Algoritmo RSA

Delfrate Riccardo

5B Informatica e telecomunicazione articolazione informatica

A.S 2020/2021

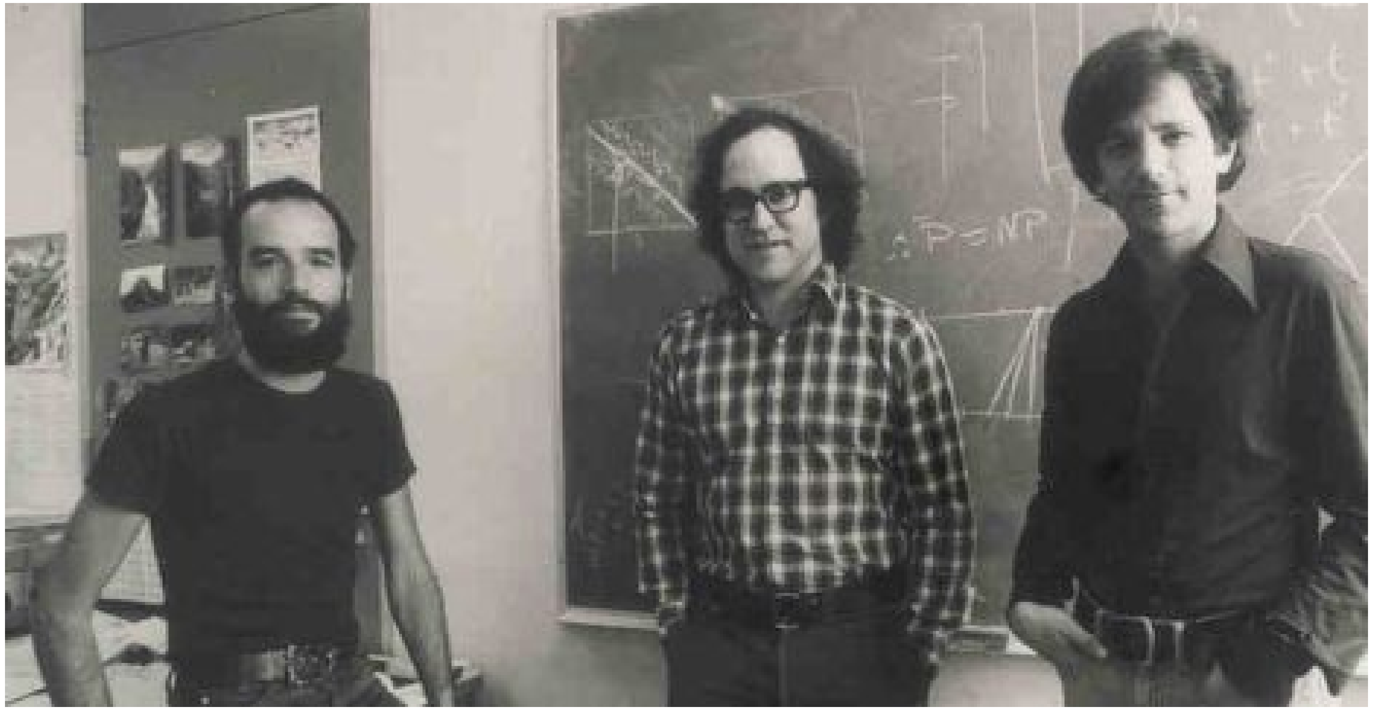
Introduzione

- Creato da Ronald Rivest, Adi Shamir e Leonard Adleman nel 1977.

- Basato sulla fattorizzazione di **numeri primi**.

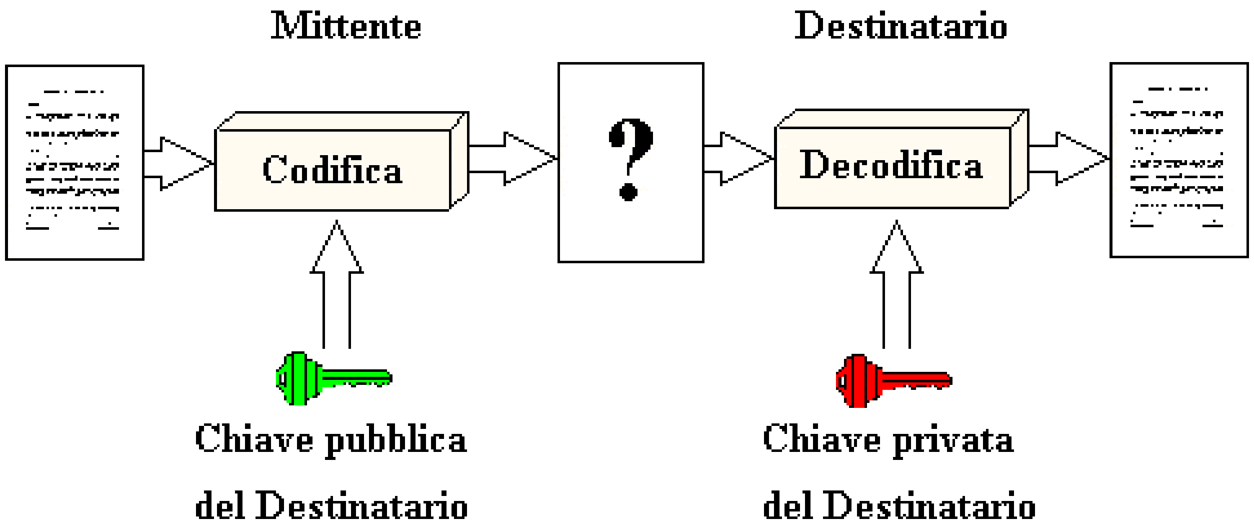
- Algoritmo a **chiave asimmetrica**.

- Largamente diffuso per il **salvataggio di dati sensibili**.



Algoritmo asimmetrico

- Basato sull'esistenza di **due chiavi distinte**, che vengono usate per cifrare e decifrare.
- Se la prima chiave viene usata per la cifratura, la seconda deve necessariamente essere utilizzata per la decodifica e viceversa.
- Le chiavi sono dipendenti ma non è possibile risalire alla prima avendo la seconda e vice versa.



Premessa

L’aritmetica che serve per implementare l’algoritmo RSA è quella della struttura algebrica Zn (Numeri interi) in cui sono definite la somma e il prodotto nel seguente modo:

- $a+b = \mathrm{mod}(a+b, n)$
- $a \cdot b = \mathrm{mod}(a \cdot b, n)$

Operazioni possibili:

Somma Z_9

+	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8	0
2	2	3	4	5	6	7	8	0	1
3	3	4	5	6	7	8	0	1	2
4	4	5	6	7	8	0	1	2	3
5	5	6	7	8	0	1	2	3	4
6	6	7	8	0	1	2	3	4	5
7	7	8	0	1	2	3	4	5	6
8	8	0	1	2	3	4	5	6	7

Moltiplicazione Z_9

x	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	4	6	8	1	3	5	7
3	3	6	0	3	6	0	3	6
4	4	8	3	7	2	6	1	5
5	5	1	6	2	7	3	8	4
6	6	3	0	6	3	0	6	3
7	7	5	3	1	8	6	4	2
8	8	7	6	5	4	3	2	1

Elevamento a potenza Z_{11}

^	1	2	3	4	5	6	7	8	9	10
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	4	1
4	4	5	9	3	1	4	5	9	3	1
5	5	3	4	9	1	5	3	4	9	1
6	6	3	7	9	10	5	8	4	2	1
7	7	5	2	3	10	4	6	9	8	1
8	8	9	6	4	10	3	2	5	7	1
9	9	4	3	5	1	9	4	3	5	1
10	10	1	10	1	10	1	10	1	10	1

Premessa

Teorema di Fermat

Se e solo se n è primo, per ogni a $\in Z_n$ non nullo risulta $a^{n-1} = 1$.

Teorema di Eulero
Per ogni n ≥ 2 e per ogni a $\in Z_n$ invertibile risulta $a\varphi(n)=1$, dove $\varphi(n)$ è il numero di naturali compresi tra 1 e n che sono primi con n.

Funzione di Eulero φ(n)
Associa ad ogni numero naturale n (composto da 2 numeri primi grandi) il numero di numeri a ∈{1, 2, ..., n} tali che MCD(a, n) = 1, **presi 2 numeri p e q primi distinti, allora φ (pq) = (p−1) · (q−1)**.

N=15

φ (3*5) = 8 → (3-1)*(5-1) = 8

Algoritmo di Euclide
L'algoritmo di Euclide è un algoritmo molto efficiente per il calcolo del MCD tra due numeri naturali a e b in tempi molto ragionevoli.

Generazione chiavi

- Si scelgono a caso due numeri primi, p e q abbastanza grandi da garantire la sicurezza dell'algoritmo (hanno più di 300 cifre ciascuno).
- Si calcola **n = p*q**.
- Si calcola **φ(n) = (p-1)(q-1)**.
- **La fattorizzazione di n (ovvero p e q) è segreta**.
- Si sceglie poi un numero **e** (chiamato esponente pubblico), coprimo (non ha divisori in comune) con φ(n) e più piccolo di φ(n) .
- Si calcola il numero **d** (chiamato esponente privato) tale che il suo prodotto con e sia congruo a 1 Mod(φ(n)).

La chiave **pubblica** è **(n,e)**, mentre la chiave **privata** è **(n,d)**.

Criptazione e Decriptazione

Esempio criptazione ‘A’ corrispondente al numero 65 nella tabella ASCII.

Inserimento dati	
p (1° numero primo)	43
q (2° numero primo)	47
m messaggio chiaro ($m < N$)	65
Cifra	

Procedura			
Descrizione	**Variabile**	**Formula**	**Valore**
Alice sceglie due numeri primi $p = 43$ e $q = 47$ e li moltiplica pubblicando il risultato	N *chiave pubblica*	$p \times q = 43 \times 47$	2021
Alice, conoscendo p e q calcola facilmente la funzione di Eulero $\Phi(N)$ che deve restare segreta.	$\Phi(N)$ *chiave privata*	$\Phi(N) = (p-1)(q-1) = \Phi(2021) = 42 \times 46$	1932
Alice calcola la seconda chiave e, primo numero primo con $\Phi(N)$ e la pubblica.	e *chiave pubblica*	primo numero tale che $MCD(e, \Phi(N)) = 1$	5
Alice calcola la chiave segreta d, che è l´inverso di e modulo $\Phi(N)$ e NON la pubblica.	d *chiave privata*	d tale che $e \times d \equiv 1 \pmod{\Phi(N)}$	773
Bruno per inviarle un messaggio m usando le chiavi pubbliche di Alice, N ed e, calcola la potenza $c = m^e \pmod{N}$ ed invia c per un canale pubblico.	c *cifrato*	$c \equiv m^e \pmod{N} = 65^5 \pmod{2021} =$	168
Alice e solo Alice che conosce la chiave segreta d può ora decifrare il messaggio m semplicemente calcolando la potenza inversa.	m *decifrato*	$m \equiv c^d \pmod{N} = 168^{773} \pmod{2021} =$	65

Il testo criptato C=168 riconvertito corrisponde al carattere ‘Ç’.

Implementazione

```php
<?php
include 'phpseclib/Crypt/RSA.php';

$numero = $_POST['numero'];
$nome = $_POST['nome'];
$mese = $_POST['mese'];
$anno = $_POST['anno'];
$cvc = $_POST['cvc'];


$rsa = new Crypt_RSA();
extract($rsa->createKey(2048));
```

```php
//----- DECRIPTAZIONE -----
$rsa->loadKey($privatekey);
$cript64Dec = base64_decode($ciphertext64);
$decriptato = $rsa->decrypt($cript64Dec);
//----- FINE DECRIPTAZIONE -----
```

```php
//----- CRIPTAZIONE -----
$plaintext = "$numero-$nome-$mese-$anno-$cvc";

$rsa->loadKey($publickey);
$ciphertext = $rsa->encrypt($plaintext);
$ciphertext64 = base64_encode($ciphertext);

$update3 = "UPDATE utente SET cartaCredito = '$ciphertext64' WHERE idUtente = $idUtente";
if (!mysqli_query($conn, $update3)) {
    ?>
    <script>
        Swal.fire({
            icon: 'error',
            title: 'Ops...',
            text: 'Cè stato un problema in fase di registrazione',
            allowEscape: false,
            allowOutsideClick: false,
            confirmButtonText: "<a href='../login/login.php'>Riprova</a>"
        });
    </script>
<?php
}
//----- FINE CRIPTAZIONE -----
```

Violazione algoritmo

Esempio elenco telefonico:

Trovare il numero di telefono dal nome di un utente è immediato mentre trovare il nome dato il numero di telefono richiederebbe un tempo x, non infinito ma da considerarsi tale con grandi numeri (Fattorizzazione di N).

- Per violare l’algoritmo RSA bisogna scoprire la fattorizzazione di N.
- Attacco **Brute Force**, ovvero vado a tentativi per indovinare la combinazione.
- Algoritmo di **Shor** (quantistico).