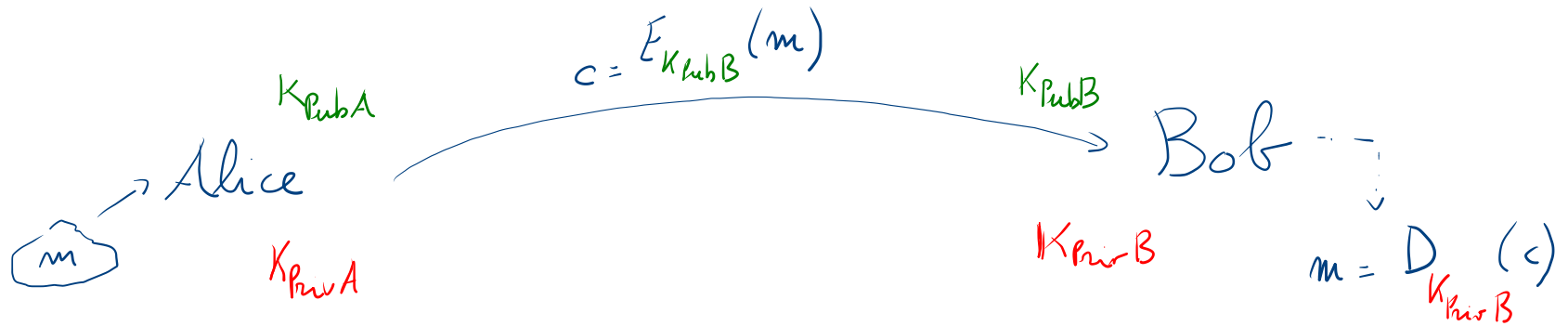


Crittografia a chiave pubblica

Scenario



Alice usa K_{PubB} per cifrare il messaggio m per Bob

Bob usa K_{PrivB} per decifrare il cifrato c ricevuto

Definition formale

DEFINITION 11.1 A public-key encryption scheme is a triple of probabilistic polynomial-time algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ such that:

1. The key-generation algorithm Gen takes as input the security parameter 1^n and outputs a pair of keys (pk, sk) . We refer to the first of these as the public key and the second as the private key. We assume for convenience that pk and sk each has length at least n , and that n can be determined from pk, sk .
2. The encryption algorithm Enc takes as input a public key pk and a message m from some message space (that may depend on pk). It outputs a ciphertext c , and we write this as $c \leftarrow \text{Enc}_{pk}(m)$. (Looking ahead, Enc will need to be probabilistic to achieve meaningful security.)
3. The deterministic decryption algorithm Dec takes as input a private key sk and a ciphertext c , and outputs a message m or a special symbol $\perp$ denoting failure. We write this as $m := \text{Dec}_{sk}(c)$.

It is required that, except possibly with negligible probability over (pk, sk) output by $\text{Gen}(1^n)$, we have $\text{Dec}_{sk}(\text{Enc}_{pk}(m)) = m$ for any (legal) message m .

Osservazioni

Comparazione con gli schemi simmetrici

- incasenti multipli : nessuna chiave da memorizzare
- risolto il problema della condizione iniziale
- chiave segreta mai "esposta"
- performance peggiori
 - operazioni più lente di 2 o 3 ordini di grandezza

Sicurezza rispetto ad attacchi CPA

Rivisitiamo le definizioni del contesto simmetrico

The eavesdropping indistinguishability experiment $\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) .
2. Adversary $\mathcal{A}$ is given pk , and outputs a pair of equal-length messages m_0, m_1 in the message space.
3. A uniform bit $b \in \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \text{Enc}_{pk}(m_b)$ is computed and given to $\mathcal{A}$. We call c the challenge ciphertext.
4. $\mathcal{A}$ outputs a bit b' . The output of the experiment is 1 if $b' = b$, and 0 otherwise. If $b' = b$ we say that $\mathcal{A}$ succeeds.

DEFINITION 11.2 A public-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has indistinguishable encryptions in the presence of an eavesdropper if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is a negligible function negl such that

$$\Pr[\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Osservazione: rispetto all'esperimento $\text{PubK}_{\mathcal{A}, \Pi}^{\text{eav}}$,
l'avversario $\mathcal{A}$ riceve la chiave pubblica pk

$\Rightarrow$ equivale ad aver accesso ad un oracolo
per la cifratura ($\text{PubK}_{\mathcal{A}, \Pi}^{\text{CPA}}$)

$\Rightarrow$ la definizione coincide con quella di
sicurezza rispetto ad attacchi CPA!

Proposizione. Se uno schema di cifratura a chiave pubblica ha cifrature indistinguibili in presenza di un avversario di ascolto, allora è CPA-sicuro

Nota: ciò contrasta con lo scenario simmetrico

Abbiamo visto che esistono schemi che

- hanno cifrature indistinguibili
- sono insicuri rispetto ad attacchi CPA

Impossibilità di sicurezza perfetta

Ogni avversario A con potenza di calcolo illimitata, data pk ed un cifrato c , può determinare m con probabilità 1.

Sketch: assumiamo che Enc_{pk} usi n random bit
A prova a cifrare m con tutte le 2^n
possibili random string, fino ad ottenere c
La condizione di correttezza garantisce che $\exists$
 m_0, m_1 che possono essere cifrati con lo stesso c

Osservazioni aggiuntive

Come nel contesto simmetrico, nessuno schema di cifratura deterministico ha cifrature indistinguibili

- A ha sempre la capacità di stabilire se un messaggio è stato inviato due volte
- se lo spazio dei messaggi da cifrare è piccolo, A può recuperare sempre m da c con probabilità 1.

Ciphertext Multiple

Usiamo lo stesso approccio del contesto simmetrico

The LR-oracle experiment $\text{PubK}_{\mathcal{A}, \Pi}^{\text{LR-cpa}}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) .
2. A uniform bit $b \in \{0, 1\}$ is chosen.
3. The adversary $\mathcal{A}$ is given input pk and oracle access to $\text{LR}_{pk, b}(\cdot, \cdot)$.
4. The adversary $\mathcal{A}$ outputs a bit b' .
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. If $\text{PubK}_{\mathcal{A}, \Pi}^{\text{LR-cpa}}(n) = 1$, we say that $\mathcal{A}$ succeeds.

DEFINITION 11.5 A public-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has indistinguishable multiple encryptions if for all probabilistic polynomial-time adversaries A there exists a negligible function negl such that:

$$\Pr[\text{PubK}_{\mathcal{A}, \Pi}^{\text{LR-cpa}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Fortunatamente vale il seguente

Teorema: Se Π ha cifrature indistinguibili (i.e., CPA-SICURO)
allora ha anche cifrature multiple
indistinguibili (i.e., è CPA-SICURO rispetto a
cifrature multiple)

Cifratura di messaggi di lunghezza arbitraria

$\Pi = (\text{Gen}, \text{Enc}, \text{Dec}) \rightarrow$ messaggi di 1 bit

$\Pi' = (\text{Gen}', \text{Enc}', \text{Dec}') \rightarrow$ messaggi di ℓ bit ($\forall \ell$)

$\rightarrow \text{Gen}' \equiv \text{Gen}$

$\begin{array}{ccc} \text{bit} & \text{bit} & \\ \downarrow & \downarrow & \\ m = m_1 m_2 \dots m_\ell \end{array}$

$\rightarrow \text{Enc}'_{pk}(m) = \text{Enc}_{pk}(m_1) \dots \text{Enc}_{pk}(m_\ell) = c_1 c_2 \dots c_\ell = c$

$\rightarrow \text{Dec}'_{sk}(c) = \text{Dec}_{sk}(c_1) \dots \text{Dec}_{sk}(c_\ell) = m_1 m_2 \dots m_\ell = m$

Attacchi di tipo chosen ciphertext

The CCA indistinguishability experiment $\text{PubK}_{\mathcal{A}, \Pi}^{\text{cca}}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) .
2. The adversary $\mathcal{A}$ is given pk and access to a decryption oracle $\text{Dec}_{sk}(\cdot)$. It outputs a pair of messages m_0, m_1 of the same length. (These messages must be in the message space associated with pk .)
3. A uniform bit $b \in \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \text{Enc}_{pk}(m_b)$ is computed and given to $\mathcal{A}$.
4. $\mathcal{A}$ continues to interact with the decryption oracle, but may not request a decryption of c itself. Finally, $\mathcal{A}$ outputs a bit b' .
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

DEFINITION 11.8 A public-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has indistinguishable encryptions under a chosen-ciphertext attack (or is CCA-secure) if for all probabilistic polynomial-time adversaries A there exists a negligible function negl such that

$$\Pr[\text{PubK}_{A,\Pi}^{\text{cca}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Fortunatamente, anche per la nozione di sicurezza CCA, vale un analogo del teorema precedente per la sicurezza CPA

Cifrare indistinguibili
rispetto ad attacchi CCA

$\Rightarrow$

Cifrare MULTIPLE indistinguibili
rispetto ad attacchi CCA

Nota

La tecnica di composizione utilizzata prima per costruire uno schema di cifratura Π' per msg di lunghezza arbitraria a partire da Π per msg di lunghezza fissa NON funziona nel caso CCA

Π CPA-sicuro $\Rightarrow \Pi'$ CPA-sicuro

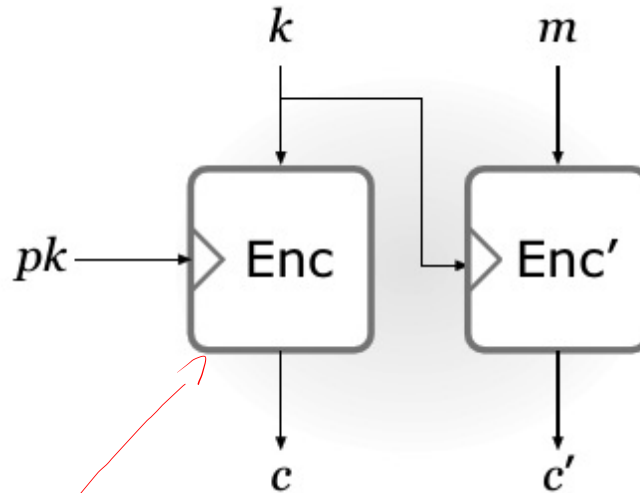
Π CCA-sicuro $\not\Rightarrow \Pi'$ CCA-sicuro

Cifrari ibridi

Utilizziamo cifratura a chiave pubblica e cifratura simmetrica

Approccio di base

Enc cifra
la chiave k



cifratura a chiave pubblica

cifratura simmetrica
(operazioni di cifratura e
decifratura più veloci)

Enc cifra
il messaggio m



Approcci più diretti per "racchiudere" tutto in un passo sono detti KEM - meccanismi di incapsulamento della chiave

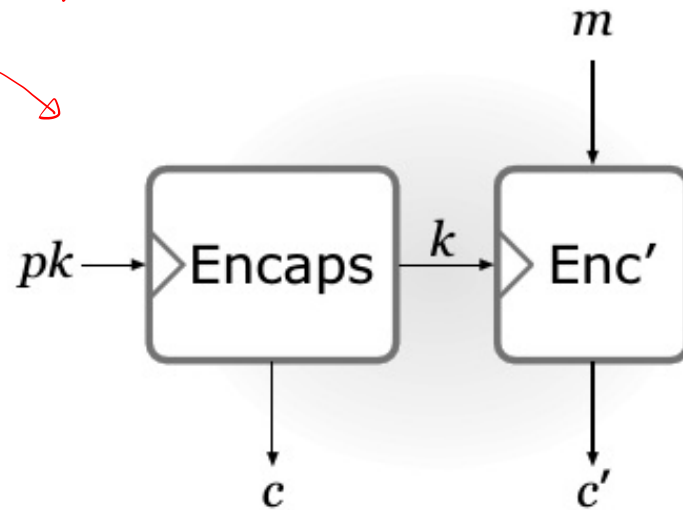
DEFINITION 11.9 A key-encapsulation mechanism (KEM) is a tuple of probabilistic polynomial-time algorithms $(\text{Gen}, \text{Encaps}, \text{Decaps})$ such that:

1. The key-generation algorithm Gen takes as input the security parameter 1^n and outputs a public-/private-key pair (pk, sk) . We assume pk and sk each has length at least n , and that n can be determined from pk .
2. The encapsulation algorithm Encaps takes as input a public key pk and the security parameter 1^n . It outputs a ciphertext c and a key $k \in \{0, 1\}^{\ell(n)}$ where ℓ is the key length. We write this as $(c, k) \leftarrow \text{Encaps}_{pk}(1^n)$.
3. The deterministic decapsulation algorithm Decaps takes as input a private key sk and a ciphertext c , and outputs a key k or a special symbol $\perp$ denoting failure. We write this as $k := \text{Decaps}_{sk}(c)$.

It is required that with all but negligible probability over (sk, pk) output by $\text{Gen}(1^n)$, if $\text{Encaps}_{pk}(1^n)$ outputs (c, k) then $\text{Decaps}_{sk}(c)$ outputs k .

Cifratura ibrida: modello rivestito

$$\text{Encaps}(pk, 1^n) \rightarrow (c, K)$$



Meccanismo di
incapsulamento dei
dati (DEM)

Alice : usa $\text{Encaps}(pk, 1^n)$ ed ottiene (c, K)

calcola $c' = \text{Enc}'_K(m)$ ed invia (c, c') a Bob

Bob : usa $\text{Decaps}(sk, c) = K$ e $\text{Dec}'_K(c') = m$

Costruzione di uno schema ibrido

CONSTRUCTION 11.10

Let $\Pi = (\text{Gen}, \text{Encaps}, \text{Decaps})$ be a KEM with key length n , and let $\Pi' = (\text{Gen}', \text{Enc}', \text{Dec}')$ be a private-key encryption scheme. Construct a public-key encryption scheme $\Pi^{\text{hy}} = (\text{Gen}^{\text{hy}}, \text{Enc}^{\text{hy}}, \text{Dec}^{\text{hy}})$ as follows:

- Gen^{hy} : on input 1^n run $\text{Gen}(1^n)$ and use the public and private keys (pk, sk) that are output.
- Enc^{hy} : on input a public key pk and a message $m \in \{0, 1\}^*$ do:
 1. Compute $(c, k) \leftarrow \text{Encaps}_{pk}(1^n)$.
 2. Compute $c' \leftarrow \text{Enc}'_k(m)$.
 3. Output the ciphertext $\langle c, c' \rangle$.
- Dec^{hy} : on input a private key sk and a ciphertext $\langle c, c' \rangle$ do:
 1. Compute $k := \text{Decaps}_{sk}(c)$.
 2. Output the message $m := \text{Dec}'_k(c')$.

Sicurezza degli schemi di cifratura ibridi con KEM

Introduciamo una nozione di sicurezza per KEM

The CPA indistinguishability experiment $\text{KEM}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) . Then $\text{Encaps}_{pk}(1^n)$ is run to generate (c, k) with $k \in \{0, 1\}^n$.
2. A uniform bit $b \in \{0, 1\}$ is chosen. If $b = 0$ set $\hat{k} := k$. If $b = 1$ then choose a uniform $\hat{k} \in \{0, 1\}^n$.
3. Give $(pk, c, \hat{k})$ to $\mathcal{A}$, who outputs a bit b' . The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

Definizione

DEFINITION 11.11 A key-encapsulation mechanism Π is CPA-secure if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function negl such that

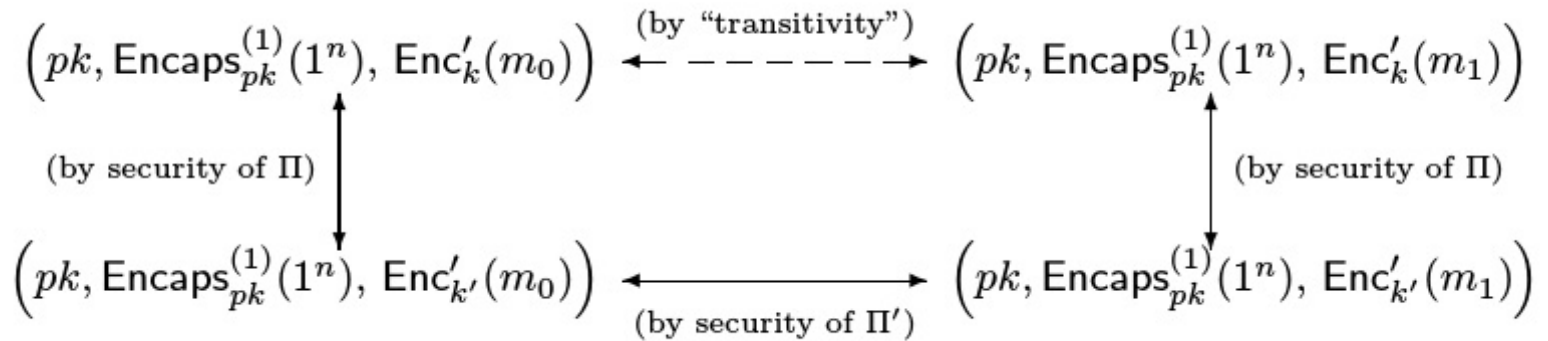
$$\Pr[\text{KEM}_{\mathcal{A}, \Pi}^{\text{cpa}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Teorema : Π è un KEM CPA-sicuro

Π' schema di cifratura simmetrico con cifrature indistinguibili rispetto ad un osservatore passivo

$\Rightarrow \Pi^{\text{hy}}$ a chiave pubblica CPA-sicuro

Struttura della prova



Procede in tre passi, sfrutta le ipotesi di ricorrenza di Π e Π' , e ottiene il risultato "per transitività"

Osservazione

Se Π' non è CCA-sicuro, indipendentemente dalle proprietà del KEM utilizzato nella costruzione

Π^{hy} NON è CCA-sicuro

$\Rightarrow$ la sicurezza CCA richiede che

Π' sia CCA-sicuro

Sicurezza CCA

The CCA indistinguishability experiment $\text{KEM}_{\mathcal{A}, \Pi}^{\text{cca}}(n)$:

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) . Then $\text{Encaps}_{pk}(1^n)$ is run to generate (c, k) with $k \in \{0, 1\}^n$.
2. A uniform bit $b \in \{0, 1\}$ is chosen. If $b = 0$ set $\hat{k} := k$. If $b = 1$ then choose a uniform $\hat{k} \in \{0, 1\}^n$.
3. $\mathcal{A}$ is given $(pk, c, \hat{k})$ and access to an oracle $\text{Decaps}_{sk}(\cdot)$, but may not request decapsulation of c itself.
4. $\mathcal{A}$ outputs a bit b' . The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

DEFINITION 11.13 A key-encapsulation mechanism Π is CCA-secure if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there is a negligible function negl such that

$$\Pr[\text{KEM}_{\mathcal{A}, \Pi}^{\text{cca}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

Vale il seguente:

Teorema. Π è un KEM CCA-secure

Π' è uno schema di cifatura simmetrico
CCA-secure

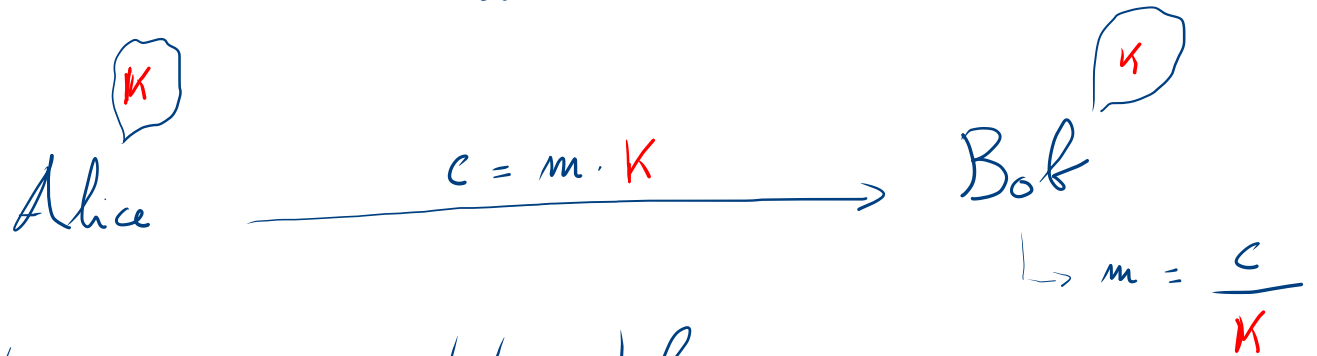
$\Rightarrow \Pi^h$ è uno schema di cifatura
a chiave pubblica CCA-secure

Costruzioni basate sulle assunzioni CDH/DDH

Cifratura El Gamal

Proposto nel 1985 da Taher El-Gamal, è una semplice modifica dello scambio di chiavi Diffie-Hellman

Intuizione: Sia K la chiave stabilita tra Alice e Bob e sia m il messaggio che Alice vuole inviare a Bob



K indistinguibile da chiave casuale $\Rightarrow c$ è indistinguibile da valore casuale $\Rightarrow$ no information

In modo più formale:

LEMMA 11.15 *Let $\mathbb{G}$ be a finite group, and let $m \in \mathbb{G}$ be arbitrary. Then choosing uniform $k \in \mathbb{G}$ and setting $k' := k \cdot m$ gives the same distribution for k' as choosing uniform $k' \in \mathbb{G}$. Put differently, for any $\hat{g} \in \mathbb{G}$ we have*

$$\Pr[k \cdot m = \hat{g}] = 1/|\mathbb{G}|,$$

where the probability is taken over uniform choice of $k \in \mathbb{G}$.

PROOF Let $\hat{g} \in \mathbb{G}$ be arbitrary. Then

$$\Pr[k \cdot m = \hat{g}] = \Pr[k = \hat{g} \cdot m^{-1}].$$

Since k is uniform, the probability that k is equal to the fixed element $\hat{g} \cdot m^{-1}$ is exactly $1/|\mathbb{G}|$. ■

El-Gamal

Sia $\mathcal{G}(1^n)$ un algoritmo ppt che su input 1^n dà in output la descrizione di un gruppo ciclico $\mathbb{G}$, di ordine q , (con $|q|=n$), ed un generatore g

CONSTRUCTION 11.16

Let $\mathcal{G}$ be as in the text. Define a public-key encryption scheme as follows:

- Gen: on input 1^n run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Then choose a uniform $x \in \mathbb{Z}_q$ and compute $h := g^x$. The public key is $\langle \mathbb{G}, q, g, h \rangle$ and the private key is $\langle \mathbb{G}, q, g, x \rangle$. The message space is $\mathbb{G}$.
- Enc: on input a public key $pk = \langle \mathbb{G}, q, g, h \rangle$ and a message $m \in \mathbb{G}$, choose a uniform $y \in \mathbb{Z}_q$ and output the ciphertext

$$\langle g^y, h^y \cdot m \rangle.$$

- Dec: on input a private key $sk = \langle \mathbb{G}, q, g, x \rangle$ and a ciphertext $\langle c_1, c_2 \rangle$, output

$$\hat{m} := c_2 / c_1^x.$$

The El Gamal encryption scheme.

Correzione

Dati $\langle c_1, c_2 \rangle = \langle g^y, h^y \cdot m \rangle$ con $h = g^x$, risulta

$$\hat{m} = \frac{c_2}{c_1^x} = \frac{h^y \cdot m}{(g^x)^x} = \frac{(g^x)^y \cdot m}{(g^x)^x} = \frac{\cancel{g^{xy}} \cdot m}{\cancel{g^{xy}}} = m$$

Esempio numerico. Siano $q = 83$, $p = 2q + 1 = 167$ e sia
 G il sottogruppo dei residui quadratici in $\mathbb{Z}_p^*$

Sia $g = 2^2 = 4 \pmod{167}$. Il ricevente sceglie $x = 37 \in \mathbb{Z}_{83}$.

La chiave pubblica diventa:

$$pk = \langle p, q, g, h \rangle = \langle 167, 83, 4, [4^{37} \pmod{167}] \rangle = \underline{\langle 167, 83, 4, 17 \rangle}$$

Un mittente, per cifrare $m = 65 \in G$ ($65 = 30^2 \bmod 167$)
sceglie y , diciamo $y = 71$, e calcola

$$\langle \underbrace{4^{71} \bmod 167}_{132}, \underbrace{76^{71} \cdot 65 \bmod 167}_{44} \rangle$$

Per decifrare, il ricevente prima calcola

$$[132^{37} \bmod 167] = 124$$

Poi, essendo $124^{-1} \bmod 167 = 66$, calcola

$$[44 \cdot 66 \bmod 167] = 65 = m$$

Teorema. DDH difficile $\Rightarrow$ El. Jamal CPA - sicuro

Alcune considerazioni sugli aspetti implementativi

- Condizione dei parametri pubblici. (G, q, g)

Essati e condivisi generalmente in anticipo

Il NIST ha pubblicato un insieme di parametri consigliati

- Sclta del gruppo

Solitamente q è primo.

Gruppi frequentemente utilizzati: gruppi di punti su curve ellittiche,

sottogruppi di $\mathbb{Z}_p^*$ (p primo)

- Spazio dei messaggi (G è scomodo)

Usare $\{0, 1\}^l$ e poi servirsi di un mapping

Sketch della prova

A ppt vogliamo mostrare che $\Pr [\text{Pub}_{A, \tilde{\Pi}}^{\text{ear}}(m) = 1] \leq \frac{1}{2} + \text{negl}(n)$

$\tilde{\Pi}$ $pk = (G, q, g, h)$ $c = \langle g^r, g^z m \rangle$
(non è in realtà uno schema, non defn Bets)

In accordo al lemma precedente la seconda componente della
cifratura è indipendente da m .

$$\text{Risulta } \Pr [\text{Pub}_{A, \tilde{\Pi}}^{\text{ear}}(m) = 1] = \frac{1}{2}.$$

Consideriamo il seguente distinguisher D per DDH

D $\longrightarrow$ (Distinguish Triple DDH / casuali)

Input: $(G, q, g, \underline{h_1}, \underline{h_2}, \underline{h_3})$

- Pone $pk = (G, q, g, \underline{h_1})$ ed esegui $A(pk)$. Otherwise $m_0, m_1 \in G$
- Scegli $b \in \{0, 1\}$ uniforme. Pone $c_1 = \underline{h_2}$, $c_2 = \underline{h_3} \cdot m_b$
- Da' $\langle c_1, c_2 \rangle$ ad A e ricevi b'
- Se $b' = b$, da' in output 1; altrimenti 0

Caso 1: $\underline{h_1} = g^x, \underline{h_2} = g^y, \underline{h_3} = g^z \quad x, y, z \in_n \mathbb{Z}_q$

risulta $c = \langle g^y, g^z m \rangle \Rightarrow$

$$\Pr[D(G, q, g, g^x, g^y, g^z) = 1] = \Pr[\Pr_{A, \Pi}^{b^{car}}(m) = 1] = \frac{1}{2}$$

caso 2: $\underline{h_1} = g^x$, $\underline{h_2} = g^y$, $\underline{h_3} = g^{xy}$, $x, y \in_n \mathbb{Z}_q$
 risulta $c = \langle g^y, g^{xy} \cdot m \rangle \Rightarrow$

$$P_2[D(\kappa, q, g, g^x, g^y, g^{xy}) = 1] = P_1[\text{Pub}_{A, \Pi}^{\text{car}}(m) = 1]$$

Se DDH è difficile, allora $\exists \text{negl}(n)$:

$$|P_2[D(\kappa, q, g, g^x, g^y, g^z) = 1] - P_2[D(\kappa, q, g, g^x, g^y, g^{xy}) = 1]| \leq \text{negl}(n)$$

$$|P_2[\text{Pub}_{A, \widetilde{\Pi}}^{\text{car}}(m) = 1] - P_2[\text{Pub}_{A, \Pi}^{\text{car}}(m) = 1]| \leq \text{negl}(n)$$

$$|1/2 - P_2[\text{Pub}_{A, \Pi}^{\text{car}}(m) = 1]| \leq \text{negl}(n) \Rightarrow$$

$$P_2[\text{Pub}_{A, \Pi}^{\text{car}}(m) = 1] \leq \frac{1}{2} + \text{negl}(n) \quad \square$$

Idea alternativa

Usare El-Gamal per produrre uno schema ibrido

- il sender può scegliere $r \in G$, lo cifra con El-Gamal e lo manda al ricevente

- usa $H: G \rightarrow \{0,1\}^m$ per calcolare $K = H(r)$

- cifra m con un cifrario simmetrico usando K

→ funzione di derivazione della chiave

Cifrato completo: $\langle c_1, c_2 \rangle$, $c = Enc_K(m)$

Esistono approcci ibridi più efficienti

KEM basato su El-Gamal

CONSTRUCTION 11.19

Let $\mathcal{G}$ be as in the previous section. Define a KEM as follows:

- **Gen:** on input 1^n run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. choose a uniform $x \in \mathbb{Z}_q$ and set $h := g^x$. Also specify a function $H : \mathbb{G} \rightarrow \{0, 1\}^{\ell(n)}$ for some function ℓ (see text). The public key is $\langle \mathbb{G}, q, g, h, H \rangle$ and the private key is $\langle \mathbb{G}, q, g, x \rangle$.
- **Encaps:** on input a public key $pk = \langle \mathbb{G}, q, g, h, H \rangle$ choose a uniform $y \in \mathbb{Z}_q$ and output the ciphertext g^y and the key $H(h^y)$.
- **Decaps:** on input a private key $sk = \langle \mathbb{G}, q, g, x \rangle$ and a ciphertext $c \in \mathbb{G}$, output the key $H(c^x)$.

An “El Gamal-like” KEM.

$$\text{Encaps}(pk, 1^n) = (g^y, \underbrace{H(h^y)}_K)$$

$$\text{Decaps}(sk, c) = \underbrace{H(c^x)}_K$$

$$\Rightarrow \text{ciphertext} \quad \langle g^y, \underbrace{\text{Enc}(m)}_K \rangle$$

compleanno

Teorema. DDH difficile e H "opportunamente scelta"
 $\Rightarrow$ KEM CPA-sicuro

Come scegliere H ? $H: G \rightarrow \{0,1\}^l$

- distribuisce quasi uniformemente: $\forall K \in \{0,1\}^l$ il # elementi di G che producono K è approssimativamente lo stesso
- H è una funzione con chiave - la chiave è inclusa nella chiave pubblica del ricevente - deve comportarsi come un "estrattore forte"
- H è un oracolo casuale: in quest'ultimo caso, la costruzione può essere provata CPA-sicura nel modello ROM, basandosi sulla più debole assunzione CDH

Teorema CDH difficile e H ROM
 $\Rightarrow$ KEM CPA-sicuro

Circa la nozione CCA?

È facile vedere che El-Jamal è vulnerabile ad attacchi di tipo chosen-ciphertext. È malleabile!

A intercetta $c = \langle c_1, c_2 \rangle$, $pk = \langle G, q, g, h \rangle$

costruisce $c' = \langle c_1, c'_2 \rangle$ dove $c'_2 = c_2 \cdot d$, $d \in G$

$\Rightarrow$ c è una cifratura di m , cioè $\langle g^y, h^y \cdot m \rangle$ ($y \in \mathbb{Z}_q$)

allora $c' = \langle g^y, (h^y \cdot m) \cdot d \rangle$ risulta una cifratura di $d \cdot m$

$$c' = \langle g^y, h^y (d \cdot m) \rangle$$

Quindi A può trasformare la cifra di m (sconosciuto)
in quella di $d \cdot m$ (sconosciuto) con d noto

Lo schema KEM precedente può essere malleabile.

Dipende dalla scelta della funzione H .

Se H è modellata come un oracolo casuale, si
può dimostrare che il KEM è CCA-sicuro.

A tal fine, occorre introdurre però una nuova
assunzione crittografica

Quindi A può trasformare la cifra di m (sconosciuto)
in quella di $d \cdot m$ (sconosciuto) con d noto

Lo schema KEM precedente può essere malleabile.

Dipende dalla scelta della funzione H .

Se H è modellata come un oracolo casuale, si
può dimostrare che il KEM è CCA-sicuro.

A tal fine, occorre introdurre però una nuova
assunzione crittografica

Gap - Computational Diffie-Hellman (Gap - CDH)

"CDH è ancora difficile anche se DDH è facile"

Dati g^x e g^y , per qualche generatore g , risulta computazionalmente impraticabile calcolare g^{xy} ANCHE SE si ha accesso ad un oracolo D tale che

$$D(U, V) = 1 \quad \text{esattamente quando} \quad V = U^x$$

Teorema. Gap-CDH difficile e H ROM
 $\Rightarrow$ KEM CCA-sicuro

Mettendo assieme un po' di cose:

Π è un KEM CCA-sicuro
 Π' è uno schema di cifratura
simmetrico CCA-sicuro

} $\Rightarrow$ lo schema ibrido Π^h è CCA-sicuro

Pertanto possiamo ottenere uno schema di cifratura ibrido a chiave pubblica CCA-sicuro usando:

- KEM della costruzione precedente
- Uno schema simmetrico di cifratura autenticata

Cifratura $\langle g^Y, \underbrace{\text{Enc}_{K_E}(m)}_{c'}, \text{Mac}_{K_M}(c') \rangle$

cifra e poi autentica

Dettagli della costruzione ibrida risultante

CONSTRUCTION 11.23

Let $\mathcal{G}$ be as in the text. Let $\Pi_E = (\text{Enc}', \text{Dec}')$ be a private-key encryption scheme, and let $\Pi_M = (\text{Mac}, \text{Vrfy})$ be a message authentication code. Define a public-key encryption scheme as follows:

- **Gen:** On input 1^n run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. choose a uniform $x \in \mathbb{Z}_q$, set $h := g^x$, and specify a function $H : \mathbb{G} \rightarrow \{0, 1\}^{2n}$. The public key is $\langle \mathbb{G}, q, g, h, H \rangle$ and the private key is $\langle \mathbb{G}, q, g, x, H \rangle$.
- **Enc:** On input a public key $pk = \langle \mathbb{G}, q, g, h, H \rangle$, choose a uniform $y \in \mathbb{Z}_q$ and set $k_E \| k_M := H(h^y)$. Compute $c' \leftarrow \text{Enc}'_{k_E}(m)$, and output the ciphertext $\langle g^y, c', \text{Mac}_{k_M}(c') \rangle$.
- **Dec:** On input a private key $sk = \langle \mathbb{G}, q, g, x, H \rangle$ and a ciphertext $\langle c, c', t \rangle$, output $\perp$ if $c \notin \mathbb{G}$. Else, compute $k_E \| k_M := H(c^x)$. If $\text{Vrfy}_{k_M}(c', t) \neq 1$ then output $\perp$; otherwise, output $\text{Dec}'_{k_E}(c')$.

DHIES / ECIES

Lo schema risultante corrisponde a quanto avviene negli schemi ibridi:

- DHIES (Diffie-Hellman Integrated Encryption Scheme)
- ECIES (Elliptic Curve Integrated Encryption Scheme)

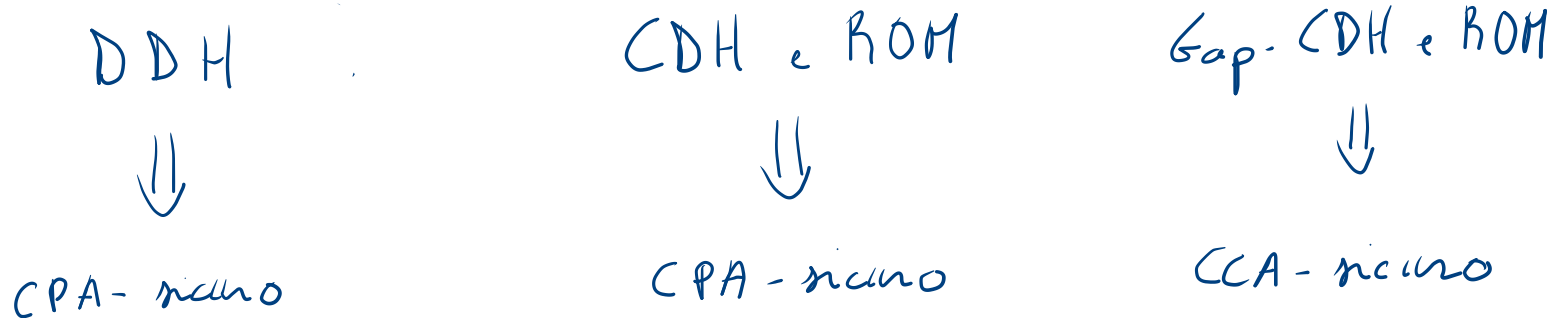
G è un sottogruppo ciclico di un campo finito

G è un gruppo di punti di una curva ellittica

Corollario. Se Π_E è CPA-sicuro, Π_M è un MAC fortemente sicuro, H è un oracolo casuale e Gap-CDH è difficile relativamente a f , allora

$\Pi^{hy}(\text{DHIES/ECIES})$ è CCA-sicura

Nota: per il KEM basato su El-Gamal vale



Teorema CDH difficile e H ROM
 $\Rightarrow$ KEM e CPA-sicuro

(Esempio non banale di prova nel ROM)

Dim. Sia A ppt. Vogliamo mostrare che $\exists \text{negl}(n)$:

A riceve $\langle pk, \epsilon, \hat{k} \rangle$

$$P_2 [KEM_{A, \pi}^{cpa}(n) = 1] \leq \frac{1}{2} + \text{negl}(n)$$

$\hat{k}$ random

ϵ random

Sia $pk = \langle G, q, g, h \rangle$ e $c = g^y$

e sia Query l'evento "A chiede l'hash di h^y ad H "

otherwise dall'oracolo l'hash del valore Diffie-Hellman

Nota: per far questa query A dovrebbe essere in grado di risolvere CDH

$$\begin{aligned}
 P_2 [KEM_{A,\Pi}^{cpa}(u)=1] &= P_2 [KEM_{A,\Pi}^{cpa}(u)=1 \wedge \overline{Query}] + P_2 [KEM_{A,\Pi}^{cpa}(u)=1 \wedge Query] \\
 &\leq P_2 [KEM_{A,\Pi}^{cpa}(u)=1 \mid \overline{Query}] + P_2 [Query]
 \end{aligned}$$

Nell'esperimento $KEM_{A,\Pi}^{cpa}(u)$ A dispone di $\langle pk, c, \hat{k} \rangle$
 dove $\hat{k}$ è o $H(h^x)$ oppure un valore casuale

Se l'evento $Query$ non si verifica, per le proprietà del random oracle (1^a proprietà) $H(h^x)$ è uniformemente distribuita e, quindi, per A completamente indistinguibile da un valore casuale

$$\Rightarrow \underline{P_2 [KEM_{A,\Pi}^{cpa}(u)=1 \mid \overline{Query}] = 1/2}$$

$$\Rightarrow \Pr[KEM_{A, \Pi}^{cpa}(n) = 1] \leq \frac{1}{2} + \Pr[\text{Query}]$$

possiamo far vedere che $\Pr[\text{Query}] \leq \text{negl}(n)$.

Siamo nel ROM, quindi A può effettuare query all'oracolo H .

Sia t # poly di query che A effettua

Vogliamo sfruttare A (che gioca nell'esperimento

$KEM_{A, \Pi}^{cpa}(n)$) per costruire un A' che

risolve il problema CDH.

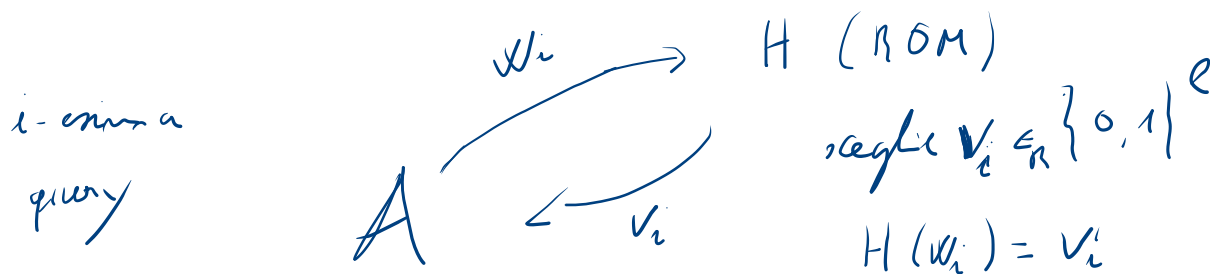
A'

Input $(\ell, q, g), \underline{h}, \underline{c}$

$\rightarrow (A' \text{ calcola } DH(h, c) = h')$

- Pone $pk = \langle \ell, q, g, \underline{h} \rangle$ e sceglie K uniformemente
 $\rightarrow$ stringa di ℓ -bit
- Esegue $A(pk, \underline{c}, K)$

Quando A effettua query ad H , A' le intercetta e simula le risposte inviando stringhe di ℓ -bit scelte uniformemente a caso



- Al termine dell'esecuzione di A , siano $w_1, w_2, \dots, w_t$ le query che A ha fatto all'oracolo H . A sceglie $i \in_R \{1, \dots, t\}$ e dà in output w_i

↓
(A scommette che al passo i -esimo A abbia chiesto ad H l'hash del valore Diffie-Hellman di h e c , che sarebbe appunto w_i)

Equivalentemente, possiamo dire che A scommette che Query ricevrà alla i -esima query di A

Qual è $\Pr[A'(t, q, g, h, c) = h^Y]$?
 $\xrightarrow{\text{red}} \text{DH}(h, c)$

Osservazioni: l'evento Query nella simulazione di A' realizza
 si verifica con la stessa probabilità con cui
 si verifica nell'esperimento $\text{KEM}_{A, \Pi}^{\text{cpa}}$

(... fino a quando
 non si verifica
 l'evento Query, de:
 - in $\text{KEM}_{A, \Pi}^{\text{cpa}}(\cdot)$ fa
 ricorere a A il
 valore reale $H(h^Y)$
 uguale a K
 con prob. $1/k$
 - in A' fa ricorere a A
 un valore uniforme)

Vista di A come subroutine A' $\equiv$ Vista di A in $\text{KEM}_{A, \Pi}^{\text{cpa}}$

Pertanto, κ l'evento Query si verifica

$$h^Y \in \{w_1, w_2, \dots, w_k\}$$

$\Rightarrow A'$ dà in output il valore corretto con probabilità $1/t$. Pertanto

$$P_r[A'(b, q, g, h, c) = h^y] \geq \frac{P_r[\text{Query}]}{t}$$

Ma α CDH è difficile, $\exists \text{negl}(n)$ tale che

$$P_r[A'(b, q, g, h, c) = h^y] \leq \text{negl}(n)$$

$$\Rightarrow P_r[\text{Query}] \leq \underbrace{t \cdot \text{negl}(n)}_{\text{poly} * \text{negl} \Rightarrow \text{negl}(n)} = \text{negl}(n)$$

$$\text{poly} * \text{negl} \Rightarrow \text{negl}(n)$$