COMBINATORIA E CRITTOGRAFIA

Implementazione del crittosistema a chiave pubblica di ElGamal.

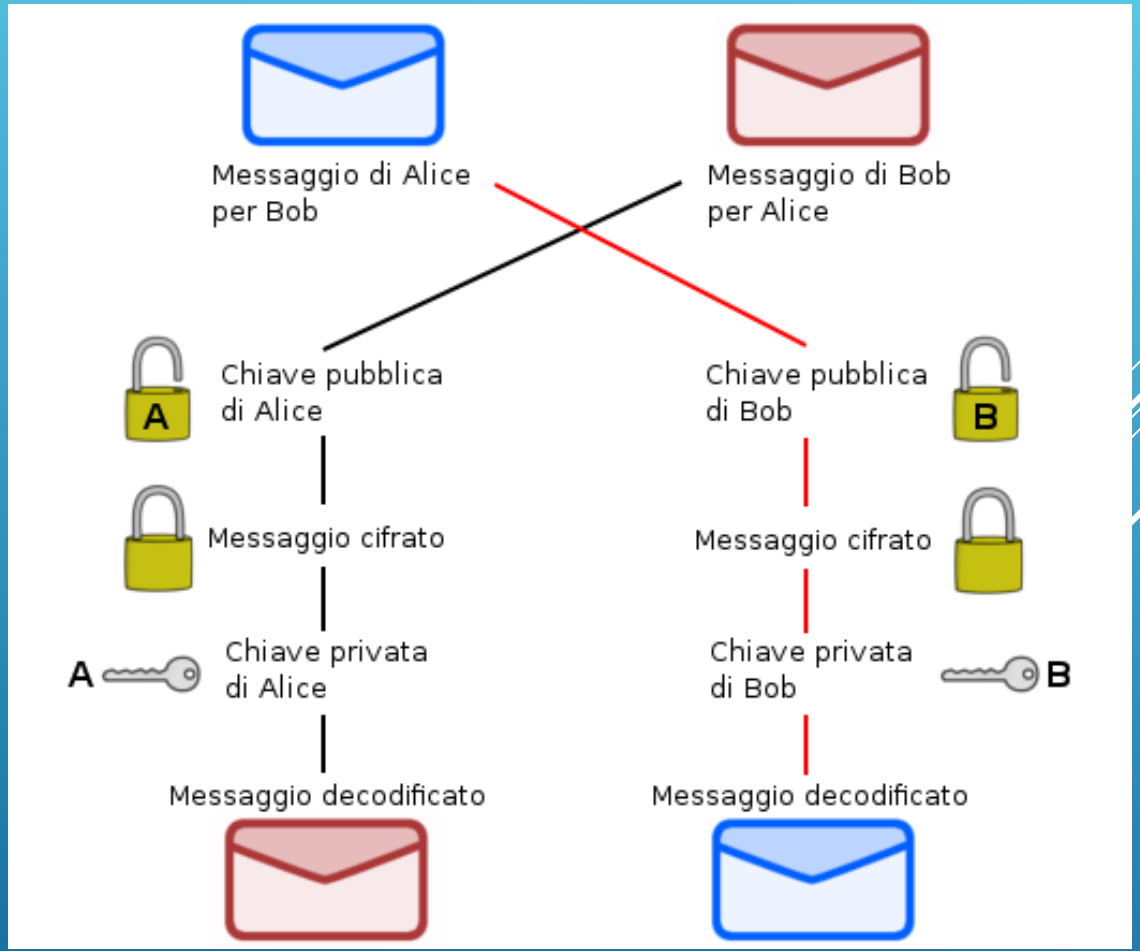
Luca Di Vita
mat.: 210430
Ing. dell'Informazione

Sistemi a chiave asimmetrica

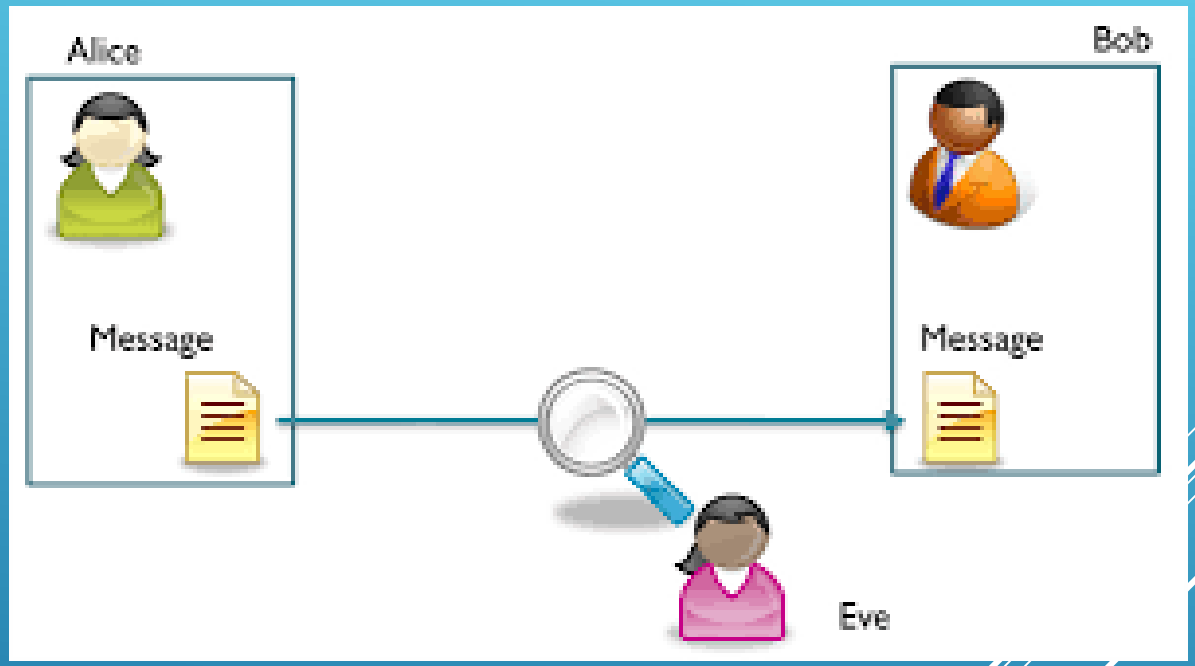
Il crittosistema di ElGamal è un sistema di cifratura a chiave pubblica (o asimmetrico) ideato da Taher ElGamal nel 1984. Ad ogni attore coinvolto, è associata una coppia di chiavi :

- Chiave Pubblica : Che viene distribuita pubblicamente.
- Chiave Privata : Rimane segreta all'attore coinvolto.

In questo modo, si evita ogni problema relativo allo scambio in modo sicuro dell'unica chiave usata per cifratura e decifratura. Il meccanismo si basa sul fatto che con la chiave pubblica si codifica, e la decodifica può essere fatta unicamente conoscendo la chiave privata.



Il problema principale degli algoritmi a chiave asimmetrica è la loro lentezza in quanto onerosi dal punto di vista computazionale. Per questo motivo sono spesso associati ad algoritmi a chiave simmetrica. Cioè si usa un algoritmo a chiave asimmetrica per lo scambio della chiave relativa all'algoritmo a chiave simmetrica precedentemente concordato e, una volta che entrambi gli attori l'hanno ricevuta, lo scambio di dati avviene con l'algoritmo simmetrico che è più veloce. In questo modo un eventuale intruso nel canale non ha modo di conoscere la chiave di cifratura dell'algoritmo utilizzato successivamente.



La sicurezza dell'algoritmo si basa sul problema di Diffie-Hellman che a sua volta si basa sulla complessità computazionale del calcolo del logaritmo discreto.

LOGARITMO DISCRETO

Fissati un primo p, siano α e β due interi non nulli modulo p tali che $\beta \equiv \alpha^x (mod\ p)$, il problema di trovare x è chiamato problema del logaritmo discreto. Se n è il più piccolo intero positivo per cui $\alpha^n \equiv 1 (mod\ p)$, si può assumere $0 \leq x < n$ e scrivere $x = L_\alpha(\beta)$ che è chiamato logaritmo discreto di β rispetto ad α.
Spesso α è una radice primitiva in quanto se così non fosse, il logaritmo non sarebbe definito per certi valori di x. Quando p è piccolo è facile calcolare il logaritmo mediante una ricerca esaustiva di tutti i possibili esponenti. Quando invece è grande non è più fattibile, per cui in genere il logaritmo discreto $f(x)$ è una funzione unidirezionale.

ALGORITMO DI POHLIG-HELLMAN

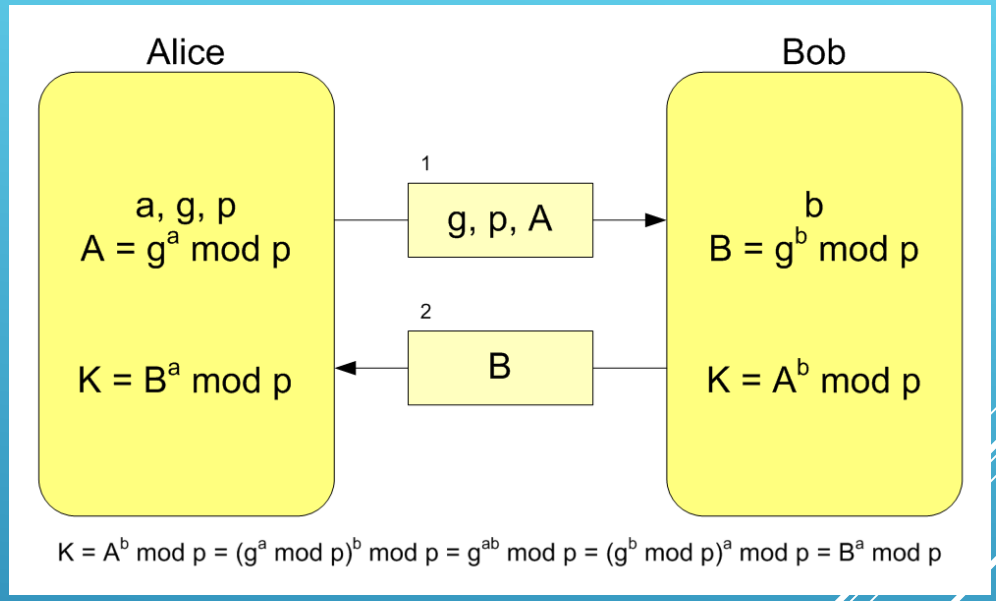
L'algoritmo di Pholig-Hellman permette di calcolare i logaritmi discreti quando $p\text{-}1$ ha solo fattori piccoli. Per cui si calcola la scomposizione in fattori primi di $p\text{-}1$: $p-1=\prod_i q_i^{r_i}$. Se il logaritmo può essere calcolato per ogni $q_i^{r_i}$ allora il logaritmo può essere calcolato mettendo insieme le soluzioni mediante il teorema cinese del resto. Sia $x=x_0+x_1q+x_2q^2+\cdots$ con $0\leq x_i\leq q-1$, sia ha che : $x\left(\frac{p-1}{q}\right)=x_0\left(\frac{p-1}{q}\right)+(p-1)(x_1+x_2q+\cdots)=x_0\left(\frac{p-1}{q}\right)+(p-1)n$. Elevando entrambi i membri di $\beta\equiv\alpha^x$ alla potenza $^{(p-1)}/_q$ si ottiene $\beta^{(p-1)/q}\equiv\alpha^{x(p-1)/q}\equiv\alpha^{x_0(p-1)/q}(\alpha^{p-1})^n\equiv\alpha^{x_0(p-1)/q}(mod\ p)$ dove l'ultima congruenza deriva dal teorema di Fermat. Per trovare x_0 basta cercare tra le potenze di $\alpha^{k(p-1)/q}(mod\ p)$ con $k=0,1,2,\ldots,q-1$, quella in corrispondenza della quale si ha $\beta^{(p-1)/q}$ da cui $x_0=k$.

Si può estendere questo discorso anche agli altri coefficienti. Assumendo che $q^2|p-1$, elevando entrambi i membri del logaritmo discreto a ${}^{(p-1)}/_{q^2}$ si ottiene $\beta_1^{{}^{k(p-1)}/_{q^2}} \equiv \alpha^{{}^{(p-1)(x_1+x_2q+\cdots)}/_{q}} \equiv \alpha^{{}^{x_1(p-1)}/_{q}} (mod\ p)$. Analogamente al caso precedente per trovare x_1 basta cercare tra le potenze di $\alpha^{{}^{k(p-1)}/_{q}} (mod\ p)$ con $k = 0,1,2, \dots, q-1$, quella per cui si ha $\beta_1^{{}^{(p-1)}/_{q^2}}$ allora $x_1 = k$. Si procede così per trovare tutti i fattori che poi saranno messi insieme col teorema cinese del resto.

SCAMBIO DI DIEFFIE-HELLMAN

Permette a due attori di stabilire in sicurezza un chiave privata *K* attraverso un canale pubblico.
Segue il seguente algoritmo:

1. Alice (o Bob) sceglie un primo p grande e sicuro e una radice primitiva $\alpha\ (mod\ p)$. α e p sono resi pubblici.
2. Alice sceglie un x segreto con $1 \leq x \leq p-2$. Anche Bob sceglie un y con $1 \leq y \leq p-2$.
3. Alice manda $\alpha^x\ (mod\ p)$ a Bob e Bob invia $\alpha^y\ (mod\ p)$ ad Alice.
4. Usando i messaggi ricevuti, entrambi calcolano la chiave *K*. Alice calcola $K \equiv (\alpha^y)^x (mod\ p)$ con α^y che ha ricevuto da Bob, e Bob calcola $K \equiv (\alpha^x)^y (mod\ p)$ con α^x ricevuto da Alice.



Ora entrambi hanno lo stesso numero *K* e la possono usare con un sistema concordato preventivamente. Anche se Eva ascoltasse il canale per scoprirebbe unicamente α^x e α^y. Se fosse in grado di calcolare i logaritmi discreti, potrebbe scoprire x e y.

ALGORITMO DI ELGAMAL

Alice vuole mandare un messaggio m a Bob. Quest'ultimo sceglie un primo p grande e una radice primitiva α. m va preso $0 \leq m < p$ e se $m > p$ va spezzato in blocchi più piccoli. Bob inoltre sceglie un intero segreto a e calcola $\beta \equiv \alpha^a (mod\ p)$. L'informazione (p, α, β) è la chiave pubblica di Bob e come tale è accessibile da chiunque.
Alice segue la procedura:

1. Scarica (p, α, β).
2. Sceglie a caso un intero k segreto e calcola r $\equiv \alpha^k (mod\ p)$.
3. Calcola $t \equiv \beta^k m (mod\ p)$.
4. Manda a Bob la coppia (r, t).

Bob decifra il messaggio calcolando $tr^{-a} \equiv m (mod\ p)$. Questo funziona perché:
$tr^{-a} \equiv \beta^k (\alpha^k)^{-a} m (mod\ p) \equiv (\alpha^a)^k (\alpha^k)^{-a} m \equiv m (mod\ p)$.
Se Eva riuscisse a determinare a potrebbedecifrare il messaggio con la stessa procedura utilizzata da Bob. Quindi tenere a segreto è importante. La sicurezza di a è data dalla difficoltà di calcolare il logaritmo discreto.

ATTACCHI AL CRITTOSISTEMA

Sono possibili, escludendo il caso in cui vengano scelti esponenti bassi che rende possibile la forza bruta e provare tutte le combinazioni per il logaritmo discreto, quattro attacchi al sistema descritto:

1. Attacco dovuto alla non variazione della chiave segreta k.
2. Attacco del compleanno.
3. Baby Step, Giant Step.
4. Attacco tramite l'ausilio delle curve ellittiche.

Mentre l'attacco 2. e l'attacco 3. prevedono di trovare una soluzione, in tempi ragionevoli, al logaritmo discreto, l'attacco 4. permette di trovare una soluzione al problema di decisione di Diffie-Hellman che dice che: « Sia p primo, α una radice primitiva modulo p. Dati $\alpha^x mod\ p$, $\alpha^y mod\ p$ e $c \neq 0 mod\ p$, decidere se $c = \alpha^{xy} mod\ p$.
Si passa ora alla discussione dei singoli attacchi 1. 2. e 3. trascurando il 4. in quanto non trattato.

1. ATTACCO DOVUTO ALLA NON VARIAZIONE DI K

Come si era detto Alice calcola r $\equiv \alpha^k (mod\ p)$ e $t \equiv \beta^k m (mod\ p)$ per cui se Eva riuscisse a scoprire k potrebbe calcolare $t\beta^{-k} \equiv m\ (mod\ p)$. Si supponga che Alice codifichi due differenti messaggi m_1 e m_2 con una stessa chiave k, per cui r sarà la stessa per entrambi i messaggi e i testi cifrati saranno (r, t_1) e (r, t_2). Se Eva conosce il testo in chiaro m_1, può determinare anche m_2, infatti conoscendo t_1 e t_2 si ha: ${}^{t_1}/_{m_1} \equiv \beta^k \equiv {}^{t_2}/_{m_2}\ (mod\ p)$ da cui si può calcolare $m_2 \equiv t_2\ {}^{m_1}/_{t_1}\ (mod\ p)$.
Una contromisura per evitare questo attacco è cambiare k ad ogni messaggio.

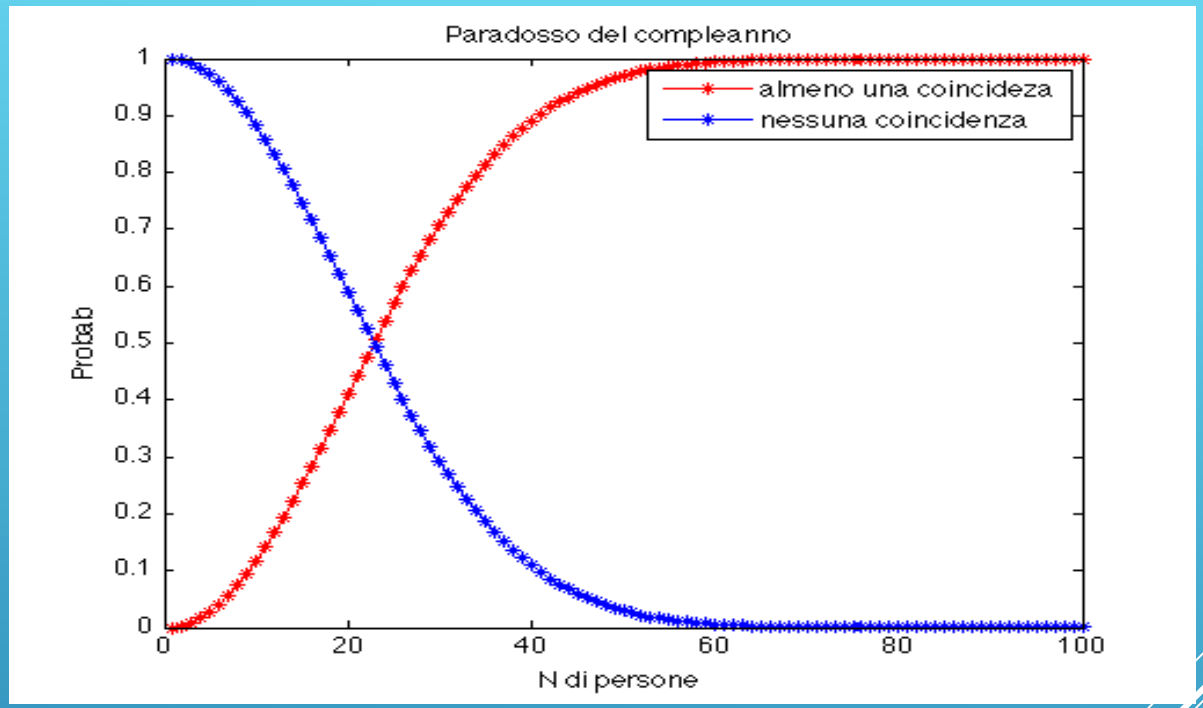
2. ATTACCO DEL COMPLEANNO

E' un attacco di tipo probabilistico. Esso consiste, dato un primo p, nel cercare di risolvere il logaritmo discreto $\alpha^x \equiv \beta \ (mod\ p)$, con alta probabilità. Esso si svolge così:
si costruiscono due liste, entrambe di lunghezza $\sqrt{p}$ circa.

- La prima lista contiene i numeri $\alpha^k (mod\ p)$ per circa $\sqrt{p}$ valori scelti a caso di k.
- La seconda lista contiene i numeri $\beta\alpha^{-l} (mod\ p)$ per circa $\sqrt{p}$ valori di l scelti a caso.

C'è una buona probabilità che esista una corrispondenza tra qualche elemento
della prima lista con qualche elemento della seconda. Se è così si ha che vale la
seguente relazione : $\alpha^k \equiv \beta\alpha^{-l}$ da cui $\alpha^{k+l} \equiv \beta (mod\ p)$ quindi $x = k + l (mod\ p - 1)$ è il
logaritmo discreto cercato.

Questo attacco di basa sul paradosso che se in una stanza ci sono 23 persone, la probabilità che due persone facciano il compleanno lo stesso giorno è più del 50%. Infatti:
$(1 - {}^{1}/_{365})(1 - {}^{2}/_{365}) \dots (1 - {}^{22}/_{365}) = 0.493$ è la probabilità che le 23 persone facciano il compleanno in giorni diversi, da cui troviamo la probabilità inversa che è $1 - 0.493 = 0.507$, la probabilità cercata.



Come contromisura a questo attacco si possono usare p molto grandi in quanto, dovendo implementare una lista per questo attacco, la cosa è fattibile fino ad un massimo ordine di $p \approx 10^{20}$ che memorizza $N = \sqrt{p} \approx 10^{10}$. Quindi per p ancora più grandi diventa problematico gestire tali liste.

3. BABY STEP, GIANT STEP

In maniera analoga all'attacco del compleanno, si prefigge di risolvere il logaritmo discreto, ma a differenza dello stesso è di tipo deterministico e non probabilistico.

Per prima cosa si sceglie un intero N con $N^2 \geq p - 1$, ad esempio $N = \sqrt{p-1} + 1$, poi si compongono le liste seguenti :

- $\alpha^j (mod\ p)$ per $0 \leq j < N$ chiamata lista «Baby Steps».
- $\beta\alpha^{-Nk} (mod\ p)$ $0 \leq k < N$chiamata lista «Giant Steps».

Infine si cerca una coincidenza tra le liste. Se se ne trova una vale la relazione : $\alpha^j \equiv \beta\alpha^{-Nk}$ da cui $\alpha^{j+Nk} \equiv \beta$ e quindi $x = j + Nk$ risolve il logaritmo discreto. La Prima lista deve essere calcolata al completo, ma non c'è bisogno di fare altrettanto con la seconda in quanto basta anche una sola coincidenza. Per cui una volta calcolata la prima lista, per ogni elemento calcolato della Giant Steps si può eseguire un controllo, ed il calcolo termina una volta trovata una corrispondenza.

Analogamente all'attacco del compleanno, una contromisura è usare p molto elevati per gli stessi motivi sopra menzionati.

FINE PRESENTAZIONE

Grazie per l'attenzione