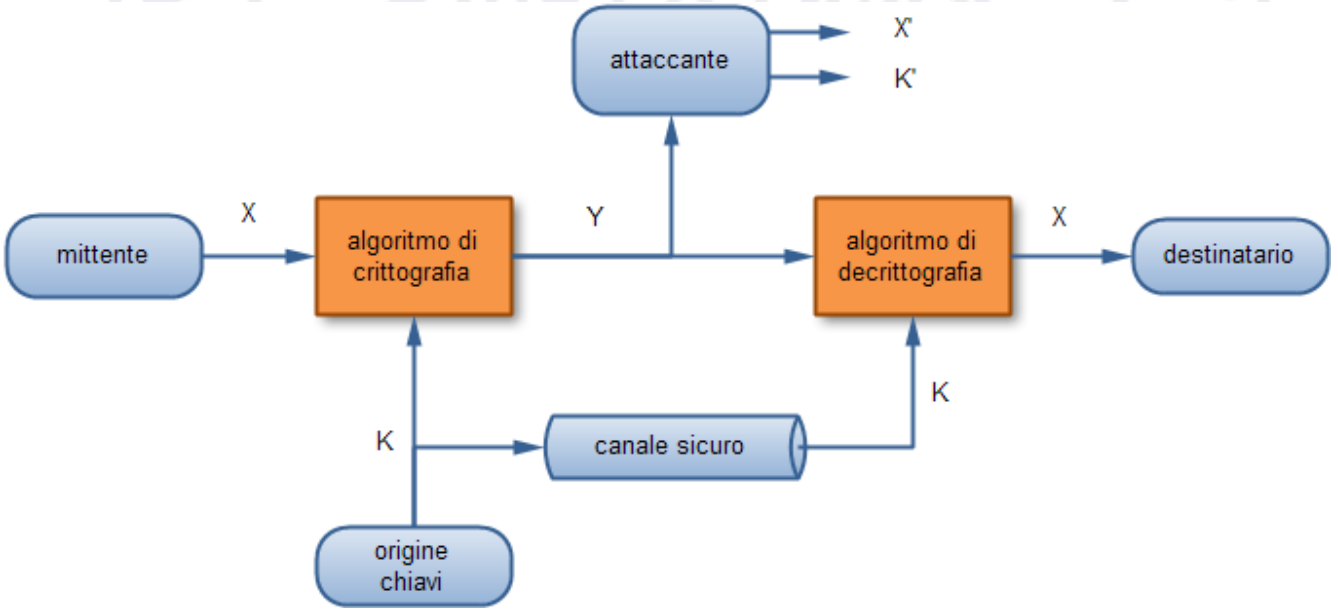
Crittografia asimmetrica (a chiave pubblica)

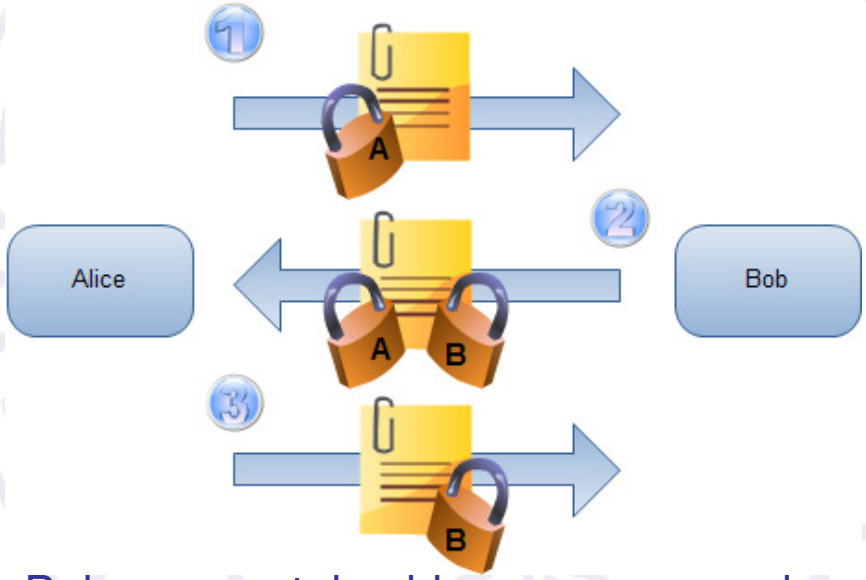
Problemi legati alla crittografia simmetrica

- Il principale problema della crittografia simmetrica sta nella necessità di disporre di un canale sicuro per la trasmissione della chiave
- Inoltre, se *n* persone devono comunicare tra loro, sono necessarie *n(n-1)/2* chiavi



Trasmissione della chiave

- E' possibile risolvere il problema dello scambio delle chiavi?
- Un esempio con chiavi «fisiche»:



- Alice manda a Bob una scatola chiusa con un suo lucchetto
- Bob aggiunge il proprio lucchetto e rispedisce la scatola ad Alice
- Alice rimuove il proprio lucchetto e rispedisce la scatola a Bob
- Il contenuto della scatola è stato inviato in maniera sicura senza alcuno scambio di chiavi!

Tecnica dei due lucchetti

- Si può applicare questa idea alla crittografia?
- Il problema sta nell'ordine delle operazioni:

  D2(D1(E2(E1(M)))) = M ?
- Solitamente no! Si ha solo che

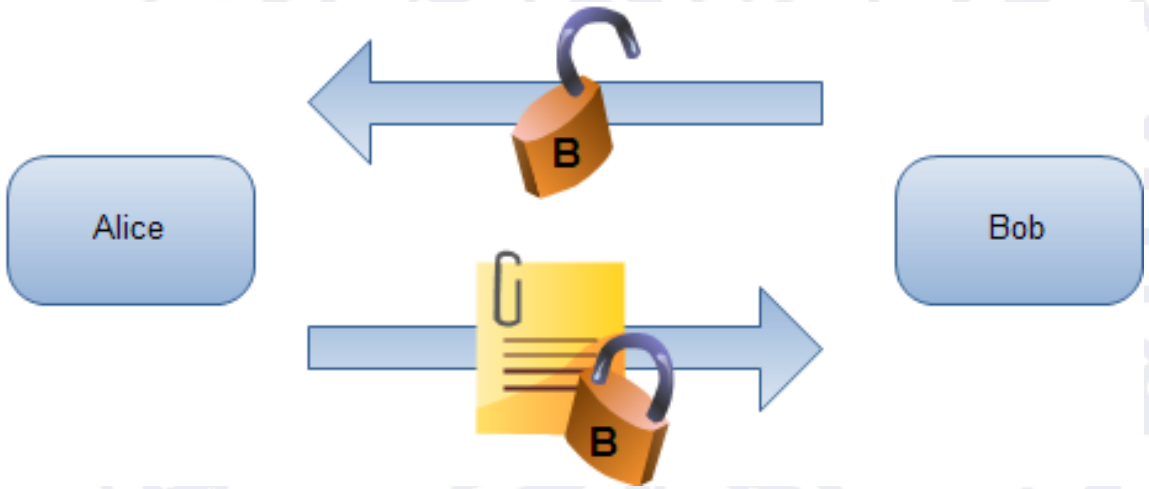
  D1(D2(E2(E1(M)))) = M

M = messaggio da trasmettere
E1 = encrypt di Alice
E2 = encrypt di Bob
D1 = decrypt di Alice
D2 = decrypt di Bob

Funziona solo con i cosiddetti *cifrari commutativi*, come quello di Cesare

Un altro esempio…

- In realtà si può ipotizzare uno scenario ancora più semplice:



- Bob spedisce ad Alice il suo lucchetto aperto (non serve un canale sicuro!)
- Alice lo usa per chiudere la scatola contenente il messaggio segreto e la spedisce a Bob

(al solito, sicuro solo per attacchi passivi)

Formalizzazione del problema

- Per applicare lo schema visto in precedenza, è necessario trovare una funzione (il lucchetto) la cui trasmissione su canali insicuri non compromette l'algoritmo, che sia facile da applicare (chiudere il lucchetto) ma difficile da invertire (aprire il lucchetto) a meno di non possedere un determinato elemento (la chiave del lucchetto)

$C=F(M)$	facile
$M=F^{-1}(C)$	difficile se non si conosce la chiave

La ricerca di una funzione con tali caratteristiche è stata la grande sfida per i crittografi degli anni '70

Cenni di aritmetica modulare

- a mod m = resto della divisione a/m
- a ≡ b (mod m) significa a mod m = b mod m
  (e si legge "'a' congruo a 'b' modulo m")
- Aritmetica modulare:

[(a mod m) + (b mod m)] mod m = (a+b) mod m

[(a mod m) - (b mod m)] mod m = (a-b) mod m

[(a mod m) · (b mod m)] mod m = (a · b) mod m

[esempio]

540 mod 17 = ?

540 / 17 = 31,764....

31 * 17 = 527

540 – 527 = 13

[esempio]

$$[(a \bmod m) + (b \bmod m)] \bmod m = (a+b) \bmod m$$

$$(6+7) \bmod 5 = 3$$

$$(6 \bmod 5 + 7 \bmod 5) \bmod 5$$

$$(1 + 2) \bmod 5 = 3$$

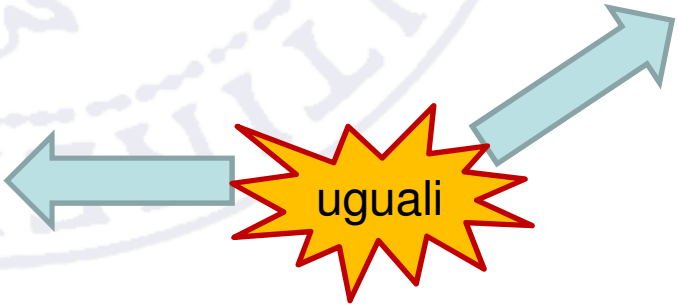
Cenni di aritmetica modulare / 2

- Conseguenza dell'ultima proprietà
- $[(a \bmod m)^k] \bmod m = a^k \bmod m$

Ad esempio

$[(9 \bmod 5)^2] \bmod 5 = 4^2 \bmod 5 = 16 \bmod 5 = 1$

$9^2 \bmod 5 = 81 \bmod 5 = 1$



Cenni di aritmetica modulare / 3

- Molte funzioni normalmente invertibili, diventano non invertibili nella versione modulare
- Esempio: il logaritmo

  $a^b = c$. Trovare *b* dati *a* e *c* è computazionalmente semplice (logaritmo: $b = \log_a(c)$)

  $a^b \bmod m = c$. Trovare *b* dati *a*, *c* ed *m* è computazionalmente molto difficile! (logaritmo discreto)

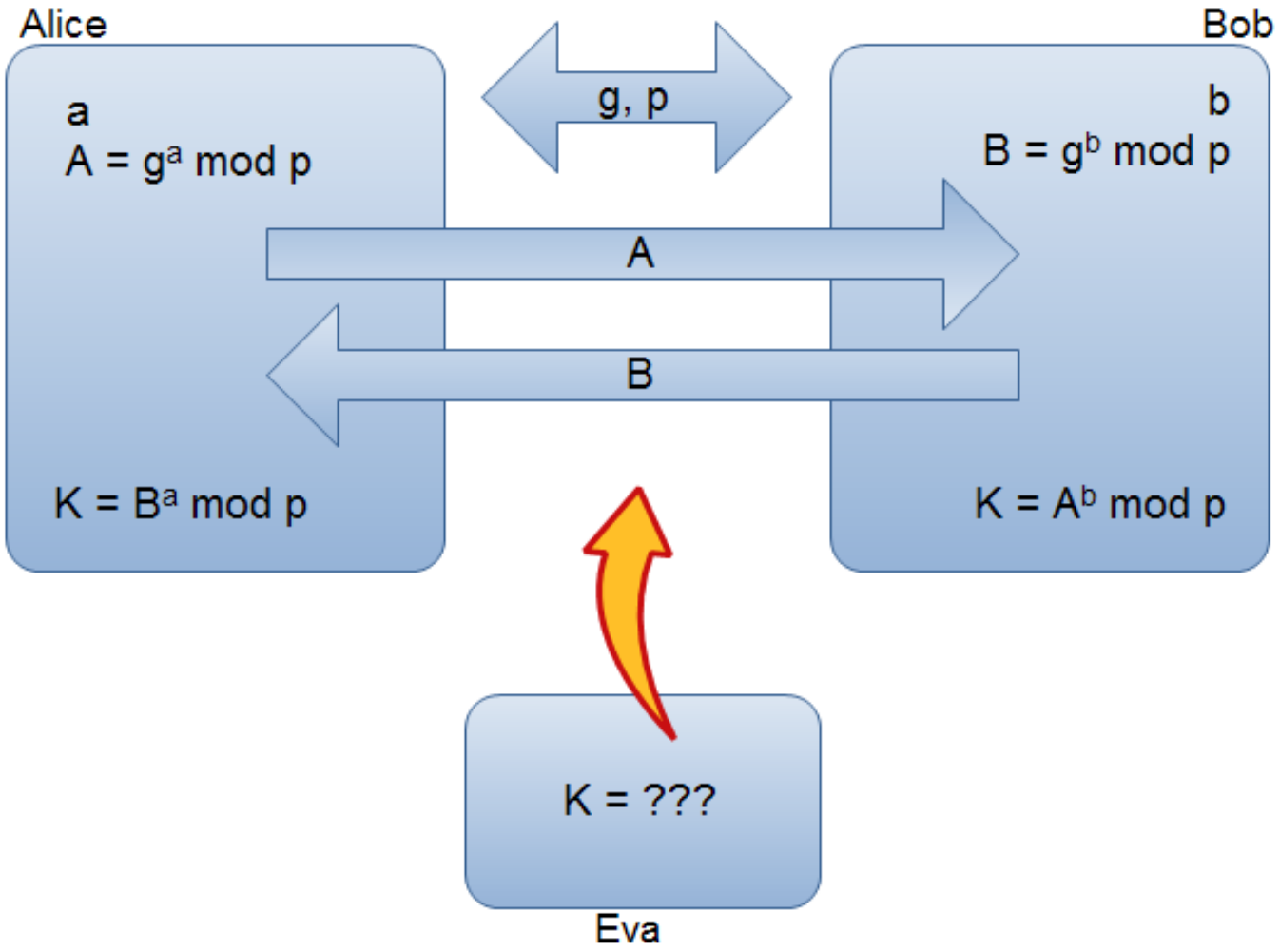
Algoritmo di Diffie-Hellman

- Supponiamo che Alice e Bob conoscano entrambi due numeri, g e p, pubblici (p numero primo). Inoltre Alice conosce un numero segreto 'a' e Bob conosce un numero segreto 'b'
- Alice calcola $A = g^a \bmod p$ e lo comunica a Bob
- Bob calcola $B = g^b \bmod p$ e lo comunica ad Alice
- Alice calcola $K = B^a \bmod p = [g^b \bmod p]^a \bmod p = g^{ab} \bmod p$
- Bob calcola $K = A^b \bmod p = [g^a \bmod p]^b \bmod p = g^{ab} \bmod p$

Alice e Bob hanno condiviso un segreto (il numero K) senza comunicarlo esplicitamente! L'attaccante Eva può osservare A, B, g, p ma questa informazione non è sufficiente per ricavare K

K è calcolabile solo conoscendo a o b, che tuttavia sono segreti e non vengono mai trasmessi. Ricavare *a* da A (o analogamente *b* da B) significa risolvere un logaritmo discreto ⇒ computazionalmente difficile!

Algoritmo di Diffie-Hellman / 2



Esempio

- g = 5, p = 23 (pubblici)
- a = 6 (Alice)
- b = 15 (Bob)

- Alice calcola A = 5^6 mod 23 = 8 e lo comunica a Bob
- Bob calcola B = 5^{15} mod 23 = 19 e lo comunica ad Alice

- Alice calcola K = 19^6 mod 23 = 2
- Bob calcola K = 8^{15} mod 23 = 2

Come calcolare le potenze

- $5^{15} \bmod 23 = 5^8 \times 5^4 \times 5^2 \times 5^1 \bmod 23$
  $= [5^8 \bmod 23 \times 5^4 \bmod 23 \times 5^2 \bmod 23 \times 5^1 \bmod 23] \bmod 23$

- $5^1 \bmod 23 = 5$
- $5^2 \bmod 23 = 25 \bmod 23 = 2$
- $5^4 \bmod 23 = (5^2)^2 \bmod 23 = (5^2 \bmod 23)^2 \bmod 23$
  $= 2^2 \bmod 23 = 4 \bmod 23 = 4$
- $5^8 \bmod 23 = (5^4)^2 \bmod 23 = (5^4 \bmod 23)^2 \bmod 23$
  $= 4^2 \bmod 23 = 16 \bmod 23 = 16$

- $5^{15} \bmod 23 = [16 \times 4 \times 2 \times 5] \bmod 23 = 640 \bmod 23 = 19$

Come calcolare le potenze / 2

- La scomposizione dell'esponente in somme di potenze di 2 è comoda perché permette di riutilizzare i calcoli precedenti, ma non è l'unica possibilità. Con un po' di allenamento si possono trovare scomposizioni più efficienti

$7^{12} \bmod 5 = [\ 7^3 \times 7^3 \times 7^3 \times 7^3\ ] \bmod 5$

$7^3 \bmod 5 = 343 \bmod 5 = 3$

$7^{12} \bmod 5 = [\ 3 \times 3 \times 3 \times 3\ ] \bmod 5 = 81 \bmod 5 = 1$

Esercizio

- g = 7, p = 31 (pubblici)
- a = 5 (Alice)
- b = 12 (Bob)
- K = ?

[esercizio]

A = g^a mod p = 7^5 mod 31

= 7^2 * 7^2 * 7 mod 31

= 49 * 49 * 7 mod 31

= [49 mod 31 * 49 mod 31 * 7 mod 31] mod 31

= [18 * 18 * 7] mod 31

= [324 * 7] mod 31

= [14 * 7] mod 31 = 98 mod 31 = 5

[esercizio]

B = g^b mod p = 7^12 mod 31

= 7^5 * 7^5 * 7^2 mod 31

= [5 * 5 * 18] mod 31

= 450 mod 31 = 16

K (Alice) = B^a mod p

= 16^5 mod 31 =

= [16^2 * 16^2 * 16] mod 31

= [256 * 256 * 16] mod 31

= [8 * 8 * 16] mod 31

= 1024 mod 31 = 1

[esercizio]

K (Bob) = A^b mod p = 5^12 mod 31

= (5^3)^4 mod 31

= 125^4 mod 31

= 1^4 mod 31 = 1

[altro esercizio]

- g=7, p=71, a=51, b=12
- A=g^a mod p = 7^51 mod 71

  = ((((7^2)^2)^2)^2)^3 * 7^3 mod 71

  ((7^2)^2) = 2401 mod 71 = 58

  (58^2) = 3364 mod 71 = 27

  (27^2) = 729 mod 71 = 19

  (19^3) = 6859 mod 71 = 43

= (43 * 7^3) mod 71 = 14749 mod 71 = 52

[altro esercizio]

- B = g^b mod p = 7^12 mod 71
  = ((7^3)^2)^2 mod 71
  = (59^2)^2 mod 71
  = 2^2 mod 71 = 4

K (Alice) = B^a mod p = 4^51 mod 71
((4^4)^4)^3 * 4^3 mod 71 =
(43^4)^3 * 4^3 mod 71 =
9^3 * 4^3 mod 71 =
19 * 64 mod 71 = 1216 mod 71 = 9

[esercizio]

- K (Bob) = A^b mod p = 52^12 mod 71 = …

Considerazioni sull'algoritmo di Diffie-Hellman

- Usando l'algoritmo sviluppato da Diffie ed Hellman nel 1976, Alice e Bob possono condividere un numero segreto senza trasmetterlo

- Se questo numero è la chiave di un algoritmo a cifratura simmetrica, si è trovato un modo per condividere la chiave senza la necessità di un canale di comunicazione sicuro!

- L'unico modo che Eva ha di trovare la chiave è quello di calcolare un logaritmo discreto. Tuttavia attualmente non sono noti algoritmi che risolvano questo problema in maniera efficiente

Considerazioni sull'algoritmo di Diffie-Hellman

- Attenzione: Alice e Bob ora condividono un numero K, ma non hanno potuto scegliere il valore di tale numero

- **In altre parole l'algoritmo di Diffie-Hellman non è un algoritmo di crittografia (non si può scegliere *quale* dato inviare) ma è esclusivamente un algoritmo di *scambio delle chiavi***

Crittanalisi dell'algoritmo di Diffie-Hellman

- La crittanalisi di D-H diventa quindi un problema matematico
- Non sono noti algoritmi generici per calcolare facilmente i logaritmi discreti per numeri grandi (solitamente p è un numero di almeno 300 cifre, a e b di almeno 100. g può assumere un valore piccolo, tipicamente 2 o 5. Con questi valori il problema è attualmente considerato praticamente insolubile)
- Esistono in realtà tecniche per il calcolo rapido dei logaritmi discreti per alcuni particolari valori di p che soddisfano determinate proprietà matematiche. Tale valore va quindi scelto accuratamente (tipicamente è un numero primo)

Crittografia asimmetrica

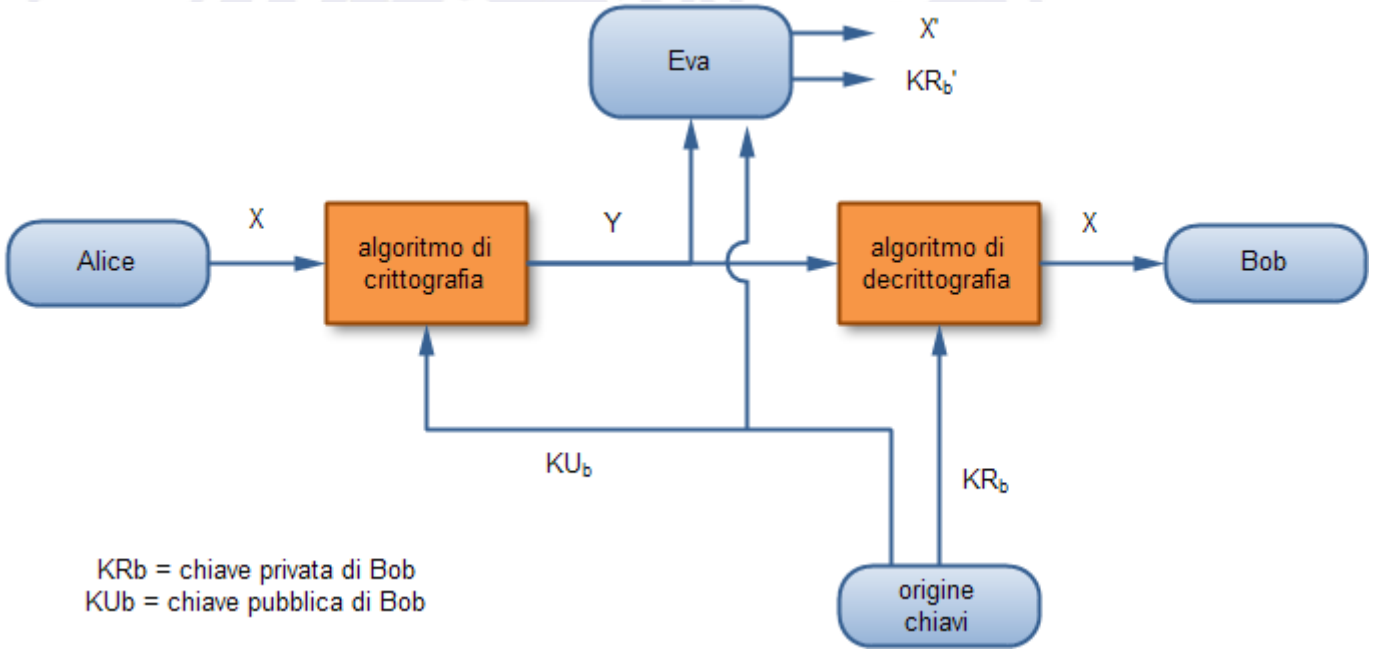
- In seguito alle scoperte di Diffie ed Hellman, la ricerca in campo crittografico divenne molto attiva

- Si giunse così all'invenzione della crittografia asimmetrica (detta anche "a chiave pubblica"), forse l'unica vera rivoluzione nella storia della crittografia. L'approccio infatti è completamente diverso da quello basato su sostituzioni e permutazioni che aveva imperversato dai tempi di Cesare fino ai giorni nostri (DES, AES, ecc…)

Crittografia asimmetrica / 2

- Idea: ogni utente ha due chiavi, una pubblica e una privata
- Alice invia a Bob un messaggio cifrandolo con la chiave pubblica di Bob
- Solo Bob può decifrarlo usando la corrispondente chiave privata

Le chiavi pubbliche si possono trasmettere su canali non sicuri!



Il problema della fattorizzazione

- Come già anticipato, il cuore della crittografia asimmetrica è una funzione facile da computare ma difficile da invertire, a meno di non conoscere un particolare dato

- Nel caso dell'algoritmo di Diffie-Hellman, questa funzione è il logaritmo discreto

- Un altro problema con caratteristiche simili è quello della fattorizzazione: dati due numeri primi p e q è facile calcolare $n = pq$, ma dato n è difficile (tempo di computazione esponenziale) risalire ai suoi fattori p e q

RSA

- L'algoritmo RSA (dal nome degli inventori Rivest, Shamir e Adleman) è il più famoso algoritmo di crittografia a chiave pubblica

- Inventato nel 1977, poco dopo l'algoritmo di Diffie-Hellman

- Due componenti principali
  - Algoritmo di generazione delle chiavi
  - Algoritmo crittografico vero e proprio

RSA – generazione delle chiavi

- Scegliere due numeri primi *p* e *q*
- Calcolare $n = pq$
- Scegliere *e*, coprimo e più piccolo di $(p-1)(q-1)$
- Calcolare *d* tale che $de \equiv 1 \mod (p-1)(q-1)$

- La coppia (n, e) è la chiave pubblica
- La coppia (n, d) è la chiave privata

- Non è possibile risalire facilmente dalla chiave pubblica a quella privata (e viceversa), in quanto servirebbe conoscere il numero $(p-1)(q-1)$, e questo implica fattorizzare *n* nei suoi fattori *p* e *q* (problema difficile)

Numeri coprimi

- Cosa significa ‘coprimo’?
- *a* è coprimo di *b* se il massimo comune divisore tra *a* e *b* è 1
- Ad es. 7 e 15 sono coprimi, mentre 8 e 10 no (hanno in comune il divisore 2)

- Nota: se *a* è primo, allora è coprimo di qualsiasi numero che non sia diviso da *a*
- Ad es. 7 è coprimo di tutti i numeri che non sono multipli di 7

RSA – esempio di generazione chiavi

- Siano p=3, q=11
- n=pq=33, (p-1)(q-1)=20
- Scegliamo e = 7 (7<20, 7 coprimo di 20)
- d = 3, infatti 3*7 = 21 ≡ 1 mod 20

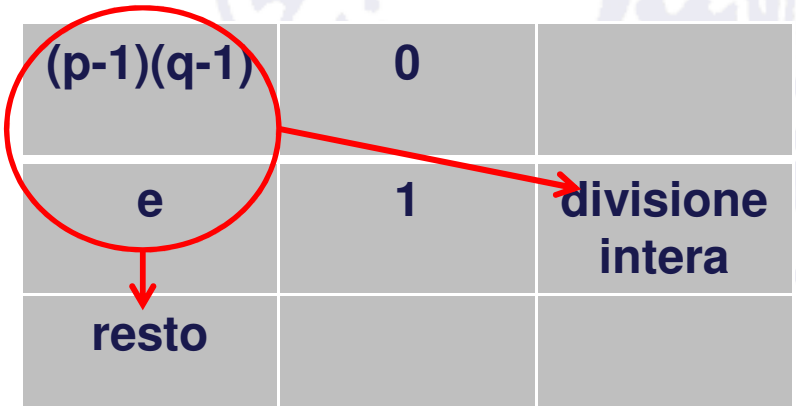
- La chiave pubblica è (33, 7)
- La chiave privata è (33, 3)

Come calcolare d?

- Metodo di Euclide esteso
- Si inizia con questa tabella:

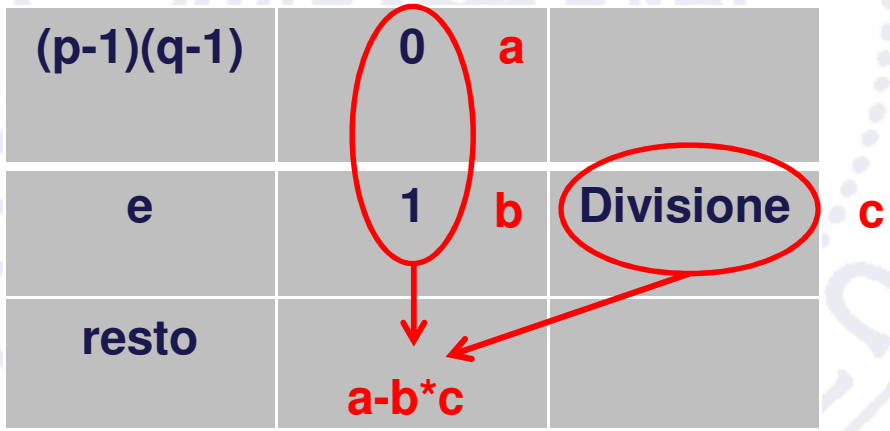
(p-1)(q-1)	0	
e	1	

- Si calcolano il risultato intero e il resto della divisione dei due numeri nella prima colonna e li si salvano nella tabella:



Come calcolare d? / 2

- Nella seconda colonna, terza riga si scrive il valore della seconda colonna, prima riga meno quello della seconda colonna, seconda riga moltiplicato per il risultato intero della divisione appena calcolato.



*Errore frequente: calcolare (a – b)*c anziché a – b*c*

Come calcolare d? / 3

- Il procedimento si ripete, aggiungendo nuove righe alla tabella e calcolandone i valori usando le due righe precedenti. Ci si ferma quando nella prima colonna compare un 1.
- Il risultato nella seconda colonna è il valore d cercato (se negativo, sommare (p-1)(q-1))

- Esempio: trovare d tale che

  $d * 7 \equiv 1 \bmod 20$

Metodo di Euclide esteso

20	0	
7	1	

Metodo di Euclide esteso

20	0	
7	1	2
6		

2 = risultato intero della divisione 20 / 7

6 = resto della divisione 20 / 7

Metodo di Euclide esteso

20	0	
7	1	2
6	-2	

$$0 - 1 * 2 = -2$$

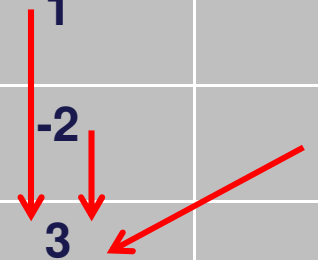
Metodo di Euclide esteso

20	0	
7	1	2
6	-2	1
1		

7/6 = 1 col resto di 1

Metodo di Euclide esteso

20	0	
7	1	2
6	-2	1
1	3	



1 - (-2*1) = 3

d = 3

Infatti $3 * 7 \equiv 1 \bmod 20$

Metodo di Euclide esteso / 2

- Altro esempio. Trovare d tale che

  d * 23 ≡ 1 mod 120

120	0	
23	1	5
5	-5	4
3	21	1
2	-26	1
1	47	

Metodo di Euclide esteso / 3

- Altro esempio. Trovare d tale che

  d * 7 ≡ 1 mod 60

60	0	
7	1	8
4	-8	1
3	9	1
1	-17	

d = -17 + 60 = 43 (se il numero è negativo, si somma il modulo)

(un altro metodo semplice per calcolare d…)

- de mod (p-1)(q-1) = 1
- de = k(p-1)(q-1) + 1
- d = (k(p-1)(q-1) + 1) / e

Provo k=1, 2, 3 …

Finché non trovo un valore INTERO per d

Esercizio

- p=7, q=13, e=…
- Calcolare la chiave pubblica (n,e) e privata (n,d)

[esercizio]

- p=7, q=13, e=11
- n = p*q = 91
- (p-1)*(q-1) = 72
- d*11 mod 72 = 1
- d = -13 + 72 = 59

72	0	
11	1	6
6	-6	1
5	7	1
1	-13	

- 59*11 mod 72 = 1
- Pubblica: (91,11) privata: (91, 59)

RSA – cifratura e decifratura

- Dato un messaggio *m* (0 < m < n)
- Cifratura: calcolare $c = m^e \bmod n$
- Decifratura: calcolare $m = c^d \bmod n$

(n,d) (n,e)
rispettivamente chiave privata e pubblica del destinatario (Bob)

(nota: si basa sull'ipotesi che l'esponenziazione modulare sia un problema difficilmente invertibile – ipotesi RSA).

RSA – Esempio di cifratura e decifratura

- Chiave pubblica: (33, 7) chiave privata: (33, 3)
- m = 15

- Cifratura: $c = m^e \bmod n = 15^7 \bmod 33 = 27$
- Decifratura: $m = c^d \bmod n = 27^3 \bmod 33 = 15$

Esercizio

- Chiave pubblica: (65,5). Privata: (65,29)
- m = 7
- Cifrare e decifrare il messaggio m

[esercizio]

- n=65, e=5, d=29, m=7
- Cifratura: c = 7^5 mod 65 =

  16807 mod 65 = 37
- Decifratura: 37^29 mod 65

  = (((37^3)^3)^3) * 37^2 mod 65

  = ((18^3)^3) * 37^2 mod 65

  = (47^3) * 37^2 mod 65

  = 18 * 4 mod 65

  = 72 mod 65 = 7

Dimostrazione del funzionamento di RSA

- **Teorema del toziente di Eulero**

Se *m* è coprimo di *pq* allora…

$$m^{(p-1)(q-1)} \equiv 1 \bmod pq$$

Dimostrazione del funzionamento di RSA

- $c^d \bmod n = [m^e \bmod n]^d \bmod n = m^{de} \bmod n$
- Siccome $de \equiv 1 \mod (p-1)(q-1)$ allora
- $m^{de} \bmod n = m^{k(p-1)(q-1)+1} \bmod n$

$$= m \cdot [m^{(p-1)(q-1)}]^k \bmod n$$

Per il teorema del toziente(*), ricordando che $n=pq$

$$= m \cdot 1^k \bmod n$$

$$= m \bmod n$$

- Quindi $c^d \bmod n = m \bmod n = m$ (perché $m<n$)

(*) se m è coprimo di n. In realtà il procedimento vale per tutti gli m, ma la dimostrazione in questo caso va oltre gli scopi di questo corso

Applicazioni reali di RSA

- Per garantire la non invertibilità delle funzioni utilizzate, è importante usare numeri sufficientemente grandi. Nelle applicazioni attuali, solitamente n è un numero di almeno 1024 bit (poco più di 300 cifre decimali).

Crittografia ibrida

Crittografia simmetrica	Crittografia asimmetrica
Pro: molto veloce	*Pro:* non serve un canale sicuro per lo scambio delle chiavi
Contro: problema dello scambio delle chiavi	*Contro:* molto lenta, a causa dei calcoli complessi da effettuare

Approccio ibrido:

- si usa la crittografia a chiave pubblica solo per la trasmissione di una chiave
- Il resto del messaggio è cifrato con una tecnica simmetrica usando la chiave trasmessa in precedenza

Firme digitali

- Le tecniche di crittografia a chiave pubblica mostrate finora garantiscono la riservatezza del messaggio (sicurezza contro attacchi *passivi*) ma non garantiscono nulla sull'identità del mittente, in quanto chiunque ha accesso alla chiave pubblica del destinatario (vulnerabilità ad attacchi *attivi*)

- Soluzione: Alice invia un messaggio cifrato con la propria chiave PRIVATA. Bob potrà decifrarlo usando la chiave pubblica di Alice

- Cifratura: $c = m^d \bmod n$
- Decifratura: $m = c^e \bmod n$

(n,d) (n,e)
rispettivamente chiave privata e pubblica del mittente (Alice)

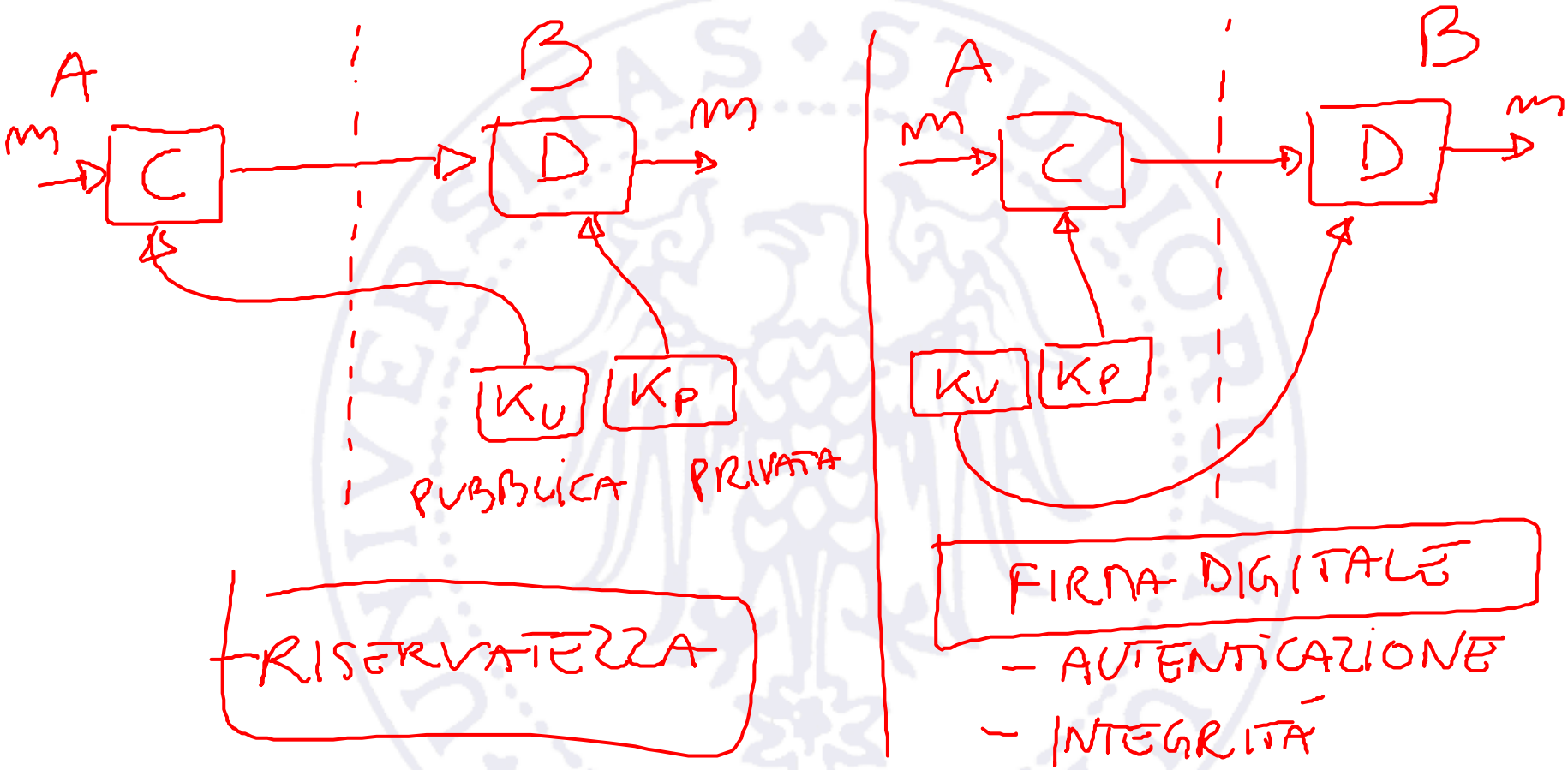
Firme digitali / 2

- In questo modo **non** si garantisce la segretezza del messaggio, in quanto chiunque può decifrarlo usando la chiave pubblica di Alice. Tuttavia si garantisce …

- **Autenticazione**: Bob è sicuro che il mittente sia davvero Alice, perché solo lei possiede la chiave privata corrispondente alla chiave pubblica usata da Bob per decifrare il messaggio

- **Integrità:** il messaggio non è stato alterato da terzi, in quanto una sua modifica richiederebbe la conoscenza della chiave privata di Alice

- **Non ripudiabilità**: Alice non potrà negare di essere l'autrice del messaggio (conseguenza di autenticazione + integrità)


A
B
M
C
D
M
KU
KP
PUBBLICA
PRIVATA
RISERVATEZZA
A
B
M
C
D
M
KU
KP
FIRMA DIGITALE
– AUTENTICAZIONE
– INTEGRITÁ
– NON RIPUDIABILITÁ

Autenticazione e segretezza

- Il messaggio può essere cifrato due volte, prima con la chiave privata di Alice e poi con quella pubblica di Bob (o viceversa)
- In questo modo è garantita sia la segretezza (grazie alla cifratura con la chiave pubblica di Bob)…
- …sia l'autenticazione/integrità, grazie alla cifratura con la chiave privata di Alice

Scambio delle chiavi

- Il meccanismo a chiave pubblica/privata permette di trasmettere le chiavi su canali non sicuri (l'intercettazione della chiave non mette a rischio il messaggio crittato)
- Tuttavia sorge un altro problema: come garantire la paternità delle chiavi? (es: Eva manda a Bob la propria chiave pubblica, spacciandola per quella di Alice)

- Necessità di un sistema di PKI (Public Key Infrastructure), ovvero una infrastruttura per la gestione e lo scambio delle chiavi

PKI

- Annuncio pubblico: ogni entità gestisce autonomamente la diffusione della propria chiave pubblica.

- Elenco pubblico: ogni entità trasmette la propria chiave pubblica ad un sistema centralizzato di gestione delle chiavi (una sorta di “elenco telefonico” delle chiavi pubbliche).

- L’elenco è gestito da un’autorità fidata, che si occupa di verificare la paternità della chiave.

PKI / 2

- Autorità certificante (certificate authority): la paternità di una chiave è garantita da un *certificato*, firmato digitalmente da una autorità certificante nota e fidata.

- Web of Trust: una rete distribuita di certificazioni, in cui ognuno si fa garante dell'autenticità delle chiavi di cui è in grado di verificare la paternità

- Ad es. Alice certifica la chiave di Bob, e Bob certifica quella di Carol. Alice può "fidarsi" della chiave di Carol anche se non può verificarne direttamente la paternità. La paternità è garantita dall'amico in comune, Bob.

Crittanalisi

- La sicurezza degli algoritmi a chiave pubblica si basa sulla presunta impossibilità pratica di invertire alcune funzioni (problema del logaritmo discreto, problema della fattorizzazione…)
- Purtroppo tale impossibilità non è mai stata dimostrata formalmente

- Inoltre, nuovi paradigmi di computazione potrebbero rendere “facile” ciò che attualmente è computazionalmente “difficile” (computer quantistici)

- In ogni caso, la matematica offre molti modi per affrontare i problemi suddetti con tecniche più efficienti della forza bruta. Per questo la dimensione delle chiavi nella crittografia asimmetrica è considerevolmente maggiore di quella usata per la crittografia simmetrica (es. chiavi da 1024 o 2048 bit)