Introduzione alla crittografia asimmetrica

Contrariamente a quanto stabilito con la crittografia simmetrica, che prevede lo scambio di una chiave fra le due parti che intendono comunicare, la crittografica asimmetrica si basa sul concetto di usare una coppia di chiavi, pubblica e privata, proposto da Diffie-Hellman nel 1976. Perché lo schema funzioni deve essere impossibile, con delle risorse limitate, ricavare l'altra chiave conoscendone solo una. Tuttavia questa limitazione ci pone subito in un ambito che esclude il raggiungimento della perfetta sicurezza.
La crittografia a chiave pubblica si basa quindi sul gap computazionale tra algoritmi efficienti per determinate operazioni e intrattabilità delle operazioni inverse in assenza della chiave privata. Mentre molte delle nozioni ad alto livello continueranno ad essere valide in questo ambito, la costruzione degli schemi e gli strumenti teorici utilizzati per dimostrare la loro efficacia sono profondamente diversi, basandosi molto spesso sull'intrattabilità computazionale.

Funzioni unidirezionali

La crittografia asimmetrica si basa sul concetto di funzioni unidirezionali, cioè funzioni che siano facili da calcolare ma molto difficili da invertire. Più formalmente, definita μ come una funzione trascurabile, per cui $\forall c \geq 0,\ \exists k_c : \mu(k) \leq k^{-c} \quad \forall k \geq k_c$, una funzione $f : \{0,1\}^* \to \{0,1\}^*$ è una unidirezionale se

- esiste un algoritmo efficiente che dato un $x \in \{0,1\}^*$ calcoli $f(x)$
- per ogni algoritmo efficiente A esiste una funzione trascurabile μ_A tale che per k sufficientemente grande $Pr[f(z) = y : x \in_R \{0,1\}^*; f(x) = y; A(k,y) = z] \leq \mu_A(k)$. In altre parole, la probabilità di invertire la funzione diventa trascurabilmente piccola

Funzioni unidirezionali trapdoor

Si tratta di funzioni unidirezionali con la seguente proprietà: se si conosce una specifica informazione (trapdoor) esse divengono facili da invertire. Si tratta di primitive estremamente interessanti, poiché partendo da esse è possibile costruire cifrari in maniera semplice.

Schema di cifratura asimmetrico

Uno schema di cifratura asimmetrico è simile ad uno simmetrico se non per la presenza di due chiavi, una pubblica (**pk**) ed una privata o segreta (**sk**). La prima è usata per cifrare il messaggio, ed essendo pubblica questa operazione è consentita a chiunque senza particolari restrizioni, mentre la seconda, custodita gelosamente, permette di decifrare i messaggi cifrati con la corrispondente **sk**.

Uno schema di cifratura asimmetrico $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ è una tripla di funzioni con i seguenti ruoli:

- $\mathcal{K} : \emptyset \to (\{0,1\}^*, \{0,1\}^*)$ è una funzione che non prende alcun input e restituisce una coppia di valori, cioè le chiavi **sk** e **pk**
- $\mathcal{E} : \{0,1\}^* \times M \to C$ è una funzione che prende in input una chiave pubblica **pk** ed un messaggio e restituisce il corrispondente crittotesto. Può essere randomizzato ma non a stati, poiché la comunicazione può giungere da più fonti e una sincronizzazione fra tutti i mittenti sarebbe inutilmente complicata se non irrealizzabile
- $\mathcal{D} : \{0,1\}^* \times C \to M$ è una funzione che prende in input una chiave privata **sk** ed un crittotesto $\neq \perp$ e restituisce il messaggio originale corrispondente. Deterministico e privo di stati

Lo spazio dei messaggio associato a **pk** è $M = \{m : \mathcal{E}_{pk}(m) \neq \perp\}$. L'algoritmo di decifratura è corretto se $\forall m \in M, \forall c \in \mathcal{E}_{pk}(m) \quad \mathcal{D}_{sk}(c) = m$.

Nozioni di sicurezza

Le due nozioni di sicurezza relative agli schemi di cifratura simmetrici erano relativi ad attacchi di tipo **cpa** [1] e **cca** [2]. Nel caso dei cifrari asimmetrici, si può fare una simile distinzione, tenendo però a mente che l'avversario ha sempre a disposizione **pk**, ed è quindi in grado di produrre il crittotesto di un messaggio in completa autonomia. Nel contesto asimmetrico, i **cca** diventano quindi ancora più rilevanti e più complessi da neutralizzare.

Sicurezza contro attacchi a messaggio scelto (ind-cpa)

Si fissi uno schema di cifratura asimmetrico $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ e si definisca una funzione $LR(x_0, x_1, b) = \begin{cases} x_0 & \text{se } b = 0 \\ x_1 & \text{se } b = 1 \end{cases}$

Si consideri un avversario A che riceve come input la **pk**. L'avversario A ha disposizione un oracolo $\mathcal{E}_{pk}(LR(.,.,b))$, che cifra uno dei due messaggi che A fornisce come input, a seconda del bit b.
L'obiettivo di A è fornire in output un bit che rappresenta il suo tentativo di indovinare il bit utilizzato dalla funzione LR.

Viene quindi definita la seguente coppia di esperimenti:

```
ESP_{AE}^{ind-cpa-1}(A):
    (pk, sk) <-R- KeyGen()
    b <- A^{Enc_{pk}}(LR(.,.,1))
    return b

ESP_{AE}^{ind-cpa-0}(A):
    (pk, sk) <-R- KeyGen()
    b <- A^{Enc_{pk}}(LR(.,.,0))
    return b
```

La nozione di vantaggio in senso **ind-cpa** dell'avversario A è così definita:

$$Adv_{\mathcal{AE}}^{ind-cpa}(A) = |Pr[ESP_{\mathcal{AE}}^{ind-cpa-1}(A) = 1] - Pr[ESP_{\mathcal{AE}}^{ind-cpa-0}(A) = 1]|$$

Lo schema di cifratura asimmetrico $\mathcal{AE}$ è sicuro in senso **ind-cpa** se il vantaggio di ogni possibile avversario polinomialmente limitato è prossimo a 0.

Sicurezza contro attacchi a crittotesto scelto (ind-cca)

Si fissi uno schema di cifratura asimmetrico $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ e si definisca una funzione $LR(x_0, x_1, b) = \begin{cases} x_0 & \text{se } b = 0 \\ x_1 & \text{se } b = 1 \end{cases}$

Si consideri un avversario A che riceve come input la **pk**. L'avversario A ha disposizione due oracoli: il primo $\mathcal{E}_{pk}(LR(.,.,b))$, che cifra uno dei due messaggi che A fornisce come input, a seconda del bit b, il secondo $\mathcal{D}_{pk}(.)$ che prende in input un crittotesto e restituisce il messaggio che ha lo ha originato.
L'obiettivo di A è fornire in output un bit che rappresenta il suo tentativo di indovinare il bit utilizzato dalla funzione LR.
Al fine di evitare soluzioni banali, si stabilisce che A imbroglia se chiede al secondo oracolo la decifratura dell'output ottenuto dal primo.

Viene quindi definita la seguente coppia di esperimenti:

```
ESP_{AE}^{ind-cca-1}(A):
    (pk, sk) <-R- KeyGen()
    b <- A^{Enc_{pk}}(LR(.,.,1)), Dec_{sk}(.)}
    if A cheated:
        return 0
    return b

ESP_{AE}^{ind-cca-0}(A):
    (pk, sk) <-R- KeyGen()
    b <- A^{Enc_{pk}}(LR(.,.,0)), Dec_{sk}(.)}
    if A cheated:
        return 0
    return b
```

La nozione di vantaggio in senso **ind-cca** dell'avversario A è così definita:

$$Adv_{\mathcal{AE}}^{ind-cxa}(A) = |Pr[ESP_{\mathcal{AE}}^{ind-cca-1}(A) = 1] - Pr[ESP_{\mathcal{AE}}^{ind-cca-0}(A) = 1]|$$

Lo schema di cifratura asimmetrico $\mathcal{AE}$ è sicuro in senso **ind-cca** se il vantaggio di ogni possibile avversario polinomialmente limitato è prossimo a 0.

Non malleabilità

La sicurezza in senso **ind-cca** garantisce anche la non malleabilità dello schema di cifratura. Si consideri infatti M lo spazio dei messaggi ed una funzione $F : M \to M$ facilmente computabile. Se lo schema $\mathcal{AE}$ fosse malleabile rispetto ad F, chiunque potrebbe calcolare $\mathcal{E}_{pk}(F(m))$ dati $\mathcal{E}_{pk}(m)$ ed F. Si può quindi definire una funzione $MAUL(\mathcal{E}_{pk}(m), F) = \mathcal{E}_{pk}(F(m))$.

Per vedere come questo fatto rompe l'esperimento, si consideri il seguente avversario:

```
A(AE, F):
    (m_0, m_1) <-R- M
    c <- O_E(m_0, m_1)
    c` <- MAUL(c, F)
    m <- O_D(c`)
    if m == m_0:
        return 0
    return 1
```

Vantaggio legato a query multiple

Negli esperimenti precedenti l'avversario ha la possibilità di effettuare un numero arbitrario q_e di domande ad ogni oracolo. Si dimostra il seguente lemma:

$$\exists A : \text{avversario che fa una singola domanda con tempo di esecuzione } t_e$$
$$\exists B : \text{avversario che fa } q_e \text{ domande con tempo di esecuzione } t_e$$
$$Adv_{\mathcal{AE}}^{ind-cpa}(B) \leq q_e \cdot Adv_{\mathcal{AE}}^{ind-cpa}(A)$$

Lo stesso dicasi per **ind-cca**. Ne segue che un cifrario asimmetrico per il quale è autorizzata una sola domanda ha una sicurezza paragonabile a quella di un sistema in cui sono autorizzate un numero arbitrario di domande.

Cifratura ibrida

Nonostante si abbiano a disposizione cifrari asimmetrici molto sofisticati, nella pratica l'approccio che si preferisce è quello di una cifratura ibrida, che utilizza il cifrario asimmetrico per lo scambio della chiave per il cifrario simmetrico che verrà utilizzato da quel momento per l'effettiva comunicazione di messaggi.

Dati due schemi di cifratura $\mathcal{AE} = (\mathcal{K}^a, \mathcal{E}^a, \mathcal{D}^a), \mathcal{SE} = (\mathcal{K}^s, \mathcal{E}^s, \mathcal{D}^s)$ rispettivamente asimmetrico e simmetrico, si può definire uno schema di cifratura ibrido $\mathcal{HE} = (\mathcal{K}^a, \mathcal{E}^h, \mathcal{D}^h)$ che si comporta nella seguente maniera:

```
Enc^h(pk, m):
    k <-R- K^s()
    c_s <- E^s(k, m)
    c_a <- E^a(pk, k)
    return (c_s, c_a)

Dec^h(sk, (C_s, c_a)):
    k <- Dec^a(sk, c_a)
    m <- Dev^s(k, c_s)
    return m
```

Il motivo di questa scelta è da ricercarsi nel costo in termini di tempo delle operazioni da affrontare, che nel caso di cifratura asimmetrica, sono ben più dispendiose.

Sicurezza contro attacchi a messaggio scelto (cpa)

Siano $\mathcal{AE} = (\mathcal{K}^a, \mathcal{E}^a, \mathcal{D}^a), \mathcal{SE} = (\mathcal{K}^s, \mathcal{E}^s, \mathcal{D}^s)$ due schemi di cifratura asimmetrico e simmetrico e $\mathcal{HE} = (\mathcal{K}^a, \mathcal{E}^h, \mathcal{D}^h)$ lo schema ibrido risultante dalla loro combinazione.

$$\exists A_{00,01} : \text{avversario che attacca } \mathcal{AE} \text{ con } q \text{ domande}$$
$$\exists A_{11,10} : \text{avversario che attacca } \mathcal{AE} \text{ con } q \text{ domande}$$
$$\exists A : \text{avversario che attacca } \mathcal{SE} \text{ con una sola domanda}$$
$$\exists B : \text{avversario che attacca } \mathcal{HE} \text{ in senso ind-cpa}$$
$$Adv^{ind-cpa}_{\mathcal{HE}}(B) \leq Adv^{ind-cpa}_{\mathcal{AE}}(A_{00,01}) + Adv^{ind-cpa}_{\mathcal{AE}}(A_{11,10}) + Adv^{ind-cpa}_{\mathcal{SE}}(A)$$

Poiché l'avversario A è limitato ad una sola domanda, anche se lo schema di cifratura simmetrico non è particolarmente forte, lo schema ibrido potrebbe continuare ad essere sicuro. Un discorso simile può essere fatto per la sicurezza in senso **ind-cca**.

Schema El Gamal

Sia G un gruppo ciclico con generatore g, per cui $G = \{g^0, g^1, \ldots, g^{n-1}\}$, dove $n = |G|$ è l'ordine di G.

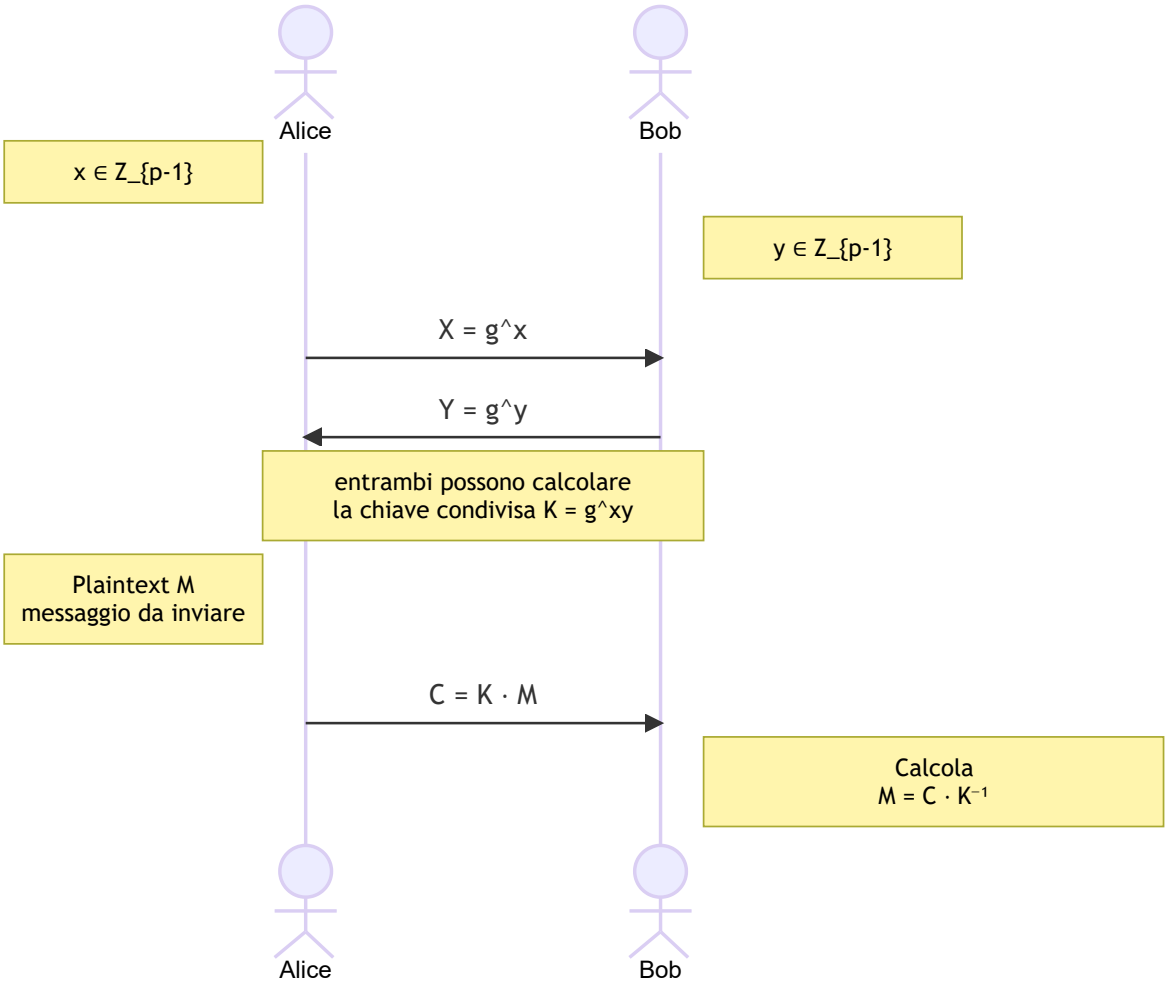
Si ricordino le funzioni

$$DExp_{G,g} : Z_n \to G$$
$$DExp(x) = g^x$$

$$DLog_{G,g} : G \to Z_n$$
$$DLog_{G,g}(g^x) = x$$

La funzione $DExp_{G,g}$ è unidirezionale, stando alla congettura che ipotizza $DLog_{G,g}$ sia non trattabile nei giusti gruppo G, come ad esempio $G = Z_p^*$, dove p è un numero primo e $p - 1$ ha un fattore di scomposizione molto grande.

Ipotizziamo quindi uno scambio di messaggi in cui il mittente e il destinatario si procurano entrambi due valori $x, y \in Z_{p-1}$. Entrambi calcolano poi rispettivamente $X = g^x, Y = g^y$ e si scambiano questi risultati così da poter calcolare $K = g^{xy}$, avendo tutti gli strumenti per farlo. Quando il mittente vuole mandare un messaggio $M \in G$, gli basta calcolare $C = M \cdot K$ ed inviare questo valore al destinatario, che potrà recuperare il messaggio originale calcolando $M = C \cdot K^{-1}$.



In questo schema un potenziale attaccante può avere accesso solo ai valori X, Y, C. Poiché abbiamo ipotizzato che $DLog_{G,g}$ sia intrattabile in questo gruppo, non può utilizzare X, Y per calcolare x, y. Se anche il `problema computazionale Diffie-Hellman` [3] è intrattabile, cosa che possiamo assumere dato che l'unico modo per risolverlo sembra essere quello di risolvere il `logaritmo discreto`, allora l'avversario non può nemmeno ottenere K conoscendo solo X, Y.

Definizione El-Gamal

Sia G un gruppo ciclico di ordine n e con generatore g. Lo schema di cifratura asimmetrico **El-Gamal** $\mathcal{AE}_{EG} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ è costruito nella seguente maniera:

```
K():
    x <-R- Z_n
    X <- g**x
    return (X, x)

E(X, M): /* con X chiave pubblica del mittente - M messaggio da inviare */
    if M not in G:
        return "error"
    y <-R- Z_n
    Y <- g**y
    K <- X**y
    C <- M * K
    return (Y, C)

D(x, Y, C): /* con x esponente privato - Y e C ricevuti dal mittente */
    K <- Y**x
    M <- C/K
    return M
```

Lo schma **El-Gamal** è sicuro in senso **ind-cpa** se si utilizza un gruppo in cui il `problema decisionale Diffie-Hellman`[4] è intrattabile. Per le sue proprietà di malleabilità rispetto alla moltiplicazione, invece, non può mai essere sicuro in sendo **ind-cca**.

Schema Pallier

Preliminari matematici

Un elemento $y \in Z^*_{N^2}$ e' detto N-residuo $\mod N^2$ se $\exists x \in Z^*_{N^2} : y \equiv x^N \mod N^2$. Un elemento N-residuo ammette N radici N-esime distinte.

Si consideri l'insieme $T = \{(1 + xN) \mod N^2 : x \in Z_N\}$. Ogni elemento $z \in T : z^N \equiv 1 \mod N^2$.
L'ordine di $Z^*_{N^2}$ è $\phi(N^2) = (p^2 - p)(q^2 - q) = \phi(N) \cdot N$. Dunque $\forall x \in Z_{N^2}$ si ha che $x^{\phi(N^2)} \equiv x^{\phi(N)\cdot N} \equiv 1 \mod N^2$.

Ogni elemento $y \in Z^*_{N^2}$ può inoltre essere scritto come $(1 + xN)w^N$, con $x \in Z_N, w \in Z^*_{N^2}$.
Questo ci consente di divirere $Z^*_{N^2}$ in classi di equivalenza $a \equiv b$ se ab^{-1} è un N-Residuo in $Z^*_{N^2}$.

Problema della N-residuosità

Dato un elemento $w \xleftarrow{\$} Z^*_{N^2}$, determinare se w è un elemento N-residuo o meno è il `problema della N-residuosità`. Si congettura che, se la fattorizzazione di N è ignota, il problema è intrattabile.

Per descrivere formalmente questo problema si può utilizzare il seguente coppia di esperimenti.

Sia $G = Z^*_{N^2}$ un gruppo ciclico di ordine $m = \phi(N^2) = \phi(N) \cdot N$ e g un generatore di G. Si consideri quindi l'elemento $x \xleftarrow{\$} G$.

Un avversario A riceve come input, a seconda del mondo in cui si trova, un input W così definito:

$$W = \begin{cases} x^N \mod N^2 & \text{se siamo nel ESP-1} \\ \xleftarrow{\$} Z^*_{N^2} & \text{se siamo nel ESP-0} \end{cases}$$

L'obiettivo di A è quello di determinare in quale esperimento si trova tramite un bit di output.

```
ESP_N^{DCRA-1}(A):
    x <-R- G
    w <- x**N % N**2
    d <- A(w)
    return d

ESP_N^{DCRA-1}(A):
    w <-R- G
    d <- A(w)
    return d
```

La nozione di vantaggio in senso **dcra** dell'avversario A è così definita:

$$Adv^{dcra}_N(A) = |Pr[ESP^{dcra-1}_N(A) = 1] - Pr[ESP^{dcra-0}_N(A) = 1]|$$

Risolvere il `problema della N-residuosità` è intrattabile se il vantaggio di ogni possibile avversario polinomialmente limitato è prossimo a 0.

Questo non è vero se l'avversario conosce la fattorizzazione di N, da cui può ricavare $\phi(N)$, valore che è possibile usare per discernere facilmente i valori N-Residui.

Definizione Paillier

Sia $G = Z^*_{N^2}$ un gruppo con generatore g. Lo schema di cifratura asimmetrico **Paillier** $\mathcal{AE}_{EG} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ è costruito nella seguente maniera:

$\mathcal{K}$ è molto simile ad **RSA**, in quanto non prende input e restituisce un $N = pq$, con N che diventa la chiave pubblia e p, q mantenuti segreti. Non serve calcolare alcun esponente aggiuntivo, al contrario che con **RSA**.
Lo spazio dei messaggi è Z_n, lo spazio dei crittotesti $Z^*_{N^2}$.

```
ENC(N, M):
    y <-R- G
    C <- (1 + m*N) * y**N % N**2
    return C

DEC(p, q, C):
    phi <- (p - 1) * (q - 1)
    m <- c**phi
    d <- MOD-INV(phi, N)
    m <- m**d
    m <- (m - 1)/N
    return m
```

La decifratura di questo cifrario è un po' più complicata, ma andando passo passo:

$$\text{Ricordando che il crittotesto in input}$$
$$c = (1 + mN)y^N \mod N^2$$

$$\text{Si calcola } c^{\phi(N)}$$
$$c^{\phi(N)} = ((1 + mN)y^N)^{\phi(N)} \mod N^2 = (1 + mN)^{\phi(N)} \mod N^2$$

$$\text{Tramite la funzione MOD-INV}(\phi(N), N) \text{ si calcola l'inverso di } \phi(N) \text{ in } Z_n \to d$$
$$(c^{\phi(N)})^d = ((1 + mN)^{\phi(N)})^d \mod N^2 = (1 + mN) \mod N^2$$

$$\text{Per finire, si ottiene } m, \text{ il messaggio originale}$$
$$m = \frac{(c^{\phi(N)})^d - 1}{N} = \frac{(1 + mN) - 1}{N}$$

La cosa interessante del cifrario **Paillier** è che è additivamente omoformico:
$\mathcal{E}_N(m_1) + \mathcal{E}_N(m_2) = \mathcal{E}_N(m_1 + m_2)$. Tale proprietà è molto utile in pratica, come ad esempio per il voto elettronico.

Sicurezza dello schema Paillier

La sicurezza in senso **ind-cpa** dello schema **Paillier** è maggiorata dal problema della N-residuosità in questa maniera: $Adv^{inc-cpa}_{N,Paillier}(A) \leq 2Adv^{dcra}_N(B)$.
In altre parole, se un avversario è in grado di rendere insicuro lo schema **Paillier**, anche il problema della N-Residuosità diventa trattabile.
Segue un esempio di un avversario B che cerca di risolvere il `problema della N-Residuosità` utilizzando un avversario A che attacca lo schema **Paillier**.

```
B(N, w): /* w è un quadrato residuo? */
    (m_0, m_1, st) <- A(N) /* st è lo stato dell'avversario A */
    b <-R- {0, 1}
    c <- (1 + m_b*N)*w % N**2 /* cifratura valida dello schema Paillier, ammesso
che w sia un N-Residuo */
    b` <- A(st, c)
    if b` == b:
        return 1
    return 0
```

OAEP (Optimal Asymmetric Encryption Padding)

Proposto inizialmente nel 1994, **OAEP** è un metodo per costruire un cifrario sicuro a partire da una qualsiasi permutazione trapdoor con qualche proprietà peculiare. Nel 2001, però, è stato scoperto un errore nella dimostrazione, che è stato possibile aggirare solo per **RSA**.

RSA-OAEP è di fatto lo standard per quanto riguarda la cifratura **RSA**. La dimostrazione di sicurezza rimane comuque, in un certo senso, euristica. Si potrebbe dimostrare la sua sicurezza in maniera matematica solo nel cosiddetto *Random Oracle model*.

Nel modello *RO*, sarebbe necessario avere una funzione hash casuale, e si ipotizza che anche utilizzando una funzione hash lo schema rimanga sicuro. Tale ipotesi è dimostrabilmente falsa, ma nonostante ciò lo schema viene comunque utilizzato.

Preliminari RSA-OAEP

Sia k la taglia del modulo **RSA** da usare, e $k_0, k_1 : k_0 + k_1 < k$. Lo spazio dei messaggi è $\{0,1\}^n$, con $n = k - k_0 - k_1$.
Abbiamo anche bisogno di due funzioni hash:

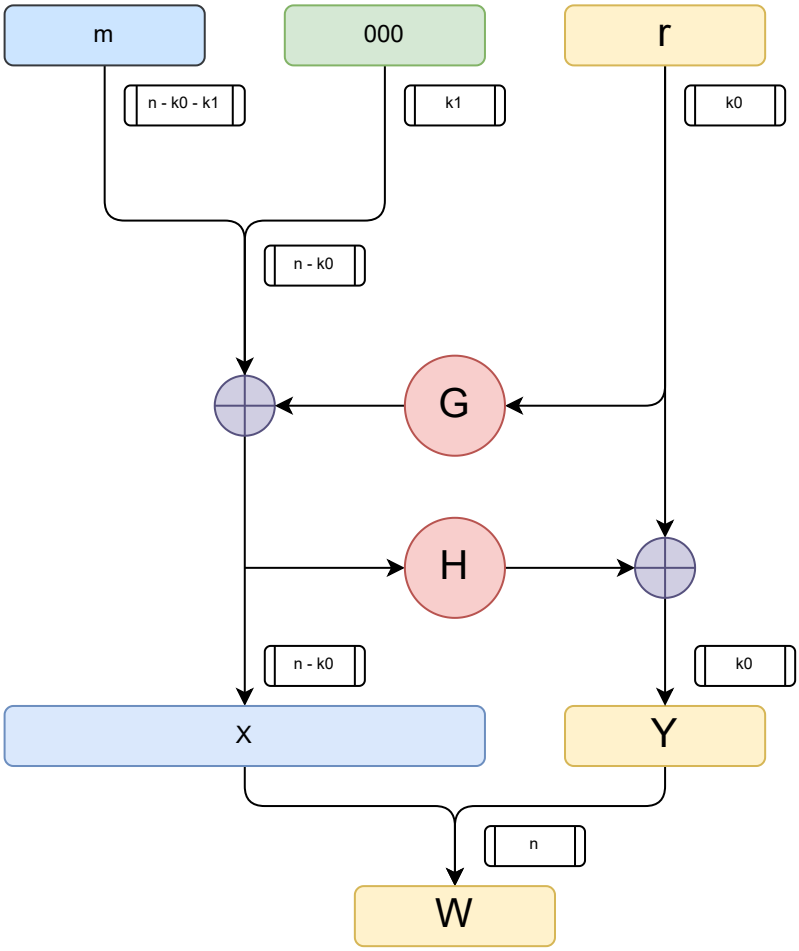
$$G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$$
$$H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$$

L'algoritmo per la generazione delle chiavi è lo stesso usato da **RSA**.

```
ENC(N, m):
    r <-R- {0, 1}^{k_0}
    X <- G(r) xor (m||0^k_1) /* |X| = n - k0 */
    Y <- H(s) xor r /* |Y| = k0 */
    W <- X||Y  /* |W| = n */
    y <- RSA(W)
    return

DEC(N, c):
    W <- INV-RSA(c)
    X || Y <- W /* |X| = n - k0, |Y| = k0 */
    r <- H(X) xor Y
    z <- G(r) xor X
    m || v <- z /* |m| = n - k1 - k0, |v| = k1 */
    if v != 0:
        return "error"
    return m
```

Di seguito lo schema riassuntivo di `ENC`.



Identity based encription

L'obiettivo è quello di utilizzare una qualsiasi stirnga come chiave pubblica in uno schema di cifratura asimmetrico.Questo vuol dire che il mittente è in grado di inviare messaggi ancora prima che il ricevente abbia generato la corrispondente chiave. Inoltre è possibile realizzare facilmente chiavi pubbliche con una scadenza, ad esempio aggiungento il timestamp nella chiave.

Funzioni bilineari

Siano G_1, G_2, G_T dei gruppi con le seguenti proprietà:

- G_1, G_2 sono gruppi di ordine q primo. G_1 è generato da P, G_2 da P'
- Esiste un isomorfismo $p : G_2 \to G_1,\ p(P') = p$
- Esiste una mappa bilineare $e : G_1 \times G_2 \to G_T$
- Le operazioni in tutti i gruppi devono essere efficienti

$$\text{Bilinearità: } \forall U \in G_1,\ V \in G_2,\ a, b \in \mathbb{Z}$$
$$e(U^a, V^b) = e(U, V)^{ab}$$
$$\text{Non deve essere degenere: } e(P, P') \neq \mathbf{1}_{G_T}$$

Definizione formale

Uno schema di identity based encription è una quadrupla così definita:
$\mathcal{IBE} = (\mathcal{Setup}, \mathcal{KeyDer}, \mathcal{Enc}, \mathcal{Dec})$.

Si presume l'esistenza di un'autorità T che conosce una chiave segreta master dalla quale è possibile generare altre chiavi segrete locali. Senza la master key è estremamente difficile fare altrettanto. La definizione di sicurezza **ibe** è leggermente diversa da quella classica. L'attaccante potrebbe avere a disposizione già un certo numero di chiavi private. Il sistema deve continuare a rimanere sicuro.

In aggiunta agli oracoli già visti per **ind-cpa**, l'avversario A ha la possibilità di fare un numero di domande arbitrario ma limitato ad un ulteriore oracolo, $Extract(ID) \rightarrow SK_{ID}$ che restituisce una chiave privata identificata da uno specifico ID. Può poi scegliere di essere sfidato su una ID a sua scelta, sulla quale però non può aver chiesto la chiave segreta corrispondente all'oracolo, che sarebbe imbrogliare.

```
ESP^{ind-id-cpa-1}(A):
    (pk, msk) <- SETUP()
    b <- A^{ENC(id, LR(., ., 1)), KeyDer(.)}
    if a imbroglia:
        return 0
    return b

ESP^{ind-id-cpa-0}(A):
    (pk, msk) <- SETUP()
    b <- A^{ENC(id, LR(., ., 0)), KeyDer(.)}
    if a imbroglia:
        return 0
    return b
```

La nozione di vantaggio in senso **ibe** dell'avversario A è così definita:

$$Adv^{ibe}(A) = |Pr[ESP^{ibe-1}(A) = 1] - Pr[ESP^{ibe-0}(A) = 1]|$$

Lo schema di cifratura asimmetrico è sicuro in senso **ibe** se il vantaggio di ogni possibile avversario polinomialmente limitato è prossimo a 0.

Lo schema Boneh-Franklin

Si tratta di uno schema $\mathcal{IBE} = (\mathcal{Setup}, \mathcal{KeyDer}, \mathcal{Enc}, \mathcal{Dec})$ che utilizza una mappa bilineare tra due gruppi G_1, G_2 di ordine q e con generatori rispettivamente P, P'. Inoltre, in G_1, una variante del `problema decisione di Diffie-Hellman` deve essere intrattabile.

Si definiscano inoltre due funzioni hash $H_1 : \{0,1\}^* \rightarrow G_1,\ H_2 : G_T \rightarrow \{0,1\}^*$.

- $\mathcal{Setup}$: l'autorità T sceglie msk a caso in $[1, 2, \ldots, q]$. La chiave pubblica è $Pk = (P')^{msk}$
- $\mathcal{KeyDer}$: la chiave segreta dell'utente *id* è usk, calcolata come $P_{id} = H_1(id),\ usk = P_{id}^{msk} = H_1(id)^{msk}$.

```
ENC(m, id, Pk):
    r <-R- range(1, q)
    p_id <- H_1(id)
    k <- e(P_id, Pk)**r
    c1 <- (P`)**r
    c2 <- m xor H_2(k)
    return (c1, c2)

DEC(c1, c2), id, usk):
    k <- e(usk, c1)
    m <- c2 xor H_2(k)
    return m
```

Correttezza $\mathcal{E}nc - \mathcal{D}ec$

Si ricordi che $usk = P_{id}^{msk} = H_1(id)^{msk}$, che $Pk = (P')^{msk}$, ed inoltre $e(U^a, V^b) = e(U, V)^{ab}$. Escludendo l'ultimo passo banale, va verificato che $e(usk, c_1) = e(P_{id}, Pk)^r$.

$$e(usk, c_1) = e(P_{id}^{msk}, (P')^r) = e(P_{id}, (P')^{msk})^r = e(P_{id}, Pk)^r$$

Problema Diffie-Hellman Bilineare Decisionale

L'esistenza della mappe bilineare rende il `problema decisionale Diffie-Helman` trattabile nei grupp G_1, G_2, nonostante il problema computazionale `Diffie-Hellman` continui a non esserlo. Si consideri infatti il seguente avversario, che ha conoscenza di tutti i valori pubblici utilizzati in uno schema di **ibe** e che attacca lo schema in senso **ddh**:

```
A(X, Y, Z):
    c <- e(X, Y)
    d <- e(P, Z)
    if c == d:
        return 1
    return 0
```

Si noti che $Adv^{ddh}(A) = 1$.

Per ovviare a ciò, si introduce il `problema decisionale Diffie-Hellman bilineare` (**bddh**). Si tratta di una variazione rispetto al precedente, per cui un avversario A riceve in input $P^a, P^b, P^c \in G_1$ e t, che a seconda dell'esperimento in cui si trova sarà

$$t = \begin{cases} e(P, P)^{abc} & \text{se siamo nel mondo 1} \\ \xleftarrow{\$} G & \text{se siamo nel mondo 0} \end{cases}$$

L'obiettivo di A è determinare in che mondo si trova.

Difficoltà dei problemi

Definendo anche il `problema computazionale Diffie-Hellman bilineare`, ci si potrebbe domandare in che tipo di relazione siano rispetto alla loro difficoltà tutti i problemi che abbbiamo definito fin'ora. Si ottiene quindi la seguente scala:

---

1. Chosen Plaintext Attack
2. Chosen Chypertext Attack
3. Vedi appunti "Primitive asimmetriche" - capitolo "Il problema computazionale Diffie-Hellman"
4. Vedi appunti "Primitive asimmetriche" - capitolo "Il problema decisionale Diffie-Hellman"