Un algoritmo di crittografia

28 maggio 2012

Inviare messaggi

Situazione:

Alice deve inviare un messaggio a **B**ob.

Inviare messaggi

Situazione:
Alice deve inviare un messaggio a **B**ob.

Problema:
Come fare in modo che nessuno, eccetto **A** e **B**, comprenda il messaggio?

Inviare messaggi

Situazione:
Alice deve inviare un messaggio a **B**ob.

Problema:
Come fare in modo che nessuno, eccetto **A** e **B**, comprenda il messaggio?

Soluzione:
Utilizzare una **chiave** per *codificare* il messaggio.

Inviare messaggi

Situazione:
Alice deve inviare un messaggio a **B**ob.

Problema:
Come fare in modo che nessuno, eccetto **A** e **B**, comprenda il messaggio?

Soluzione:
Utilizzare una **chiave** per *codificare* il messaggio.

Chiave asimmetrica

Quando si usa la stessa chiave per effettuare sia la cifratura che la decifratura si parla di metodo *a chiave simmetrica*.

Chiave asimmetrica

Quando si usa la stessa chiave per effettuare sia la cifratura che la decifratura si parla di metodo *a chiave simmetrica*.

Problema:
Come fare se non si dispone di una chiave condivisa?

Chiave asimmetrica

Quando si usa la stessa chiave per effettuare sia la cifratura che la decifratura si parla di metodo *a chiave simmetrica*.

Problema:
Come fare se non si dispone di una chiave condivisa?

Nascono i sistemi *a chiave asimmetrica*, composti da una **chiave pubblica**, nota a tutti ed usata per la cifratura, e da una **chiave privata**, nota ad una sola persona ed usata per la decifratura.

Chiave asimmetrica

Quando si usa la stessa chiave per effettuare sia la cifratura che la decifratura si parla di metodo *a chiave simmetrica*.

Problema:
Come fare se non si dispone di una chiave condivisa?

Nascono i sistemi *a chiave asimmetrica*, composti da una **chiave pubblica**, nota a tutti ed usata per la cifratura, e da una **chiave privata**, nota ad una sola persona ed usata per la decifratura.

Osservazione:
Di solito si utilizza un sistema a chiave asimmetrica per scambiare una chiave (simmetrica) con la quale viene cifrato il messaggio.

Chiave asimmetrica

Ora, se **A** vuole mandare un messaggio (o una chiave simmetrica) a **B** può procedere nel modo seguente:

Chiave asimmetrica

Ora, se **A** vuole mandare un messaggio (o una chiave simmetrica) a **B** può procedere nel modo seguente:

1. **A** codifica il proprio messaggio utilizzando la chiave pubblica di **B**;

Chiave asimmetrica

Ora, se **A** vuole mandare un messaggio (o una chiave simmetrica) a **B** può procedere nel modo seguente:

1. **A** codifica il proprio messaggio utilizzando la chiave pubblica di **B**;
2. **A** invia il messaggio a **B**: nessuno, conoscendo la chiave pubblica, può decodificarlo;

Chiave asimmetrica

Ora, se **A** vuole mandare un messaggio (o una chiave simmetrica) a **B** può procedere nel modo seguente:

1. **A** codifica il proprio messaggio utilizzando la chiave pubblica di **B**;
2. **A** invia il messaggio a **B**: nessuno, conoscendo la chiave pubblica, può decodificarlo;
3. **B** codifica il messaggio attraverso la propria chiave privata.

Chiave asimmetrica

Ora, se **A** vuole mandare un messaggio (o una chiave simmetrica) a **B** può procedere nel modo seguente:

1. **A** codifica il proprio messaggio utilizzando la chiave pubblica di **B**;
2. **A** invia il messaggio a **B**: nessuno, conoscendo la chiave pubblica, può decodificarlo;
3. **B** codifica il messaggio attraverso la propria chiave privata.

Osservazione:
Un messaggio di testo può essere convertito in un numero (ad esempio assegnando ad ogni lettera il proprio codice ASCII), pertanto il problema si riduce ad effettuare la **codifica e decodifica di un numero**.

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

- È "**semplice**" generare dei numeri primi di una data grandezza.
- È "**semplice**" moltiplicare due numeri e fare elevamenti a potenza.
- È "**molto difficile**" fattorizzare un numero intero.

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

- È "**semplice**" generare dei numeri primi di una data grandezza.
- È "**semplice**" moltiplicare due numeri e fare elevamenti a potenza.
- È "**molto difficile**" fattorizzare un numero intero.

L'algoritmo restituisce una chiave pubblica ed una privata tali per cui:

- con la sola conoscenza della chiave pubblica non si possa decifrare il messaggio;
- il messaggio sia decifrabile con la chiave privata;
- dalla conoscenza della chiave pubblica non si possa risalire alla chiave privata.

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

- È "**semplice**" generare dei numeri primi di una data grandezza.
- È "**semplice**" moltiplicare due numeri e fare elevamenti a potenza.
- È "**molto difficile**" fattorizzare un numero intero.

L'algoritmo restituisce una chiave pubblica ed una privata tali per cui:

- con la sola conoscenza della chiave pubblica non si possa decifrare il messaggio;
- il messaggio sia decifrabile con la chiave privata;
- dalla conoscenza della chiave pubblica non si possa risalire alla chiave privata.

L'algoritmo RSA è costruito in modo che l'ultimo punto sia, **in pratica**, garantito.

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

- È "**semplice**" generare dei numeri primi di una data grandezza.
- È "**semplice**" moltiplicare due numeri e fare elevamenti a potenza.
- È "**molto difficile**" fattorizzare un numero intero.

L'algoritmo restituisce una chiave pubblica ed una privata tali per cui:

- con la sola conoscenza della chiave pubblica non si possa decifrare il messaggio;
- il messaggio sia decifrabile con la chiave privata;
- dalla conoscenza della chiave pubblica non si possa risalire alla chiave privata.

L'algoritmo RSA è costruito in modo che l'ultimo punto sia, **in pratica**, garantito. Ma **non in teoria**.

RSA

Nel 1977 Ronald Rivest, Adi Shamir e Leonard Adleman inventarono un sistema di crittografia a chiave asimmetrica basato sui seguenti fatti:

- È "**semplice**" generare dei numeri primi di una data grandezza.
- È "**semplice**" moltiplicare due numeri e fare elevamenti a potenza.
- È "**molto difficile**" fattorizzare un numero intero.

L'algoritmo restituisce una chiave pubblica ed una privata tali per cui:

- con la sola conoscenza della chiave pubblica non si possa decifrare il messaggio;
- il messaggio sia decifrabile con la chiave privata;
- dalla conoscenza della chiave pubblica non si possa risalire alla chiave privata.

L'algoritmo RSA è costruito in modo che l'ultimo punto sia, **in pratica**, garantito. Ma **non in teoria**. *Forse*.

Due teoremi utili

Teorema (Piccolo Teorema di Fermat)

Sia p un numero primo e sia a un numero intero non divisibile per p. Allora

$$a^{p-1} \equiv 1 \mod p$$

Due teoremi utili

Teorema (Piccolo Teorema di Fermat)

Sia p un numero primo e sia a un numero intero non divisibile per p. Allora

$$a^{p-1} \equiv 1 \mod p$$

Corollario

Sia p un numero primo e sia a un numero intero. Allora

$$a^{p} \equiv a \mod p$$

Due teoremi utili

Teorema (Teorema Cinese del Resto)

Siano $\{n_i\}$ dei numeri naturali a due a due coprimi e siano $\{k_i\}$ numeri interi, allora il seguente sistema di congruenze ha sempre soluzione:

$$\begin{cases} x \equiv k_1 & \mod n_1 \\ \vdots & \\ x \equiv k_t & \mod n_t \end{cases}$$

Inoltre se x ed y sono due soluzioni del sistema predecente, allora $x \equiv y \mod n$ dove $n = n_1 \cdot \ldots \cdot n_t$.

Due teoremi utili

Teorema (Teorema Cinese del Resto)

Siano $\{n_i\}$ dei numeri naturali a due a due coprimi e siano $\{k_i\}$ numeri interi, allora il seguente sistema di congruenze ha sempre soluzione:

$$\begin{cases} x \equiv k_1 & \mod n_1 \\ \vdots & \\ x \equiv k_t & \mod n_t \end{cases}$$

Inoltre se x ed y sono due soluzioni del sistema predecente, allora $x \equiv y \mod n$ dove $n = n_1 \cdot \ldots \cdot n_t$.
In altri termini, si ha un isomorfismo

$$\mathbb{Z}/n\mathbb{Z} \xrightarrow{\cong} \mathbb{Z}/n_1\,\mathbb{Z} \times \ldots \times \mathbb{Z}/n_t\,\mathbb{Z}$$

La funzione totiente di Eulero

Sia n un numero intero positivo. Si definisce una funzione

$$\varphi : \quad n \quad \mapsto \quad \left|\left\{m \in \mathbb{N} : \ (m, n) = 1\right\}\right|$$

ed essa viene detta *funzione totiente* di Eulero.

La funzione totiente di Eulero

Sia n un numero intero positivo. Si definisce una funzione

$$\varphi : \quad n \quad \mapsto \quad \left|\left\{m \in \mathbb{N} : \ (m, n) = 1\right\}\right|$$

ed essa viene detta *funzione totiente* di Eulero.

Osservazione:

$$\varphi(n) \quad = \quad \left|(\mathbb{Z}_n)^*\right|$$

La funzione totiente di Eulero

Sia n un numero intero positivo. Si definisce una funzione

$$\varphi : \quad n \quad \mapsto \quad \left|\left\{m \in \mathbb{N} : \ (m, n) = 1\right\}\right|$$

ed essa viene detta *funzione totiente* di Eulero.

Osservazione:

$$\varphi(n) \quad = \quad \left|(\mathbb{Z}_n)^*\right|$$

Osservazione:

Se p è un numero primo, allora $\varphi(p) = p - 1$.

La funzione totiente di Eulero

Sia n un numero intero positivo. Si definisce una funzione

$$\varphi : \quad n \quad \mapsto \quad \left|\left\{m \in \mathbb{N} : \ (m,n) = 1\right\}\right|$$

ed essa viene detta *funzione totiente* di Eulero.

Osservazione:

$$\varphi(n) \quad = \quad \left|(\mathbb{Z}_n)^*\right|$$

Osservazione:

Se p è un numero primo, allora $\varphi(p) = p - 1$.

Proposizione

φ è moltiplicativa, ovvero se $p, q \in \mathbb{N}$ sono primi tra di loro, allora

$$\varphi(pq) \quad = \quad \varphi(p) \cdot \varphi(q)$$

La funzione totiente di Eulero

Questo risultato deriva dal fatto che l'isomorfismo del Teorema Cinese del Resto ne determina uno tra i gruppi degli elementi invertibili:

$$\left(\mathbb{Z}/n\mathbb{Z}\right)^* \xrightarrow{\cong} \left(\mathbb{Z}/p\,\mathbb{Z}\right)^* \times \left(\mathbb{Z}/q\,\mathbb{Z}\right)^*$$

La funzione totiente di Eulero

Questo risultato deriva dal fatto che l'isomorfismo del Teorema Cinese del Resto ne determina uno tra i gruppi degli elementi invertibili:

$$\left(\mathbb{Z}/n\mathbb{Z}\right)^* \xrightarrow{\cong} \left(\mathbb{Z}/p\,\mathbb{Z}\right)^* \times \left(\mathbb{Z}/q\,\mathbb{Z}\right)^*$$

Osservazione:
Se quindi p e q sono due numeri *primi* ed $n = pq$, allora vale che

$$\begin{aligned} \varphi(n) &= \varphi(p) \cdot \varphi(q) \\ &= (p-1)(q-1) \end{aligned}$$

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

- Scegliere due numeri primi p e q tali per cui $n = pq$ sia maggiore di m e né p né q divida m.

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

- Scegliere due numeri primi p e q tali per cui $n = pq$ sia maggiore di m e né p né q divida m.
- Scegliere un numero intero e che sia coprimo con $\varphi(n)$.

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

- Scegliere due numeri primi p e q tali per cui $n = pq$ sia maggiore di m e né p né q divida m.
- Scegliere un numero intero e che sia coprimo con $\varphi(n)$.
- Determinare, attraverso l'algoritmo generalizzato per il massimo comun divisore, un intero d tale per cui $de \equiv 1 \mod \varphi(n)$.

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

- Scegliere due numeri primi p e q tali per cui $n = pq$ sia maggiore di m e né p né q divida m.
- Scegliere un numero intero e che sia coprimo con $\varphi(n)$.
- Determinare, attraverso l'algoritmo generalizzato per il massimo comun divisore, un intero d tale per cui $de \equiv 1 \mod \varphi(n)$.
- La coppia (n, e) è la chiave pubblica, mentre la coppia (n, d) è la chiave privata.

L'algoritmo

L'algoritmo pensato per cifrare un numero positivo m è il seguente:

- Scegliere due numeri primi p e q tali per cui $n = pq$ sia maggiore di m e né p né q divida m.
- Scegliere un numero intero e che sia coprimo con $\varphi(n)$.
- Determinare, attraverso l'algoritmo generalizzato per il massimo comun divisore, un intero d tale per cui $de \equiv 1 \mod \varphi(n)$.
- La coppia (n, e) è la chiave pubblica, mentre la coppia (n, d) è la chiave privata.

Ora l'applicazione dell'una o l'altra chiave si effettua tramite l'elevamento a potenza modulo n, ovvero $c = m^e \pmod n$ è il messaggio cifrato, e si ha che $m = c^d \pmod n$ effettua la decifrazione.

Funzionamento dell'algoritmo

Lemma

Siano p, q, n ed m come in precedenza, allora vale che

$$m^{\varphi(n)} \equiv 1 \mod n$$

Funzionamento dell'algoritmo

Lemma

Siano p, q, n ed m come in precedenza, allora vale che

$$m^{\varphi(n)} \equiv 1 \mod n$$

Dimostrazione.

Vale che

$$\begin{aligned} m^{\varphi(n)} &= m^{(p-1)(q-1)} \\ &= \left(m^{q-1}\right)^{p-1} \\ &= \left(m^{p-1}\right)^{q-1} \end{aligned}$$

Funzionamento dell'algoritmo

Lemma

Siano p, q, n ed m come in precedenza, allora vale che

$$m^{\varphi(n)} \equiv 1 \mod n$$

Dimostrazione.

Vale che

$$\begin{aligned} m^{\varphi(n)} &= m^{(p-1)(q-1)} \\ &= \left(m^{q-1}\right)^{p-1} \\ &= \left(m^{p-1}\right)^{q-1} \end{aligned}$$

ed allora per il Piccolo Teorema di Fermat:

$$\begin{aligned} m^{\varphi(n)} &\equiv 1 \mod p \\ m^{\varphi(n)} &\equiv 1 \mod q \end{aligned}$$

Funzionamento dell'algoritmo

Allora si hanno i due sistemi:

$$\begin{cases} m^{\varphi(n)} \equiv 1 \mod p \\ m^{\varphi(n)} \equiv 1 \mod q \end{cases} \qquad \begin{cases} 1 \equiv 1 \mod p \\ 1 \equiv 1 \mod q \end{cases}$$

Funzionamento dell'algoritmo

Allora si hanno i due sistemi:

$$\begin{cases} m^{\varphi(n)} \equiv 1 \mod p \\ m^{\varphi(n)} \equiv 1 \mod q \end{cases} \qquad \begin{cases} 1 \equiv 1 \mod p \\ 1 \equiv 1 \mod q \end{cases}$$

Dunque per il Teorema Cinese del Resto:

$$m^{\varphi(n)} \equiv 1 \mod pq = n$$

Funzionamento dell'algoritmo

Allora si hanno i due sistemi:

$$\begin{cases} m^{\varphi(n)} \equiv 1 \mod p \\ m^{\varphi(n)} \equiv 1 \mod q \end{cases} \qquad \begin{cases} 1 \equiv 1 \mod p \\ 1 \equiv 1 \mod q \end{cases}$$

Dunque per il Teorema Cinese del Resto:

$$m^{\varphi(n)} \equiv 1 \mod pq = n$$

□

Corollario

Con le notazioni precedenti

$$m^{de} \equiv 1 \mod n$$

Sicurezza dell'algoritmo

Problema:

Non ci sono argomenti **teorici** che garantiscano la sicurezza di questo algoritmo!

Sicurezza dell'algoritmo

Problema:
Non ci sono argomenti **teorici** che garantiscano la sicurezza di questo algoritmo!

La sicurezza si basa sul fatto che a partire da (n, e) non si riesca a risalire ad (n, d) ovvero, in ultima istanza, sul fatto che fattorizzare un numero intero sia **molto dispendioso** dal punto di vista **computazionale**.

Sicurezza dell'algoritmo

Problema:
Non ci sono argomenti **teorici** che garantiscano la sicurezza di questo algoritmo!

La sicurezza si basa sul fatto che a partire da (n, e) non si riesca a risalire ad (n, d) ovvero, in ultima istanza, sul fatto che fattorizzare un numero intero sia **molto dispendioso** dal punto di vista **computazionale**.

Infatti per ottenere e partendo da n e da d bisogna passare per la conoscenza di $\varphi(n)$, che dunque necessita la determinazione esplicita di p e q, ovvero la fattorizzazione di n.