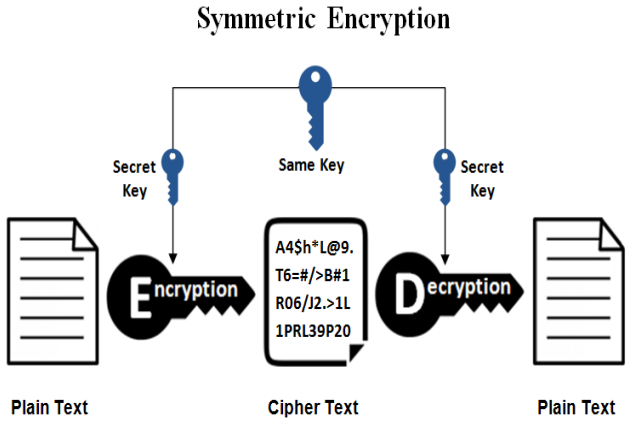
Schemi di cifratura a chiave publica e Diffie-Hellman

Indice

- Introduzione
- Requisiti crittografici
- Teoria dei numeri
- Due breaktrough
  - Diffie-Hellman
  - RSA
- Attacchi e fattorizzazione

Crittografia a chiave simmetrica

La crittografia a chiave simmetrica si riferisce a metodi in cui sia il mittente che il destinatario condividono la stessa chiave.



- Chiamata anche crittografia a chiave singola o convenzionale
- L'unico tipo di cifratura in uso prima della fine degli anni '70
- Il più diffuso tra i due tipi di crittografia

Crittografia a chiave simmetrica

- Plaintext: il messaggio originale dato come input all'algoritmo.
- Chiave segreta: Un altro input all'algoritmo di cifratura
- Algoritmo di cifratura: esegue varie sostituzioni e trasformazioni in chiaro utilizzando la chiave segreta per ottenere un testo cifrato
- Ciphertext: il messaggio criptato prodotto come output; dipende dal testo in chiaro e dalla chiave segreta
- Algoritmo di decifratura: è il simmetrico dell'algoritmo di cifratura; dal testo cifrato produce quello originale in chiaro.

Crittografia a chiave simmetrica

Gli schemi di cifratura (tradizionali) a chiave privata/segreta/singola:

- richiedono che il mittente e il destinatario abbiano ottenuto una copia della chiave segreta in modo sicuro e la tengano al sicuro.
- non proteggono un mittente da un destinatario che falsifica un messaggio e che poi asserisce che la richiesta è stata inviata dal mittente.

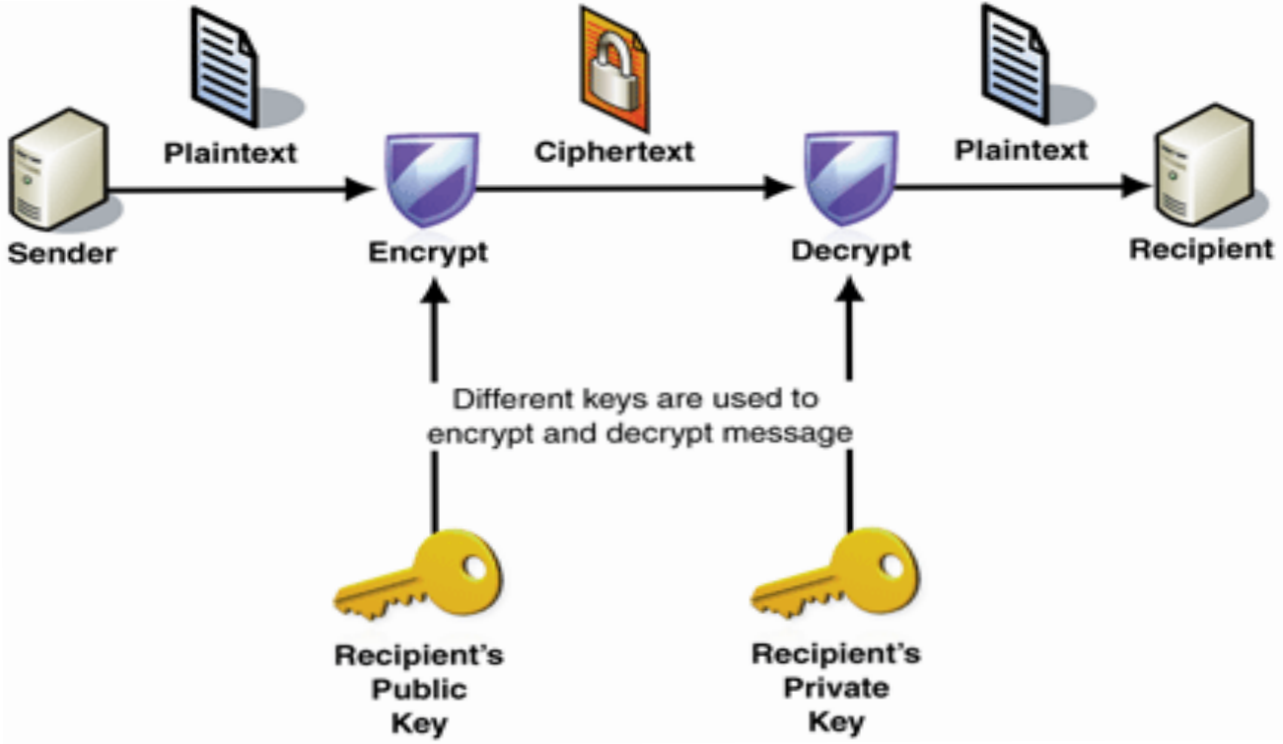
Crittografia a doppia chiave

- Gli schemi di cifratura (nuovi) a chiave pubblica / doppia / asimmetrica:
  - Prevedono l'uso di due chiavi:
    - una *chiave pubblica*, che può essere conosciuta da chiunque e che può essere utilizzata per criptare i messaggi e verificare le firme
    - una corrispondente *chiave privata*, nota solo al destinatario, utilizzata per decifrare i messaggi e firmare (creare) firme
  - Realizzano uno scambio asimmetrico: chi codifica i messaggi o verifica le firme non può decodificare i messaggi o creare firme

Crittografia a doppia chiave

- Sviluppato per affrontare due questioni important:
  - distribuzione delle chiavi: garantire comunicazioni sicure con una chiave prersonale senza dipendere da un centro distribuzuone chiavi o fidarsi del comportamento di altri.
  - firme digitali: verificare che un messaggio provenga intatto dal mittente dichiarato
- Complementa piuttosto che sostituire la crittografia a chiave privata
- Si basa su proprietà garantite dalla teoria dei numeri piuttosto che sull'uso permutazioni e sostituzioni

Cifratura asimmetrica



Cifratura asimmetrica vs. cifratura simmetrica

- **Crittografia simmetrica:** stesso algoritmo usato per cifrare e decifrare con la stessa chiave. Chiave e algoritmo sono condivisi da mittente e ricevente. La chiave deve essere tenuta segreta. Quasi impossibile decifrare un messaggio se si conoscono solo l'algoritmo e il testo cifrato.
- **Crittografia asimmetrica:** stesso algoritmo usato per cifrare e decifrare cifrare ma si usano due chiavi: una per cifrare e una per decifrare. Il mittente e il ricevente devono avere ognuno una chiave che fa coppia con l'altra (non la stessa). Quasi impossibile decifrare un messaggio se si conoscono solo l'algoritmo, il testo cifrato e una delle chiavi (da una non si può risalire all'altra).

Cifratura asimmetrica vs. simmetrica

Crittografia simmetrica

- Stesso algoritmo usato per cifrare e decifrare con la stessa chiave.
- Chiave e algoritmo sono condivisi da mittente e ricevente.
- La chiave deve essere tenuta segreta.
- Quasi impossibile decifrare un messaggio se si conoscono solo l'algoritmo e il testo cifrato.

Crittografia asimmetrica

- Stesso algoritmo usato per cifrare e decifrare cifrare ma si usano due chiavi: una per cifrare e una per decifrare.
- Mittente e ricevente devono avere ognuno una chiave che fa coppia con l'altra (non la stessa).
- Quasi impossibile decifrare un messaggio se si conoscono solo l'algoritmo, il testo cifrato e una delle chiavi (da una non si può risalire all'altra).

Principi di crittografia a doppia chiave

- Si distingue tra la *chiave pubblica* di un soggetto (che viene divulgata) e la *chiave privata* che viene mantenuta segreta dal soggetto.
- Deve essere computazionalmente difficile ricavare la chiave di decifratura conoscendo l'algoritmo e la chiave di cifratura.
- Una qualunque delle due chiavi può essere usata per cifrare e l'altra per decifrare
- La cifratura con chiave pubblica garantisce **confidenzialità.**
- La cifratura con chiave privata garantisce **autenticazione**.

Crittografia a doppia chiave

- Plaintext: il messaggio originale dato come input all'algoritmo.
- Chiavi: una coppia di chiavi pubblica/privata generate in modo che una venga utilizzata per la cifratura e l'altra per la decifratura
- Algoritmo di cifratura: esegue sostituzioni e trasformazioni sul testo in chiaro utilizzando una delle due chiavi per ottenerne uno cifrato
- Ciphertext: il messaggio criptato prodotto come output; dipende dal testo in chiaro e da una delle due chiavi
- Algoritmo di decifratura: accetta il testo cifrato e le chiavi corrispondenti e produce il testo in chiaro originale.

Gestione delle chiavi

- Ogni utente genera una coppia di chiavi da utilizzare per la cifratura e la decifratura dei messaggi:
  - La chiave pubblica viene inserita in un registro pubblico o in un altro file accessibile
  - La chiave “gemella” è tenuta privata;
  - Ciascun utente mantiene una collezione di chiavi pubbliche di altri
- Se Bob cripta un messaggio usando la chiave pubblica di Alice, solo Alice potrà decifrarlo usando la propria chiave privata
- Se Bob cripta il messaggio utilizzando la sua chiave privata, Alice e chiunque conosca la chiave pubblica di Bob potrà decriptare il messaggio

Impatto della doppia chiave

Resa pubblica da Diffie e Hellman nel 1976

Basata su proprietà dei numeri (primi)

Utilizza due chiavi diverse ma collegate: pubblica e privata

Il suo uso ha avuto profonde conseguenze su: riservatezza, distribuzione delle chiavi, autenticazione

Principali proposte per PKC

- D-H (Diffie-Hellman 1976): Primo algoritmo a chiave pubblica reso pubblico. Permette a due utenti di condividere un segreto da utilizzare come chiave per successive cifratura simmetrica dei messaggi
- RSA (Rivest, Shamir, Adleman 1977): L'approccio più usato per la crittografia a chiave pubblica
- ECC (Koblitz and Miller 1985) - Crittografia a curva ellittica proposta come alternativa a RSA, con lo stesso livello di sicurezza ma con chiavi molto più piccole
- DSS (U.S.NIST 1991) Standard di firma digitale rivisto spesso fino al 2013. Fornisce solo la funzione di firma digitale, non può essere utilizzata per cifrare o scambiare chiavi.

Principali proposte per PKC

Algorithm	Digital Signature	Key Exchange	Encryption / Decryption
Rivest, Shamir, Adleman (RSA)	Yes	Yes	Yes
Diffie-Hellman (D-H)	No	Yes	No
Digital Signature Standard (DSS)	Yes	No	No
Elliptic Curve Cryptography (ECC)	Yes	Yes	Yes

Requisiti per le funzioni crittografiche

- Computazionalmente:
  - poco costoso creare coppie di chiavi
  - poco costoso cifrare messaggi per il mittente che conosce la chiave pubblica e decifrare messaggi per il destinatario che conosce la chiave private
  - difficile per un avversario scoprire la chiave privata conoscendo la chiave pubblica e decifrare un messaggio senza conoscere la chiave privata
- Deve essere possible una delle due chiavi correlate per la cifratura, e l'altra per la decifratura

Requisiti per le funzioni crittografiche/2

Gli schemi a chiave pubblica (PKC) dipendono da appropriate funzioni trap-door one-way

- Funzioni one-way
  - $Y = f(X)$ Facile!
  - $X = f^{-1}(Y)$ Difficile - non fattibile!
- Una funzione trap-door one-way deve essere tale che
  - $Y = f_k(X)$ è facile se k e X sono noti
  - $X = f_k^{-1}(Y)$ è facile se k e y sono noti
  - $X = f_k^{-1}(Y)$ è non fattibile, se Y è noto ma k non lo è

N.B. Un problema è facile se può essere risolto in tempo polinomiale in funzione della lunghezza dell'ingresso

Un esempio di funzione one-way

Dato il numero 6895601 stabilire se esso è il prodotto di due numeri primi, e quali sono questi numeri.

- Una soluzione naturale sarebbe quella di provare a dividere 6895601 per diversi numeri primi più piccoli del numero in considerazione fino a trovare la risposta.
- Ma se si sa che 1931 è uno dei numeri, si può trovare la risposta utilizzando una qualsiasi calcolatrice per calcolare 6895601 ÷ 1931

Attacchi

- Attacchi di forza bruta a PKC sono teoricamente possibili
- Per difendersi sono necessarie chiavi molto grandi: uno schema a chiave privata a 64 bit ha una sicurezza simile a ad una di 512 bit nella PKC RSA
- L'attacco è noto, ma è reso sufficientemente difficile da renderne la risoluzione impraticabile usando numeri molto grandi
- È più lento degli schemi a chiave privata

Teoria dei numeri

- La teoria dei numeri è fondamentale per affrontare le sfide della crittografia asimmetrica.
- Gli ingredienti chiave per lo sviluppo di una teoria della crittografia a doppia chiave sono:
  - I numeri primi
  - Aritmetica modulare
  - Esponenziazione e logaritmi

Numeri primi

- Un numero primo è un numero naturale maggiore di 1 che non può essere formato moltiplicando due numeri naturali.
- Un teorema fondamentale: ogni numero naturale è un numero primo o può essere ottenuto come prodotto delle potenze dei numeri primi:
  - $91 = 7 \times 13$
  - $3600 = 2^4 \times 3^3 \times 5^2$
  - $11011 = 7 \times 11^2 \times 13$

Numeri e numeri primi

- Teorema: Se P è l'insieme dei numeri primi, ogni generico numero intero positivo a può essere scritto come il prodotto di esponenziali di numeri primi

$$a = \prod_{p \in P} p^{a_p} \qquad \text{where each } a_p \geq 0$$

- N.B.: Per qualsiasi numero specifico, per la maggior parte dei numeri primi p della formula, l'esponente corrispondente sarà 0.

Numeri e numeri primi

- Corollario: Per moltiplicare due numeri è sufficiente sommare i corrispondenti esponenti.
- Esempio
  - Siccome: $91 = 7 \times 13$ e $11011 = 7 \times 11^2 \times 13$
  - Abbiamo: $91 \times 11011 = 7^2 \times 11^2 \times 13^2$
  - Verificatelo! …

MCM e MCD

- Il *Minimo Comune Multiplo* di due interi a e b - MCM(a, b), è il più piccolo intero positivo che è divisibile sia per a che per b.
  - MCM(4,6) = 12
    - Multiple of 4: 4, 8, 12, 16.....
    - Multiple of 6: 6, 12, 18....
- Il *Massimo Comun Divisore* di due interi a e b - MCD(a, b), è il più grande intero positivo che divide sia a che b.
  - MCD(54,24) = 6
    - 54 x 1 = 27 x 2 = 18 x 3 = 9 x 6 - i divisori di 54: 1, 2, 3, 6, 9, 18, 27,54
    - 24 x 1 = 12 x 2 = … 3 x 8 … - i divisori di 24 sono: 1,2,3,4,6,8,12,24

Aritmetica Modulare

- È un sistema di aritmetica per numeri interi, dove i numeri si "avvolgono" al raggiungimento di un certo valore - il modulo
- Si basa su una relazione di congruenza sugli interi che sia compatibile con le operazioni di addizione, sottrazione e moltiplicazione.
- Due numeri a e b sono detti congruenti modulo n ($a \equiv b \bmod n$), se la loro differenza a - b è un multiplo intero di n.
- $a \equiv b \bmod n$ stabilisce che a e b hanno lo stesso resto se divise per n, cioè $a = pn + r$, $b = qn + r$

Aritmetica modulare

- Esempio:
  - $38 \equiv 14 \pmod{12}$ perché
    - $38 - 14 = 24$, che è un multiplo di 12
    - Sia 38 che 14 hanno lo stesso resto (2) se divisi per 12.
- Properties:
  - Riflessività : $a \equiv a \pmod{n}$
  - Symmetria: $a \equiv b \pmod{n}$ se e solo se $b \equiv a \pmod{n}$
  - Transitività: Se $a \equiv b \pmod{n}$ e $b \equiv c \pmod{n}$, allora $a \equiv c \pmod{n}$

Aritmetica Modulare

- **Esempio**: 38 ≡ 14 mod 12 perché
  - 38 − 14 = 24, che è un multiplo di 12
  - sia 38 sia 14 hanno lo stesso resto 2 se divisi per 12.
- La relazione di congruenza soddisfa la seguenti proprietà:
  - **Riflessività**: a ≡ a (mod n)
  - **Simmetria**: a ≡ b (mod n) se e solo se b ≡ a (mod n)
  - **Transitività**: Se a ≡ b (mod n) e b ≡ c (mod n), allora a ≡ c (mod n)

Congruence for Modular Arithmetic

Due termini congruenti possono essere utilizzati in qualsiasi contesto in modo intercambiabile

- Se $a_1 \equiv b_1 \pmod{n}$ e $a_2 \equiv b_2 \pmod{n}$ allora:
  - $a_1 + a_2 \equiv b_1 + b_2 \pmod{n}$
  - $a_1 - a_2 \equiv b_1 - b_2 \pmod{n}$
  - $a_1 a_2 \equiv b_1 b_2 \pmod{n}$
- Se $a \equiv b \pmod{n}$ allora:
  - $a^k \equiv b^k \pmod{n}$ per qualsiasi numero intero non negativo k

Il piccolo teorema di Fermat

- Il piccolo teorema di Fermat : Dato un intero a e un primo p con a non divisibile per p, abbiamo : $a^{p-1} = 1 \pmod p$
- Un esempio : $7^{18} \equiv 1 \pmod{19}$

$$
\begin{aligned}
&a = 7, p = 19 \\
&7^2 = 49 \equiv 11 \pmod{19} \\
&7^4 \equiv 121 \equiv 7 \pmod{19} \\
&7^8 \equiv 49 \equiv 11 \pmod{19} \\
&7^{16} \equiv 121 \equiv 7 \pmod{19} \\
&a^{p-1} = 7^{18} = 7^{16} \times 7^2 \equiv 7 \times 11 \equiv 1 \pmod{19}
\end{aligned}
$$

Picture from: *W. Stalling: Cryptography and Network Security, International Edition, Pearson*

Il piccolo teorema di Fermat

Una variante:

Dato un intero a e un primo p:

- $a^p = a \pmod p$

$$p = 5, a = 3 \qquad a^p = 3^5 = 243 \equiv 3 (\mathrm{mod}\, 5) = a (\mathrm{mod}\, p)$$
$$p = 5, a = 10 \qquad a^p = 10^5 = 100000 \equiv 10 (\mathrm{mod}\, 5) \equiv 0 (\mathrm{mod}\, 5) = a (\mathrm{mod}\, p)$$

N.B.: In questo caso non è necessario che a non sia divisibile per p

Picture from: *W. Stalling: Cryptography and Network Security, International Edition, Pearson*

Numeri Coprimi

- Due interi a and b sono detti relativamente primi, reciprocamente primi, o coprimi se l'unico intero positivo che li divide entrambi è 1.
- Qualsiasi numero primo che divide un numero primo su due numeri coprimi non divide l'altro.
- Il massimo comun divisore (MCD) di due numeri coprimi è 1.

Teorema di Eulero– Totiente ϕ

- Dato un intero n, la funzione totiente di un numero n - $\phi(n)$ - corrisponde al numero di interi più piccoli di n che sono coprimi a n.
  - $\phi(15) = \#\{1,2,4,78,11,13,14\} = 8$
  - $\phi(17) = 16$ perché tutti i numeri interi da 1 a 16 sono coprimi con 17.
- Se n è primo allora $\phi(n) = n-1$
- Dati due numeri primi diversi p e q:

$$\text{Se } n = p \times q \text{ allora } \phi(n) = (p-1) \times (q-1)$$

Teorema di Eulero

- Teorema:

  Dati due numeri interi a e n che sono coprimi:

  $a^{\phi(n)} = 1 \pmod{n}$

- Una variante:

  Dati due numeri interi a e n che sono coprimi:

  $a^{\phi(n)+1} = a \pmod{n}$

Esempi per il teorema di Eulero

$$a^{\phi(n)} = 1 \pmod{n}$$

Due esempi

- Dati a = 3 e n = 10
  - $\phi(10) = \#\{1,3,7,9\} = 4$
  - $a^{\phi(10)} = 3^4 = 81 = 1 \pmod{10}$
- Dati a = 2 e n = 11,
  - $\phi(11) = 10$
  - $a^{\phi(10)} = 2^{10} = 1024 = 1 \pmod{11}$

Logaritmi Discreti

- Il logaritmo $\log_b a$ è un numero x tale che $b^x = a$
- Il logaritmo discrete $\log_b a$ è un intero k tale che $b^k = a$
- Non è noto alcun metodo efficiente per il calcolo dei logaritmi in generale.
- Importanti algoritmi nella crittografia a chiave pubblica basano la loro sicurezza sul presupposto che il problema del logaritmo discreto quando si usa l'aritmetica modulare non ha una soluzione efficiente.

Radici primitive

- Un numero g è una radice primitiva modulo n se ogni numero a coprimo di n è congruente ad una potenza di g modulo n.
- g è una radice primitiva modulo n se per ogni numero a coprimo di n esiste un numero intero k tale che
  $g^k \equiv a \pmod{n}$.
- Tale valore k è chiamato indice o logaritmo discreto di a in base g modulo n.

Il problema del logaritmo discreto

- Ci sono metodi efficienti per calcolare la radice primitive.
- Il calcolo del logaritmo discreto è l'operazione inversa del calcolo della radice primitiva: Dato un numero segreto b che soddisfa

  $$b^e \equiv c \pmod n$$

  bisogna trovare *b* dati *c*, *e* e *n*. NON FACILE!
- Senza la funzione di modulo si può sfruttare la corrispondenza

  $$\log_b(c) = e$$

  ma l'aritmetica modulare impedisce di utilizzare efficacemente il calcolo dei logaritmi.

Due Breaktrough

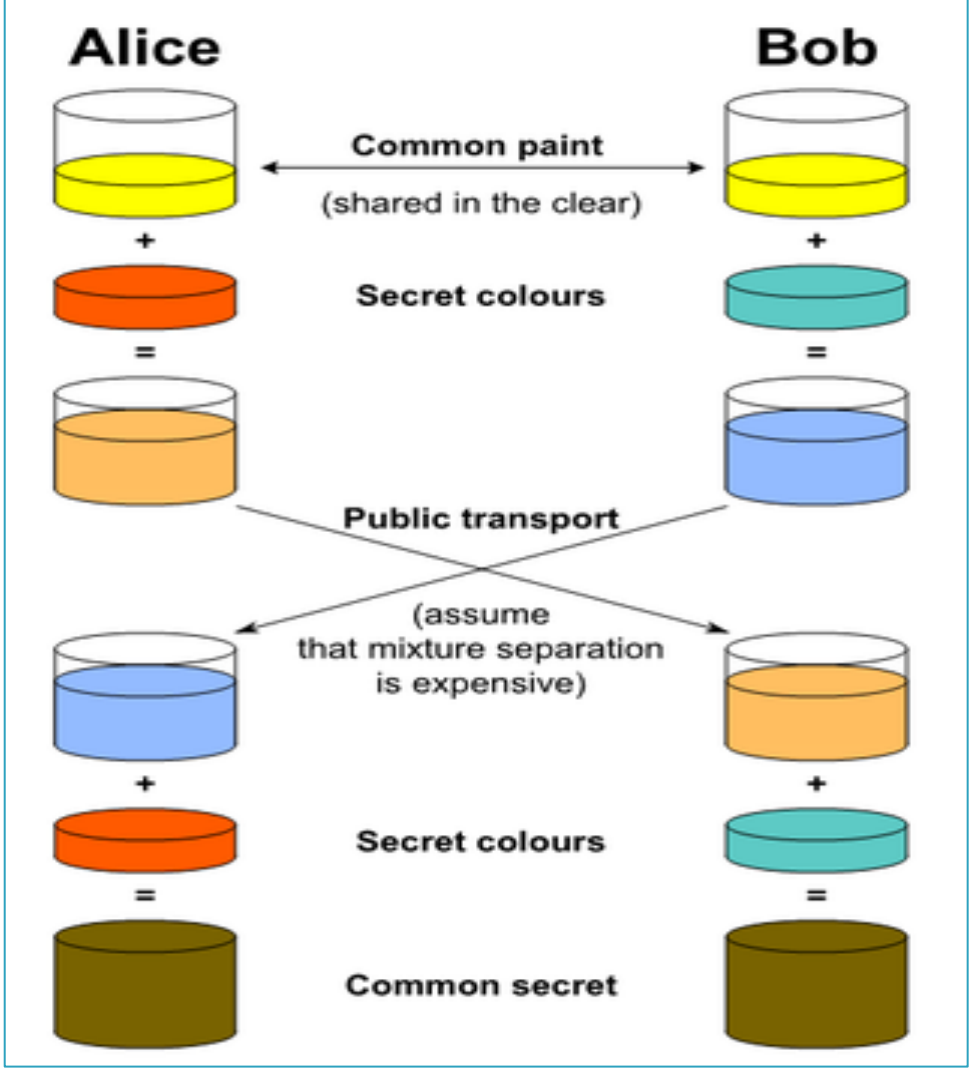
- Algoritmo per lo scambio di chiave (Diffie-Hellman-Merkle 1976):
  - L'algoritmo è stato progettato per consentire agli utenti di raggiungere un accordo sicuro su un segreto condiviso da utilizzare come chiave per le successive cifrature simmetriche.
  - Il brevetto USA 4.200.770 del 1977, ora scaduto, descrive l'algoritmo di dominio pubblico.
- Schema a chiave pubblica (RSA - Rivest, Shamir, Adleman 1977)
  - L'approccio più ampiamente accettato e implementato alla crittografia a chiave pubblica
  - la chiave di cifratura è pubblica e distinta dalla chiave di decifratura che è tenuta segreta (privata).

Diffie-Hellman

- L’algoritmo di Diffie-Hellman è un metodo per lo scambio sicuro di chiavi crittografiche su un canale pubblico ed è stato il primo protocollo a chiave pubblica ad essere reso noto.
- Tradizionalmente, la comunicazione criptata sicura tra due parti richiedeva che le chiavi venissero scambiate prima attraverso un qualche canale fisico (ad esempio un corriere di fiducia).
- Il metodo di scambio di chiavi Diffie-Hellman permette a due parti che non hanno alcuna conoscenza reciproca, di stabilire una chiave segreta condivisa su un canale non sicuro.
- Hellman ha suggerito di chiamare l'algoritmo Diffie-Hellman-Merkle

Diffie-Hellman Intuizione

Come ottenere un colore condiviso partendo da una vernice concordata e mantenendone due segret.



Diffie-Hellman: matematicamente

- Alice genera:
  - un numero primo p molto grande (1024 bit, circa 300 cifre decimali)
  - Una radice primitiva α di q più piccola di q
  - un numero segreto casuale a
- Alice calcola $\alpha^a \bmod q$ e lo manda a Bob (eventualmente insieme ad α e q).
- Bob sceglie un numero segreto casuale b, e calcola $\alpha^b \bmod q$ e lo manda ad Alice

Perché Diffie-Hellman funziona

- Alice calcola $\alpha^a \bmod q$ e lo manda a Bob (eventualmente insieme ad α e q).
- Bob sceglie un numero segreto casuale b, e calcola $\alpha^b \bmod q$ e lo manda ad Alice
- Con calcoli successivi i due ottengono la stessa chiave!
  - K_A
    $= (\alpha^b \bmod q)^a \bmod q$
    $= \alpha^{ba} \bmod q$
    $= \alpha^{ab} \bmod q$
    $= (\alpha^a \bmod q)^b \bmod q$
    $= K_B$

Diffie-Hellman: un esempio

- Alice e Bob concordano di basarsi su un gruppo ciclico finito G di ordine n e sull'elemento generatore g in G (scelgono i primi p = 23 e g = 5).
- Alice sceglie un valore casuale a = 6, calcola $A = 5^6 \bmod 23$ (= 8) e lo invia a Bob insieme a 5 e 23.
- Bob sceglie un valore casuale b = 15, calcola $B = \bmod 23$ (= 19) lo invia ad Alice
- Con successivi calcoli i due ottengono la stessa chiave $K_A = K_B$
  - Alice calcola $K_A = 19^6 \bmod 23 = 2$
  - Bob calcola $K_B = 8^{15} \bmod 23 = 2$
  - $K_B\ (g^a \bmod p)^b \bmod p = K_B$
- I segreti sono 6 (a), 15 (b) e soprattutto, 2 (g^{ab} e g^{ba})
- Eva (la cattiva) senza a e b, e con le sole A e B non può far nulla

RSA

- Basato sull'esponenziazione sugli interi modulo un numero primo - n
  - cifratura e decifratura sono singole esponenziazioni mod (n): l`esponenziazione è facile richiede $O((\log n)^3)$ operazioni.
  - la sicurezza garantita dal costo della fattorizzazione difficile di grandi numeri, richiede $O(e^{\log n \log \log n})$ operazioni.
  - Vengono usati interi molto grandi (tipicamente 1024 bit)

RSA

- Il testo in chiaro è criptato in blocchi, usando blocchi con un valore binario inferiore a n.
- I blocchi di messaggi sono stringhe di 1024 bit. Ogni singolo blocco è un numero decimale con 309 cifre($2^{1024} \cong 10^{309}$)
- Mittente e destinatario devono conoscere il valore di n, e le rispettive chiavi pubbliche
- Il messaggio M (un numero intero) deve essere più piccolo del modulo n (eventualmente viene diviso in blocchi).
- I passi cruciali sono la scelta del modulo e degli esponenti.

RSA cifratura e decifratura

- Chiave Pubblica - PU = {e, n}
- Chiave Privata - PR = {d, n}

- Per cifrare un messaggio M il mittente:
  - Si procura la chiave **pubblica de destinatario key** PU = {e,n}
  - Calcola $C = M^e \bmod n$, con $0 \leq M < n$

- Per decifrare C il destinatario:
  - usa la sua chiave privata PR = {d,n}
  - Calcola: $M = C^d \bmod n$

- La “magia” è nella scelta del modulo e degli esponenti che sono tali che

$$(M^e \bmod n)^d \bmod n = M$$

Generazione di chiavi con RSA

Un utente genera una coppia di chiavi pubbliche/private come segue:

- Sceglie a caso due numeri primi : p, q
- Calcola n = p X q e $\phi(n) = (p-1) X (q-1)$ (ϕ: totiente di Eulero)
- Sceglie a caso la chiave pubblica e tale che
  - $1 < e < \phi(n)$ con e e $\phi(n)$ coprimi($gcd(e, \phi(n))=1$)
- Determina la chiave privata d risolvendo l'equazione
  - $(e \times d) \bmod \phi(n) = 1$ with $0 \leq d \leq n$
- Pubblica la chiave pubblica (PU={e,n}) e tiene segreta la chiave privata (PR={d,n}).

Perché RSA Funziona

- Il teorema di Eulero :
  - $a^{\phi(n)} \bmod n = 1$ if $\gcd(a, n)=1$
- In RSA abbiamo :
  - $n = p \times q$ e $\phi(n)=(p-1)\times(q-1)$
- Le chiavi della coppia (e, d) sono inverse mod $\phi(n)$
  - $e \times d = 1 + (k \times \phi(n))$ per qualche k

Perché RSA Funziona

- Euler's theorem:
  - $a^{\phi(n)} \bmod n = 1$ se $\gcd(a, n)=1$
- Così: (lavorando mod n)

$C^d = M^{(e \times d)}$ poiché $C = M^e$

$= M^{(1+(k \times \phi(n)))}$ poiché (e, d) sono inverse mod $\phi(n)$

$= M^1 \times (M^{\phi(n)})^k$ con semplice calcoli

$= M^1 \times (1)^k$ per il teorema di Eulero

$= M^1 = M$

Un esempio in RSA – Setup delle chiavi

- Seleziona i primi: p = 17 and q = 11
- Calcola $n = p \times q = 17 \times 11 = 187$
- Calcola $\phi(n) = (p-1) \times (q-1) = 16 \times 10 = 160$
- Seleziona e: GCD(e,160)= 1; e = 7
- Determina d < 160 tale che (d x e) mod 160 = 1 - Il valore è d = 23 perché 23 x 7 = 161 = 160 + 1
- Pubblica la chiave pubblica PU = {7,187}
- Mantieni segreta la chiave privata PR={23,187}

Un esempio in RSA - De/Cifratura

- Public key = {7, 187}
- Private key = {23, 187}
- Given M = 88 (88 < 187)
- Cifratura di M:
  - $C = 88^7 \bmod 187 = 11$
- Decifratura di C:
  - $M = 11^{23} \bmod 187 = 88$

Uso di RSA

Cifratura usa l`esponenziazione alla potenza di e

- se e è piccolo, l'esponenziazione è veloce, ma se e è troppo piccolo (tipo e = 3) il sistema è attaccabile
- Per fissare e bisogna accertarsi che MCD(e,ϕ(n)) = 1, rifiutando tutti quei p e q che non sono relativamente primi con e

Decifratura usa l`esponenziazione alla potenza di d

- d deve essere molto grande, se non lo è il sistema è insicuro
- Si usa il teorema cinese del resto per calcolare mod p e mod q separatamente.
- solo il proprietario della chiave privata che conosce p e di q può utilizzare questa tecnica

Teorema cinese del resto: se si conoscono i resti della divisione di n per diversi interi, allora si può determinare in modo univoco il resto della divisione di n per il prodotto di questi numeri, se i divisori sono coprimi a coppie.

Attacchi di forza bruta a RSA

Con chiavi corte, il modulo può essere fattorizzato con attacchi di forza bruta.

- Un modulo a 256 bit può essere calcolato in un paio di minuti.
- Un modulo a 512 bit richiede diverse settimane per l'hardware di consumo moderno.
- La fattorizzazione di chiavi a 1024 bit non è sicuramente possibile in un tempo ragionevole con mezzi ragionevoli, ma può essere possibile per attaccanti ben equipaggiati.
- Il modulo a 2048 bit è sicuro contro attacchi alla fattorizzazione con forza bruta.
- I computer quantistici potrebbero cambiare lo scenario: " *How to factor 2048 bit RSA Integers in 8 Hours using 20 million noisy qubits*" (arxiv.org/abs/1905.09749)

Idee sbagliate

- La crittografia a chiave pubblica è più sicura dalla crittoanalisi che la crittografia simmetrica
  - Non c'è nulla in linea di principio che renda l'uno superiore all'altro dal punto di vista della resistenza alla crittoanalisi
- La crittografia a chiave pubblica ha reso obsoleta la crittografia simmetrica
  - L'overhead computazionale della crittografia a chiave pubblica suggerisce che la crittografia simmetrica non sarà abbandonata
- La distribuzione delle chiavi è banale per la crittografia a chiave pubblica, mentre l`uso di centri di distribuzione delle chiavi per la crittografia simmetrica è pesante
  - Anche per la cifratura a chiave pubblica, sono necessary protocolli che spesso coinvolgono un agente centrale, e le procedure non sono più semplici.

Progressi nella fattorizzazione

Number of Decimal Digits	Approximate Number of Bits	Date Achieved	MIPS-years	Algorithm
100	332	April 1991	7	quadratic sieve
110	365	April 1992	75	quadratic sieve
120	398	June 1993	830	quadratic sieve
129	428	April 1994	5000	quadratic sieve
130	431	April 1996	1000	generalized number field sieve
140	465	February 1999	2000	generalized number field sieve
155	512	August 1999	8000	generalized number field sieve
160	530	April 2003	—	Lattice sieve
174	576	December 2003	—	Lattice sieve
200	663	May 2005	—	Lattice sieve

MIPS-year è il numero di istruzioni eseguite durante un anno di calcolo eseguendo un milione di istruzioni al secondo.

Un anno MIPS corrisponde a circa 31,5 trilioni di istruzioni

Progressi nella fattorizzazione

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	US$5,898[4]	July 9, 1993[6]	T. Denny *et al.*
RSA-129 [**]	129	426	US$100	April 26, 1994[5]	Arjen K. Lenstra *et al.*
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA-150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA-155	155	512	US$9,383[4]	August 22, 1999	Herman te Riele *et al.*
RSA-160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [*]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230 [*]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA-232 [*]	232	768		February 17, 2020[9]	N. L. Zamarashkin, D. A. Zheltkov and S. A. Matveev.
RSA-768 [*]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*
RSA-240 [*]	240	795		Dec 2, 2019[10]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA-250 [*]	250	829		Feb 28, 2020[11]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann

RSA Factoring Challenge.
Da Wikipedia

Approfondimento:
Principi di crittografia a chiave pubblica

- Gli algoritmi a chiave pubblica, usando differenti chiavi per cifratura e decifratura, hanno due importanti caratteristiche:
  - È computazionalmente difficile ricavare la chiave di **decifratura** conoscendo l'algoritmo e la chiave di cifratura (pensate un po'… si, entrano in gioco proprio i numeri primi e la difficoltà nel fattorizzarli!).
  - (opzionale) Qualunque delle due chiavi può essere usata per cifrare e l'altra per decifrare.
- La cifratura a chiave pubblica permette non solo di garantire **confidenzialità** ma anche **autenticazione**, in funzione del tipo di chiave usata. In particolare, si distingue tra la *chiave pubblica* di un soggetto (che viene divulgata al mondo) e la *chiave privata* che viene mantenuta segreta dal soggetto.
- Vediamo ora in dettaglio gli ingredienti di un cifrario a chiave pubblica.

Approfondimento: Cifratura asimmetrica vs. cifratura simmetrica

- Prima di entrare più in dettaglio sugli algoritmi a chiave pubblica, facciamo un rapido confronto tra questa e la cifratura simmetrica.
- **Crittografia simmetrica:** viene usato lo stesso algoritmo per criptare e decriptare con la STESSA chiave. Chiave e algoritmo sono condivisi da mittente e ricevente. La chiave DEVE essere tenuta SEGRETA. Deve essere quasi impossibile decifrare un messaggio se nessun'altra informazione è nota. La conoscenza dell'algoritmo più pezzi di ciphertext non devono essere sufficienti per determinare la chiave.
- **Crittografia asimmetrica:** viene usato un solo algoritmo per cifrare e decifrare ma si usano DUE chiavi: una per criptare e una per decriptare. Il mittente e il ricevente devono avere ognuno una chiave che fa coppia con l'altra (NON la stessa). Una delle due chiavi è SEGRETA, l'altra è PUBBLICA. Deve essere quasi impossibile decifrare un ciphertext se non sono note altre informazioni. La conoscenza dell'algoritmo più pezzi di ciphertext più una delle due chiavi non deve permettere di risalire all'altra chiave.

Approfondimento:
Principi di crittografia a chiave pubblica/2

- Come si può vedere dall'immagine precedente, ci sono sei ingredienti principali nella crittografia a chiave pubblica:
  - Il **plaintext**, il **ciphertext**, **l'algoritmo di cifratura**, **l'algoritmo di decifratura**
  - La chiave **pubblica** e **privata** usate per cifrare e decifrare (una opposta all'altra).

- I passi base sono i seguenti:
  - Ogni utente genera (vedremo poi che vuol dire!) una coppia di chiavi.
  - Ogni utente mette una delle due chiavi su un registro pubblico, a disposizione, potenzialmente, di tutti gli utenti. La seconda (la privata) la conserva e non la divulga.
  - Se Bob vuole mandare un messaggio **confidenziale** a Alice, cifra il messaggio con **la chiave pubblica di Alice**, in modo tale che possa essere decifrato soltanto da chi disponga della corrispondente chiave privata (ovvero solo Alice, visto che quella chiave non è stata divulgata!)
  - Alla ricezione, Alice decifra con la sua chiave privata e ottiene il plaintext.

Approfondimento: Cifratura a chiave pubblica/3

- L'avvento della crittografia a chiave pubblica, dove si usa una chiave per cifrare ed un'altra per e un'altra per decifrare è stata a tutti gli effetti una rivoluzione strutturale per la crittografia di fine anni '70, soprattutto per gli usi (come vedremo) che tale strumento supporta.
- La crittografia a chiave pubblica (asimmetrica) è stata introdotta da *Diffie ed Hellman nel 1978*. Gli algoritmi a chiave pubblica si basano principalmente su funzioni matematiche anziché su permutazioni e sostituzioni come quelli a chiave simmetrica.
- Ci sono 3 luoghi comuni **FALSI** nel rapporto tra crittografia simmetrica e asimmetrica:
  1. La crittografia asimmetrica è più robusta contro la crittoanalisi. FALSO!
  2. La crittografia asimmetrica rende quella simmetrica obsoleta: no, tranquilli durerà ancora per molto!
  3. La distribuzione delle chiavi nella crittografia asimmetrica è più sicura rispetto alle chiavi simmetriche che dovevano venir distribuite con algoritmi ad hoc. Anche questo è falso!

Approfondimento:
Cifratura asimmetrica: confidenzialità

- Entriamo ora più in dettaglio sul comportamento dei protocolli asimmetrici per garantire la confidenzialità del messaggio.

Dato un messaggio X da parte di A da inviare a B e nota ad A la chiave pubblica di B (come si vede in figura B genera due chiavi, conserva la privata PR_b e distribuisce ad A la pubblica PU_b), A usa l'algoritmo di cifratura e PU_b per calcolare il ciphertext Y:

$Y=E(PU_b, X)$

Alla ricezione del messaggio,

- B usa la sua **chiave privata**
- per decifrarlo: **$X=D(PR_b, Y)$**

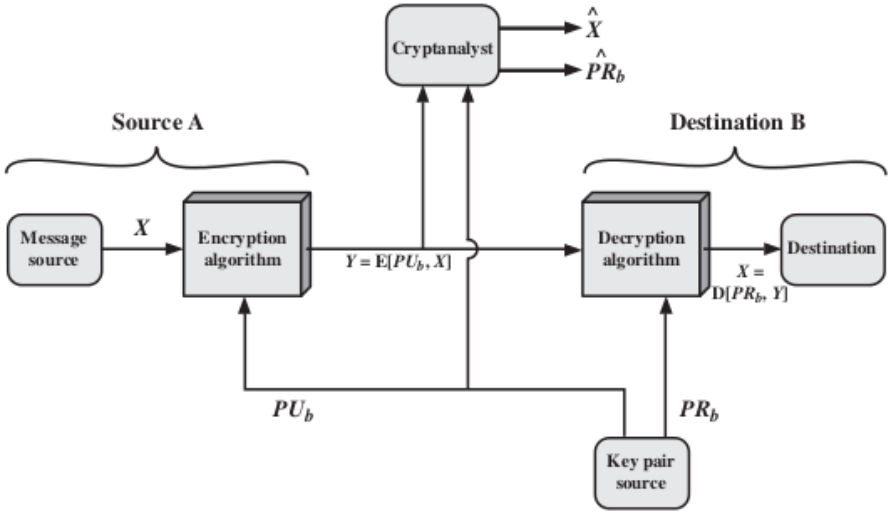


Immagine tratta da *W. Stalling: Cryptography and Network Security, International Edition, Pearson*

Approfondimento: Crittoanalisi per violare confidenzialità

- Nello schema precedente, un attaccante che conosce PU_b e Y, deve cercare di trovare X o PR_b. Si suppone, chiaramente, che l'algoritmo di cifratura e decifratura sia noto all'attaccante.
- Qualora l'attaccante sia interessato allo specifico messaggio, i suoi sforzi si concentreranno solamente su X, cercando di calcolarne una approssimazione $\underline{X}$.
- Tuttavia, molto più spesso l'attaccante è interessato a poter leggere tutta la corrispondenza. In questo caso, l'attaccante cercherà di derivare PR_b
- La difficoltà nel ricavare (almeno una buona approssimazione di) PR_b dipende dalla robustezza dell'algoritmo.
- In particolar modo (come vedremo) dalla robustezza dei «numeri primi» alla base dell'algoritmo.

Approfondimento:
Cifratura asimmetrica: autenticazione

- La cifratura asimmetrica può essere usata per l'autenticazione anziché per la confidenzialità se invertiamo le chiavi usate. Guardiamo lo schema sottostante:

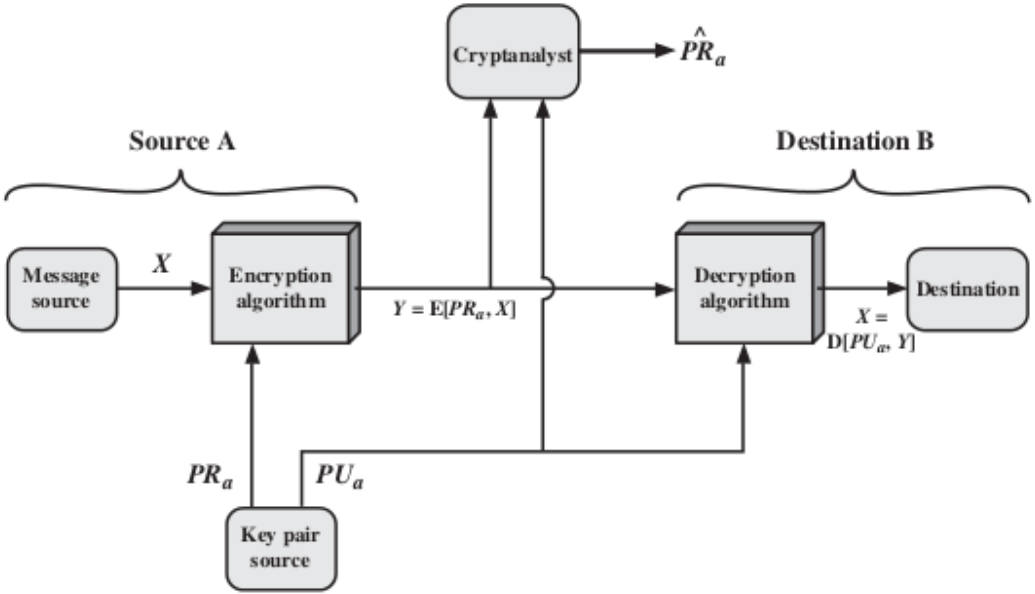


Immagine tratta da *W. Stalling: Cryptography and Network Security, International Edition, Pearson*

Approfondimento:
Cifratura asimmetrica: autenticazione /2

- In questo caso, vogliamo che il ricevente B sia in grado di riconoscere che il messaggio inviato (in chiaro) provenga effettivamente da A. In questo caso, quindi è un problema di *autenticazione* e *integrità*. In questo caso, occorre usare una chiave che «solo il legittimo possessore possa usare», pertanto una chiave privata: quella di colui che deve autenticarsi, ovvero A.
- Pertanto, A usa la sua chiave privata per cifrare il messaggio: **$Y=E(PR_a, X)$**.
- A sua volta B decifrerà il messaggio con la chiave pubblica di A: **$X=(PU_a, Y)$.**
- Se il messaggio è «leggibile» allora è stato sicuramente cifrato con la chiave pubblica di A. Se non è leggibile, o non è stato firmato da A oppure è stato alterato (non è integro).
- Tenete presente che tale cifratura non garantisce confidenzialità perché chiunque può decifrare il messaggio visto che la chiave pubblica di A è, appunto, pubblica.
- Di solito per l'autenticazione è uno spreco *cifrare tutto il messaggio*, visto che non si ottiene confidenzialità. Pertanto, per «firmare» il messaggio si usa (come vedremo) cifrare solo una sottoparte del messaggio (ne parleremo in termini di **firme digitali**).

Approfondimento: Cifratura + Autenticazione

- Possiamo chiaramente garantire sia la confidenzialità che l'autenticazione, applicando due volte l'algoritmo con le chiavi opportune. Lo studente guardi lo schema qui sotto: riconosce le due fasi e l'uso delle rispettive chiavi?

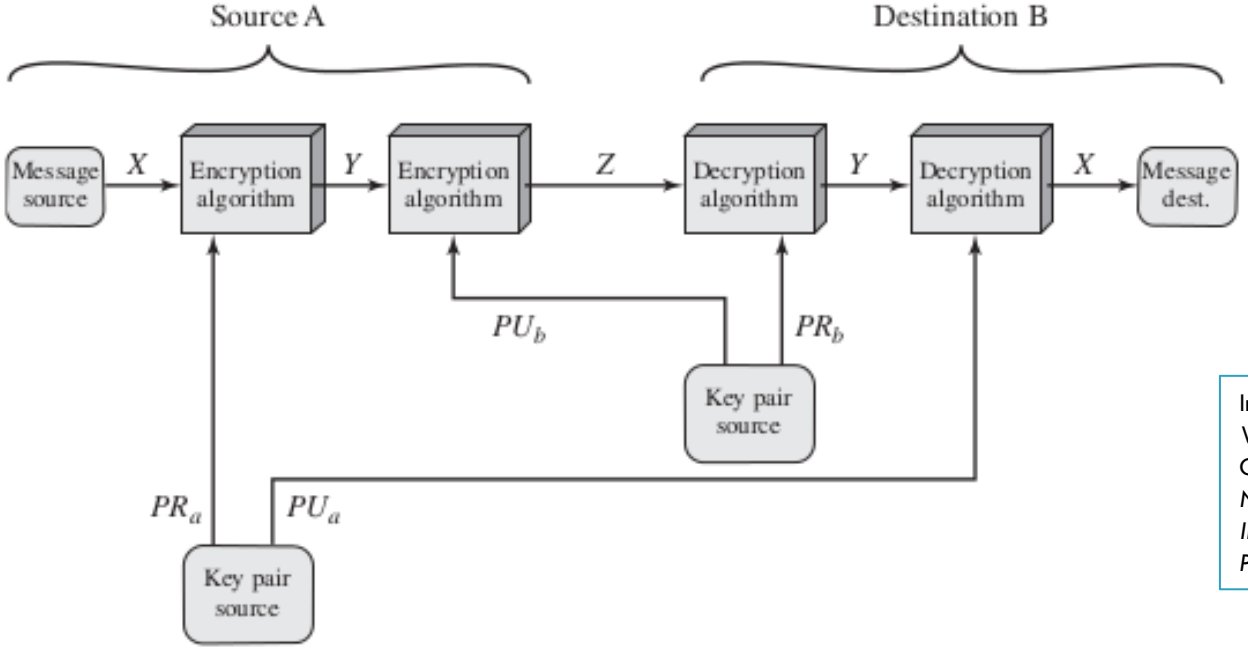


Immagine tratta da *W. Stalling: Cryptography and Network Security, International Edition, Pearson*

Approfondimento: Requisiti per le funzioni crittografiche

- The requirements boil down to the need for a trap-door one-way function. A one-way function is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy whereas the calculation of the inverse is infeasible:
  - $Y = f(X)$ easy
  - $X = f^{-1}(Y)$ infeasible
- Generally, *easy* is defined to mean a problem that can be solved in polynomial time as a function of input length. The term *infeasible* is a much fuzzier concept. In general, we can say a problem
- Now consider a trap-door one-way function, which is easy to calculate in one direction and infeasible to calculate in the other direction unless certain additional information is known. With the additional information the inverse can be calculated in polynomial time. We can summarize as follows: A trap-door one-way function is a family of invertible functions f_k, such that:
  - $Y = f_k(X)$ easy, if k and X are known
  - $X = f_k^{-1}(Y)$ easy, if k and Y are known
  - $X = f_k^{-1}(Y)$ infeasible, if Y known but k not known
- Thus, the development of a practical public-key scheme depends on discovery of a suitable trap-door one-way function.

Approfondimento. Attacchi

- Gli schemi a chiave pubblica non sono più o meno sicuri degli schemi a chiave privata - in entrambi i casi è la dimensione della chiave a determinare la sicurezza. Come nel caso della cifratura simmetrica, uno schema a chiave pubblica è vulnerabile a un attacco di forza bruta. La contromisura è la stessa: utilizzare chiavi di grandi dimensioni. Tuttavia, c'è da considerare tra sicurezza e costo.
- I sistemi a chiave pubblica dipendono dall'uso di una sorta di funzione matematica invertibile. La complessità del calcolo di queste funzioni può non scalare linearmente con il numero di bit della chiave, ma crescere più rapidamente. Pertanto, la dimensione della chiave deve essere abbastanza grande da rendere l'attacco di forza bruta poco pratico, ma abbastanza piccolo per la cifratura e la decifratura pratica.
- In pratica, le dimensioni delle chiavi proposte rendono gli attacchi di forza bruta poco pratici, ma comportano velocità di cifratura/decifratura troppo lente per un uso generico. Invece, come si è detto in precedenza, la cifratura a chiave pubblica è attualmente limitata alle applicazioni di gestione di chiavi e di firma. Un'altra forma di attacco consiste nel trovare un modo per calcolare la chiave privata a partire dalla chiave pubblica. Finora non è stato dimostrato matematicamente che questa forma di attacco sia impossibile per un determinato algoritmo a chiave pubblica.
- Non è possibile confrontare la sicurezza in base alle dimensioni delle chiavi - uno schema di chiavi private a 64 bit ha una sicurezza molto simile a quella di una RSA a 512 bit. Entrambe potrebbero essere rotte se si dispone di risorse sufficienti. Ma con gli schemi a chiave pubblica almeno c'è di solito una base teorica più solida per determinare la sicurezza, poiché si basa su problemi di teoria dei numeri ben noti e ben studiati.

Approfondimento: Teorema di Fermat

L'excursus matematico precedente ci serve per introdurre due teoremi fondamentali per la crittografia a chiave pubblica (o asimmetrica, i termini sono intercambiabili): il Teorema di Fermat e il Teorema di Eulero.

Teorema di Fermat

Dati un intero a e un primo p con a *non divisibile* per p abbiamo $a^{p-1} = 1 \bmod p$

- Esempio: Se a = 7 e p = 19 abbiamo che $7^{18} = 1 \bmod 19$
- Ma come calcolare 7^{18}?
- $7^2 = 49 = 11 \bmod 19$ ---- $7^4 = 11 \times 11 = 121 \bmod 19 = 7 \bmod 19$
- $7^8 = 7 \times 7 \bmod 19 = 49 \bmod 19 = 11 \bmod 19$ ----- $7^{16} = 11 \times 11 = 121 \bmod 19 = 7 \bmod 19$
- $7^{18} = 7^{16} \times 7^2 = 11 \bmod 19 = 7 \bmod 19 \times 11 \bmod 19 = 77 \bmod 19 = 1 \bmod 19$

Versione alternativa del Teorema di Fermat

Dati un intero a e un primo p con a *non divisibile* per p abbiamo $a^p = a \bmod p$

- p = 5 e a = 3 $a^p = 3^5 = 245 = 3 \bmod 5$
- p = 5 e a = 10 $a^p = 10^5 = 10000 = 10 \bmod 5 = 0$

Approfondimento: Teorema di Eulero

- Prima di poter introdurre il teorema, dobbiamo aggiungere una funzione importante: **la funzione totiente FTot(n)**.
- Dato un intero n, la funzione totiente corrispondente fornisce il numero di interi minori di n che sono primi relativamente ad n.
  - Ftot(15) = #{1,2,4,8,11,13,14} = 7
  - Ftot(17) = 16 perché tutti gli interi da 1 a 16 sono primi relativamente a 17.
  - Infatti Ftot(n) = n-1 se n è primo
- Dati due numeri primi p e q, tra loro diversi, abbiamo che se n=pxq, allora Ftot(n) = (p-1) x (q-1)
- **Teorema di Eulero.**
- *Dati due interi a e n relativamente primi tra loro* : $a^{Ftot(n)} = 1 \bmod n$
- a = 3 e n = 10 Ftot(10) = #{1,3,7,9} = 4 → $a^{Ftot(10)} = 3^4 = 81 = 1 \bmod 10$
- a = 2 e n = 11 Ftot(11) = 10 → $a^{Ftot(10)} = 2^{10} = 1024 = 1 \bmod 11$
- **Variante del Teorema di Eulero.**
- *Dati due interi a e n relativamente primi tra loro* : $a^{Ftot(n)+1} = a \bmod n$

Perché RSA Funziona

- Teorema di Eulero:
  - $a^{\phi(n)} \bmod n = 1$ se $GCD(a, n)=1$
- Thus: (working mod n)

$C^d = M^{(e \times d)}$ siccome $C = M^e$

$= M^{(1+(k \times \phi(n)))}$ siccome (e, d) sono inversi mod $\phi(n)$

$= M^1 \times (M^{\phi(n)})^k$ con semplice calcoli

$= M^1 \times (1)^k$ per il teorema di Eulero

$= M^1 = M$