttato per risolvere MBDH in G e G_T .

2. Si definisca formalmente il concetto di indistinguibilità contro attacchi a crittotesto scelto **ind-cca** per cifrari asimmetrici.
3. (a) Si presenti lo pseudocodice del cifrario Paillier, discutendo il funzionamento degli algoritmi KeyGen, Enc e Dec.
(b) Si dimostri che il cifrario Paillier non è sicuro in senso IND-CCA.
4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k sceglie due primi p, q tali che $|p| = |q| = k/2$ e un intero e tale che $\gcd(e, \phi(N)) = 1$. Inoltre sia g un elemento in \mathbb{Z}_N^* (di ordine sufficientemente grande). Infine, sia $\mathcal{M} = \mathbb{Z}_N^*$ lo spazio dei messaggi. L'algoritmo restituisce $VK = (N, \mathcal{M}, e, g)$ come chiave pubblica e $SK = (p, q)$ come chiave privata.

Algoritmo di firma

```

Sign( $SK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \mathbb{Z}_N^*$ 
   $y \leftarrow \sqrt[e]{rg^m} \bmod N;$ 
  return  $\sigma \leftarrow (y, r)$ 

```

Algoritmo di verifica

```

Verify( $VK, m, \sigma$ )
  If  $\sigma = \perp$  return 0;
  If  $rg^m = y^e \bmod N$  return 1
  else return 0

```

m°1

H_p: ESISTE B CHE RISOLVE DL

TS: ESISTE A CHE RISOLVE MBDH

A(g^a, g^b, g^c):

$a \leftarrow B(g^a);$

$b \leftarrow B(g^b);$

$c \leftarrow B(g^c);$

$c^{-1} = \text{MOD-INV}(c);$

return $e(g^{ab}, g^{c^{-1}});$

$\text{Ad}^{\sqrt{\text{MBDH}}}(A) = \text{Ad}^{\sqrt{DL}}(B) \Rightarrow \text{TESI}$

A(g^a, g^b, g^c):

$c \leftarrow B(g^c);$

$c^{-1} = \text{MOD-INV}(c);$

$z = e(g^a, g^b)^{\frac{1}{c}}$

return z;

m° 5

$$r \leftarrow \mathbb{Z}_n^*$$

$$y \leftarrow \sqrt[m]{r g^m} = (r g^m)^{\frac{1}{m}} = r^{\frac{1}{m}} g^{\frac{m}{m}}$$

CONSIDERO:

- m^2
- $y^m = r^{\frac{m}{m}} \cdot g^{\frac{m}{m}}$
- r^m

ALGORITMO:

$$\text{VF}(\forall n, m^2, (y^m, r^m)) = 1$$

A($\forall n$):

$$m \leftarrow \mathbb{Z}_n^*$$

$$(y, r) \leftarrow \text{Osign}_n(m)$$

$$\text{VF}(\forall n, m^2, (y^m, r^m))$$

$$\text{Ad}_{\forall^{uf-cma}}(A) = 1$$

Corso di Crittografia

Prova in Itinere del 18 Giugno 2012

1. Si definisca formalmente il concetto di indistinguibilità contro attacchi a crittotesto scelto (**ind-cca**) per cifrari asimmetrici.
2. Si consideri il seguente seguente cifrario asimmetrico. L'algoritmo di generazione delle chiavi, su input un parametro di sicurezza k , produce un modulo $N = p \cdot q$, tale che $|p| = |q| = k$. Come esponente pubblico si consideri $e = N$ (si noti che $\gcd(N, \phi(N)) = 1$). La chiave pubblica è (N, e) , la corrispondente chiave privata è il valore d tale che $e \cdot d = 1 \pmod{\phi(N)}$. Lo spazio dei messaggi è l'insieme $\mathcal{M} = \{0, 1\}$.

L'algoritmo di cifratura funziona nel seguente modo

```
Enc( $N, m$ )
  if ( $m \notin \mathcal{M}$ ) return  $\perp$ 
   $C_1 \leftarrow r^N \pmod{N}; C_2 \leftarrow (r + 1)^N m \pmod{N};$ 
  Return  $C = (C_1, C_2);$ 
```

L'algoritmo di decifratura, invece, opera come segue

```
Dec( $d, C = (C_1, C_2)$ )
   $r \leftarrow C_1^d \pmod{N}$ 
   $m \leftarrow C_2 \cdot ((r + 1)^N)^{-1} \pmod{N}$ 
  Return  $m.$ 
```

Dimostrare che tale cifrario non è sicuro in senso IND-CPA.

3. Si consideri il seguente problema computazionale (che chiameremo Strong Diffie-Hellman Problem), definito su gruppi per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine primo q) come quella discussa a lezione. Sia g un generatore di G , definiamo il seguente esperimento

```
Esp $G, g$ SDH( $\mathcal{A}$ )
   $a \leftarrow_R \mathbb{Z}_q^*; A_1 \leftarrow g^a; A_2 \leftarrow g^{a^2}; A_3 \leftarrow g^{a^3}$ 
   $x \leftarrow \mathcal{A}(A_1, A_2, A_3);$ 
  If  $x = e(g, g)^{\frac{1}{a}}$  return 1 else return 0
    // La quantità  $\frac{1}{a}$  all'esponente è equivalente a  $a^{-1} \pmod{q}$ 
```

Il vantaggio di \mathcal{A} è definito come

$$\mathbf{Adv}_{G,g}^{\text{SDH}} = \Pr [\mathbf{Esp}_{G,g}^{\text{SDH}}(\mathcal{A}) = 1]$$

Dimostrare che il problema SDH non può essere più difficile del problema del logaritmo discreto (DL) in G . In altre parole, si dimostri che, se esiste un avversario B capace di risolvere DL in G , tale avversario può essere sfruttato per risolvere SDH in G o G_T .

4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si considerino, inoltre, due gruppi G e G_T per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine p). Sia g un generatore di G . L'algoritmo procede scegliendo $x \in \{1, \dots, p-1\}$ (a caso) e ponendo $h = g^x$. La chiave pubblica è quindi $VK = (h, g, e, G, G_T)$, mentre la chiave privata è $SK = x$. Lo spazio dei messaggi è G .

Algoritmo di firma

```

Sign( $SK, m$ )
  If  $m \notin G$  return  $\perp$ 
   $\sigma = m^x$ 
  return  $\sigma$ 

```

Algoritmo di verifica

```

Verify( $VK, m, \sigma$ )
  If  $\sigma = \perp$  return 0;
  If  $e(\sigma, g) = e(m, h)$  return 1
  else return 0

```

Si noti che una firma valida viene sempre verificata correttamente. Infatti per la bilinearità di e si ha

$$e(\sigma, g) = e(m^x, g) = e(m, g)^x, \quad e(m, h) = e(m, g^x) = e(m, g)^x$$

$m^o 2$

$$M = \{0, 1\} \quad N = p \cdot q \quad e = N$$

$$\text{Enc}(N, m) = (r^N, (r+1)^N m)$$

A(PK):

$$m_0 = 0, m_1 = 1;$$

$$(c_1, c_2) \leftarrow \text{OEnc}(m_0, m_1);$$

if $c_2 \neq 0$ return 1

else return 0

$$\Pr_{r} [\text{ESp}^{\text{ind-CPA-1}}(A) = 1] = 1$$

$$\Pr_{r} [\text{ESp}^{\text{ind-CPA-0}}(A) = 1] = \frac{1}{N}$$

$$\text{Adv}_{\mathcal{V}}^{\text{ind-CPA}}(A) = \left| 1 - \frac{1}{N} \right| \approx 1$$

m°3

H_p: ESISTE B RISOLVE DL

T_S: ESISTE A RISOLVE SDH

A(g^a, g^{a^2}, g^{a^3}):

$a \leftarrow B(g^a);$

$a^2 \leftarrow B(g^{a^2});$

$a^{-1} \leftarrow a/a^2;$

return $e(g, g)^{a^{-1}}$;

$\text{Adv}^{\text{SDH}}(A) = \text{Adv}^{\text{2DC}}(B) \Rightarrow \text{TESI}$

m° 5

$$h = g^\times \quad Sk = \times \quad \nu u = (h, g, e, G, Gr) \quad M = G$$

$$e(a, g) = e(m, h)$$

$$e(m, g)^\times = e(m, g)^\times$$

A(vu):

$$VF(vu, g, h)$$

$$\text{Ad}_v^{\text{uf-cma}}(A) = 1$$

Corso di Crittografia

Prova in Itinere del 30 Gennaio 2014

1. Si definisca formalmente il concetto di indistinguibilità contro attacchi a crittotesto scelto **ind-cca** per cifrari asimmetrici.
2. Si consideri la seguente variante del cifrario El Gamal.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave prende in input un parametro k e sceglie due primi q, p tali che $|q| = k$ e q divide $p - 1$. Quindi procede come segue. Detto G un sottogruppo di \mathbb{Z}_p^* di ordine q , pone $\mathcal{M} = \{1, \dots, 100\}$ come spazio dei messaggi. Quindi sceglie un generatore g di G , sceglie (a caso secondo la distribuzione uniforme) $x \in \{1, \dots, q\}$ e pone $h = g^x \bmod p$. Infine, restituisce $PK = (p, q, g, h, \mathcal{M})$ come chiave pubblica e $SK = x$ come chiave privata.

Algoritmo di cifratura

```
Enc( $PK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \{1, \dots, q\}; C_1 \leftarrow g^r \bmod p;$ 
   $C_2 \leftarrow h^r g^m \bmod p$ 
  return  $(C_1, C_2)$ .
```

Algoritmo di decifratura

```
Dec( $SK, C_1, C_2$ )
   $A \leftarrow C_1^x \bmod p;$ 
   $Y \leftarrow C_2/A \bmod p$ 
   $m = 0$ 
  For  $i = 1$  to  $100$  do
    if  $(Y = g^i \bmod p)$   $m \leftarrow i$ ;
  If  $m \notin \mathcal{M}$  return  $\perp$ 
  else return  $m$ 
```

Dimostrare che tale cifrario non è sicuro in senso ind-cca.

3. Descrivere dettagliatamente il cifrario RSA-OAEP. In particolare, descrivere l'algoritmo di generazione della chiave, l'algoritmo di cifratura e l'algoritmo di decifratura.

4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si considerino, inoltre, due gruppi G e G_T per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine p). Siano g, h due generatori di G . L'algoritmo procede scegliendo $x \in \{1, \dots, p-1\}$ (a caso) e ponendo $T = g^x$. La chiave pubblica è quindi $VK = (T, h, g, e, G, G_T)$, mentre la chiave privata è $SK = x$. Lo spazio dei messaggi è $\mathcal{M} = \{1, \dots, p-1\}$.

Algoritmo di firma

```

Sign( $SK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
  Sia  $t \leftarrow_R \mathbb{Z}_p$ 
   $\sigma_1 = h^t$ 
   $\sigma_2 = (h^t g^m)^x$ 
  return  $(\sigma_1, \sigma_2)$ 

```

Algoritmo di verifica

```

Verify( $VK, m, (\sigma_1, \sigma_2)$ )
  If  $\sigma_1 = \perp \vee \sigma_2 = \perp$  return 0;
  If  $e(\sigma_2, g) = e(\sigma_1 g^m, T)$  return 1
  else return 0

```

Si noti che una firma valida viene sempre verificata correttamente. Infatti per la bilinearità di e si ha

$$e(\sigma_2, g) = e((h^t g^m)^x, g) = e(h^t g^m, T) = e(\sigma_1 g^m, T)$$

m^o2

$$M = \{1, \dots, 100\}$$

$$\text{Enc}(m) = (g^{n \bmod p}, h^n \cdot g^{m \bmod p});$$

$$m_0 = 1, m_1 = 2$$

$$C_2' \leftarrow C_2 \cdot g = h^n \cdot g^{m+1} \bmod p$$

$$\text{Dec}(C_1, C_2) = m + 1$$

A(PK):

$$m_0 = 1;$$

$$m_1 = 2;$$

$$(C_1, C_2) \leftarrow \text{OEnc}(m_0, m_1);$$

$$C_2' \leftarrow C_2 \cdot g;$$

$$m \leftarrow \text{ODec}(C_1, C_2');$$

if $m - 1 = m_1$ return 1;

else return 0;

$$\text{Adv}^{\text{ind-CCA}}(A) = 1;$$

m°3

IN RSA-OAEP SI CONSIDERA UN PARAMETRO $K \in K_0 + K_1 < K$
TALE CHE $m = K - K_0 - K_1$ E $\Pi = \{0,1\}^m$ È LO SPAZIO DEI
MESSAGGI. INOLTRE SONO PRESENTI DUE FUNZIONI HASH
(RANDOM ORACCE) COSÌ DEFINITE:

$$G: \{0,1\}^{K_0} \rightarrow \{0,1\}^{m+K_1}$$
$$H: \{0,1\}^{m+K_1} \rightarrow \{0,1\}^{K_0}$$

KeyGen(κ):

$((N, e), (N, p, q, d)) \leftarrow \text{KeyGen}_{\text{RSA}}(\kappa);$
return $((N, e), (N, p, q, d));$

Enc(PK, m): // $m \in \Pi$

$r \in_R \{0,1\}^{K_0};$
 $s \leftarrow G(r) \oplus (m || 0^{K_1});$
 $t \leftarrow H(s) \oplus r;$
 $y \leftarrow \text{Slt}(t);$
 $c \leftarrow \text{RSA}(y);$
return $c;$

Dec(SK, c):

$y \leftarrow \text{RSA}^{-1}(c); // y = \text{Slt}(t);$
 $r \leftarrow H(s) \oplus t;$
 $m' \leftarrow s \oplus G(r);$
 $m' \parallel e = m';$
if $e == 0^K$ return $m;$
else return $\perp;$

m° 5

$|G| = |G_r| = p$ $g, h \in G$ sono suoi GENERATORI
 $V_N = (\tau, h, g, e, G, G_r)$ $S_N = X$ $H = \{1, \dots, p-1\}$

$$e(\sigma_2, g) = e(\sigma_2 \cdot g^m, g^\times)$$

A(V_N):

return(1, 1, τ);

CONSIDERANDO $m=1$:

$$\begin{aligned} e(\tau, g) &= e(1 \cdot g^1, \tau) \\ e(g, g)^\times &= e(g, g)^\times \quad \checkmark \end{aligned}$$

$$\text{Adr}^{\text{uf-cma}}(A) = 1$$

Corso di Crittografia

Prova in Itinere del 29 Gennaio 2016

1. Si definisca formalmente il concetto di indistinguibilità `ind-id-cpa` per cifrari basati sull'identità.
2. Si fornisca lo pseudo-codice e si spieghi il funzionamento dell'algoritmo Miller-Rabin.
3. Si consideri la seguente variante del cifrario Paillier.

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave è identico a quello del cifrario Paillier. Esso quindi restituisce in output un modulo $N = pq$; la chiave segreta è (p, q) , lo spazio dei messaggi è \mathbb{Z}_N .

Algoritmo di cifratura. Detto ENC_P l'algoritmo di cifratura del cifrario Paillier, l'algoritmo è il seguente.

```
Enc( $N, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \mathbb{Z}_N$ ;  $C_1 \leftarrow (m - r) \bmod N$ ;
   $C_2 \leftarrow \text{ENC}_P(r)$ ;
  return  $(C_1, C_2)$ 
```

Algoritmo di decifratura. Detto DEC_P l'algoritmo di decifratura del cifrario Paillier, l'algoritmo è il seguente.

```
Dec( $SK, C_1, C_2$ )
   $r \leftarrow \text{DEC}_P(C_2)$ ;
   $m \leftarrow C_1 + r \bmod N$ ;
  return  $m$ 
```

Dimostrare che tale cifrario non è sicuro in senso ind-cca.

4. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
5. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si considerino, inoltre, due gruppi G e G_T per i quali è disponibile una funzione bilineare $e : G \times G \rightarrow G_T$ (sia G che G_T si suppongono essere gruppi ciclici aventi ordine p). Siano g, h due generatori di G . L'algoritmo procede scegliendo $x \in \{1, \dots, p-1\}$ (a caso) e ponendo $T = g^x$. La chiave pubblica è quindi $VK = (T, h, g, e, G, G_T)$, mentre la chiave privata è $SK = x$. Lo spazio dei messaggi è $\mathcal{M} = \{1, \dots, p-1\}$.

Algoritmo di firma

```

Sign( $SK, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
  Sia  $t \leftarrow_R \mathbb{Z}_p^*$ 
   $\sigma = (h^t g^m)^x$ 
  return  $(\sigma, t)$ 

```

Algoritmo di verifica

```

Verify( $VK, m, (\sigma_1, \sigma_2)$ )
  If  $\sigma = \perp \vee t \notin \mathbb{Z}_p^*$  return 0;
  If  $e(\sigma, g) = e(h^t g^m, T)$  return 1
  else return 0

```

$$\sigma_1 \leftarrow (h^{t_1} g^{m_1})^x \quad (\sigma_1, t_1)$$

m°2

L'ALGORITMO DI MILLER RABIN È UN TEST DI PRIMALITÀ PROBABILISTICO, ovvero:

- SE L'OUTPUT RESTITUISCE COMPOSITO, ALLORA IL VALORE IN INPUT È COMPOSITO CON PROBABILITÀ 1.
- SE RESTITUISCE PRIMO, ALLORA LA PROBABILITÀ DI AVER FATTO UN ERRORE È 2^{-2^S} , CON S IL NUMERO DI VOLTE CHE ESEGUIANO IL TEST.

L'ALGORITMO È COSÌ DEFINITO:

MILLER-RABIN(p): // p DEVE ESSERE DISPARI

$$p = 2^k \cdot m + 1; \quad (m \text{ dispari})$$

$$a \in \{1, \dots, p-1\}$$

$$b = a^m \bmod p;$$

if $b == 1 \bmod p$ return "PRIMO";

for $i=0$ to $k-1$:

if $b == -1 \bmod p$ return "PRIMO";

else:

$$b = b^2 \bmod p;$$

return "COMPOSITO";

$a^{p-1} \equiv 1 \bmod p$ SE QUESTO È VERO ALLORA p È PRIMO.

PER FARE CIÒ L'ALGORITMO SELEZIONA A CASUALMENTE IN $\{1, \dots, p-1\}$ E CONSIDERIAMO $a^m \bmod p$ SE VACE SIN DA SUBITO CHE $a^m \equiv 1 \bmod p$ ALLORA L'ALGORITMO TORNA PRIMO. ALTRIMENTI PROVEREMO:

$$a^{2^m} \bmod p \stackrel{?}{=} 1$$

SE UNA DEGLIE DUE RADICI È -1 ALLORA TORNERETTO "PRIMO";
ALTRIMENTI SI PROSEGUITA FINO A:

$$\alpha^{2^m} = \alpha^{p-1} \equiv 1 \pmod{p}$$

SE ANCHE IN QUESTO CASO NESSUNA DEGLIE DUE RADICI È
1 o -1 ALLORA SIAMO CERTI CHE SIA COMPOSITO.

n°3

$$c_1 = (m - n) \bmod N$$

$$c_2 = (1 + rN) \cdot y^N \bmod N^2$$

A (PK) :

$$m_0, m_2 \in \mathbb{Z}_N^*;$$

$$(c_1, c_2) \leftarrow \text{OEnc}_{\text{PK}}(m_0, m_2);$$

$$c'_1 \leftarrow c_1 + 1;$$

$$m' \leftarrow \text{ODEcs}_{\text{A}}(c'_1, c_2);$$

if $m' - 1 = m_0$ return 1;

else return 0;

$$\text{Adv}^{\text{ind-CCA}}(\text{A}) = 1$$

m° 5

Dobbiamo gestire $\sigma \neq t$

$$e(\sigma, g) = e(h^t g^m, g^\times)$$

Poniamo $t = p-1$, $\sigma = T$ e $m = 1$;

$$e(T, g) = e(g, T)$$

A($\vee\wedge$):

$$t \leftarrow p-1;$$

$$m \leftarrow 1;$$

$$\sigma \leftarrow T;$$

return ($m, (\sigma, t)$);

$$\text{Adv}^{\text{uf-Cma}}(A) = 1$$

SQUARE - DH

ESP^{SQUARE}(A) : g, G, p
 $a \leftarrow \mathbb{Z}_p^*$
 $g_1 \leftarrow g^a$;
 $z \leftarrow A(g_1)$
if ($z == g_1^{a^2}$) return 1;
else return 0;

B DEVE USARE A SDH

$B(g, g^a, g^b) : \rightarrow g^{ab}$

$$A(g^{a+b}) = g^{a^2 + b^2 + 2ab};$$

$$A(g^a) = g^{a^2};$$

$$A(g^b) = g^{b^2};$$

IPOTESI DI EVENTI INDEPENDENTI

RSA

Alice e Bob N
 ℓ_1 ℓ_2 $\gcd(\ell_1, \ell_2) = 1$
 d_1 d_2

$$m^{\ell_1} \bmod N \quad m^{\ell_2} \bmod N$$

oSCAR ANDA m:

$$\begin{aligned} c_1 &= m^{\ell_1} \bmod N \\ c_2 &= m^{\ell_2} \bmod N \quad (\ell_1, \ell_2) \\ c_1, c_2 &\end{aligned}$$

$$\lambda, \mu \quad \lambda \ell_1 + \mu \ell_2 = 1$$

$$\begin{aligned} c_1^\lambda \cdot c_2^\mu &= (m^{\ell_1})^\lambda \cdot (m^{\ell_2})^\mu \bmod N = \\ &= m \bmod N\end{aligned}$$

SQUARE AND MULTIPLY

$$x^e \bmod N$$

$$e = e_n \cdot 2^n + e_{n-1} \cdot 2^{n-1} + \dots + e_1 \cdot 2 + e_0$$

$$x^e = (x^{2^n})^{e_n} \cdot (x^{2^{n-1}})^{e_{n-1}} \cdot \dots \cdot (x^2)^{e_1} + x^{e_0}$$

1. SE $e_n = 1$ ALLORA \downarrow

2. SE $e_i = 0$ SI EFFETTUO A UN QUADRATO;

3. SE $e_i = 1$ SI EFFETTUO UN QUADRATO E UNA MOLTIPLICAZIONE PER x .

S-M(x, e):

$$d = 1$$

for i=n down to 0 do:

$$d = d^2 \bmod N;$$

if $e_i = 1$ then:

$$d = d \cdot x \bmod N;$$

return d;

Corso di Crittografia

Prova del 20 Gennaio 2021 – Gruppo A

1. Si definisca formalmente il concetto di indistinguibilità contro attacchi a messaggio scelto **ind-id-cpa** per cifrari basati sull'identità.
2. Si consideri il seguente cifrario asimmetrico

Algoritmo di Generazione della Chiave. L'algoritmo di generazione della chiave è analogo a quello della funzione RSA. Esso restituisce in output un modulo $N = pq$, e un intero e tale che $\text{MCD}(\phi(N), e) = 1$.

Lo spazio dei messaggi è $\mathcal{M} = \{0, 1, 2, 3, 4, 5\}$. La chiave pubblica è $\mathbf{pk} = (N, e, \mathcal{M})$. La chiave privata è $\mathbf{sk} = d$, tale che $ed = 1 \pmod{\phi(N)}$.

Algoritmo di cifratura. L'algoritmo riceve in input un messaggio m

```
Enc( $\mathbf{pk}, m$ )
  If  $m \notin \mathcal{M}$  return  $\perp$ 
   $r \leftarrow_R \mathbb{Z}_N^*$ ;  $C_1 \leftarrow r^e \pmod{N}$ ;
   $C_2 \leftarrow (r + m)^e \pmod{N}$ ;
  return  $(C_1, C_2)$ 
```

Algoritmo di decifratura. L'algoritmo di decifratura è il seguente.

```
Dec( $\mathbf{sk}, (C_1, C_2)$ )
  If  $(C_1 \notin \mathbb{Z}_N^* \text{ or } C_2 \notin \mathbb{Z}_N^*)$  return  $\perp$ 
   $r \leftarrow C_1^d \pmod{N}$ ;
   $a \leftarrow C_2^d \pmod{N}$ ;
  Output  $(a - r) \pmod{N}$ ;
```

Dimostrare che tale cifrario non è sicuro in senso ind-cpa.

3. Definire formalmente il concetto di sicurezza (ovvero non falsificabilità relativamente ad attacchi a messaggio scelto) per schemi di firma digitale.
4. Si dimostri che il seguente schema di firma non è sicuro.

Algoritmo di Generazione delle Chiavi. L'algoritmo prende in input un parametro k e sceglie un primo p tale che $|p| = k$. Si consideri, inoltre, un gruppo G ciclico avente ordine p . Sia g un generatore di G . L'algoritmo procede scegliendo $x \in \{1, \dots, p-1\}$ (a caso) e ponendo $h = g^x$. Sia inoltre

$H : \{0, 1\}^* \rightarrow \{1, \dots, p - 1\}$ una funzione hash resistente alle collisioni. La chiave pubblica è quindi $VK = (H, h, g, G)$, mentre la chiave privata è $SK = (x)$. Lo spazio dei messaggi è $\mathcal{M} = \{0, 1\}^*$ (i messaggi sono stringhe di bit di lunghezza arbitraria).

Algoritmo di firma

```

Sign( $SK, m$ )
   $r \leftarrow_R \mathbb{Z}_p^*$ 
   $R \leftarrow g^r;$ 
   $s \leftarrow (r + H(m)x) \bmod p;$ 
  return  $\sigma = (R, s)$ 

```

Algoritmo di verifica

```

Verify( $VK, m, (\sigma)$ )
  Sia  $\sigma = (R, s)$ ;
  If  $Rh^{H(m)} = g^s$  return 1
  else return 0

```

5. Si consideri il seguente problema computazionale, che chiameremo Quadratic-Diffie-Hellman (QDH), definito su un gruppo ciclico G avente ordine primo q . Sia g un generatore di G , definiamo il seguente esperimento

```

EspQDH $G, g$ ( $\mathcal{B}$ )
   $a \leftarrow_R \mathbb{Z}_q^*$ ;  $g_1 \leftarrow g^a$ ;
   $y \leftarrow \mathcal{B}(g_1)$ ;
  If ( $y = g^{a^2}$ ) return 1 else return 0

```

Il vantaggio di \mathcal{B} è definito come

$$\text{Adv}_{G,g}^{\text{QDH}} = \Pr [\text{Esp}_{G,g}^{\text{QDH}}(\mathcal{B}) = 1]$$

Si dimostri che, se esiste un avversario \mathcal{B} capace di risolvere il problema QDH in G , tale avversario può essere sfruttato per risolvere il problema computazionale Diffie-Hellman studiato a lezione.

//Suggerimento: Si noti che g è noto a tutti. Si sfrutti la formula del quadrato di un binomio.

$m \circ 2$

$$N = pq, e \in \mathbb{Z}_{\phi(N)}^*, M = \{0, 1, 2, 3, 4, 5\}$$

$$r \in \mathbb{Z}_n^*$$

$$c_1 \leftarrow r^e \bmod N$$

$$c_2 \leftarrow (r + m)^e \bmod N$$

$$m_0 = 1, m_1 = 0$$

$$\text{Enc}(m_1) = (r^e, r^e)$$

$$\text{Enc}(m_0) = (r^e, (r+1)^e)$$

A(AE):

$$m_0 = 1;$$

$$m_1 = 0;$$

$$(c_1, c_2) \leftarrow \text{O}_{\text{Enc}}(m_0, m_1);$$

if ($c_1 == c_2$) return 1;

else return 0;

$$\text{Adv}^{\text{ind-CPA}}(A) = 1$$

m^o4

$$g \in G, h = g^{\times}, H: \{0,1\}^* \rightarrow \{1, \dots, p-1\}$$
$$\sigma = (R, s) \quad R = g^{\textcircled{n}} \rightarrow n \in \mathbb{Z}_p^*$$
$$R^{h^{H(m)}} = g^s$$

A(vk):

$$m \leftarrow \{0,1\}^*$$

$$a \leftarrow h^{H(m)}$$

$$R \leftarrow a^{-2} // h^{-H(m)}$$

$$s = 0;$$

return (m, (R, s));

$$\text{Adv}^{\text{uf-cma}}(A) = 1$$

$m^o S$

$|G| = q$, $g \in G$ GENERATORE

H_P: $\exists B$ CHE RISOLVE QDH

T_S: $\exists A$ CHE RISOLVE CDH

A(g^x, g^y):

$$g^z \leftarrow B(g^x, g^y); // g^{x^2 + y^2 + 2xy}$$

$$g^{x^2} \leftarrow B(g^x);$$

$$g^{y^2} \leftarrow B(g^y);$$

$$g^{xy} = [g^z / g^{x^2} \cdot g^{y^2}]^{\frac{1}{2}};$$

return $g^{xy};$

$$\text{Ad}_v(A) = \text{Ad}(B)^3 \Rightarrow \underline{\text{TESI}}$$