

CRITTOGRAFIA

Risoluzione prove

d'esame

Prof. Bernasconi

INDICE

- Pag.6 KOLMOGOROV → Sequenze casuali
- Pag.8 MILLER-RABIN → Numeri primi (es.1)
- Pag.10 MILLER-RABIN → Numeri primi (es.2)
- Pag.12 CIFRARIO AFFINE → Cifrari storici
- Pag.14 ALBERTI → Cifrari storici
- Pag.15 VIGENÈRE → Cifrari storici
- Pag.17 ALBERTI, VIGENÈRE → Cifrari storici
- Pag.19 CIFRARIO A GRIGLIA → Cifrari storici (es.1)
- Pag.21 CIFRARIO A GRIGLIA → Cifrari storici (es.2)
- Pag.22 CRITTOANALISI STATISTICA → Cifrari storici
- Pag.23 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.1)
- Pag.24 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.2)
- Pag.25 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.3)
- Pag.26 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.4)
- Pag.27 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.5)
- Pag.28 CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti (es.6)
- Pag.29 DES, ATTACCHI, MEET IN THE MIDDLE → Cifrari simmetrici
- Pag.31 AES → Cifrari simmetrici (es.1)
- Pag.33 AES → Cifrari simmetrici (es.2)
- Pag.34 AES → Cifrari simmetrici (es.3)
- Pag.35 DES, AES → Cifrari simmetrici
- Pag.36 CIFRARI A COMPOSIZIONE DI BLOCCHI → Cifratura a blocchi
- Pag.37 CIFRARI A COMPOSIZIONE DI BLOCCHI → Cifrari a composizione
- Pag.38 TRE CIFRARI A CHIAVE PUBBLICA → Cifrari
- Pag.40 EULERO, EUCLIDE ESTESO → Complessità in algebra
- Pag.41 EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia (es.1)
- Pag.42 EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia (es.2)
- Pag.44 EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia (es.3)
- Pag.45 RSA, RSA ATTACCHI → RSA (es.1)
- Pag.47 RSA, RSA ATTACCHI → RSA (es.2)
- Pag.49 RSA, RSA ATTACCHI → Numeri primi

INDICE

- Pag.50 RSA, RSA ATTACCHI → Complessità in algebra
- Pag.51 RSA, RSA ATTACCHI → RSA (es.1)
- Pag.52 RSA, RSA ATTACCHI → RSA: attacchi
- Pag.53 RSA, RSA ATTACCHI → RSA (es.2)
- Pag.57 RSA, RSA ATTACCHI → Chiave pubblica
- Pag.60 RSA, RSA ATTACCHI → RSA (es.2)
- Pag.61 DIFFIE-HELLMAN → Scambio di chiavi (es.1)
- Pag.63 DIFFIE-HELLMAN → Scambio di chiavi (es.2)
- Pag.64 DIFFIE-HELLMAN → Scambio di chiavi (es.3)
- Pag.65 CURVE ELLITTICHE → Curve ellittiche
- Pag.68 CURVE ELLITTICHE → Crittografia ellittica (es.1)
- Pag.69 CURVE ELLITTICHE → Crittografia ellittica (es.2)
- Pag.70 CURVE ELLITTICHE → Scambio di chiavi su curve ellittiche
- Pag.72 KOBLEZ, SCAMBIO MESSAGGI, ELGAMAL → Crittografia ellittica (es.1)
- Pag.74 KOBLEZ, SCAMBIO MESSAGGI, ELGAMAL → Curve ellittiche
- Pag.76 KOBLEZ, SCAMBIO MESSAGGI, ELGAMAL → Crittografia ellittica (es.2)
- Pag.78 HASH, FIRMA DIGITALE → Funzioni hash e autenticazione di messaggi
- Pag.79 HASH, FIRMA DIGITALE → RSA, Complessità in algebra (es.1)
- Pag.80 HASH, FIRMA DIGITALE → Funzioni hash
- Pag.81 HASH, FIRMA DIGITALE → RSA, Complessità in algebra (es.2)
- Pag.82 HASH, FIRMA DIGITALE → Funzioni hash e autenticazione di messaggi
- Pag.83 HASH, FIRMA DIGITALE → Firma digitale
- Pag.85 HASH, FIRMA DIGITALE → Certificato digitale
- Pag.86 ZERO KNOWLEDGE → Identificazione (es.1)
- Pag.88 ZERO KNOWLEDGE → Identificazione (es.2)
- Pag.89 SSL → Protocolli
- Pag.93 BB84 → Scambio di chiavi (es.1)
- Pag.96 BB84 → Scambio di chiavi (es.2)
- Pag.99 BB84 → Scambio di chiavi (es.3)
- Pag.103 BITCOIN → Funzioni hash e firma digitale
- Pag.104 BITCOIN → Moneta elettronica

KOLMOGOROV → Sequenze casuali

Dare la definizione di sequenza casuale secondo Kolmogorov, **illustrandone** il significato, e dimostrare l'esistenza di sequenze casuali secondo Kolmogorov di ogni lunghezza n .

- UNA SEQUENZA h È DEFINITA CASUALE SECONDO KOLMOGOROV SE VALE CHE:

$$\rightarrow K(h) \geq |h| - \lceil \log_2 |h| \rceil, \text{ DANE ABBIANO CHE:}$$

- $K(h)$ È LA COMPLESSITÀ DI KOLMOGOROV SU h , cioè LA LUNGHEZZA DEL PROGRAMMA PIÙ CORTO CHE GENERA h NEL SISTEMA UNIVERSALE S_U $\rightarrow K(h) = K_{S_U}(h)$

- S_U RAPPRESENTA LA SIMULAZIONE DI DETERMINATI SISTEMI DI CALCOLO S_i
// CHE IN QUESTO CASO SONO USATI PER h

- QUINDI, LA CASUALITÀ È UNA PROPRIETÀ DELLA SEQUENZA. NON DIPENDE DA ALTRO COME, AD ESEMPIO, POTREBBE ESSERE LA SORGENTE CHE GENERA h

- PER OGNI VALORE DI m SI DEMONSTRA L'ESISTENZA DI SEQUENZE CASUALI LUNGHE (m) .

- INIZIAMO COL FISSARE UN CERTO (m)

- INDICHiamo CON $S = 2^m$ IL NUMERO DI SEQUENZE (BINARIE) DI LUNGHEZZA (m)
// TUTTE LE SEQUENZE LUNGHE (m)

- INDICHiamo CON T IL NUMERO DI SEQUENZE DI LUNGHEZZA (m) NON CASUALI

\rightarrow SI SOGLIA DEMONSTRARE CHE $T < S$

// PER QUEL VALORE DI (m) SE VALE CHE $T < S$, ALMENO UNA SEQUENZA CASUALE ESISTE.

INVECE, SE NON SI VERIFICA LA DISUGUAGLIANZA (ES. $T = S$) \rightarrow LE SEQUENZE SONO TUTTE NON CASUALI

- INDICHiamo CON N IL NUMERO DI SEQUENZE BINARIE DI LUNGHEZZA $< m - \lceil \log_2 m \rceil$.

// I PROGRAMMI GENERERANNO SEQUENZE (h) :

- CASUALI SE $\rightarrow |h| \geq m - \lceil \log_2 m \rceil$

- NON CASUALI SE $\rightarrow |h| < m - \lceil \log_2 m \rceil$ (COME IN QUESTO CASO)

QUESTE (N) SEQUENZE SONO DATE DALLA SOMMATORIA $\rightarrow \sum_{i=0}^{m - \lceil \log_2 m \rceil - 1} 2^i = 2^{m - \lceil \log_2 m \rceil} - 1$

$$\text{// ABBIANO } N = 2^{m - \lceil \log_2 m \rceil} - 1 < m - \lceil \log_2 m \rceil < 2^m = S$$

\rightarrow QUINDI, ABBIANO OTTENUTO $\rightarrow N < S$ *

- TRA LE (N) SEQUENZE CI SONO I PROGRAMMI PER GENERARE LE (T) SEQUENZE NON CASUALI DI LUNGHEZZA (m) . QUINDI, VALE $\rightarrow T \leq N$ *



- DALLE DUE DISUGUAGLIANZE $\textcircled{*}$ E $\textcircled{*} \textcircled{*}$ OTTENIAMO $\rightarrow T \leq N < S$, DA CUI SI HA LA TESI:

$$\rightarrow T < S$$

// RICARTELLAZIONE → ABBIAMO DIMOSTRATO CHE:

- ESISTE UN $N < S$
- N POSSIEDE QUALCHE T , ANCHE SE COME INSIEME È MAGGIORE VISTO CHE POSSIEDE TUTTE LE POSSIBILI SEQUENZE FINO A $m \cdot \lceil \log_2 m \rceil$ $\rightarrow T \leq N$
- DALLE DUE PRECEDENTI $\rightarrow T \leq N < S$
- INFINE, LA TESI $\rightarrow T < S$

MILLER-RABIN → Numeri primi

Descrivere il test di primalità di Miller-Rabin e discuterne la complessità.

Si ricordi che dati un numero N e un intero arbitrario y , $2 \leq y \leq N-1$, se N è un numero primo, devono essere veri i due predicati:

P1: $\text{mcd}(N, y) = 1$

P2: $(y^z \bmod N = 1)$ or (esiste un valore i , $0 \leq i \leq w-1$, tale che $y^{2^i z} \bmod N = -1$)

dove z e w sono definiti da $N-1 = z \cdot 2^w$ con z dispari.

- PRIMA DI RAPPRESENTARE IL TEST DI MILLER-RABIN, SCRIVIAMO LA SEGUENTE FUNZIONE:

VERIFICA (N, y) {

if (P1 == FALSE OR P2 == FALSE)

return 1;

else return 0;

}

FUNZIONE CHE PRENDE IN INPUT \textcircled{N} PER LA PRIMALITÀ E \textcircled{Y} COME CERTIFICATO, E CONTROLLA LA VALIDITÀ DI TALE CERTIFICATO.

IL CONTROLLO VIENE FATTO SUI DUE PREDICATI P1 E P2, VERIFICANDO SE:

- UNO O ENTRAMBI I PREDICATI SONO FALSI \rightarrow CERTIFICATO VALIDO, QUINDI, \textcircled{N} È COMPOSTO E VIENE RESTITUITO 1
- ENTRAMBI I PREDICATI SONO VERI \rightarrow CERTIFICATO NON VALIDO, QUINDI, \textcircled{N} È PROBABILMENTE PRIMO CON UNA PERCENTUALE BASSISSIMA DI ERRORE

// P1 == TRUE E P2 == TRUE \rightarrow CONDIZIONE NECESSARIA PER LA PRIMALITÀ

- ADESSO CONSIDERIAMO L'ALGORITMO DI MILLER-RABIN :

TEST MR (N, k) {

for ($i=1; i \leq k; i++$) {

SCELGI A CASO $y \in [2, N-1]$;

if (VERIFICA (N, y) == 1) RETURN 0;

}

RETURN 1;

}

\textcircled{N} e \textcircled{K} sono i valori in input, dove \textcircled{K} è scelto dall'utente.

Dopo la scelta casuale di \textcircled{Y} , si controlla se è valida la verifica:

- Verifica valida ("if VERIFICATO == 1") → si restituisce \textcircled{O} , poiché il numero è composto
- Verifica non valida ("if NONVERIFICATO") → si itera il ciclo scegliendo un'altra \textcircled{Y} , e poi si ricontrolla la verifica.

Se ci troviamo nel caso di dover restituire \textcircled{A} , vuol dire che non esiste una \textcircled{Y} che rende falsi i due predicati

Ciò implica che la condizione necessaria di primalità è rispettata, e si restituisce \textcircled{A} .

Poiché \textcircled{N} è probabilmente primo con probabilità di errore $< \left(\frac{1}{4}\right)^k$

- Quanto a complessità abbiamo che:

- il ciclo si ripete al massimo \textcircled{K} volte
 - Dentro il ciclo ci sono operazioni di complessità polinomiale $p(m)$ nella dimensione \textcircled{m} dei dati
- Quindi, la complessità dell'algoritmo è polinomiale di ordine $\rightarrow O(K \cdot p(m))$

MILLER-RABIN → Numeri primi

Spiegare come si può utilizzare il test di Miller-Rabin per generare numeri primi grandi in crittografia e spiegare perché tale metodo è considerato efficiente.

Si ricordi che dati un numero N e un intero arbitrario y , $2 \leq y \leq N-1$, se N è un numero primo, devono essere veri i due predicati:

P1: $\text{mcd}(N, y) = 1$

P2: $(y^z \bmod N = 1)$ or (esiste un valore i , $0 \leq i \leq w-1$, tale che $y^{2^i z} \bmod N = -1$)

dove z e w sono definiti da $N-1 = z \cdot 2^w$ con z dispari.

- DEFINIAMO IL SEGUENTE ALGORITMO PER LA GENERAZIONE DI UN NUMERO PRIMO BINARIO DI ALMENO m bit, CON m VALORE FISSATO DALL'UTENTE: // m GRANDE

PRIMO (m, k) {

S = SEQUENZA DI $m-2$ bit PRODOTTI DA UN GENERATORE BINARIO PSEUDO-CASUALE // 1

$N = (1 S 1)_2 ;$ // 2

while (TEST_MR(N, k) == 0) { // 3

$N = N + 2;$

}

RETURN $N;$

}

1) Si levano 2 bit, perché tali sequenza dovrà contenere successivamente

① bit iniziale pari a 1 ED ② bit finale pari ad 1 // VEDI PUNTO SEGUENTE

2) CONCATENO ① CON ① INIZIALE E ① FINALE, ED IL RISULTATO SI INSERISCE DENTRO N , QUINDI, N SARÀ DI m bit

3) UTILIZZO TEST DI MILLER-RABIN.

FINCHÉ IL TEST FORNISCE 0, CIOÈ N COMPOSTO, ALLORA SI INCREMENTA N DI 2

4) SIAMO USCITI DAL WHILE, PERCHÉ ABBIANO TROVATO UN NUMERO PRIMO (TEST_MR(N, k) == 1). QUINDI, SI RESTITUISCE N .

- IN QUANTO AD EFFICIENZA ABBIANO CHE :

- L'ALGORITMO DI MILLER-RABIN HA COSTO POLINOMIALE IN (M)
 - OGNI ITERAZIONE CON ANNESSO INCREMENTO RICHIESTE TEMPO POLINOMIALE
- COMPRESSIVAMENTE L'INTERO ALGORITMO AVRA' TEMPO POLINOMIALE

CIFRARIO AFFINE → Cifrari storici

Sia c la cifra decimale di valore maggiore del proprio numero di matricola, e sia $k = 25 + c$.

Si consideri un cifrario affine in cui si lavora modulo k , e si determini il numero di chiavi possibili.

Si scelga infine una chiave e si cifri il proprio cognome.

- INIZIAMO COL CONSIDERARE COME CIFRARIO AFFINE → CIFRARIO A SOSTITUZIONE MONOALFABETICA, CHE PERMETTE DI MAPPARE OGNI LETTERA DI UN DETERMINATO MESSAGGIO (m) IN UN'ALTRA POSIZIONE DELL'ALFABETO. // "MAPPARE" EQUIVCOME A CIFRARE
- LA FUNZIONE DI QUESTO CIFRARIO È → $\text{Pos}(y) = (\alpha \cdot \text{Pos}(x) + b) \bmod k$, DOVE ABBIANO:
 - $\text{Pos}(x)$ → POSIZIONE DELLA LETTERA PRESA IN QUESTIOTNE DEL MESSAGGIO DA MASCHERARE
 - $\text{Pos}(y)$ → NUOVA POSIZIONE CHE DORRÀ ASSUMERE LA LETTERA PRESA IN QUESTIOTNE DEL MESSAGGIO DA MASCHERARE
 - α → COMPONENTE DELLA CHIAVE $\text{Key} = \langle \alpha, b \rangle$ CHE DEVE ESSERE CO-PRIMO CON k
// k NON È LA CHIAVE MA IL MOLTIPLICATORE DEL MODULO.
 - ② CO-PRIMO CON k IMPLICA CHE → $\text{MCD}(\alpha, k) = 1$A CAUSA DELLA CO-PRIMALITÀ ② POTRÀ ASSUMERE UN CERTO NUMERO DI VALORI, A PARTIRE DA ①.TALE NUMERO LO INDICHERETTO CON → α'
 - b → COMPONENTE DELLA CHIAVE $\text{Key} = \langle \alpha, b \rangle$, E POTRÀ ASSUMERE UN CERTO NUMERO DI VALORI IN $[0, k-1]$. TALE NUMERO LO INDICHERETTO CON → b'
- IL NUMERO DI CHIANI POSSIBILI È DATO DA → # Key Possibili = $\alpha' \cdot b'$
- PER RISOLVERE L'ESERCIZIO PROCEDIAMO COSÌ:
 - 1) PRENDO IL VALORE MAGGIORDE DEL MIO NUMERO DI MATRICOLA, CHE È 7 → $C = 7$
 - 2) PONGO k COME DATOSTO DELL'ESERCIZIO → $k = 25 + 7 = 32$
 - 3) CERCO UN CIFRARIO AFFINE CHE LAVORI $\bmod 32$, PER CI:
 - ① DEVE ESSERE CO-PRIMO CON 32 → $\text{MCD}(\alpha, 32) = 1$.
AD ESEMPIO, SCEGLIAMO $\alpha = 3$.
ESSENDO $32 = 2^5$, cioè UN MULTIPLO DI ② → ② NON DORRÀ ASSUMERE NUMERI pari, quindi i VALORI POSSIBILI DA ASSUMERE DA ① A 32 SARANNO 16 → $\alpha' = 16$
 - ② ASSUMERÀ TUTTI I VALORI IN $[0, 31]$ → $b' = 32$.
AD ESEMPIO, SCEGLIAMO $b' = 1$

4) IL NUMERO DI CHIAVI POSSIBILI SARÀ DATO DA $\rightarrow \text{Key Possibili} = 16 \cdot 32 = 512$

OSS IN REALTÀ SONO 511, PERCHÉ LA COPPIA DI CHIAVE Key<1,0> LASCIA INALTERATO IL MESSAGGIO SE SI APPLICA LA FUNZIONE DI CIFRATURA.
L'ALFABETO CHE VIENE USATO È QUELLO INGLESE DA 26 LETTERE

5) CIFRO IL MIO COGNOME "COLI" SCELGENDO COME CHIAVE $\rightarrow \text{Key } \langle 3,1 \rangle$. // a=3, b=1

CONSIDERANDO CHE (A) È MAPPATA SU ⑥, CIOÈ $\rightarrow \text{Pos}('A') = 0$, SI PROSCDE ALLA CIFRATURA

LETTERA PER LETTERA:

• $\text{Pos}(y) = (3 \cdot \text{Pos}('c') + 1) \bmod 32$ // $\text{Pos}('c') = 2$

$\rightarrow 7 \bmod 32 = 7$ // QUOTIENTE ⑥, RESTO 7

$\rightarrow \text{Pos}(7) = H$ // NUOVA LETTERA OTTENUTA

• $\text{Pos}(y) = (3 \cdot \text{Pos}('o') + 1) \bmod 32$ // $\text{Pos}('o') = 14$

$\rightarrow 43 \bmod 32 = 11$ // QUOTIENTE ①, RESTO 11

$\rightarrow \text{Pos}(11) = L$ // NUOVA LETTERA OTTENUTA

• $\text{Pos}(y) = (3 \cdot \text{Pos}('l') + 1) \bmod 32$ // $\text{Pos}('l') = 11$

$\rightarrow 34 \bmod 32 = 2$ // QUOTIENTE ①, RESTO 2

$\rightarrow \text{Pos}(2) = C$ // NUOVA LETTERA OTTENUTA

• $\text{Pos}(y) = (3 \cdot \text{Pos}('i') + 1) \bmod 32$ // $\text{Pos}('i') = 8$

$\rightarrow 25 \bmod 32 = 25$ // QUOTIENTE ⑥, RESTO 25

$\rightarrow \text{Pos}(25) = Z$ // NUOVA LETTERA OTTENUTA

6) DAL PUNTO PRECEDENTE OTTIENIAMO CHE IL MIO COGNOME "COLI", CIFRATO CON Key <3,1>, D'ISULTA:

$\rightarrow \underline{\text{HLCZ}}$

Facendo riferimento al seguente allineamento iniziale di un disco cfrante di Alberti

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z	1	2	3	4	5
S	D	T	K	B	J	O	R	H	Z	C	U	N	Y	E	P	X	V	F	W	A	G	I	Q	L	M

cifrare il proprio nome e cognome (senza spazi) con il metodo dell'*indice mobile* e con due **cambi di chiave**.

- Si considera il nome e il cognome come messaggio da cifrare:

→ m: LUCA PAOLI

- Considerata la griglia, dobbiamo distinguere le righe di lettere:

- Quella sopra → rappresenta le lettere del messaggio di partenza
- Quella sotto → rappresenta le lettere del crittogramma, che vanno associate alle lettere del messaggio di partenza, cioè la chiave // allineamento 'A-S'

- L'utilizzo dell'indice mobile prevede l'inserimento di una coppia di valori all'interno del messaggio → (N, L), in cui:

- N (numero) → in fase di cifratura, indica che dopo (N) caratteri bisogna cambiare chiave. Inoltre, tale numero va cifrato
- L (lettera) → valore irrilevante per il crittogramma finale, ma si cifra ugualmente; rappresenta la lettera della seconda riga da allineare alla lettera iniziale della prima riga.
oss (ovviamente, si allineano anche le lettere successive, oltre ad L).

- Nel caso di questo esercizio, i cambi di chiave sono (2), quindi, le coppie (N, L) da inserire saranno (2) // all'interno di (m)

- Date considerazioni precedenti scriviamo il messaggio → m: LUCA DPAZOLBI

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z	1	2	3	4	5
D	T	K	B	J	O	R	H	Z	C	U	N	Y	E	P	X	V	F	W	A	G	I	Q	L	M	S

→ 1° Cambio di chiave, in cui ho allineato la (D) alla (A)

A	B	C	D	E	F	G	H	I	L	M	N	O	P	Q	R	S	T	U	V	Z	1	2	3	4	5
B	J	O	R	H	Z	C	U	N	Y	E	P	X	V	F	W	A	G	I	Q	L	M	S	D	T	K

→ 2° Cambio di chiave, in cui ho allineato la (B) alla (A)

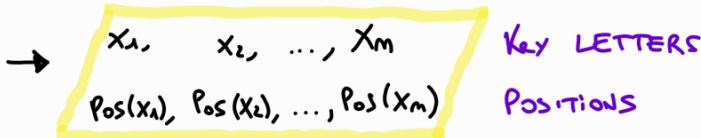
→ OTTENIAMO IL CRIPTOGRAFFMA FINALE → C: ZFTGSB ED QYCSN

LUCA DPAZOLBI
 ↑ ↑
 inizio 1° Cambio di chiave inizio 2° Cambio di chiave

VIGENÈRE → Cifrari storici

Utilizzando la cifratura di Vigenère, cifrare la frase "appello straordinario di crittografia" utilizzando come chiave il proprio cognome.

- DATO UN MESSAGGIO m SI CONSIDERA UNA CHIAVE K COMPOSTA DA $x_1 \dots x_m$ LETTERE
- OGNI SUA LETTERA ANDÀ UNA POSIZIONE $\rightarrow \text{Pos}(x_i)$ (SI CONSIDERA L'ALFABETO CHE HA $\rightarrow \text{Pos}('A') = 0$)



- PER OGNI LETTERA DEL MESSAGGIO, SI SCRIVONO SOTTO AD ESSE, LE LETTERE DELLA CHIAVE KEY.

GUARDO LE LETTERE DELLA CHIAVE FINISCONO PRIMA CHE FINISCA IL MESSAGGIO \rightarrow SI RIFERISCE DA CASO KEY

- OGNI LETTERA DI m , SARÀ TRASLATA DI UN CERTO NUMERO DI POSIZIONI, RAPPRESENTATO DAL VALORE DI $\text{Pos}(x_i)$, SOTTO ALLA LETTERA PRESA IN CONSIDERAZIONE DEL MESSAGGIO m

OSS (GLI SPAZI VUOTI NON SI CONSIDERANO)

- SUPPOSTO IL MESSAGGIO $m = m_1 m_2 m_3 m_4 m_5 m_6$ E LA CHIAVE $\text{Key} = x_1 x_2 x_3$, SI APPLICA IL CIFRARIO

→ $m_1 m_2 m_3 \quad m_4 m_5 m_6$

$$\begin{array}{ccc|ccc} & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \\ \rightarrow & x_1 & x_2 & x_3 & x_1 & x_2 & x_3 \end{array}$$

\rightarrow E SI OTTERRÀ IL NUOVO MESSAGGIO CIFRATO CON VIGENÈRE $\rightarrow m' = m'_1 m'_2 m'_3 \quad m'_4 m'_5 m'_6$ FACENDO:

→ $m'_1 \quad m'_2 \quad m'_3 \quad m'_4 \quad m'_5 \quad m'_6$

$$\begin{array}{cccccc} m'_1 & m'_2 & m'_3 & m'_4 & m'_5 & m'_6 \\ " & " & " & " & " & " \\ \rightarrow & \text{Pos}(m_1) & \text{Pos}(m_2) & \text{Pos}(m_3) & \text{Pos}(m_4) & \text{Pos}(m_5) & \text{Pos}(m_6) \\ + & + & + & + & + & + \\ \text{Pos}(x_1) & \text{Pos}(x_2) & \text{Pos}(x_3) & \text{Pos}(x_1) & \text{Pos}(x_2) & \text{Pos}(x_3) \end{array}$$

→ NUOVO MESSAGGIO CIFRATO

→ FASE DI TRASLATORIE

- PER RISOLVERE L'ESERCIZIO PROCEDIAMO COSÌ:

1) INIZIO CONSIDERANDO COME CHIAVE IL MIO COGNOME $\rightarrow \text{Key} = \text{"COLI"}$

2) CIASCUNA LETTERA DELLA CHIAVE ANDÀ UNA SUA POSIZIONE $\text{Pos}(x_i)$:

→ $\begin{array}{c} \text{C O L I} \\ \hline 2 \ 1 \ 4 \ 1 \ 1 \ 7 \end{array}$

3) SI CONSIDERA IL MESSAGGIO $m = \text{"APPELLO STRAORDINARIO DI CRITTOGRAFIA"}$

4) SI ASSOCIA LA CHIAVE (CON ANNESSA POSIZIONE) AL MESSAGGIO:

→ $\begin{array}{c} \text{A P P E L L O \ S T R A O R D I N A R I O \ D I \ C R I T T O G R A F I A} \\ \hline \text{C O L I C O L I \ i C O L I C O L I C O L I \ C O \ L I C O L I \ C O L I C O} \end{array}$

2 1 4 1 1 7 2 1 4 1 1 7 2 1 4 1 1 7 2 1 4 1 1 7 2 1 4 1 1 7 2 1 4



5) MAPPANDO IL MESSAGGIO, ATTRaverso la chiave, si ottiene il messaggio cifrato:

→ C D A L N Z Z Z V F L V T R T U C F T V F W N Y K H E V I F L M K O

2	3	4	14	13	23	25	25	24	5	11	21	19	13	13	20	25	13	21	5	22	13	24	10	7	4	21	8	5	11	12	10	14		
0	15	15	4	11	11	14	18	13	17	0	14	17	3	8	13	0	17	8	19	3	8	2	17	8	13	13	14	6	17	0	5	8	0	
+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2	14	11	7	2	14	11	7	2	14	11	7	2	14	11	2	2	19	11	7	2	14	11	7	2	14	11	7	2	14	11	7	2	14	

OSS (QUANDO NELLA SOMMA SI VA oltre LA POSIZIONE DELLA LETTERA 'Z' → Si riparte dalla
POSIZIONE DELLA LETTERA 'A')

ALBERTI, VIGENÈRE → Cifrari storici

Confrontare il cifrario polialfabetico di Alberti con quello di De Vigenère dal punto di vista della sicurezza e della praticità.

- IN QUANTO A PRATICITÀ, INIZIAMO COL CONSIDERARE IL CIFRARIO DI ALBERTI, LA CUI CHIAVE:

- È CONCORDATA DA MITTENTE E DESTINATARIO
- È RAPPRESENTATA DALL'ALLINEAMENTO DI DUE DISCHI (ROTANTI).

A B C D E F G H I L M N O P Q R S T U V Z 1 2 3 4 5
E Q H C W L M V P D N X A O G Y I B Z R J T S K U F

IN FASE DI CIFRATURA (E DECIFRAZIONE), OGNI VOLTA CHE SI INCONTRA UN CARATTERE SPECIALE ALL'INTERNO DEL MESSAGGIO SPEDITO (O RICEVUTO) → SI CAMBIA LA CHIAVE, QUINDI, L'ALLINEAMENTO DEI DUE DISCHI

- IL CIFRARIO DI VIGENÈRE, INVECE, È PIÙ PRATICO PERCHE' HA LE SEGUENTI CARATTERISTICHE:

- È BASATO SU UNA TABELLA T PUBBLICA, DI DIMENSIONI 26x26
- OGNI RIGA i DELLA TABELLA, CONTIENE L'ALFABETO DI 26 LETTERE ROTATO VERSO SINISTRA DI (i-1) POSIZIONI

A	B	C	...	X	Y	Z
B	C	D	...	Y	Z	A
C	D	E	...	Z	A	B
...
X	Y	Z	...	U	V	W
Y	Z	A	...	V	W	X
Z	A	B	...	W	X	Y

- LA CHIAVE È UNA PAROLA SEGRETA K.

PUÒ ESSERE PIÙ CORTA DEL MESSAGGIO m, E IN QUESTO CASO SI RIPETE FINO ALLA FINE DI m

- PER LA CIFRATURA DISPONIAMO m E K SU DUE RIGHE ADIACENTI, COSÌ OGNI LETTERA x DEL MESSAGGIO È ALLINEATA CON UNA LETTERA y DELLA CHIAVE.
Poi CIASCUA x SI CIFRA CON LA LETTERA CHE SI TROVA NELLA CELLA DELLA MATRICE T ALL'INCROCIO TRA LA RIGA CHE INIZIA CON x E LA COLONNA CHE INIZIA CON y
- // DISCORSO ANALOGO PER LA DECIFRAZIONE

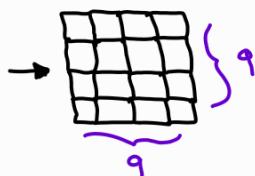
I L D E L F I N O → m
A B R A A B R A A → K (RIPETUTA)
↓ ↓ ↓ ↓ ↓ ↓ ↓
I M U E L G Z N O → CRIPTOGRAMMA

- IN QUANTO A SICUREZZA, IL CIFRARIO DI ALBERTI PUÒ ESSERE DIFFICILE DA ATTACCARE SE SI INSERISCONO SPESSO I CARATTERI SPECIALI NEL MESSAGGIO CHE DEVE ESSERE INVIATO → CI PRODURREBBE TANTI CAMBI DI CHIAVE
- IL CONTINUO CAMBIO DI CHIAVE RENDE INUTILI GLI ATTACCHI BASATI SULLA FREQUENZA DI CARATTERI;
- INVECE, PER IL CIFRARIO DI VIGENÈRE LA SICUREZZA DIPENDE DALLA LUNGHEZZA DELLA CHIAVE.
SE LA CHIAVE È LUNGA m, LE APPARIZIONI DI UNA STESSA LETTERA DISTANTI UN MULTIPLO DI m NEL MESSAGGIO IN CHIAVE m → SARANNO TRASFORMATE NELLA STESSA LETTERA CIFRATA
POICHÉ SI SOVRAPPONGONO ALLA STESSA LETTERA DELLA CHIAVE

CIFRARIO A GRIGLIA → Cifrari storici

In riferimento ai cfrari a griglia, spiegare in cosa consiste un tale cifrario.

- INNANZITUTTO, CONSIDERANO UNA GRIGLIA $q \times q$, CON q VALORE pari:



- LA CHIAVE PER IL CIFRARIO A GRIGLIA → È LA GRIGLIA STESSA CON UN DETERMINATO MECCANISMO DI FUNZIONAMENTO PER LA CIFRATURA:

- $\frac{1}{4}$ DEL TOTALE DELLE CELLE DELLA GRIGLIA DEVONO ESSERE TRASPARENTE, E INDICHEREMO CON

$$\rightarrow S = \frac{q^2}{4}.$$

- $\frac{3}{4}$ DEL TOTALE DELLE CELLE DELLA GRIGLIA DEVONO ESSERE OPACHE



- LA GRIGLIA VIENE POSTA SOPRA UNA PAGINA P , DI $q \times q$ DI dimensioni.

- SI SCRIVONO IN SEQUENZA I PRIMI S CARATTERI DEL MESSAGGIO IN CORRISPONDENZA ALLE CELLE TRASPARENTE DELLA GRIGLIA → E QUESTI CARATTERI FINIRANNO SULLA PAGINA P .

LA SEQUENZA DI SCRITTURA VA DA SINISTRA A DESTRA, E DALL'ALTO VERSO IL BASSO

- L'OPERAZIONE PRECEDENTE SI RIPETE ALTRE 3 VOLTE RUOTANDO, PER OGNI VOLTA, LA GRIGLIA IN IN SENSO ORARIO DI 90°

OSS / LA GRIGLIA È SCELTA IN MODO TAU CHE LE CELLE TRASPARENTE, IN FASE DI ROTAZIONE, NON SI SOVRAPPONGANO ALLE POSIZIONI CHE ESSE AVRANNO IN PRECEDENZA.

SE, IN FASE DI SCRITTURA DENTRO LE CELLE TRASPARENTE, SI FINISCE DI SCRIVERE IL MESSAGGIO PRIMA CHE LE CELLE TRASPARENTE SIANO STATE USATE TUTTE → SI COMPLETANO CON IL CARATTERE *

→ ALLA FINE, LA PAGINA P CONTERÀ IL MESSAGGIO CIFRATO CON TUTTE LE LETTERE DEL MESSAGGIO, DISPOSTE IN MANIERA CASUA agli occhi di tutti

- LA DECIFRAZIONE AVVIENE IN MANIERA "SIMILE":

- SI PRENDE LA GRIGLIA E SI DISPOSTE COME ERA NELLA POSIZIONE INIZIALE DURANTE LA FASE DI CIFRATURA

- SI SOVRAPPONE LA GRIGLIA SUA PAGINA P

- SI LEGGONO I CARATTERI SOTTO LE CELLE TRASPARENTE, ANDANDO DA SINISTRA VERSO DESTRA E DALL'ALTO VERSO IL BASSO

- LE ULTIME $\textcircled{2}$ OPERAZIONI SI RIPETONO PER ALTRE $\textcircled{3}$ VOLTE, ROTANDO DI (90°) LA GRIGLIA
IN SENSO ORARIO; CI SI FERMA QUANDO:
- ABBIANO LETTO TUTTE LE CELLE TRASPARENTE: DOPO ANEL FATTO TUTTE E $\textcircled{3}$ LE ROTAZIONI.
//QUINDI, IL MESSAGGIO RIEMPIE TUTTA LA GRIGLIA
 - ABBIANO TROVATO IL PRIMO $\textcircled{*}$ //CHE INDICA CHE IL MESSAGGIO È TERMINATO

ESEMPIO

- $m = L'ASSASSINO E' ARCHIMEDES TARRINGTON$

- $q = 6$

- $s = 3$

	L	A		
	S	S		
			A	
		S		
S			I	
			N	

rotazione 0

	O			E
			A	
	R		C H	
I	M	E		

rotazione 1

D	L	G	A	T	O
E	O	S	S	S	E
N	*	T	*	A	A
A	*	R	S	C	H
*	S	R	R	*	I
I	M	I	E	N	N

crittogramma

D				
E			S	
		T		
A				
	R	R		
	I		N	

rotazione 2

		G	T	O
N	*		*	
	*			
*			*	

rotazione 3

CIFRARIO A GRIGLIA → Cifrari storici

In riferimento ai cifrari a griglia, dimostrare quante chiavi diverse si possono costruire per una griglia $q \times q$.

- PER UN CIFRARIO A GRIGLIA ($q \times q$) SI POSSONO OTTENERE UN TOTALE DI $\rightarrow G = 4^S$ CHIAVI,
E CIÒ È CHIAVI SEGRETE DIVERSE.

IL VALORE S RAPPRESENTA $\frac{1}{4}$ DELLE CELLE DELLA GRIGLIA, QUELLE CHE DEVONO ESSERE
TRASPARENTE, E SONO $\rightarrow S = \frac{q^2}{4}$

// IL TOTALE DI CHIAVI SI PUÒ SCRIVERE IN MANIERA ALTERNATIVA $\rightarrow G = 4^{\frac{q^2}{4}}$

ESEMPIO

- PER $q=6$ ABBIAMO LA DIMENSIONE DELLA GRIGLIA DATA DA:

$$\rightarrow q \times q = 36$$

- IL NUMERO DI CELLE TRASPARENTE SARÀ $\rightarrow S = \frac{36}{4} = 9$

- DAI VALORI PRECEDENTI OTTERNAMO CHE IL NUMERO DI CHIAVI (GRIGLIE) POSSIBILI È $\rightarrow G = 4^9 = 262144$

CRITTOANALISI STATISTICA → Cifrari storici

Spiegare come utilizzare la crittoanalisi statistica per attaccare il cifrario di De Vigenère.

— INNANZITUTTO, FACCIA MO UNA CONSIDERAZIONE SUL CIFRARIO DI DE VIGENÈRE:

- OGNI LETTERA y DEL CRITTOGRAMMA DIPENDE DA UNA COPPIA DI LETTERE (x, k) PROVENIENTI DAL MESSAGGIO E DALLA CHIAVE → QUESTO, SERVE AD ALTERARE LA FREQUENZA DELLE LETTERE NEL CRITTOGRAMMA

// PER NON CONSENTIRE UNA DECIFRAZIONE DIRETTA

- LA CHIAVE È UNICA ED È RIPETUTA PIÙ VOLTE
- SU QUEST'ULTIMO PUNTO VIENE ATTACCATO IL CIFRARIO //DEBOLEZZA
- SE LA CHIAVE CONTIENE h CARATTERI, IL CRITTOGRAMMA PUÒ ESSERE DECOMPOSTO IN (h) SOTTOSEQUENZE.
- CIASCUNA DI QUESTE È OTTENUTA PER SOSTITUZIONE MONOALFABETICA
- LA CRITTOANALISI SI BASA SULLO SCOPRIRE IL VALORE DI (L) PER SCOMPORRE IL CRITTOGRAMMA E CONTINUARE LA DECIFRAZIONE CON IL METODO MONOALFABETICO
- OSSERVIAMO CHE IL MESSAGGIO CONTIENE QUASI SICURAMENTE GRUPPI DI LETTERE ADIACENTI RIPETUTI PIÙ VOLTE → SONO I TRIGRAMMI PIÙ FREQUENTI NELLA LINGUA.
- QUINDI, APPARIZIONI DELLA STESSA SOTTOSEQUENZA ALLINEATE CON LA STESSA PORZIONE DELLA CHIAVE SONO TRASFORMATE NEL CRITTOGRAMMA IN SOTTOSEQUENZE IDENTICHE
- DALLE CONSIDERAZIONI PRECEDENTI:
- SI CERCANO NEL CRITTOGRAMMA COPPIE DI POSIZIONI p₁ E p₂ IN CUI INIZIANO LE SOTTOSEQUENZE IDENTICHE
 - LA DISTANZA d = p₂ - p₁ È PROBABILMENTE UGUALE ALLA LUNGHEZZA (h) DELLA CHIAVE, O AD UN SUO MULTIPLO
- LA CRITTOANALISI STATISTICA SI APPLICA, QUINDI, VALUTANDO I DUE PUNTI PRECEDENTI

CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti

Definire i cifrari perfetti, e spiegare a parole il significato di tale definizione.

- UN CIFRARIO È **PERFETTO** SE LA SICUREZZA È GARANTITA QUALUNQUE SIA L'INFORMAZIONE PRESA SUL CANALE DI COMUNICAZIONE. IN TERMINI "MATEMATICI":

→ UN CIFRARIO È **PERFETTO** $\forall m \in \text{MSG}, \forall c \in \text{CRITTO}$ se $P(M=m | C=c) = P(M=m)$

→ LA PROBABILITÀ CHE IL MESSAGGIO INVIAUTO SIA m POSTO CHE SUL CANALE TRANSITA IL CRITTOGRAMMA

c È UGUALE ALLA PROBABILITÀ CHE IL MESSAGGIO INVIAUTO SIA m

// LA CONOSCENZA DI UN CRITTOANALISTA NON AUMENTA DOPO CHE EGLI HA OSSERVATO UN CRITTOGRAMMA SUL CANALE → m E c RISULTANO AL CRITTOANALISTA SCARRELLI FRA LORO

CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti

Dimostrare il Teorema di Shannon, cioè che in un cifrario perfetto il numero delle chiavi deve essere maggiore o uguale al numero di messaggi possibili.

- PER LA DIMOSTRAZIONE INIZIAMO INDICANDO CON N_m IL NUMERO DI MESSAGGI POSSIBILI, CIÒE':

$$\rightarrow m \in \text{MSG} : P(M=m) > 0$$

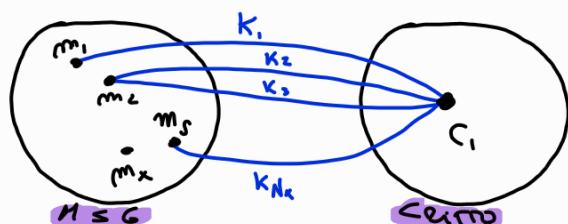
// I MESSAGGI POSSIBILI SONO QUEI MESSAGGI CHE HANNO PROBABILITÀ NON NULLA DI ESSERE INTIPI.

- INDICHIANO CON N_k IL NUMERO DELLE CHIAVI

- DIMOSTRIAMO PER ASSURDO, SUPPONENDO CHE $N_k < N_m$: // CHE VA CONTRO ALLA TESI

- PER UN CRIPTOGRAFMA C , CON $P(C=c) > 0$, SI CONTANO QUANTI SONO I MESSAGGI CHE POTREBBERO CORRISPONDERE AD ESSO → INDICHEREMO CON S TALE CONTEGGIO

- CONSIDERANDO IL SEGUENTE SCHEMA D'ESEMPIO:



// m_x È IL MESSAGGIO IN PIÙ CHE FA SÌ CHE SI SUPERI IL NUMERO DELLE CHIAVI

→ È POSSIBILE CHE, DECODIFICANDO CON CHIAVI DIVERSE, OTTENGA LO STESSO MESSAGGIO IN CHIAVI, QUINDI, ABBIAVANO CHE IL NUMERO DEI MESSAGGI OTTENUTI (S) AL MASSIMO SARÀ COME IL NUMERO DELLE CHIAVI (N_k) → $S \leq N_k$

- Poiché $N_k < N_m$, per l'assurdo, dal punto precedente OTTENIAMO → $S < N_m$ // $S \leq N_k < N_m$

// IL NUMERO DI MESSAGGI CORRISPONDENTI A (C) È MINORE DEL NUMERO DI MESSAGGI POSSIBILI

- QUINDI, ESISTERÀ ALMENO UN MESSAGGIO (m) CON $P(M=m > 0)$ NON OTTENIBILE DA (C)

// DECODIFICANDO IL CRIPTOGRAFMA CON TUTTE LE CHIAVI POSSIBILI RIMANE QUAUCHE MESSAGGIO ESCURO, IN QUESTO CASO m_x

- IL PUNTO PRECEDENTE IMPLICA CHE → $P(M=m | C=c) = 0$ (SU m_x), E CIÒ DIOSTRA CHE IL CIFRARIO NON È PERFETTO PER $N_k < N_m$

// AVENDO DIMOSTRATO CHE PER $N_k < N_m$ IL CIFRARIO NON È PERFETTO, ANORA VERRÀ CHE PER $N_k > N_m$ IL CIFRARIO È PERFETTO (TESI DI SHANNON)

CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti

Definire il cifrario One-Time Pad e le assunzioni standard su di esso.

- IL CIFRARIO ONE-TIME PAD È UN CIFRARIO PERFETTO, CHE UTILIZZA COME CHIAVE UNA SEQUENZA DI bit IN UN BLOCCO (PAD).

TALE SEQUENZA VIENE CONSUMATA VIA VIA NELLA CIFRATURA, E NON VIENE PIÙ RIUTILIZZATA → SI USA UNA SOLO VOLTA (ONE-TIME).

- PER QUESTO CIFRARIO GLI SPAZI MSG, CRITTO E KEY SONO SPAZI DI SEQUENZE DI m bit ($m > 0$):

• PER MSG $\rightarrow m \in \{0,1\}^m$

• PER KEY $\rightarrow K \in \{0,1\}^m$

• PER CRITTO \rightarrow LA SEQUENZA DI m bit È OTTENUTA CIFRANDO m CON LA CHIAVE K

- LA CIFRATURA AVVIENE FACENDO LO XOR bit A bit TRA $m \in \mathbb{K}$ $\rightarrow c = C(m, K) = m \oplus K$

- LA DECIFRAZIONE AVVIENE IN MODO SIMILE $\rightarrow m = D(c, K) = c \oplus K$.

SCRIVIAMO LA DECIFRAZIONE IN MANIERA PIÙ ESTESA TRAMITE I SUCCESSIVI PASSAGGI:

1. $(m \oplus K) \oplus K$, AL POSTO DI c SI METTE $m \oplus K$, PER COME È NEGLIA FUNZIONE DI CIFRATURA

2. $m \oplus (K \oplus K)$, UTILIZZO DELLA PROPRIETÀ ASSOCIAZIONE

3. $m \oplus 0 = m$, PERCHÉ $K \oplus K = 0$

OSS LA TABELLA DI VERITÀ DELLO XOR bit A bit È LA SEGUENTE

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti

Nel cifrario *One-Time Pad* si sostituisca l'operatore XOR con NAND (AND negato).

Spiegare se il protocollo funziona con le stesse proprietà del cifrario originale.

- INIZIAMO COL CONSIDERARE LE TABELLE DI VERITÀ DELLO XOR (⊕) E DEL NAND (↑)

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	$A \uparrow B$
0	0	1
0	1	1
1	0	1
1	1	0

- SIMULIAMO, ATTRAVERSO UN ESEMPIO, L'UTILIZZO DEL ONE-TIME PAD SIA CON LO XOR CHE CON IL NAND

ESEMPPIO

- SI CONSIDERANO $M=5$, $M=10110$ E $K=00011$ // 5 È IL NUMERO DI bit

XOR

• CIFRATURA $\rightarrow C = C(M, K) = M \oplus K$

$$10110 \oplus$$

$$\begin{array}{r} \rightarrow 00011 = \\ \hline 10101 \end{array}$$

$$\rightarrow C = 10101$$

• DECIFRAZIONE $\rightarrow M = D(C, K) = C \oplus K$

$$10101 \oplus$$

$$\begin{array}{r} \rightarrow 00011 = \\ \hline 10110 \end{array}$$

$$\rightarrow M = 10110$$

// MESSAGGIO UGUALE A QUELLO INIZIALE

NAND

• CIFRATURA $\rightarrow C = C(M, K) = M \uparrow K$

$$10110 \uparrow$$

$$\begin{array}{r} \rightarrow 00011 = \\ \hline 11101 \end{array}$$

$$\rightarrow C = 11101$$

• DECIFRAZIONE $\rightarrow M = D(C, K) = C \uparrow K$

$$11101 \uparrow$$

$$\begin{array}{r} \rightarrow 00011 = \\ \hline 11110 \end{array}$$

$$\rightarrow M = 11110$$

// MESSAGGIO DIVERSO DA QUELLO INIZIALE

- METTENDO NAND AL POSTO DELLO XOR → IL PROTOCOLLO NON FUNZIONA ALLA STESSA MANIERA.

- INOLTRE, SCRIVENDO LA DECIFRAZIONE IN MANIERA PIÙ ESTESA TRAMITE I SUCCESSIVI PASSAGGI:

1. $(M \uparrow K) \uparrow K$, AL POSTO DI ② SI METTE $M \uparrow K$, COME NELLA CIFRATORA

2. $M \uparrow (K \uparrow K)$, USO DI ASSOCIAZIONE E CI FERMiamo QUA, SENZA OTTENERE SUBITO (M) PERCHÉ $K \uparrow K \neq 0$

$$\begin{array}{l} // 00011 \uparrow 00011 = 11100 \end{array}$$

Dimostrare che il cifrario *One-Time Pad* è un cifrario perfetto.

- PRIMA DELLA DEMOSTRAZIONE, INIZIANO COL DIRE CHE LA TESI È VERIFICATA SOTTO LE SEGUENTI IPOTESI:

1) TUTTI I MESSAGGI HANNO LUNGHEZZA (m) .

SE COSÌ NON FOSSE VERIFICHIANO SE, DATO IL MESSAGGIO (m) , VALE CHE:

- $m < m \rightarrow$ AGGIUNTA DI bif CASUALI A FINE MESSAGGIO

- $m > m \rightarrow$ DIVISIONE DEL MESSAGGIO IN BLOCCHI LUNGHI (m) , ED INIZIO DEL MESSAGGIO A BLOCCHI

2) TUTTE LE SEQUENZE DI (m) bif SONO MESSAGGI POSSIBILI.

QUALUNQUE SEQUENZA HA PROBABILITÀ, NON NULLA, DI ESSERE INVIAITA; POSSONO ESSERE INViate pure le frasi

PRIVE DI SIGNIFICATO, ANCHE SE ANBANNO UNA PROBABILITÀ PIÙ BASSA RISPETTO AGLI ALTRI MESSAGGI

//PROBABILITÀ PIÙ BASSA DI INIZIO DI MESSAGGI = "INVIARE OGNI TANTO"

3) PER OGNI MESSAGGIO BISOGNA USARE UNA CHIAVE SCELTA PERFETTAMENTE A CASO

- PER LA DEMOSTRAZIONE DOBBIANO PROVARE CHE IL CIFRARIO È PERFETTO, E CIÒ È:

$\rightarrow \forall m \in MSG, \forall c \in CRITTO \rightarrow P(M=m | C=c) = P(M=m)$, E SI PROCEDA NELLA SEGUENTE MANIERA:

- Si considera il termine a SINISTRA dell'identità precedente e si applica la DEFINIZIONE DI:

$$\text{PROBABILITÀ CONDIZIONALE} \rightarrow P(M=m | C=c) = \frac{P(M=m, C=c)}{P(C=c)} \quad \text{(*)}$$

OSS $\left(\begin{array}{l} P(M=m, C=c) \text{ INDICA CHE IL MITTENTE HA GENERATO IL MESSAGGIO } (m) \text{ E L'HA CIFRATO COME} \\ \text{CRIPTOGRAFIA } (c) \end{array} \right)$

- PER LA DEFINIZIONE DI XOR, FISSATO IL MESSAGGIO, ABBIAMO CHE:

- CHIAVI DIVERSE DANNO ORIGINE A CRIPTOGRAFII DIVERSI.

- Ogni CHIAVE PUÒ ESSERE GENERATA CON PROBABILITÀ $\rightarrow \left(\frac{1}{2}\right)^m$

//TUTTI I CRIPTOGRAFII HANNO LA STESSA PROBABILITÀ DI APPARIRE

- FISSATO (m) , PER OGNI (c) AVEMMO CHE $\rightarrow P(C=c) = \left(\frac{1}{2}\right)^m$, CIÒ È LA PROBABILITÀ DEI EVENTO $\{C=c\}$ È COSTANTE E INDEPENDENTE DA (m) .

QUINDI, GLI ELEMENTI $\{M=m\}$ E $\{C=c\}$ SONO INDEPENDENTI TRA LORO E ANBEMO CHE:

$$\rightarrow P(M=m, C=c) = P(M=m) \times P(C=c) \quad \text{(*)} \quad \text{(*)}$$

- SOSTITUENDO (*) (*) IN (*) AVEMMO $\rightarrow P(M=m | C=c) = \frac{P(M=m) \times P(C=c)}{P(C=c)}$

- INFINE, OTTEREMO $\rightarrow P(M=m | C=c) = P(M=m)$, CHE È LA DEFINIZIONE DI CIFRARIO PERFETTO

CIFRARI PERFETTI, SHANNON, ONE-TIME PAD → Cifrari perfetti

Spiegare se si può usare un attacco esauriente sulle chiavi per attaccare il cifrario One-Time Pad.

— PER IL CIFRARIO ONE-TIME PAD UN ATTACCO DI TIPO BRAVEFORCE NON HA NUOVO SENSO IN QUANTO:

- ALL'IMPROVVISABILITÀ COMPUTAZIONALE → TEMP. ESPONENZIALE
- AL FATTO CHE OGNI CHIAVI DECIFRA UN CRIPTOGRAFIA \textcircled{C} IN UN MESSAGGIO POSSIBILE.

DECIFRANDO CON TUTTE LE CHIAVI RITRANO $\{0,1\}^m$, cioè lo spazio dei messaggi → QUINDI, TUTTI SONO MESSAGGI POSSIBILI ED ERANO GIÀ NOTI AL CRİTTANALISTA PRIMA CHE EGLI VEDESSE \textcircled{C}

//NESSUNA ULTERIORE ACQUISIZIONE DI CONOSCENZA DA PARTE DEL CRİTTANALISTA

DES, ATTACCHI, MEET IN THE MIDDLE → Cifrari simmetrici

Siano C_1 e C_2 due diversi cifrari simmetrici, e siano k_1 e k_2 le rispettive chiavi, di pari lunghezza l .

Si considera la doppia cifratura di un messaggio m

$$c = C_2(C_1(m, k_1), k_2).$$

- Spiegare come si esegue la decifrazione del crittogramma c
- Descrivere un attacco che riduca la sicurezza della doppia cifratura, e discuterne il costo computazionale.

- COME CIFRARIO SIMMETRICO, FACCIAMO RIFERIMENTO AL CIFRARIO DES PER C_1 , E AL CIFRARIO AES PER C_2

- SIANO D_1 E D_2 DUE FUNZIONI DI DECIFRAZIONE, LA DECIFRAZIONE DEL CRITTOGRAMMA (C) AVVIENE COSÌ:

$$\rightarrow m = D_1(D_2(c, k_2), k_1) \quad // D_1, D_2 \rightarrow \text{FUNZIONI DI DECIFRAZIONE DEL 2DES}$$

// PRIMA SI DECIFRA IL CRITTOGRAMMA CON k_2 IN MODO DA AVERE $C_1(m, k_1)$, E Poi SI DECIFRA IL CRITTOGRAMMA CON k_1 PER AVERE IL MESSAGGIO IN CHIARO

- IN MERITO ALLA SICUREZZA DELLA CIFRATURA MULTIPIA PER CIFRARI SIMMETRICI, ABBIAMO UNO SPAZIO DI CHIAVI EQUIVALENTE A $\rightarrow 2^{112}$. // $56 \text{ bit} + 56 \text{ bit}$

PERÒ ESISTE UN TIPO DI ATTACCO A QUESTI CIFRARI, DETTO MEET-IN-THE-MIDDLE, CHE NOSTRA CHE LA REALE SICUREZZA NON È DI 112 bit MA DI $\rightarrow 57 \text{ bit}$

MEET-IN-THE-MIDDLE

- CONSIDERIAMO IL CRITTOGRAMMA (C) DELL'ESERCIZIO, AVENNE (K_1) E (K_2) CHIAVI DI 56 bit

- QUANDO DECIFRAMO (C) CON k_2 , cioè $\rightarrow D_2(c, k_2) = C_1(m, k_1)$, ABBIAMO L'INTERVENTO DEL CRITTOANALISTA

- IL CRITTOANALISTA, INTERVIENE IN MEZZO ALLA COMUNICAZIONE ("IN-THE-MIDDLE") E PROCEDE COSÌ:

- PER OGNI POSSIBILE SCelta DI k_1 SI CALCOLA E SI SALVA IN UNA LISTA TUTTE LE CIFRATURE DI (m) FATTE CON $k_1 \rightarrow C_1(m, k_1)$.

QUINDI, UN TOTALE DI 2^{56} OPERAZIONI DI CIFRATURA

// ENUMERAZIONE DI TUTTE LE POSSIBILI CHIAVI k_1 LUNGHE 56 bit

- PER OGNI POSSIBILE VALORE k_2 , PRESSO DUE SEQUENZE BINARIE DI 56 bit , SI CALCOLA $\rightarrow D_2(c, k_2)$.

- PRENDE IL VALORE OTTENUTO E VA A VEDERE SE SI TROVA NELLA LISTA DI TUTTE LE CIFRATURE FATTE CON k_1 ($\otimes \otimes$):

→ PER COME È COSTRUITO IL CIFRARIO DOVRÀ ESSERE PER FORZA UNA COPPIA DI CHIAVI $\langle k_1, k_2 \rangle$, CHE

SODDISFA L'UGUAGLIANZA \otimes , E QUINDI, VERIFICANDO QUEST'ULTIMA FASE

- Dopo i punti precedenti, abbiamo che la sicurezza dell'attacco è di $2^{56} + 2^{56} = 2^{57}$, il cui costo:
 $\bullet 2^{56}$ → fase in cui si enumerano le chiavi la prima volta per calcolare tutte le possibili cifrature del messaggio
 - $\bullet 2^{56}$ → fase per enumerare tutte le chiavi lunghe 56 bit
 - Indicando con $\rightarrow N = 2^{56}$ cifrature + $O(N)$ decifrati, abbiamo che:
 $\rightarrow N^2 = 2^{112}$ cifrature + $O(N^2)$ decifrati, che corrisponde ad enumerare tutte le copie (K_1, K_2) con sequenze lunghe 112 bit
 - prendendo il costo dell'attacco, 2^{57} lo possiamo scrivere come $\rightarrow 2^{57} = 2^{56} + 2^{56} = 2(2^{56}) = 2N$
- // N descritto al punto precedente
- Abbiamo, quindi, la riduzione dei bit di sicurezza per questo tipo di attacco, poiché vede:
- $\rightarrow 2N \ll N^2$
- // $2^{57} \ll 2^{112}$

AES → Cifrari simmetrici

Descrivere il cifrario AES, in particolare le quattro operazioni eseguite in ciascuna fase, specificando il motivo per cui sono state utilizzate.

- CONSIDERIAMO IL CIFRARIO SIMMETRICO AES NELLA VERSIONE CON CHIAVE A 128 bit
- PER QUESTA VERSIONE IL CIFRARIO PREvede 10 FASI
- OGNI FASE OPERA SU UN BLOCCO DI 128 bit ORGANIZZATO COME UNA MATRICE Bidimensionale B di 16 byte

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

// $b_{i,j}$ → SONO i byte DELLA MATRICE
B

- IL BLOCCO DELLA PRIMA FASE È ESTRATTO DAL MESSAGGIO
- IL BLOCCO FINALE COSTITUISCE IL CRITTOGRAMMA
- LE CHIAVI LOCALI, cioè le chiavi di ogni fase COSTRUITE A PARTIRE DALLA CHIAVE SEGRETA, SONO DIVISE IN 16 byte POSTI IN CORRISPONDENZA AI byte DI B
- CONSIDERIAMO L'ORGANIZZAZIONE A GRANDI LINEE

TRASFORMAZIONE INIZIALE:

- IL MESSAGGIO (m) È CARICATO NELLA MATRICE B
- OGNI byte DI B È POSTO IN XOR (bit a bit) CON IL CORRISPONDENTE byte DELLA CHIAVE INIZIALE

FASE

- SEGUE LA TRASFORMAZIONE INIZIALE ED È RIPETUTA PER 10 VOLTE
- È SUDDIVISA IN 4 OPERAZIONI

OP1 → OGNI byte DELLA MATRICE B È TRASFORMATO MEDIANTE UNA S-box:
 $\rightarrow b_{i,j} \leftarrow S_{i,j}(b_{i,j})$

OP2 → LA MATRICE OTTENUTA VIENE PERMUTATA MEDIANTE SHIFT CICLICI SULLE RIGHE:

- LA PRIMA RIGA DELLA MATRICE RESTA INALTERATA //RIGA 0
- i byte CONTENUTI NELL'ALTRA RIGHE SONO SOGGETTI AD UNO SHIFT CIClico VERSO SINISTRA DI 1

OP 3 → LE COLONNE RISULTANTI VENGONO TRASFORMATE CON UN' OPERAZIONE ALGEBRICA:

- OGNI COLONNA DELLA MATRICE, TRATTATA COME VETTORE DI 4 ELEMENTI, VIENE MOLTIPLICATA PER UNA MATRICE PREFISSATA M DI 4×4 byte
- LA MOLTIPLICAZIONE TRA byte È ESEGUITA MODULO 2^8
- L'ADDITIONE TRA byte È ESEGUITA COME XOR

OP 4 → OGNI byte DELLA MATRICE RISULTANTE È POSTO IN XOR CON UN byte DELLA CHIAVE LOCALE PER QUELLA FASE → $b_{i,j} \leftarrow b_{i,j} \oplus K_{i,j}$

// $K_{i,j}$ È IL CORRISPONDENTE byte DELLA CHIAVE LOCALE

AES → Cifrari simmetrici

Il cifrario AES soddisfa la proprietà di non linearità. Cosa significa formalmente?

- LA PROPRIETÀ DI NON LINEARITÀ VIENE APPLICATA DALL'AES IN OGNI SUA FASE, IN PARTICOLARE NELL'OPERAZIONE 1 QUANDO INTERVIENE LA S-box
- LA S-box INTERVIENE CON UNA TRASFORMAZIONE NON LINEARE SULLA T-_{esima} CHIAVE DEL CIFRARIO NEI PROCEDIMENTI DI:
 - CIFRATURA
 - ESTRAZIONE DELLA SOTTOCHIAVE DI FASE
- LA TRASFORMAZIONE NON LINEARE È UNA FUNZIONE T APPLICATA NELLA FORMULA:
$$\rightarrow w(t) = T(w(t-1)) \oplus w(t-4)$$
 DONDE:
 - T È MULTIPLO DI 4 E $t > 4$
 - w(t) → T-_{esima} CHIAVE
- QUESTO TIPO DI TRASFORMAZIONE È UTILE CONTRO GLI ATTACCHI DI CRIPTOANALISI LINEARE, I QUALI CERCANO DI FARLE APPROXIMAZIONI LINEARI PER INDOVINARE I bif DELLA CHIAVE OSS (SI USA IN PARTICOLAR MODO PER CREARE CONFUSIONE NEI DATI DI INPUT)

Spiegare perché la proprietà di non linearità dell'AES è fondamentale per la sua sicurezza, illustrando come attaccare un cifrario lineare.

- LA PROPRIETÀ DI NON LINEARITÀ DELL'AES È FONDAMENTALE SUGLI ATTACCI DI CRITTOANALISI LINEARE DI TIPO TESTO IN CHIARO SCELTO // CHOSEN PLAIN TEXT
- IN TALI TIPOLOGIE DI ATTACCO SI CERCA DI COMPROMETTERE LA SICUREZZA DEL CIFRARIO RIDUCENDONE LO SPAZIO DELLE CHIAVI // ESPLORABILE PER UN CRITTOANALISTA

FUNZIONAMENTO ATTACCO

- IL CRITTOANALISTA SI PROCURA ALCUNE COPPIE MESSAGGIO/CRITTOGRAMMA COSÌ FORMATE:
→ $\langle m, c_1 \rangle$ e $\langle \bar{m}, c_2 \rangle$ // \bar{m} È IL COMPLEMENTARE DI m
- CONSIDERANDO UNA DI TALI COPPIE, SCELTA LA CHIAVE K , SI CALCOLA LA FUNZIONE DI CIFRATURA → $C(m, K)$
- SE $C(m, K) = c_1$, ALLORA PROBABILMENTE K È LA CHIAVE SEGRETA
// NON È CERTO PERCHÉ PIÙ CHIAVI POTREBBERO TRASFORMARE m IN c_1 .
POTREBBE ESSERE STATA TROVATA UNA CHIAVE CHE NON È DI QUEL CIFRARIO
- SI PROVA ALLORA K SU ALTRE COPPIE MESSAGGIO/CRITTOGRAMMA
- SE IL SUCCESSO SI RIPETE PER ALCUNE VOLTE CONSECUTIVE, ALLORA HA LA PRATICA CERTEZZA DI AVER TROVATO LA CHIAVE
- SE INVECE $C(m, K) = \bar{c}_2$, ALLORA PROBABILMENTE \bar{K} È LA CHIAVE SEGRETA PERCHÉ GENERIREBBERE CORRETTAMENTE LA COPPIA $\langle \bar{m}, c_2 \rangle$
- COME NEL CASO PRECEDENTE VIENE PROVATA LA CHIAVE (\bar{K}) SU ALTRE COPPIE
- SE I DUE CASI PRECEDENTI FALLISCONO, QUINDI NE \bar{K} NE \bar{K} È LA CHIAVE SEGRETA, PROCEDE PROVANDO UN'ALTRA CHIAVE
- IN CONCLUSIONE PROVATA UNA CHIAVE NON È NECESSARIO RIPETERE L'OPERAZIONE SUL SUO COMPLEMENTO, QUINDI → SI RIDUCE LO SPAZIO DELLE CHIAVI

Confrontare DES e AES dal punto di vista della sicurezza e della resistenza agli attacchi crittoanalitici noti.

- IN QUANTO A SICUREZZA, INIZIAMO COL DIRE CHE:

- IL DES UTILIZZA CHIAVI DA 56 bit // SI PRESENTA COME 2^{64} , MA 8 bit SONO DI PARITÀ
- L'AES UTILIZZA CHIAVI DISPOSTE A BLOCCI DA $128, 192, 256$ bit
// I BLOCCI DELLA CHIAVE SONO DISPOSTI IN UNA MATRICE

- GLI ATTACCHI CHE SI FANNO SUL DES SONO ATTACCHI DI TIPO BRUTEFORCE, CHE SI BASANO SULL'ESPLORAZIONE DELLO SPAZIO DELLE CHIAVI.

- UN ATTACCO CHE SI FA SU QUESTO CIFRARIO È DI TIPO CHOSEN PLAIN TEXT IN CUI:

- SI RIDUCE LO SPAZIO DELLE CHIAVI DA 2^{56} A 2^{55}
- Poi si esegue il BruteForce

- UN ALTRO TIPO DI ATTACCO CHOSSEN PLAIN TEXT È LA CRITTOANALISI DIFFERENZIALE IN CUI:

- IL CRITTOANALISTA SI PROCURA 2^{47} COPPIE $\langle m, c \rangle$ CON m SCELTO DA LUI
- SCEGLIE LE COPPIE DI MESSAGGI E ANALIZZA I CRIPTOGRAMMI, E LA DIFFERENZA TRA I RELATIVI CRIPTOGRAMMI PERMETTE DI ASSEGNARE PROBABILITÀ DIVERSE A CHIAVI DIVERSE
- LA CHIAVE CERCATA EMERGERÀ SULLE ALTRE COME QUELLA AVENTE PROBABILITÀ MASSIMA

- UN'ALTRA TECNICA DI ATTACCO DI TIPO CHOSSEN PLAIN-TEXT È LA CRITTOANALISI LINEARE DOLE:

- SI PRENDE LA FUNZIONE DI CIFRATURA E SE NE FA UN'APPROXIMAZIONE LINEARE IN MODO DA RIDURRE I BIT DELLA CHIAVE
- I bit RIMASTI VENGONO DETERMINATI CON UN BRUTEFORCE
- IL NUMERO DI COPPIE $\langle m, c \rangle$ DA ESAMINARE SI RIDURREÀ A 2^{43} .

TALE ANALISI VERRÀ FATTA ATTRAVERSO UN ATTACCO DI TIPO KNOWN PLAIN TEXT

- RIFERENDOCI ALL'AES, INVECE, ABBIAMO CHE NON È STATA COMPROMESSA NEMMENO LA VERSIONE PIÙ SEMPLICE A 128 bit

- PER TALE CIFRARIO ESISTONO SOLO ATTACCHI DI TIPO SIDE-CHANNEL CHE NON SFRUTTANO LE CARATTERISTICHE DEL CIFRARIO MA SOLTANTO LE DEBOLEZZE DELLA PIATTAFORMA IN CUI SI TROVANO

CIFRARI A COMPOSIZIONE DI BLOCCHI → Cifratura a blocchi

Descrivere cosa rappresenta e perché viene utilizzata la cifratura a composizione di blocchi di tipologia *Cipher Block Chaining* (CBC).

- CONSIDERANDO I CIFRARI SIMMETRICI (DES, AES), PER QUESTI ABBIANO CHE IL MESSAGGIO m VIENE SUDDIVISO IN UN CERTO NUMERO DI BLOCCHI DECIFRATI CON LA STESSA CHIAVE K, PER TUTTA UNA SESSIONE DI COMUNICAZIONE.

// DES (BLOCCHI 64 bit), AES (BLOCCHI DA 128/192/256 bit)

cioè CONPORTA AL CRITTOANALISTA L'OTTENIMENTO DI INFORMAZIONI IN QUANTO:

→ BLOCCI UGUALI DEL MESSAGGIO PRODUcono BLOCCI UGUALI DEL CRITTOGRAMMA //PROBLEMA!

- CON IL CIPHER BLOCK CHAINING (CBC) SI CERCA DI CREARE UNA DIPENDENZA NELLA CIFRATURA DI UN BLOCCO CON IL SUO SUCCESSIVO → L'i-ESIMO BLOCCO DEVE DIPENDERE DALLA CIFRATURA DEI BLOCCI PRECEDENTI //SEQUENZIALE

- QUINDI, OGNI VOLTA CHE SI INVIA UN MESSAGGIO m; SI AGGIUNGE UN "PREFISSO" (padding) DATO DAL CRITTOGRAMMA DEL BLOCCO PRECEDENTE (c_{i-1}) → MESSAGGI UGUALI PRODUcono CRITTOGRAMMI DIVERSI //RISOLUZIONE AL PROBLEMA INIZIALE

- DATO UN MESSAGGIO m; LA SUA CIFRATURA È DATA DA → $c_i = C(m_i \oplus c_{i-1}, K)$

// CRITTOGRAMMA PRODOTTO "UN PEZZO PER VOLTA"

- INVECE, DATO UN CRITTOGRAMMA c_i E CHIAVE K, LA DECIFRAZIONE AVVIENE FACENDO:

→ $m_i = D(c_i, K) \oplus c_{i-1}$ //PARALLELO

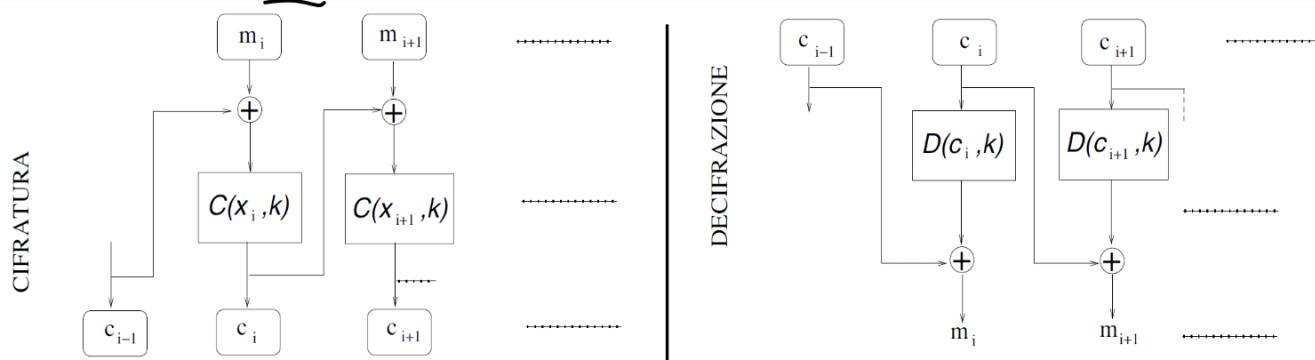
CIFRARI A COMPOSIZIONE DI BLOCCHI → Cifrari a composizione

Alice utilizza un cifrario a composizione di blocchi in modalità *Cipher Block Chaining* (CBC), con crittogramma iniziale c_0 , per cifrare un messaggio composto di N blocchi, e invia il crittogramma complessivo (c_0, c_1, \dots, c_N) a Bob.

1. Nel caso in cui il blocco di crittogramma c_1 si danneggi **nel corso della trasmissione**, quali blocchi possono essere decifrati correttamente da Bob?
2. Si assuma ora che l'errore avvenga invece **durante la cifratura** del secondo blocco di messaggio. Quali blocchi di crittogramma sono influenzati dall'errore? Quali blocchi possono essere decifrati correttamente da Bob se non si verificano altri errori?

- INIZIAMO COL CONSIDERARE LA STRUTTURA DELLE FUNZIONI DI CIFRATURA E DECIFRAZIONE

DEL METODO CBC:



- TALE METODO IMPLICA CHE:

- IL PROCESSO DI CIFRATURA SIA STRETTAMENTE SEQUENZIALE IN QUANTO IL CALCOLO DI OGNI c_i IMPIEGA IL RISULTATO c_{i-1} DEL PASSO PRECEDENTE
 - IL PROCESSO DI DECIFRAZIONE PUÒ ESSERE ESEGUITO IN PARALLELO SE TUTTI I BLOCCI CIFRATI SONO DISPONIBILI
- ① DALLE CONSIDERAZIONI PRECEDENTI E DALLO SCHEMA, IL FATTO CHE SI DANNEGGI IL BLOCCO c_1 NEL CORSO DELLA TRASMISSIONE → COMPROMETTE LA DECIFRAZIONE DELL'INTERO BLOCCO (c_1) E DEL MESSAGGIO DEL BLOCCO SUCCESSIVO $m_{i+1} \rightarrow m_2$
QUINDI, BOB DECIFRERÀ CORRETTAMENTE E IN PARALLELO I BLOCCI DEI CRYPTOGRAFFI:
→ c_0, c_3, \dots, c_N
- ② DALLE CONSIDERAZIONI PRECEDENTI E DALLO SCHEMA, IL FATTO CHE L'ERRORE SI VERIFICA DURANTE LA CIFRATURA DEL SECONDO BLOCCO DEL MESSAGGIO (m_1) → COMPROMETTE LA CIFRATURA DEL SUO BLOCCO E DI TUTTI I BLOCCI SUCCESSIVI, GIÀ → c_1, \dots, c_N .
QUINDI, L'UNICO BLOCCO CHE PUÒ ESSERE DECIFRATO CORRETTAMENTE È:
→ c_0 , INDIPENDENTEMENTE DAL FATTO CHE NON SI VERIFICINO ALTRI ERRORI
// È STATO SUPPOSTO CHE I BLOCCI DEL MESSAGGIO INIZINO DA m_0

TRE CIFRARI A CHIAVE PUBBLICA → Cifrari

Discutere in massimo trenta righe quali sono le differenze d'impiego tra i tre cfrari One-Time Pad, AES, e RSA (o più in generale i cfrari a chiave pubblica), giustificando le affermazioni fatte.

// SI DESCRIVONO I CIFRARI A CHIAVE PUBBLICA IN MODO GENERALE

- LA CRIPTOGRAFIA A CHIAVE PUBBLICA È UTILIZZATA PER LE COMUNICAZIONI DI MASSA
- SI UTILIZZANO CIFRARI ASIMMETRICI DOVE:
 - NON SI RICHIEDE UNO SCAMBIO DI CHIAVE TRA MITTENTE E DESTINATARIO
 - IL MITTENTE USERA' UNA CHIAVE PUBBLICA (K_{pub}), CHE È NOTA A TUTTI, PER CIFRARE IL MESSAGGIO.
// ANCHE LA FUNZIONE DI CIFRATURA ($C = C(m, K_{\text{pub}})$) È NOTA A TUTTI.

LA FUNZIONE DI CIFRATURA È UNA FUNZIONE ONE-WAY TRAP-DOOR, CIOÈ CALCOLARLA È FACILE MA DECIFRARLA È DIFFICILE // A MENO CHE NON SI CONOSCA LA "TRAP-DOOR", CIOÈ K_{priv}

- IL DESTINATARIO USERA' UNA CHIAVE PRIVATA (K_{priv}), CHE È NOTA SOLO A LUI, PER DECIFRARE IL MESSAGGIO
- L'OBIETTIVO È PERMETTERE A TUTTI DI INVIERE MESSAGGI CIFRATI, MA ABILITARE SOLO ① DESTINATARIO A DECIFRARLI
- IN UN SISTEMA DI M UTENTI CI SARÀ UN TOTALE DI $2M$ CHIAVI, DOVE OGNI UTENTE HA UNA COPPIA DI CHIAVI:
 $\rightarrow \langle K_{\text{pub}}, K_{\text{priv}} \rangle$.

OSS / MITTENTE E DESTINATARIO NON DOVRANNO PIÙ CONDIVIDERE UN SEGRETO, CIOÈ DI SCAMBIARSI PRIMA UNA CHIAVE, PER GORE ACCADE NEI CIFRARI SIMMETRICI

// SI DESCRIVONO LE DIFFERENZE D'IMPIEGO TRA ONE-TIME PAD, AES E RSA

- ONE-TIME PAD È UN CIFRARIO SIMMETRICO UTILIZZATO SOLO PER COMUNICAZIONI CLASSIFICATE.
LA SUA PECULIARITÀ È LA CHIAVE DA SCAMBIRE, CHE DEVE ESSERE:
 - LUNGA QUANTO IL MESSAGGIO STESO
 - GENERATA IN MODO CASUALE
 - MAI PIÙ RIUTILIZZABILE

- L'AES È UN CIFRARIO SIMMETRICO UTILIZZATO PER COMUNICAZIONI NON CLASSIFICATE.

PER QUESTO CIFRARIO, LE CHIAVI SONO POSTE IN MATRICI DA $4 \text{ byte} \times 4 \text{ byte}$, DI DIMENSIONI DI:

$\rightarrow 128 \text{ bit}$, 192 bit O 256 bit .

LE CHIAVI SONO BREVI E DEVONO ESSERE CAMBIATE PER OGNI SESSIONE DI COMUNICAZIONE

- L'RSA È UN CIFRARIO ASIMMETRICO CHE UTILIZZA IL MECCANISMO DEI CIFRARI A CHIAVE PUBBLICA, IMPLIMENTANDO L'ALGEBRA MODULARE IN FASE DI CODIFICA E DECODIFICA DUE INFORMAZIONI

- Dopo tutte le considerazioni, possiamo dire che per One-Time Pad e AES esiste un problema per lo scambio della chiave su un canale sicuro prima della comunicazione, mentre per l'RSA no.
// Tale scambio si chiama potrebbe essere gestito, ad esempio, tramite il protocollo Diffie-Hellman

EULERO, EUCLIDE ESTESO → Complessità in algebra

Dato un intero n definire la funzione di Eulero $\Phi(n)$, indicare se è noto un algoritmo efficiente per calcolarla e spiegare in termini matematici quale implicazione avrebbe questo algoritmo sul cifrario DES.

- PER UN INTERO $m > 1$ SI DEFINISCE LA FUNZIONE DI EULERO $\phi(m)$ COLE':
 - IL NUMERO DI INTERI MINORI DI m E CO-PRIMI CON ESSO, E SI INDICA COSÌ:
 $\rightarrow \phi(m) = |\mathbb{Z}_m^*|$ // $|\cdot|$ INDICA LA CARDINALITÀ E NON IL MODULO
- A SECONDA DEL VALORE CHE ASSUME m SI POSSONO PRESENTARE PIÙ CASI PER LA FUNZIONE DI EULERO:
 - SE m E' PRIMO, ALLORA VALE CHE $\rightarrow \phi(m) = m - 1$
 - SE m E' COMPOSTO, ALLORA VALE CHE $\rightarrow \phi(m) = m \left(1 - \frac{1}{p_1}\right) \dots \left(1 - \frac{1}{p_k}\right)$, DOVE p_1, \dots, p_k SONO FATTORI PRIMI DI m PRESI SENZA MOLTEPLICITÀ
 - SE m E' SEMIPRIMO, CIÒE' $m = p \cdot q$ PRODOTTO DAI NUMERI PRIMI $p \neq q$, ALLORA VALE CHE $\rightarrow \phi(m) = (p-1)(q-1)$
- UN ALGORITMO EFFICIENTE PER LA FUNZIONE DI EULERO E' L'ALGORITMO DI EUCLIDE ESTESO, USATO NEL SISTEMA DI CRIPTOGRAFIA RSA PER LE CREAZIONI DI CHIAVI ASIMMETRICHE
- LA FUNZIONE DI EULERO NON E' UTILIZZATA NEL DES

EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia

L'algoritmo di Euclide esteso EE è così definito:

```
Function EE(a, b):
    if (b == 0) return (a, 1, 0)
    else {
        (d', x', y') = EE(b, a mod b);
        (d, x, y) = (d', y', x' - ⌊a/b⌋ * y');
        return (d, x, y)
    }
```

Indicare quale problema risolve EE, cioè cosa rappresentano i valori di d, x, y all'uscita.

- QUESTO ALGORITMO È UN'ESTENSIONE DEL CLASSICO ALGORITMO DI EUCLIDE PER IL CALCOLO DEL MCD, IN CUI SI RISOLVE L'EQUAZIONE IN DUE INCognITE $\rightarrow a \cdot x + b \cdot y = \text{MCD}(a, b)$ *
- // si dovranno trovare la x e la y che soddisfano quell'equazione
- L'ALGORITMO PRENDE IN INPUT a E b E RESTITUISCE QUELLA TRIPPIA DI VALORI DOVE:
 - d È il MASSIMO COMUN DIVISORE TRA a E b
 - x, y SONO I VALORI DA DARE AGLI INCognITE PER SODDISFARE L'EQUAZIONE *

EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia

L'algoritmo di Euclide esteso EE è così definito:

```
Function EE(a, b):  
    if (b == 0) return (a, 1, 0)  
    else {  
        (d', x', y') = EE(b, a mod b);  
        (d, x, y) = (d', y', x' - ⌊a/b⌋ * y');  
        return (d, x, y)  
    }
```

Dimostrare come EE possa essere impiegato per calcolare un inverso in modulo per valori opportuni dei parametri **indicando i calcoli** eseguiti in un esempio numerico.

- LA FORMULA PER IL CALcolo DELL'INVERSO IN MODULO E' $\rightarrow X = a^{-1} \text{ mod } b$
- INNANZITUTTO, SCEGLIAMO a E b COPRIMI FRA LORO, PERCHÉ CIÒ CI GARANTISCE L'ESISTENZA E L'UNICITÀ DELL'INVERSA.
DEVE VIVERE, QUINDI, CHE $\rightarrow \text{mcd}(a, b) = 1$ (*)
- SI SCRIVE LA FORMULA INIZIALE COME LA SEGUENTE CONGRUENZA $\rightarrow ax \equiv 1 \pmod{b}$
- TALE CONGRUENZA EQUIVALE A $\rightarrow ax = bz + 1$ PER UN OPPORTUNO VALORE DI z
- PONENDO $y = -z$ E CONSIDERANDO (*) SI OTTIENE L'EQUAZIONE $\rightarrow ax + by = \text{mcd}(a, b)$
- DALLA SOLUZIONE DI QUEST'ULTIMA EQUAZIONE, TRAMITE EUCLIDE ESTESO, SI OTTIENE IL VALORE DELL'INVERSO $\rightarrow x$
//cioè il 2° VALORE DELLA TRIPLO OTTENUTA (d, x, y) , DALL'ALGORITMO 3: EUCLIDE ESTESO

ESEMPIO

- CONSIDERIAMO DI CALCOLARE L'INVERSO DI $\rightarrow X = 5^{-1} \text{ mod } 132$
- FACENDO RIFERIMENTO AI DISCORSI PRECEDENTI, OTTIENIAMO L'EQUAZIONE $\rightarrow 5x + 132y = 1$
 $a=5$, $b=132$, $\text{mcd}(5, 132) = 1$
- SI PROCEDE, APPLICANDO L'ALGORITMO DI EUCLIDE ESTESO:
 1. EE(5, 132):
 - CASO BASE NO → CHIAMATA SU $(b, a \text{ mod } b)$ $\rightarrow 132, 5 \text{ mod } 132 \rightarrow$ QUOTIENTE 0, RESIDUO 5
 2. EE(132, 5):
 - CASO BASE NO → CHIAMATA SU $(b, a \text{ mod } b)$ $\rightarrow 5, 132 \text{ mod } 5 \rightarrow$ QUOTIENTE 26, RESIDUO 2
 3. EE(5, 2):
 - CASO BASE NO → CHIAMATA SU $(b, a \text{ mod } b)$ $\rightarrow 2, 5 \text{ mod } 2 \rightarrow$ QUOTIENTE 2, RESIDUO 1
 4. EE(2, 1):
 - CASO BASE NO → CHIAMATA SU $(b, a \text{ mod } b)$ $\rightarrow 1, 2 \text{ mod } 1 \rightarrow$ QUOTIENTE 2, RESIDUO 0

5. EE(1,0) :

- SI CASO BASE \rightarrow S. CHIUSA LA CHIAMATA RESTITUENDO AL PUNTO 4. $\rightarrow \langle d, 1, 0 \rangle = \langle 1, 1, 0 \rangle$

// LA TRIPLOA DI VMORE, AUA CHIUSURA DELLA CHIAMATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

4.1. EE(2,1) :

- ESSENDO CHE $a=2, b=1, d'=1, x'=1, y'=0$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 0, 1 - \lfloor \frac{2}{1} \rfloor \cdot 0 \rangle \rightarrow \langle 1, 0, 1 - (2) \cdot 0 \rangle \rightarrow \langle 1, 0, 1 \rangle \in$$

Si RESTITUISCE AL PUNTO 3.

// LA TRIPLOA DI VMORE, AUA CHIUSURA DELLA CHIAMATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

3.1. EE(5,2) :

- ESSENDO CHE $a=5, b=2, d'=1, x'=0, y'=1$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 1, 0 - \lfloor \frac{5}{2} \rfloor \cdot 1 \rangle \rightarrow \langle 1, 1, 0 - (2) \cdot 1 \rangle \rightarrow \langle 1, 1, -2 \rangle \in$$

Si RESTITUISCE AL PUNTO 2.

// LA TRIPLOA DI VMORE, AUA CHIUSURA DELLA CHIAMATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

2.1. EE(12,5) :

- ESSENDO CHE $a=12, b=5, d'=1, x'=1, y'=-2$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, -2, 1 - \lfloor \frac{12}{5} \rfloor \cdot (-2) \rangle \rightarrow \langle 1, -2, 1 - (2)(-2) \rangle \rightarrow \langle 1, -2, 5 \rangle \in$$

Si RESTITUISCE AL PUNTO 1.

// LA TRIPLOA DI VMORE, AUA CHIUSURA DELLA CHIAMATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

1.1. EE(5,132) :

- ESSENDO CHE $a=5, b=132, d'=1, x'=-2, y'=53$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 53, -2 - \lfloor \frac{5}{132} \rfloor \cdot 0 \rangle \rightarrow \langle 1, 53, -2 - 0 \rangle \rightarrow \langle 1, 53, -2 \rangle \in$$

Si RESTITUISCONO I TRE VMORE

AUA FUNZIONE DI EUCLIDE COME $\langle d, x, y \rangle$

\rightarrow I VALORES DELL'INVERSO SARÀ QUENCO RAPPRESENTATO DA $\textcircled{x} \rightarrow 53$

EULERO, EUCLIDE ESTESO → Algoritmi per la crittografia

L'algoritmo di Euclide esteso EE è così definito:

```
Function EE(a, b):
    if (b == 0) return (a, 1, 0)
    else {
        (d', x', y') = EE(b, a mod b);
        (d, x, y) = (d', y', x' - ⌊a/b⌋ * y');
        return (d, x, y)
    }
```

Indicare un protocollo crittografico in cui EE è utilizzato.

- L'ALGORITMO DI EUCLIDE ESTESO PUÒ ESSERE UTILIZZATO NELL'RSA PER CALCOLARE L'INVERSO IN MODULO DEL DEL SEGUENTE VALORE INTERO → $d = e^{-1} \text{ mod } \phi(n)$, DONGE ABBIAMO CHE:
 - e e $\phi(m)$ DOBBERO ESSERE CO-PRIMI TRA ZORO.
 - DORÀ VAVERE → $e < \phi(m)$

QUESTA PROCEDURA AVVIENE → NELLA FASE DI COTRAZIONE DELLA CHIAVE

RSA, RSA ATTACCHI → RSA

Spiegare in cosa consiste il cifrario RSA, definendone tutti i parametri e indicando esplicitamente le operazioni eseguite per ottenerli e la loro complessità computazionale.

- L'RSA È UN CIFRARIO A CHIAVE PUBBLICA IN W SI USANO DUE CHIAVI, UNA PUBBLICA (K_{pub}) E UNA PRIVATA (K_{priv}), PER CIFRARE E DECIFRARE I MESSAGGI.
NON NECESSITA DELLO SCAMBIO DI CHIAVI, IN MANIERA PREVENTIVA, SU QUALCHE CANALE SICURO
- È CONSIDERATO UN CIFRARIO SICURO PERCHÉ BASA LA SUA DIFFICOLTÀ SULLA FATTORIZZAZIONE DI GRANDI NUMERI COMPOSTI
- CI SONO QUATTRO FASI PER QUESTO CIFRARIO:

- 1) CREAZIONE DELLA CHIAVE
- 2) STRUTTURAZIONE DEL MESSAGGIO
- 3) CIFRATURA
- 4) DECIFRAZIONE

CREAZIONE DELLA CHIAVE

- OGNI UTENTE DESTINATARIO ESEGUE LE SEGUENTI OPERAZIONI:
 - SCEGLIE I VALORI P E q PRIMI E MOLTO GRANDI. // IL LORO PRODOTTO DEVE ESSERE DI ALMENO 2048 bit.
ESSI DEVONO ESSERE GENERATI IN TEMPO POLINOMIALE
// INOLTRE, SU P E q CI SI SVOLGE IL TEST DI PRIMAVERA
 - SI CALCOLA m=p·q E LA FUNZIONE DI EUCLIDE $\phi(m)=(p-1)(q-1)$.
QUESTO PASSAGGIO RICHIESTE TEMPO POLINOMIALE NEL NUMERO DI CIFRE
 - SI SCEGLIE UN VALORE e TRA I CHE → $e < \phi(m)$.
INOLTRE, DEVE RISULTARE CHE e SIA CO-PRIMO CON $\phi(m)$, CIOÈ DEVE VIVERE CHE → $\text{mcd}(e, \phi(m))=1$.
QUESTA VERIFICA SI FA IN TEMPO POLINOMIALE
 - SI CALCOLA L'INVERSO IN MODULO FACENDO → $d = e^{-1} \bmod \phi(m)$, UTILIZZANDO L'ALGORITMO DI EUCLIDE ESTESO.
QUESTO PASSAGGIO RICHIESTE TEMPO POLINOMIALE
// L'ESISTENZA DELL'INVERSA È GARANTITA DALLA CO-PRIMAVERA TRA e E $\phi(m)$
 - OTTIENE LA COPPIA (chiave pubblica, chiave privata) COSÌ COMPOSTA:
 - $K_{\text{pub}} = \langle e, m \rangle$, CHE VIENE RESA PUBBLICA
 - $K_{\text{priv}} = \langle d \rangle$, CHE LA MANTIENE SEGRETA



STRUTTURAZIONE DEL MESSAGGIO

- UN MESSAGGIO È CODIFICATO IN BINARIO, E LO TRATTIAMO COME UN INTERO, CHE INDICHEREMO CON → m
- PER POTER USARE IL CIFRARIO DEVE VALERE CHE → $m < m'$
- SE ABBIANO CHE $m' \geq m$, ALLORA → SI DIVIDE IL MESSAGGIO IN BLOCCHI DI $b = \lceil \log_2 m \rceil$ bit
- // TUTTI I BLOCCHI SONO CIFRATI CON LA STESSA CHIAVE
- LA DIMENSIONE MASSIMA DI UN BLOCCO DIPENDE DALLA $K[\text{pub}]$ DEL DESTINATARIO, PER CIÒ:
 - SI FISSA UN LIMITE (INFERIORE) COMUNE PER IMPIEGARE BLOCCHI DELLE STESE DIMENSIONI PER TUTTI I DESTINATARI.

UN BLOCCO DI (b) bit PER I MESSAGGI COMPORTA IL SEGUENTE LIMITE → $m < 2^b \leq m'$

MASSIMO VALORE CON CUI SI PUÒ SCRIVERE IL MESSAGGIO

CIFRATURA

- IL MITTENTE COSTRUISCE UN CRITTOGRAMMA (C) IN FUNZIONE DEL MESSAGGIO (m) COSÌ:
 - $c = C(m, K[\text{pub}]) = m^e \bmod m'$ // CON LA CHIAVE PUBBLICA DEL DESTINATARIO
- Ciò RICHIEDE TEMPO POLINOMIALE UTILIZZANDO LE QUADRATURE SUCCESSIVE

DECIFRAZIONE

- IL DESTINATARIO RICEVE IL CRITTOGRAMMA (C) E LO DECIFRA USANDO LA SUA $K[\text{priv}]$, CHE È CONTENUTA IN (d), IN QUESTA MANIERA → $m = D(c, K[\text{priv}]) = c^d \bmod m'$.

ANCHE PER QUESTO PASSAGGIO SI RICHIEDE TEMPO POLINOMIALE UTILIZZANDO LE QUADRATURE SUCCESSIVE

OSS / LA DECIFRAZIONE DEVE AVERE CORrettezza NELLA FORMULA, PERCHÉ È POSSIBILE RISALIRE AL MESSAGGIO
(ELEVANDO IL CRITTOGRAMMA AL VALORE (d))

RSA, RSA ATTACCHI → RSA

I parametri del cifrario RSA, nonché il suo impiego, sono noti. Dimostrare che il cifrario funziona, cioè tutti i messaggi sono cifrati e decifrati correttamente, quindi che è corretto.

- LA CORRETTEZZA E' RAPPRESENTATA DAL FATTO CHE, DECIFRANDO CON $K[\text{priv}]$ IL CONTENUTO OTTENUTO CON $K[\text{pub}]$, SI È POSSIBILE RISENARE AD (m) → $D(C(m, K[\text{pub}]), K[\text{priv}]) = m$

- CONSIDERANDO L'RSA BISOGNA DIMOSTRARE CHE:

$$\rightarrow m = C^d \bmod M, \text{ E SCRIVENDO IL } 2^{\circ} \text{ MEMBRO COME:}$$

$$\rightarrow m = (m^e \bmod m)^d \bmod M, \text{ E RAGGRUPPANDO GLI ESPONENTI OTTENIAMO:}$$

$$\rightarrow m = m^{ed} \bmod M.$$

- PER LA DEMOSTRAZIONE BISOGNERÀ INTRODURRE IL TEOREMA DI CORRETTEZZA (DELL'RSA) :

PER QUALSIASI INTERO $m < M$ SI HA $(m^e \bmod m)^d \bmod M = m$.

$(m), (e)$ E (d) SONO I PARAMETRI DEL CIFRARIO RSA

- IN MERITO AL TEOREMA DOVEMO DISTINGUERE DUE CASI, CHE SONO RELATIVI AI VALORI (P) E (Q) SCELTI DAL DESTINATARIO:

1) (P) E (Q) NON DIVIDONO (m)

2) (P) DIVIDE (m) , MA (Q) NON DIVIDE (m) , OPPURE (P) DIVIDE (m) , MA (Q) NON DIVIDE (m)

(P) E (Q) NON DIVIDONO (m)

- IN QUESTO CASO ABBIAMO → $\text{mcd}(m, n) = 1$ // ALLORA NOTEREMO (m) DIVIDE (m) , VISTO CHE (m) È IL PRODOTTO DI $P \cdot Q$

- QUINDI, PER IL TEOREMA DI EUERO → $m^{\phi(m)} \equiv 1 \pmod{m}$

- POICHÉ (d) È L'INVERSO DI (e) IN $\mathbb{Z}_{\phi(m)}^*$, CIÒ È $d = e^{-1} \pmod{\phi(m)}$, CHE RISCRIVIAMO COME LA SEGUENTE CONGRUENZA:

$$\rightarrow ed \equiv 1 \pmod{\phi(m)}$$

- PER UN OPPORTUNO VALORE DI $r > 0$ SCRIVIAMO IL PRODOTTO PRECEDENTE COME → $ed = 1 + r\phi(m)$

- DALE CONSIDERAZIONI PRECEDENTI PONIAMO → $m^{ed} \bmod m = m^{1+r\phi(m)} \bmod m$, CHE DIVENTA:

→ $m(m^{r\phi(m)}) \bmod m$, E APPLICANDO NUOVAMENTE LE PROPRIETÀ DELLE POTENZE ASSUMO:

→ $m(m^{\phi(m)})^r \bmod m$, DA CUI UTILIZZANDO IL TEOREMA DI EUERO SI OTTIENE:

→ $m(1)^r \bmod m$, DA CUI SI TROVA IL MESSAGGIO IN CHIARO:

→ $m \bmod m = m$

// QUESTA UGUALANZA VALE PERCHÉ $m < M$

\hookrightarrow (P) DIVIDE m , MA (Q) NON DIVIDE m , OPPURE (P) DIVIDE m , MA (Q) NON DIVIDE m

- CONSIDERIAMO LA DEMOSTRAZIONE SUL 1° CASO // QUELLA SUL 2° CASO E' SPECULARE

- POICHÉ (P) DIVIDE m , PER QUALSIASI INTERO $n > 0$ ABBIAMO $\underline{m \equiv m^n \equiv 0 \pmod p}$, CHE POSSIAMO SCRIVERE:

$$\rightarrow \underline{(m^n - m) \equiv 0 \pmod p}$$

- CON PROCEDIMENTO ANALOGO AL PRECEDENTE PONIAMO $\underline{m^{ed} \pmod q = m^{1+e_2\phi(n)} \pmod q}$, CHE EQUIVALE A:

$$\rightarrow \underline{m(m^{e_2\phi(n)}) \pmod q}, \text{ ED ESPANDENDO LA FUNZIONE DI EULEO OTTENIAMO:}$$

$$\rightarrow \underline{m(m^{(q-1)(p-1)}) \pmod q}, \text{ CHE POSSIAMO SCRIVERE COME:}$$

$$\rightarrow \underline{m(m^{(q-1)})^{p-1} \pmod q} \quad (*)$$

- POICHÉ PER IL TEOREMA DI EULEO VALE $m^{(q-1)} \equiv 1 \pmod q$, SCRIVIAMO QUESTA (*) COME:

$$\rightarrow \underline{m(1)^{p-1} \pmod q}, \text{ CHE SCRIVEREMO COME:}$$

$$\rightarrow \underline{m \pmod q}$$

- ABBIAMO OTTENUTO, DOPO UNA SERIE DI PASSAGGI $\underline{m^{ed} \pmod q \equiv m \pmod q}$, CHE POSSIAMO SCRIVERE COME:

$$\rightarrow \underline{(m^{ed} - m) \equiv 0 \pmod p}$$

- NE CONSEGUO CHE $m^{ed} - m$ E' DIVISIBILE SIA PER (P) CHE PER (Q) \rightarrow QUINDI, E' DIVISIBILE ANCHE PER IL LORO PRODOTTO $\rightarrow \underline{pq = m}$

- OTTIENIAMO CHE $(m^{ed} - m) \equiv 0 \pmod m$, DA CUI DERIVA LA TESI:

Si consideri una modifica del cifrario RSA in cui si utilizza come modulo pubblico un numero primo invece di un semiprimo, cioè $n = p$ invece di $n = p q$. **Discutere** se e come questa modifica influenza la sicurezza del cifrario.

- Poiché $m = p$, questo comporta una modifica nella funzione di Eulero, che verrà calcolata come $\phi(m) = m - 1$, cioè $\phi(m) = p - 1$.
TALE CALCOLO È FACILE PER IL CRITTOANALISTA.
- Poiché è nota la e , facente parte della chiave pubblica $K[\text{pub}] = \langle e, m \rangle$, il crittoanalista si ricava facilmente la chiave privata $K[\text{priv}] = \langle d \rangle$ calcolando l'inverso in modulo $\rightarrow d = e^{-1} \bmod \phi(m)$
// UTILIZZO DELL'ALGORITMO DI EUCLIDE ESTESO
- Essendo in possesso della chiave pubblica e privata, il crittoanalista, decifrerà ogni ciphogramma c e troverà il messaggio m calcolando $\rightarrow m = c^d \bmod m$
// IL CIPHGRAMMA c ERA STATO OTTENUTO FACENDO $\rightarrow c = m^e \bmod m$
- In questo caso non abbiamo sicurezza, perché questa è basata sulla fattorizzazione // che è assente

OSS / Si suppone che un crittoanalista conosca a priori la struttura del cifrario,
quindi anche della modifica $m = p$.
Di conseguenza, riesce a ricavarsi $\phi(m)$

RSA, RSA ATTACCHI → Complessità in algebra

Dato un intero n dimostrare che dato un intero n prodotto di due numeri primi, il calcolo della funzione di Eulero $\Phi(n)$ e la fattorizzazione di n sono problemi computazionalmente equivalenti.

- CONSIDERATO (m) SEMI-PRIMO, DATO DA $m = pq$, ALLORA → IL CALCOLO DI $\phi(m)$ DA (m) E LA

FATTORIZZAZIONE DI (m) SONO PROBLEMI EQUIVALENTI.

CIASCUNO SI TRASFORMA NEGL'ALTRO IN TEMPO POLINOMIALE

- DISTINGUIAMO SEPARATAMENTE I DUE CASI :

① FATTORIZZARE (m) IMPLICA CALCOLARE $\phi(m)$

② DA (m) E $\phi(m)$ SI TROVANO (p) E (q) IN TEMPO POLINOMIALE

FATTORIZZARE (m) IMPLICA CALCOLARE $\phi(m)$

- SE SAPPIAMO FATTORIZZARE (m) ,cioè TROVARE $(p) \in (q)$ TALI CHE $m = pq$, ALLORA PER CALCOLARE $\phi(m)$ SI FA:

$$\rightarrow \phi(m) = (p-1)(q-1), \text{ CHE SI FA IN TEMPO POLINOMIALE} // ② \text{ DECREMENTI E } ① \text{ PRODOTTO}$$

DA (m) E $\phi(m)$ SI TROVANO (p) E (q) IN TEMPO POLINOMIALE

- PRENDIAMO LA FUNZIONE DI EULERO $\phi(m) = (p-1)(q-1)$ E NE SVILUPPIAMO IL PRODOTTO:

$$\rightarrow \phi(m) = pq - p - q + 1$$

- FACENDO ALCUNE MANIPOLAZIONI ALGEBRICHE, E RICORDANDO CHE $m = pq$, OTTENIAMO:

$$\rightarrow \phi(m) = m - (p+q) + 1$$

- INDICHIAMO CON x_1 LA SEGUENTE SOMMA → $x_1 = p+q$, E METTENDOLA NELLA FORMULA PRECEDENTE, OTTENIAMO:

$$\rightarrow x_1 = m - \phi(m) + 1 // \phi(m) \in \text{ANDATO A 2° MEMBRO}$$

- ADESSO, CONSIDERIAMO IL SEGUENTE QUADRATO → $(p-q)^2 = (p+q)^2 - 4pq$

- ESSENDO CHE $m = pq$, ALLORA IL PRECEDENTE QUADRATO DIVENTA → $(p-q)^2 = (p+q)^2 - 4m$

- INDICHIAMO CON x_2 LA SEGUENTE DIFFERENZA → $x_2 = p-q$, E DALL'EQUAZIONE PRECEDENTE OTTENIAMO:

$$\rightarrow (x_2)^2 = (p+q)^2 - 4m, \text{ E NETTENDO DENTRO LE PARENTESI } x_1 = p+q, \text{ OTTENIAMO:}$$

$$\rightarrow x_2 = \sqrt{x_1^2 - 4m}$$

- POICHÉ CONOSCiamo SOMMA E DIFFERENZA, Siamo IN GRADO DI TROVARE (p) E (q) PONENDOle COLE:

$$\bullet p = \frac{x_1 + x_2}{2}$$

$$\bullet q = \frac{x_1 - x_2}{2}$$

// ABBIANO TROVATO (p) E (q) DA x_1 E x_2 , i quali, USANDO $\phi(m)$ E m → TEMPO POLINOMIALE

RSA, RSA ATTACCHI → RSA

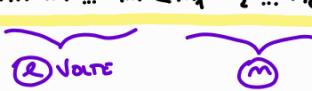
Mostrare come sia possibile attaccare il cifrario RSA quando più utenti scelgono lo stesso valore del parametro e , anche quando questo parametro è piccolo.

// L'ATTACCO È SUA CHIAVE PUBBLICA → $K[\text{pub}] = \langle e, n \rangle$

- SUPPONIAMO CHE CI SIANO SCELTI UN VALORE MOLTO PICCOLO DI \textcircled{e}
- SINCE \textcircled{e} È MOLTO PICCOLO, SUPPONIAMO CHE CI SIANO ALTRI $\textcolor{blue}{e}$ UTENTI CHE SI SONO SCELTI QUESTO STESSO VALORE
- CONSIDERANDO CHE GLI \textcircled{e} UTENTI RICEVONO LO STESSO MESSAGGIO (m) CIFRATO, ALLORA CI SARANNO $\textcolor{blue}{e}$ CRIPTOGRAFMI CON MOLTEPLICITÀ \textcircled{e} COSÌ FORMATI:

$$\begin{aligned} C_1 &= m^e \pmod{M_1} \\ C_2 &= m^e \pmod{M_2} \\ &\vdots \\ C_e &= m^e \pmod{M_e} \end{aligned}$$

// STESSO MESSAGGIO, MA CRIPTOGRAFMI TUTTI DIVERSI

- CON CERTEZZA SAPPIAMO CHE VALE → $M \leq M_i, \forall i : i \in [1, e]$
- FACCiamo UN'ULTERIORE IPOTESI, CIOÈ CHE → M_1, M_2, \dots, M_e SONO CO-PRIMI TRA loro
- PER IL TEOREMA CHINSE DEL RESTO, NOI SAPPIAMO CHE → ESISTE E SI PUÒ FACILMENTE CALCOLARE UN UNICO VALORE m' TALE CHE → $m' \leq M$ DOTE $M = M_1 \cdot M_2 \cdot \dots \cdot M_e$ E CHE SOBBISFA LA CONGRUENZA $m' \equiv m^e \pmod{M}$ $\textcircled{*}$
- // AL CRIPTODANALISTA INTERESSA CALCOLARE m' PER ARRIVARE AD (m)
- POICHÉ $M \leq M_i, \forall i$, CIOÈ IMPLICA → $M \cdot M \cdot \dots \cdot M \leq M_1 \cdot M_2 \cdot \dots \cdot M_e$, CIOÈ → $M^e \leq M$

- CONSIDERANDO LA CONGRUENZA $\textcircled{*}$, ABBIANO CHE → $m' \pmod{M} = m^e \pmod{M}$
- POICHÉ VOLETE CHE $m' \leq M$ E $M^e \leq M$ → LA RIDUZIONE IN MODULO NON SI PUÒ APPLICARE, QUINDI, ABBIANO:
 $\rightarrow m' = m^e$
- SI POSSONO DECIFRARE I CRIPTOGRAFMI E TROVARE IL MESSAGGIO COMME (m) , CHE ERA STATO INVIAVO AGLI \textcircled{e} UTENTI,
FACENDO → $m = \sqrt[m]{m'}$

RSA, RSA ATTACCHI → RSA: attacchi

Mostrare come sia possibile attaccare il cifrario RSA quando più utenti scelgono lo stesso valore del modulo n .

- IL CRITTOANALISTA INIZIA SELEZIONANDO DUE UTENTI CHE HANNO CHIAVI PUBBLICHE:

CON LO STESSO VALORE DI m :

- $\langle e_1, m \rangle$
- $\langle e_2, m \rangle$

- LA SELEZIONE DEVE ESSERE TALE CHE VALGA $\rightarrow \text{mcd}(e_1, e_2) = 1$

- IN QUESTO CASO ESISTERANNO DUE INTERI r ED s CALCOLABILI DIRETTAMENTE CON L'ALGORITMO DI EUCLIDE ESTESO PER CUI $\rightarrow e_1 r + e_2 s = 1$

- PONIAMO IL FATTO CHE:

- $e_1 < e_2$ //CASO e_1 E' SIMMETRICO

• IL CRITTOANALISTA RIESCE AD INTERCETTARE DUE CRITTOGRAMMI C_1 E C_2 RELATIVI ALLO STESSO MESSAGGIO (m)

- CI SARÀ $\rightarrow m = m^{e_1 r + e_2 s} = (C_1^r \times C_2^s) \bmod n$ CHE POTREMO SCRIVERE ANCHE COSÌ:
 $\rightarrow ((C_1^{-1})^{-r} \times C_2^s) \bmod n$ *

- POICHÉ C_1 ED n SONO PRIMI TRA LORO, IL CRITTOANALISTA PUÒ CALCOLARE $\rightarrow C_1^{-1} \bmod n$ CON L'ALGORITMO DI EUCLIDE ESTESO

- PER ESPONENTIAZIONE CALCOLERÀ $(C_1^{-1})^{-r}$ E C_2^s E RI COSTRUIRÀ (m) ATTRaverso L'ESPRESSione *

Dare un esempio di applicazione del cifrario RSA impiegando parametri numerici molto piccoli per cifrare il messaggio costituito dalle due cifre centrali del proprio numero di matricola, se le due cifre sono minori di 10, aggiungere 33. Riportare esplicitamente tutte le operazioni aritmetiche eseguite (utilizzare l'algoritmo di Euclide Esteso per il calcolo dell'inverso in modulo, e il metodo delle quadrature successive per gli elevamenti a potenza).

- SCEGLIAMO COME PARAMETRI NUMERICI I SEGUENTI NUMERI PRIMI → $p=3$ E $q=7$
- SI CONSIDERANO LE CIFRE CENTRALI DEL NUMERO DI MATRICOLA → 44.
- Poiché 44 > 10, ALWEA NON SI AGGIUNGE 33. //PER COME È DESCRITTO NEL RESTO DELL' ESECIZIO QUINDI, IL MESSAGGIO SARÀ → M=44
- CI CALCOLIAMO $m = pq \in \phi(m) = (p-1)(q-1)$, OTTENENDO → $m=21$ E $\phi(m)=12$
- SI SCEGLIE $e < \phi(m)$ E CO-PRIMO CON $\phi(m)$ → $e=5$
- SI CALCOLA L'INVERSO IN MODULO FACENDO → $d = e^{-1} \bmod \phi(m)$, E UTILIZZANDO L'ALGORITMO DI EUCLIDE ESTESO, SU: → $d=5^{-1} \bmod 12$

L'algoritmo di Euclide esteso EE è così definito:

Function EE(a, b):

```

    if ( $b == 0$ ) return ( $a, 1, 0$ )
    else {
        ( $d'$ ,  $x'$ ,  $y'$ ) = EE( $b, a \bmod b$ );
        ( $d$ ,  $x$ ,  $y$ ) = ( $d'$ ,  $y'$ ,  $x' - \lfloor a/b \rfloor * y'$ );
        return ( $d$ ,  $x$ ,  $y$ )
    }
  
```

// Poiché l'inverso si ha da → $x = a^{-1} \bmod b$, PER APPLICARE LA FORMULA SU EUCLIDE SI CONSIDERERANNO COME VALORI DI $a = 5$ E $b = 12$

1. EE(5, 12):

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → $12, 5 \bmod 12$ → QUOTIENTE 0, RESTO 5

2. EE(12, 5):

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → $5, 12 \bmod 5$ → QUOTIENTE 2, RESTO 2

3. EE(5, 2):

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → $2, 5 \bmod 2$ → QUOTIENTE 2, RESTO 1

4. EE(2, 1):

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → $1, 2 \bmod 1$ → QUOTIENTE 0, RESTO 1

5. EE(1, 0):

- SI CASO BASE → SI CHIUSA LA CHIAMATA RESTITUENDO AL PUNTO 4. → $(d, x, y) = (1, 1, 0)$

// LA TRIPLO DI VALORI, ANA CHIUSURA DELLA CHIAMATA FINIRÀ IN → (d', x', y')

4.1. EE(2,1) :

- ESSENDO CHE $a=2$, $b=1$, $d'=1$, $x'=1$, $y'=0$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:
 $\rightarrow \langle 1, 0, 1 - \lfloor \frac{2}{1} \rfloor \cdot 0 \rangle \rightarrow \langle 1, 0, 1 - (2) \cdot 0 \rangle \rightarrow \langle 1, 0, 1 \rangle \in \text{SI RESTITUISCE AL PUNTO 3.}$

// LA TRIPLOA DI VALORI, ANA CHIUSURA DELLA CHIATTA FINIRÀ IN → $\langle d', x', y' \rangle$

3.1. EE(5,2) :

- ESSENDO CHE $a=5$, $b=2$, $d'=1$, $x'=0$, $y'=1$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:
 $\rightarrow \langle 1, 1, 0 - \lfloor \frac{5}{2} \rfloor \cdot 1 \rangle \rightarrow \langle 1, 1, 0 - (2) \cdot 1 \rangle \rightarrow \langle 1, 1, -2 \rangle \in \text{SI RESTITUISCE AL PUNTO 2.}$

// LA TRIPLOA DI VALORI, ANA CHIUSURA DELLA CHIATTA FINIRÀ IN → $\langle d', x', y' \rangle$

2.1. EE(12,5) :

- ESSENDO CHE $a=12$, $b=5$, $d'=1$, $x'=1$, $y'=-2$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:
 $\rightarrow \langle 1, -2, 1 - \lfloor \frac{12}{5} \rfloor (-2) \rangle \rightarrow \langle 1, -2, 1 - (2)(-2) \rangle \rightarrow \langle 1, -2, 5 \rangle \in \text{SI RESTITUISCE AL PUNTO 1.}$

// LA TRIPLOA DI VALORI, ANA CHIUSURA DELLA CHIATTA FINIRÀ IN → $\langle d', x', y' \rangle$

1.1. EE(5,12) :

- ESSENDO CHE $a=5$, $b=12$, $d'=1$, $x'=-2$, $y'=5$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:
 $\rightarrow \langle 1, 5, -2 - \lfloor \frac{5}{12} \rfloor (5) \rangle \rightarrow \langle 1, 5, -2(0) \rangle \rightarrow \langle 1, 5, 0 \rangle.$

IL VALORE DEL' INVERSA SARÀ IN → $x=5$

- SOSTITUENDO IL VALORE TORNATO AL POSIZIONE DI \bar{e}^{-1} IN $d = \bar{e}^{-1} \pmod{\phi(n)}$, SI OTTIEDE LA CHIAVE PRIVATA:

$$\rightarrow d = 5 \pmod{12} = 5$$

- LA COPPIA (CHIAVE PUBBLICA, CHIAVE PRIVATA) SARÀ, QUINDI, RAPPRESENTATA DA:

- $K[\text{pub}] = \langle r, m \rangle$, cioè → $\langle 5, 21 \rangle$
- $K[\text{priv}] = \langle d \rangle$, cioè → $\langle 5 \rangle$

- PER POTER USARE IL CIFRARIO DEVE VALERE CHE → $m < m$, MA $44 > 21 \rightarrow$ ALLORA SI CONSIDERA LA STRUTTURA DEL MESSAGGIO ($m=44$) CODIFICATA IN BINARIO → 101100. // SU 6 b.i.f

- SI DIVIDE IL MESSAGGIO IN BLOCCHI DI $b = \lceil \log_2 m \rceil$ b.i.f → $b = \lceil \log_2 21 \rceil = \lceil 4.392... \rceil = 4$:

$$\bullet m_1 = 1011$$

$$\bullet m_2 = 00--$$

- ESSENDO CHE NEL BLOCCO FINALE DEL MESSAGGIO MANCANO 2 b.i.f, VISTO CHE IL MESSAGGIO ERA DI 6 b.i.f, SI FA UN PADDING, AGGIUNGENDO 4 b.i.f CASUALI, PER COMPLETARZI: $\langle 1011 | 0001 \rangle$ // NUOVO VALORE DI $m = 177$

$$\begin{array}{c} \text{padding} \\ \swarrow \quad \searrow \\ m_1 = 11 \quad m_2 = 1 \end{array}$$

— ADesso si fa la cifratura su entrambi i blocchi, separatamente, trovando due crittogrammi $\rightarrow C_1$ e C_2 .

Si utilizzerà la Tecnica delle quadrature successive:

// Ricorda $K[\text{pub}] = \langle \alpha, m \rangle \rightarrow \langle s, \alpha^s \rangle$

1) $C_1 = m_1^2 \pmod{m} \rightarrow C_1 = 11^5 \pmod{21}$, da qui ne scriviamo l'esponente come scomposizione di potenze (di 2):

$\rightarrow C_1 = 11^{1+4} \pmod{21}$, e da qui, si procede con il metodo:

• $C_1 = 11^1 \pmod{21} = 11$

• $C_1 = 11^2 \pmod{21} = (11)^2 \pmod{21} = 16$

• $C_1 = 11^4 \pmod{21} = (16)^2 \pmod{21} = 4$

Ci fermiamo alla potenza massima, e consideriamo solo i risultati delle potenze nella scomposizione:

$\rightarrow C_1 = (11 \cdot 1^4) \pmod{21} \rightarrow (11 \cdot 4) \pmod{21} \rightarrow 44 \pmod{21} = 2$ // l'16 di 1^2 non è stato considerato

2) $C_2 = m_2^2 \pmod{m} \rightarrow C_2 = 1^5 \pmod{21}$, da qui ne scriviamo l'esponente come scomposizione di potenze (di 2):

$\rightarrow C_2 = 1^{1+4} \pmod{21}$, e da qui, si procede con il metodo:

• $C_2 = 1^1 \pmod{21} = 1$

• $C_2 = 1^2 \pmod{21} = (1)^2 \pmod{21} = 1$

• $C_2 = 1^4 \pmod{21} = (1)^2 \pmod{21} = 1$

Ci fermiamo alla potenza massima, e consideriamo solo i risultati delle potenze nella scomposizione:

$\rightarrow C_2 = (1 \cdot 1^4) \pmod{21} \rightarrow (1 \cdot 1) \pmod{21} \rightarrow 1 \pmod{21} = 1$ // l'1 di 1^2 non è stato considerato

— IL CRITTOGRAMMA FINALE È DATO DALLA CONCATENAZIONE DEI CRITTOGRAMMI C_1 E C_2 , DOPO CHE SONO STATI CONVERTITI IN BINARIO SU $\binom{b}{b}$ b.r:

• 2 = 0010

• 1 = 0001

$\rightarrow C = 0010\ 0001$ // IN DECIMALE CORRISPONDE A 33

OSS (Si ricorda che, in caso di messaggio (m) non scomposto, la cifratura si fa $\rightarrow C = m^e \pmod{m}$)

— LA DECIFRAZIONE AVVIENE IN MANIERA ANALOGA \rightarrow Si considera il crittogramma C , scomposto nei blocchi:

C_1 e C_2 , per trovare il messaggio, scomposto anch'esso in due messaggi m_1 e m_2 .

Si utilizzerà, anche qua, la Tecnica delle quadrature successive sui due crittogrammi separatamente:

// m_1 corrisponderà alla decifrazione di C_1 ed m_2 corrisponderà alla decifrazione di C_2 .

RICORDO $K[\text{priv}] = \langle d \rangle \rightarrow \langle s \rangle$

1) $M_1 = c_1^d \pmod{m} \rightarrow m_1 = 2^5 \pmod{21}$, E DA QUI NE SCRIVIAMO L'ESPONENTE COME SCOMPOSIZIONE DI POTENZE (di 2):

$\rightarrow m_1 = 2^{1+4} \pmod{21}$, E DA QUI, SI PROCEDE CON IL METODO:

• $m_1 = 2^1 \pmod{21} = 2$

• $m_1 = 2^2 \pmod{21} = (2)^2 \pmod{21} = 4$

• $m_1 = 2^4 \pmod{21} = (4)^2 \pmod{21} = 16$

Ci FERMiamo ALLA POTENZA MASSIMA, E CONSIDERIAMO SOLO I RISULTATI DELLE POTENZE NELLA SCOMPOSIZIONE:

$\rightarrow M_1 = (2^1 \cdot 2^4) \pmod{21} \rightarrow (2 \cdot 16) \pmod{21} \rightarrow 32 \pmod{21} = 11$ // l'1 di 2^2 Non È STATO CONSIDERATO

2) $M_2 = c_2^d \pmod{m} \rightarrow m_2 = 1^5 \pmod{21}$, E DA QUI NE SCRIVIAMO L'ESPONENTE COME SCOMPOSIZIONE DI POTENZE (di 2):

$\rightarrow M_2 = 1^{1+4} \pmod{21}$, E DA QUI, SI PROCEDE CON IL METODO:

• $m_2 = 1^1 \pmod{21} = 1$

• $m_2 = 1^2 \pmod{21} = (1)^2 \pmod{21} = 1$

• $m_2 = 1^4 \pmod{21} = (1)^2 \pmod{21} = 1$

Ci FERMiamo ALLA POTENZA MASSIMA, E CONSIDERIAMO SOLO I RISULTATI DELLE POTENZE NELLA SCOMPOSIZIONE:

$\rightarrow M_2 = (1^1 \cdot 1^4) \pmod{21} \rightarrow (1 \cdot 1) \pmod{21} \rightarrow 1 \pmod{21} = 1$ // l'1 di 1^2 Non È STATO CONSIDERATO

- IL MESSAGGIO COMPLETO È DATO DALLA CONCATENAZIONE DEI MESSAGGI M_1 E M_2 , DOPO CHE SONO STATI CONVERTITI IN BINARIO SU (b) bit, E A TAIE CONCATENAZIONE SI TOGLIE LA PARTE FINALE AGGIUNTA // IL PADDING

• 11 = 1011

• 1 = 0001

$\rightarrow m = 10110001$ // IN DECIMALE CORRISPONDE A 177

$\rightarrow M = 101100$ // È STATO TOLTO IL PADDING

OSS (Si ricorda che, in caso di crittogramma non scomposto, la decifrazione si fa $\rightarrow m = c^d \pmod{m}$)

- CONVERTENDO IL MESSAGGIO IN DECIMALE, OTTENIAMO IL MESSAGGIO IN CHIARO (CORRETTO) $\rightarrow M = 44$

RSA, RSA ATTACCHI → Chiave pubblica

Si consideri il cifrario RSA con chiave pubblica $n = 187$, $e = 9$.

Forzare il cifrario e decifrare il crittogramma c composto dalle due cifre centrali del proprio numero di matricola. Riportare esplicitamente le operazioni aritmetiche eseguite.

- Innanzitutto, consideriamo le due cifre centrali di un numero di matricola $\rightarrow c=12$
// Si scelgono dei numeri piccoli per non avere i conti successivi lunghi, cioè ① e ②
- Poiché sappiamo che ① deve essere sempre → p e q dovranno essere dei numeri primi in modo da forzarlo $\rightarrow m = pq$
- DEDUCIAMO, quindi, di avere $p=11$ e $q=17$. // o viceversa
cioè ci permette di ottenere la ① dell'esercizio $\rightarrow 187 = 11 \cdot 17$ // $m = pq$
- Di conseguenza ottieniamo $\phi(m) = (p-1)(q-1)$, e cioè $\phi(m) = 10 \cdot 16 = 160$
- VERIFICIAMO SE LA SEGUENTE COMBINAZIONE È RISPETTATA $\rightarrow 2 < \phi(m) \rightarrow 9 < 160$ OK!
- INOLTRE, si verifica che ② è co-primo con $\phi(m)$ perché vale $\rightarrow \text{mcd}(e, \phi(m)) = 1$, cioè $\rightarrow \text{mcd}(9, 160) = 1$
// $9 = 3^2$ e $160 = 2^5 \cdot 5$ → ① MASSIMO DIVISORE IN COMUNE
- PER POTER FORZARE IL CIFRATO SI CALCOLA L'INVERSO IN MODULO FACENDO $\rightarrow d = 2^{-1} \pmod{\phi(m)}$, E UTILIZZANDO L'ALGORITMO DI EUCLIDE ESTESO SU $\rightarrow d = 9^{-1} \pmod{160}$

L'algoritmo di Euclide esteso EE è così definito:

Function EE(a, b):

```
if ( $b == 0$ ) return ( $a, 1, 0$ )
else {
    ( $d'$ ,  $x'$ ,  $y'$ ) = EE( $b, a \bmod b$ );
    ( $d$ ,  $x$ ,  $y$ ) = ( $d'$ ,  $y'$ ,  $x' - \lfloor a/b \rfloor * y'$ );
    return ( $d$ ,  $x$ ,  $y$ )
}
```

// Poiché l'inversa si: $m \rightarrow x = a^{-1} \pmod{b}$, per

APPLICARE LA FORMULA SU EUCLIDE SI CONSIDERERANNO COME VALORI ④ E ⑤ $\rightarrow a = 9$ E $b = 160$

1. EE (9, 160)

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → 160, $9 \bmod 160$ → QUOTIENTE ⑥, RESTO ⑦

2. EE (160, 9)

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → 9, $160 \bmod 9$ → QUOTIENTE ⑧, RESTO ⑨

3. EE (9, 7)

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → 7, $9 \bmod 7$ → QUOTIENTE ⑩, RESTO ⑪

4. EE (7, 2)

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → 2, $7 \bmod 2$ → QUOTIENTE ⑫, RESTO ⑬

5. EE (2, 1)

- CASO BASE NO → CHIAMATA SU $(b, a \bmod b)$ → 1, $2 \bmod 1$ → QUOTIENTE ⑭, RESTO ⑮

6. EE(1,0) :

- SI CASO BASE \rightarrow S. CHIUSA LA CHIARATA RESTITUENDO AL PUNTO 5. $\rightarrow \langle d, 1, 0 \rangle = \langle 1, 1, 0 \rangle$

// LA TRIPLOA DI VMORE, ANA CHIUSURA DELLA CHIARATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

5.1. EE(2,1) :

- ESSENDO CHE $a=2, b=1, d'=1, x'=1, y'=0$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 0, 1 - \lfloor \frac{2}{1} \rfloor \cdot 0 \rangle \rightarrow \langle 1, 0, 1 - (2) \cdot 0 \rangle \rightarrow \langle 1, 0, 1 \rangle \in$$
 Si RESTITUISCE AL PUNTO 4.

// LA TRIPLOA DI VMORE, ANA CHIUSURA DELLA CHIARATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

4.1. EE(7,2) :

- ESSENDO CHE $a=7, b=2, d'=1, x'=0, y'=1$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 1, 0 - \lfloor \frac{7}{2} \rfloor \cdot 1 \rangle \rightarrow \langle 1, 1, 0 - (3) \cdot 1 \rangle \rightarrow \langle 1, 1, -3 \rangle \in$$
 Si RESTITUISCE AL PUNTO 3.

// LA TRIPLOA DI VMORE, ANA CHIUSURA DELLA CHIARATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

3.1. EE(9,7) :

- ESSENDO CHE $a=9, b=7, d'=1, x'=1, y'=-3$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, -3, 1 - \lfloor \frac{9}{7} \rfloor (-3) \rangle \rightarrow \langle 1, -3, 1 - (1)(-3) \rangle \rightarrow \langle 1, -3, 4 \rangle \in$$
 Si RESTITUISCE AL PUNTO 2.

// LA TRIPLOA DI VMORE, ANA CHIUSURA DELLA CHIARATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

2.1. EE(160,9) :

- ESSENDO CHE $a=160, b=9, d'=1, x'=-3, y'=4$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, 4, -3 - \lfloor \frac{160}{9} \rfloor \cdot 4 \rangle \rightarrow \langle 1, 4, -3 - (17)(4) \rangle \rightarrow \langle 1, 4, -71 \rangle \in$$
 Si RESTITUISCE AL PUNTO 1.

// LA TRIPLOA DI VMORE, ANA CHIUSURA DELLA CHIARATA FINIRÀ IN $\rightarrow \langle d', x', y' \rangle$

1.1. EE(9,160) :

- ESSENDO CHE $a=9, b=160, d'=1, x'=4, y'=-71$ METTIAMO IN $\langle d, x, y \rangle = \langle d', y', x' - \lfloor \frac{a}{b} \rfloor y' \rangle$:

$$\rightarrow \langle 1, -71, \dots \rangle \rightarrow$$
 IL VMORE DELL'INVERSA SARÀ IN $\rightarrow x = -71$

- SOSTITUENDO IL VALORE TROVATO AL POSTO DI x' IN $d = x' \bmod \phi(n)$ SI OTTIENE LA CHIAVE PRIVATA:

$$\rightarrow d = -71 \bmod 160 = 89 \quad // -71 + \lceil \frac{-71}{160} \rceil \cdot 160 \rightarrow -71 + 1(160) = 89$$

OSS (IL CALCOLO SI RIDUCE IN MAGNITUDINE CON NUMERI NEGATIVI, come $-x \bmod y$, SI FA LA SEGUENTE OPERAZIONE:
 $\rightarrow -x + \lceil \frac{|-x|}{y} \rceil \cdot y$)

- LA COPPIA (CHIAVE PUBBLICA, CHIAVE PRIVATA) SARÀ, QUINDI, RAPPRESENTATA DA:

$$\bullet K[\text{pub}] = \langle r, m \rangle, \text{cioè} \rightarrow \langle 9, 187 \rangle$$

$$\bullet K[\text{priv}] = \langle d \rangle, \text{cioè} \rightarrow \langle 89 \rangle$$

- ADESSO SI DECIFRA IL CRIPTOGRAFIA \textcircled{C} PER TRARRE IL MESSAGGIO IN CHIARO, FACENDO:

$$\rightarrow m = c^d \bmod n. // D(c, K[\text{pub}])$$

Si utilizzerà la TECNICA DELLE QUADRATURE SUCCESSIVE per calcolare $\rightarrow m = 12^{89} \bmod 187$

- SCRIVIAMO L'ESPOLENTE IN $m = 12^{89} \bmod 187$ CONE SCOMPOSIZIONE DI POTENZE (D: ②):

$$\rightarrow m = 12^{1+8+16+64} \bmod 187, \text{ QUINDI, SI PROCEDE CON IL METODO:}$$

$$\bullet m = 12^1 \bmod 187 = 12$$

$$\bullet m = 12^2 \bmod 187 = (12)^2 \bmod 187 = 144$$

$$\bullet m = 12^4 \bmod 187 = (144)^2 \bmod 187 = 166$$

$$\bullet m = 12^8 \bmod 187 = (166)^2 \bmod 187 = 67$$

$$\bullet m = 12^{16} \bmod 187 = (67)^2 \bmod 187 = 1$$

$$\bullet m = 12^{32} \bmod 187 = (1)^2 \bmod 187 = 1$$

$$\bullet m = 12^{64} \bmod 187 = (1)^2 \bmod 187 = 1$$

Ci FERMiamo ALLA POTENZA MASSIMA, E CONSIDERIAMO SOLO I RISULTATI DELLE POTENZE NELLA Scomposizione:

$$\rightarrow m = (12^1 \cdot 12^8 \cdot 12^{16} \cdot 12^{64}) \bmod 187, \text{ cioè } \rightarrow m = (12 \cdot 67 \cdot 1 \cdot 1) \bmod 187 \rightarrow m = 804 \bmod 187 = 56$$

// L'① DI 12^{32} , IL 166 DI 12^4 E IL 144 DI 12^8 Non si contano

- IL MESSAGGIO IN CHIARO SARÀ, QUINDI $\rightarrow m = 56$

RSA, RSA ATTACCHI → RSA

Si supponga che Eve intercetti un crittogramma $c = m^e \pmod{n}$ diretto ad Alice. Si supponga inoltre che Alice sia disposta a decifrare per Eve qualsiasi crittogramma c' , a patto che c' sia diverso da c . Descrivere come Eve possa decifrare m in tempo polinomiale, richiedendo ad Alice la decifrazione del crittogramma $c' = c x^e$ dove $x < n$ è un intero casuale, co-primo con n .

— PREMESSE iniziali → DAL TESTO DELL' ESEMPIO SAPPIAMO CHE:

- ALICE È LA PROPRIETARIA DEL $K[\text{Pub}]$ E DELLA $K[\text{Priv}]$ ASSOCIATE A QUELLA (\mathbb{Q}) E A QUELLA (\mathbb{M})
- EVE INTERCETTA UN MESSAGGIO CIFRATO CHE STA ANDANDO AD ALICE, CIOÈ IL CRITTOGRAMMA (C) , DI CUI LUÌ NON NE SA NIENTE PERCHÉ CONOSCE SOLO $\underline{K[\text{Pub}]} = \langle \mathbb{x}, \mathbb{m} \rangle$

— EVE, CONSIDERATA LA DECIFRAZIONE CHIESTA AD ALICE, PER SICUREZZA STOLGE → $\text{mcd}(\mathbb{x}, \mathbb{m})$, CONTROLLANDO SE SI TROVA IN UNO DEI SEGUENTI DUE CASI:

- $\underline{\text{mcd}(\mathbb{x}, \mathbb{m})} \neq 1 \rightarrow$ Fine, perché Eve ha fattorizzato (\mathbb{M}) .

Quindi, trova la chiave privata di Alice decifrando (C) e pure tutti i crittogrammi che lei riceve in futuro.

- $\underline{\text{mcd}(\mathbb{x}, \mathbb{m})} = 1 \rightarrow$ Eve, per ottenere $\underline{m^l}$, procede con la decifrazione di $\underline{c} \rightarrow \underline{c'} = c x^e \pmod{m}$
// $\underline{m^l}$ È il risultato della decifrazione di $\underline{c'}$ chiesta da Eve

— PER TROVARE $\underline{m^l}$ si decifra $\underline{c'}$ facendo → $\underline{m^l} = (c x^e)^d \pmod{m}$, e si può scrivere così:

$$\rightarrow (\underline{c^d \pmod{m}})(\underline{x^{ed} \pmod{m}}) \pmod{m}, \text{ DA CUI SI OTTIENE:}$$

SARÀ LA DECIFRAZIONE DEL CRITTOGRAMMA CHE INTERESSA AD EVE PER SCOPRIRE IL MESSAGGIO (\mathbb{m})

→ $\underline{m x^{ed} \pmod{m}}$, SUL QUALE SI APPLICA IL TEOREMA DI EULER, VISTO CHE (\mathbb{x}) ED (\mathbb{M}) SONO CO-PRIMI:

→ $\underline{m x^{1+\frac{r}{2}\phi(m)} \pmod{m}}$, E SCOMPONENTANDO SI OTTIENE:

→ $\underline{m x x^{\frac{r}{2}\phi(m)} \pmod{m}}$, E SI OTTIENE INFINE LA SOLUZIONE:

$$\rightarrow \underline{m^l = m x \pmod{m}} \quad // \text{QUINDI, È STATO FATTO} \rightarrow \underline{m^l = (c')^d \pmod{m}}$$

— VISTO CHE EVE SA LA DECIFRAZIONE DEL CRITTOGRAMMA $\underline{c'}$, PER DECIFRARE (C) PROCEDERÀ COSÌ:

- SI FA DARE LA DECIFRAZIONE DI $\underline{c'} \rightarrow \underline{m^l = (c')^d \pmod{m}}$

- SI CALCOLA Poi L'INVERSA FACENDO → $\underline{x^{-1} \pmod{m}}$

// L'ESISTENZA E L'UNICITÀ DELL'INVERSA SONO GARANTITE DALLA CO-PRIMALITÀ TRA (\mathbb{M}) E (\mathbb{x})

- OTTIENE IL MESSAGGIO IN CHIARO (\mathbb{m}) FACENDO → $\underline{m = (x^{-1} \cdot m^l) \pmod{m}}$

// IN QUESTO ULTIMO PUNTO È STATA PRESA QUESTA $\underline{(\mathbb{x})}$ ED È STATA "MANIPOLATA" PORTANDO (\mathbb{m}) A PRIMO MEMBRO

DIFFIE-HELLMAN → Scambio di chiavi

Descrivere il protocollo Diffie-Hellman e discuterne la sicurezza.

- QUESTO PROTOCOLLO È UN ALGORITMO PER GENERARE E SCAMBIARSI UNA CHIAVE DI SESSIONE TRA DUE UTENTI.
- Poi VERRÀ USATA IN UNA SESSIONE DI COMUNICAZIONE PROTETTA TRAMITE L'USO DI UN CIFRARIO PERFETTO
- I DUE UTENTI GENERANO $K[\text{SESSION}]$ CON LA STESSA PROBABILITÀ, E IN MANIERA INCREMENTALE, INVIANOSSI IN CHIARO ALCUNI PEZZI DI essa
- QUESTI PEZZI RICOSTRUIRANNO LA CHIAVE SE VERRANNO COMBINATI CON LE INFORMAZIONI, SEGRETE E DIVERSE, DI CIASCUNO DEI DUE UTENTI

FUNZIONAMENTO PROTOCOLLO // UTENTI ALICE E BOB

- ALICE E BOB SI ACCORDANO PUBBLICAMENTE SU UN NUMERO PRIMO p MOLTO GRANDE (MIGLIAIA DI bit).
- Si ACCORDANO ANCHE SU UN GENERATORE g DI \mathbb{Z}_p^*
- \mathbb{Z}_p^* È L'INSIEME DI TUTTI GLI INTERI MINORI DI p E CO-PRIMI CON esso, cioè $\rightarrow \mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$
- LA SCELTA DI LAVORARE CON p PRIMO CI GARANTISCE L'ESISTENZA DI ALMENO UN GENERATORE g
- SE ALICE E BOB NON HANNO RISORSE CHE GLI PERMETTONO DI TROVARE LA COPPIA $\langle p, g \rangle$, POSSONO UTILIZZARE UNA COPPIA PUBBLICA MESSA A DISPOSIZIONE DAL SISTEMA // LINEE GUIDA DEL NIST
- PARTE DI COMUNICAZIONE TRA ALICE E BOB:
 - ALICE HA BISOGNO DI PREPARARE IL SUO SEGRETO, QUINDI, SCEGLIE A CASO UN $x > 0 \in \mathbb{Z}_p^*$, cioè:
 $\rightarrow 1 < x < p-1$ // \textcircled{x} È IL SEGRETO
 - SI CALCOLA $A = g^x \text{ mod } p$, E INVIA IN CHIARO \textcircled{A} A BOB TENENDOSI RISERVATA L'INFORMAZIONE SEGRETA \textcircled{x}
 - ANALOGAMENTE BOB SCEGLIE A CASO UN $y > 0 \in \mathbb{Z}_p^*$, cioè $\rightarrow 1 < y < p-1$
 - SI CALCOLA $B = g^y \text{ mod } p$, E INVIA IN CHIARO \textcircled{B} AD ALICE TENENDOSI RISERVATA L'INFORMAZIONE SEGRETA \textcircled{y}// A QUESTO PUNTO IL CRITTOANALISTA Vede IN CHIARO SOLAMENTE $\rightarrow \langle g, p, A, B \rangle$
- ALICE, DOPO AVER RICENNUTO \textcircled{B} DA BOB, SI CALCOLA $\rightarrow K[\text{SESSION}] = B^x \text{ mod } p$, E cioè $\rightarrow g^{xy} \text{ mod } p$
// Poiché \textcircled{B} ERA UN NUMERO CASUALE IN \mathbb{Z}_p^* LO È ANCHE \textcircled{B}^x
- BOB, DOPO AVER RICENNUTO \textcircled{A} DA ALICE, SI CALCOLA $\rightarrow K[\text{SESSION}] = A^y \text{ mod } p$, E cioè $\rightarrow g^{xy} \text{ mod } p$
// Poiché \textcircled{A} ERA UN NUMERO CASUALE IN \mathbb{Z}_p^* LO È ANCHE \textcircled{A}^y
- ALLA FINE DEL PROTOCOLLO OSSERVIAMO CHE ENTRambi i PARTNER HANNO LA STESSA $K[\text{SESSION}]$, LA QUALE VERRÀ UTILIZZATA PER LE CIFRATURE SIMMETRICHE SUCCESSIVE

- IN QUANTO A SICUREZZA IL PROTOCOLLO PUÒ RITENERSI SICURO.
- Ciò è dovuto al fatto che il crittoanalista, dopo aver intercettato $\langle g, p, A, B \rangle$, per trovare $K_{\text{[SESSION]}}$ dovrà calcolare il LOGARITMO DISCRETO di A o di B .
- LA RISOLVIZIONE DEL LOGARITMO DISCRETO È UN PROBLEMA COMPUTAZIONALMENTE DIFFICILE
- L'UNICA VULNERABILITÀ PER QUESTO PROTOCOLLO STA NEI ATTACCO DI TIPO MAN-IN-THE-MIDDLE.
- IN QUESTO ATTACCO IL CRIPTOANALISTA ATTIVO DOVREBBE MODIFICARE LA COMUNICAZIONE TRA I DUE PARTNER
- OSS / LA CHIAVE DI SESSIONE NON VIAGGIA MAI IN CHIARO, MA SOLO DEI SUOI PEZZI, CHE
IN QUESTO CASO SONO (A) E (B) /

DIFFIE-HELLMAN → Scambio di chiavi

Descrivere un attacco attivo al protocollo (di tipo *man-in-the-middle*).

- FUNZIONAMENTO ATTACCO MAN-IN-THE-MIDDLE :

- ALICE MANDA A BOB $\rightarrow A = g^x \text{ mod } p$
 - BOB MANDA AD ALICE $\rightarrow B = g^y \text{ mod } p$
 - EVE SOTTRAE (A) E (B) DAL CANALE E LI SOSTITUISCE CON UN SUO NUMERO (PUBBLICO) $\rightarrow E = g^z \text{ mod } p$,
Dove $z \in \mathbb{Z}_p^*$, cioè $\rightarrow 1 < z < p-1$
 - Poi INVIA LA STESSA (E) SIA AD ALICE CHE A BOB
 - LA CHIAVE DI SESSIONE CHE SI COSTRUISCE ALICE SARÀ $\rightarrow K_A = E^x \text{ mod } p$, cioè $\rightarrow g^{xz} \text{ mod } p$
 - LA CHIAVE DI SESSIONE CHE SI COSTRUISCE BOB SARÀ $\rightarrow K_B = E^y \text{ mod } p$, cioè $\rightarrow g^{yz} \text{ mod } p$
 - A QUESTO PUNTO SIA K_A CHE K_B SONO NOTE AD EVE
 - ALICE, DOPO ANER COSTRUITO LA CHIAVE DI SESSIONE K_A , INIZIA LA SESSIONE DI COMUNICAZIONE CIFRANDO CON K_A IL CRITTOGRAMMA DA MANDARE A BOB
 - EVE SOTTRAE IL CRITTOGRAMMA CHE PASSA SUL CANALE, PREPARATO DA ALICE, E LO DECIFRA CON K_A CHE GLI PERMETTE DI LEGGERE IL MESSAGGIO
 - DOPO ANER LETTO IL MESSAGGIO LO CIFRA CON K_B E LO INVIA A BOB
 - BOB, CHE HA RICEVUTO IL CRITTOGRAMMA CAMBIATO DA EVE, LO DECIFRA CON K_B E LEGGE IL MESSAGGIO
 - SE BOB RISPONDE, CIFRA IL MESSAGGIO DI RISPOSTA CON K_B E LO INVIA AD ALICE
 - A QUESTO PUNTO, EVE, INTERCETTA IL CRITTOGRAMMA E LO DECIFRA CON K_B CHE GLI PERMETTE DI LEGGERE IL MESSAGGIO
 - DOPO LA LETTURA DEL MESSAGGIO, LO CIFRA CON K_A E LO INVIA AD ALICE // E COSÌ VIA DA CAPO
- EVE HA COSTRUITO, QUINDI, QUESTE CHIAVI SIMMETRICHE PER CONVERSARE CON BOB FINGENDOSI ALICE E CON ALICE FINGENDOSI BOB

DIFFIE-HELLMAN → Scambio di chiavi

L'algoritmo DH per lo scambio pubblico di chiavi è basato sull'uso di un primo p e di un generatore g di \mathbb{Z}_p^* . Scelti $p = 13$ e $g = 2$:

1. Verificare che 2 è un generatore di \mathbb{Z}_{13}^* ;
2. Presi i due interi x, y (corrispondenti alle due cifre meno significative e maggiori o uguali a 2 del proprio numero di matricola) come scelte casuali di due partner che devono costruire una chiave comune, indicare come procede l'algoritmo per questi due valori e quale chiave si costruisce.

① PER VERIFICARE SE UN NUMERO ⑨ È UN GENERATORE DI \mathbb{Z}_p^* PROSEGUIMOSI COSÌ:

- SI CALCOLANO TUTTE LE POTENZE (IN MODULO) DI ⑨ FACENDO $\rightarrow g^K \bmod p$, CON K CHE VA DA 1 A $p-1$.

FACENDO CIÒ SI VERIFICA IL FATTO CHE SIANO DESCRIVENDO TUTTI GLI ELEMENTI DI \mathbb{Z}_p^* .

IN QUESTO CASO ABBIAMO \mathbb{Z}_{13}^* E DOBBIAMO FARE $2^K \bmod 13$, CON K CHE VA DA 1 A 12

K	1	2	3	4	5	6	7	8	9	10	11	12
$2^K \bmod 13$	2	4	8	3	6	12	11	9	5	2	7	1

- SE, SVOLGENDO LE POTENZE SI TROVA IL VALORE ① PRIMA DELLA FINE, CI DÀ LA POTENZA $(p-1)$ -ESIMA, ALLORA \rightarrow ⑨ NON È UN GENERATORE PER \mathbb{Z}_p^* .

IN CASO CONTRARIO \rightarrow ⑨ È UN GENERATORE DI \mathbb{Z}_p^* .

IN QUESTO CASO SI VERIFICA CHE ② È UN GENERATORE DI \mathbb{Z}_{13}^*

② CONSIDERIAMO LA 3° E 4° CIFRA DEL MIO NUMERO DI MATRICOLA \rightarrow 4 E 3 E SIMULIAMO IL PROTOCOLLO

DIFFIE-HELLMAN SU $x=4$ E $y=3$: // PER GENERARE LA CHIAVE DI SESSIONE

- ALICE SI CALCOLA $A = g^x \bmod p \rightarrow A = 2^4 \bmod 13 = 3$

• Poi INVIA $A=3$ A BOB

- BOB SI CALCOLA $B = g^y \bmod p \rightarrow B = 2^3 \bmod 13 = 8$

• Poi INVIA $B=8$ AD ALICE

- ALICE, DOPO ANER RICEVUTO ⑧ DA BOB, SI CALCOLA $K[\text{SESSION}] = B^x \bmod p \rightarrow K[\text{SESSION}] = 8^4 \bmod 13 = 1$

- BOB, DOPO ANER RICEVUTO ⑦ DA ALICE, SI CALCOLA $K[\text{SESSION}] = B^y \bmod p \rightarrow K[\text{SESSION}] = 3^3 \bmod 13 = 1$

- QUINDI, LA CHIAVE DI SESSIONE PER ENTRAMBI È $\rightarrow K[\text{SESSION}] = 1$

CURVE ELLITTICHE → Curve ellittiche

1. **Disegnare** la curva ellittica prima $E_{13}(a,b)$ scegliendo come coefficienti a e b le due cifre meno significative del proprio numero di matricola.
2. Calcolare l'ordine della curva ellittica prima $E_{13}(a,b)$.
3. La curva definisce un gruppo abeliano su \mathbb{Z}_{13} ?

① Scegliamo come coefficienti le due cifre meno significative della mia matricola $\rightarrow a=6$ e $b=7$

- LA CURVA ELLITTICA PRIMA SARÀ $\rightarrow E_{13}(6,7)$ CON $P=13$ // $\mathbb{Z}_P=13$

- SI SCRIVE LA CUBICA SECONDO LA FORMA NORMALE DI WIEFERSTRASS $\rightarrow y^2 \equiv (x^3 + ax + b) \pmod{P}$, cioè:

$$\rightarrow y^2 \equiv (x^3 + 6x + 7) \pmod{13} \quad (*)$$

- SI TROVANO I PUNTI NEL CAMPO \mathbb{Z}_P CHE HANNO RADICE NEL CAMPO, Detti RESIDUI QUADRATICI, CALCOLANDO:

$$\rightarrow y^2 \pmod{P}, \text{ cioè } \rightarrow y^2 \pmod{13}.$$

i VALORI ASSUNTI NEL CAMPO SONO IN $[0, P-1]$, QUINDI, y VA DA \odot A $P-1$: // $[0, 12]$

\rightarrow	y	0	1	2	3	4	5	6	7	8	9	10	11	12
	$y^2 \pmod{13}$	0	1	4	9	3	12	10	10	12	3	9	10	1

- i RESIDUI QUADRATICI SONO $\rightarrow 0, 1, 3, 4, 9, 10, 12$ // Si scrivono SOLO UNA VOLTA SENZA RIPETIZIONE

- PER COSTRUIRE LA CURVA, CONSIDERIAMO GLI \times DEL CAMPO \rightarrow cioè DA \odot A $P-1$, UNO ALLA VOLTA

E PROCEDIAMO COSÌ:

- SI PRENDE IL VALORE DI \times E SI SOSTITUISCE NELLA CUBICA $(*)$ // NEL 2° MEMBRO
- SI CONTROLLA SE IL VALORE OTTENUTO AL SECONDO MEMBRO NELLA CUBICA È UGUALE AD UNO DEI RESIDUI QUADRATICI // SVOLGENDO IL MODO
- SE LO È, PER QUEL VALORE DI \times CI SARANNO PUNTI DELLA CURVA CON QUELLA \times E CON CIASUNA SUA y ASSOCIATA (cioè il RESIDUO QUADRATICO)
- SE NON LO È, SI SCARTA LA \times E SI PROSEGUE CON LE SUCCESSIVE \times

COSTRUZIONE CURVA

$$x=0 \rightarrow y^2 \equiv (0^3 + 6 \cdot 0 + 7) \pmod{13} = 7 \rightarrow \text{NON È UN RESIDUO QUADRATICO}$$

$$x=1 \rightarrow y^2 \equiv (1^3 + 6 \cdot 1 + 7) \pmod{13} = 1 \rightarrow \text{È UN RESIDUO QUADRATICO CON } y=1 \text{ E } y=12 \text{ ASSOCIATE.}$$

i PUNTI DELLA CURVA SONO $\rightarrow (1,1)$ E $(1,12)$

$$x=2 \rightarrow y^2 \equiv (2^3 + 6 \cdot 2 + 7) \pmod{13} = 1 \rightarrow \text{È UN RESIDUO QUADRATICO CON } y=1 \text{ E } y=12 \text{ ASSOCIATE.}$$

i PUNTI DELLA CURVA SONO $\rightarrow (2,1)$ E $(2,12)$

$$x=3 \rightarrow y^2 \equiv (3^3 + 6 \cdot 3 + 7) \pmod{13} = 0 \rightarrow \text{È UN RESIDUO QUADRATICO CON } y=0 \text{ ASSOCIATA.}$$

L'UNICO PUNTO DELLA CURVA È $\rightarrow (3,0)$

$x=4 \rightarrow y^2 \equiv (4^3 + 6 \cdot 4 + 7) \pmod{13} = 4 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=2$ ASSOCIATA.

L'UNICO PUNTO DELLA CURVA È $\rightarrow (4, 2)$

$x=5 \rightarrow y^2 \equiv (5^3 + 6 \cdot 5 + 7) \pmod{13} = 6 \rightarrow$ NON È UN RESIDUO QUADRATICO

$x=6 \rightarrow y^2 \equiv (6^3 + 6 \cdot 6 + 7) \pmod{13} = 12 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=5$ E $y=8$ ASSOCIATE.

I PUNTI DELLA CURVA SONO $\rightarrow (6, 5)$ E $(6, 8)$

$x=7 \rightarrow y^2 \equiv (7^3 + 6 \cdot 7 + 7) \pmod{13} = 2 \rightarrow$ NON È UN RESIDUO QUADRATICO

$x=8 \rightarrow y^2 \equiv (8^3 + 6 \cdot 8 + 7) \pmod{13} = 8 \rightarrow$ NON È UN RESIDUO QUADRATICO

$x=9 \rightarrow y^2 \equiv (9^3 + 6 \cdot 9 + 7) \pmod{13} = 10 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=6$, $y=7$ E $y=11$ ASSOCIATE.

I PUNTI DELLA CURVA SONO $\rightarrow (9, 6)$, $(9, 7)$ E $(9, 11)$

$x=10 \rightarrow y^2 \equiv (10^3 + 6 \cdot 10 + 7) \pmod{13} = 1 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=1$ E $y=12$ ASSOCIATE.

I PUNTI DELLA CURVA SONO $\rightarrow (10, 1)$ E $(10, 12)$

$x=11 \rightarrow y^2 \equiv (11^3 + 6 \cdot 11 + 7) \pmod{13} = 0 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=0$ ASSOCIATA.

L'UNICO PUNTO DELLA CURVA È $\rightarrow (11, 0)$

$x=12 \rightarrow y^2 \equiv (12^3 + 6 \cdot 12 + 7) \pmod{13} = 0 \rightarrow$ È UN RESIDUO QUADRATICO CON $y=0$ ASSOCIATA.

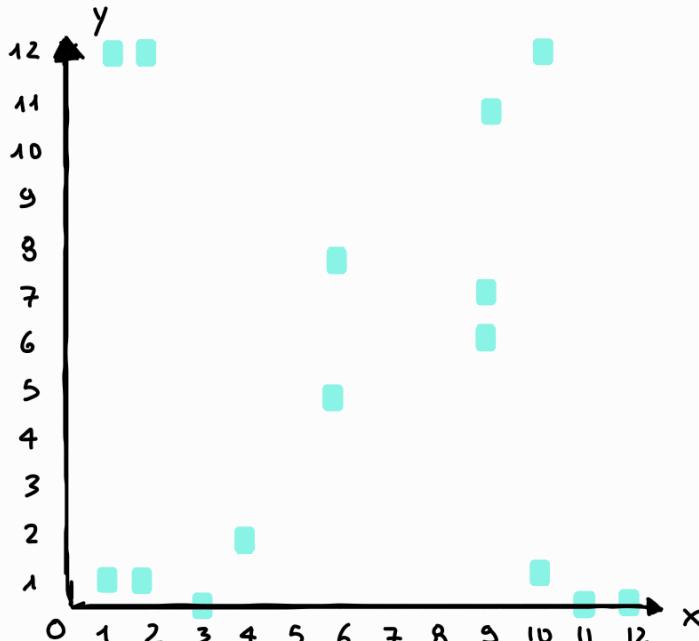
L'UNICO PUNTO DELLA CURVA È $\rightarrow (12, 0)$

- LA CURVA ELLITTICA SARÀ COSÌ COMPOSTA:

$$\rightarrow E_{13}(6, 7) = \{(1, 1), (1, 12), (2, 1), (2, 12), (3, 0), (4, 2), (6, 5), (6, 8), (9, 6), (9, 7), (9, 11), (10, 1), (10, 12), (11, 0), (12, 0)\} \cup \mathcal{O}$$

// \mathcal{O} È IL PUNTO ALL'INFINITO CHE VA SEMPRE AGGIUNTO

- SI PROCEDE COL DISEGNARE LA CURVA NEL QUADRANTE POSITIVO CON $x, y \in \mathbb{Z}_p$ // CHE VANNNO DA \mathcal{O} A 12



// I PUNTI DELLA CURVA SONO \rightarrow

② L'ORDINE DELLA CURVA È IL NUMERO TOTALE DI PUNTI CHE COMPOGGONO LA CURVA, COMPRENSO IL PUNTO ALL'INFINITO.

IN QUESTO CASO → 16

③ AFFINCHÉ UNA CURVA ELLITTICA POSSA DEFINIRE UN GRUPPO ABELIANO DEVE VALERE CHE:

$$\rightarrow (4a^3 + 27b^2) \bmod p \neq 0.$$

$$\text{IN QUESTO CASO ABBIANO} \rightarrow (4(6^3) + 27(7^2)) \bmod 13 = 3.$$

QUINDI, ESSENDO $\neq 0$ LA CURVA $E_{13}(6,7)$ DEFINISCE IL GRUPPO ABELIANO SU \mathbb{Z}_{13}

CURVE ELLITTICHE → Crittografia ellittica

Impiegando una curva ellittica $E_p(a,b)$ su un campo finito, sia k l'intero formato dalla prima e dalla terza cifra del proprio numero di matricola. Dato un punto P della curva, con quante operazioni (raddoppi e somme di punti) è possibile calcolare il punto $Q = kP$? **Mostrare** le operazioni che devono essere eseguite (senza calcolare i risultati).

- PER CALCOLARE $Q = kP$ DOBBIANO UTILIZZARE L'ALGORITMO DEI RADDOPPI RIPETUTI
- CONSIDERIAMO LE CIFRE DELLA MATRICOLA (1) E (5) CHE FORMANO $(K) \rightarrow K=15$
- QUINDI, VOGLIAMO CALCOLARE IL PUNTO $\rightarrow Q = 15P$, E PROCEDE REMO COSÌ:
 - Si scrive (K) COME SOMMA DI POTENZE DI $(2) \rightarrow 15 = 1 + 2 + 4 + 8$.
 - IL PUNTO DA CALCOLARE DIVENTA $\rightarrow Q = (1 + 2 + 4 + 8)P$
 - Si considerano i raddoppi fino alla potenza massima:
 1. DA (P) si calcola il raddoppio $\rightarrow 2(P) = 2P$
 2. DA $2P$ si calcola il raddoppio $\rightarrow 2(2P) = 4P$
 3. DA $4P$ si calcola il raddoppio $\rightarrow 2(4P) = 8P$.AHBIAMO RAGGIUNTO LA POTENZA MASSIMA FERMANDOCI A (3) RADDOPPI
 - IL PUNTO $Q = 15P$ si ottiene con (3) somme facendo $\rightarrow Q = P + 2P + 4P + 8P$
 - DALLE CONSIDERAZIONI PRECEDENTI SI HA CHE $Q = 15P$ SI FA CON (3) RADDOPPI E (3) SOMME

CURVE ELLITTICHE → Crittografia ellittica

Impiegando una curva ellittica $E_p(a, b)$ su un campo finito, spiegare cosa si intende per “logaritmo discreto” (se esiste) di un punto Q in base P .

- IL PROBLEMA DEL LOGARITMO DISCRETO PER LE CURVE ELLITTICHE È L'OPERAZIONE INVERSA DELLA

MOLTIPLICAZIONE SCALARE SU UNA CURVA ELLITTICA, cioè :

→ DATI DUE PUNTI (P) E (Q) TROVARE, SE ESISTE, IL PIÙ PICCOLO INTERO (k) TALE CHE $\rightarrow Q = kP$.

IL NUMERO (k) QUANDO È DEFINITO È $\rightarrow k = \log_P Q$

- QUESTO È UN PROBLEMA COMPUTAZIONALMENTE DIFFICILE.

L'UNICO METODO PER RISOLVERLO È IL BRUTE FORCE, IL QUALE SI BASA SUL CALCOLO DI TUTTI I MULTIPLI DI (P)

FINO A TROVARE (Q) // CON AD ESEMPIO L'ALGORITMO DEI RADOPPI RIPETUTI

- IL CALCOLO DEL LOGARITMO DISCRETO È UNA FUNZIONE DI TIPO ONE-WAY TRAP-DOOR.

SU TALE FUNZIONE SI BASA LA SICUREZZA DELLA CRYPTOGRAFIA SU CURVE ELLITTICHE

CURVE ELLITTICHE → Scambio di chiavi su curve ellittiche

Descrivere il protocollo DH su curve ellittiche per lo scambio di chiavi segrete e spiegare per quale motivo il protocollo può ritenersi sicuro.

- Si premette che Alice e Bob vogliono costruirsi insieme una chiave segreta di sessione ($K_{[SESSION]}$) da utilizzare per comunicazioni successive protette con un cifrario simmetrico

FUNZIONAMENTO PROTOCOLLO

- Alice e Bob scelgono una curva ellittica e un punto (B) della curva di ordine (m) molto grande.

L'ordine (m) del punto (B) è il più piccolo intero tale che $\rightarrow mB = O$. // $O \rightarrow$ PUNTO ALL'INFINITO
COME CURVA ELLITTICA POSSONO SCEGLIERE ANCHE UNA DELLE CURVE RACCOMANDATE DAL NIST.

Inoltre, il punto (B) corrisponde al generatore (g) del protocollo DH "standard"

- Ad Eve sono noti sia la curva ellittica che il punto (B)
- Alice estrae un intero positivo casuale $m_A < m$ come propria chiave privata
- Poi, tramite il calcolo dei raddoppi ripetuti genera la chiave pubblica $\rightarrow P_A = m_A B$ e la invia in chiaro a Bob.

Dunque, la chiave pubblica di Alice corrisponde ad un punto della curva ellittica scelta casualmente

- Analogamente, Bob estrae un intero positivo casuale $m_B < m$ come propria chiave privata
- Anche Bob, tramite il calcolo dei raddoppi ripetuti genera la chiave pubblica $\rightarrow P_B = m_B B$ e la invia in chiaro ad Alice.

Dunque, anche la chiave pubblica di Bob corrisponde ad un punto della curva ellittica scelta casualmente

- Alice riceve P_B e si calcola un punto S della curva usando la sua chiave privata m_A :
 $\rightarrow S = m_A P_B$, che equivale a fare $S = m_A m_B B$
- Bob riceve P_A e si calcola un punto S della curva usando la sua chiave privata m_B :
 $\rightarrow S = m_B P_A$, che equivale a fare $S = m_B m_A B$
- Abbiamo che sia Alice che Bob condividono lo stesso punto (S) della curva determinato dalle scelte casuali di entrambi $\rightarrow m_A m_B B = m_B m_A B$
- Per trasformare (S) in una chiave segreta ($K_{[SESSION]}$), da usare nella cifratura simmetrica convenzionale, considereremo la sua ascissa e i suoi ultimi 256 bit $\rightarrow K_{[SESSION]} = x_S \bmod 256$

- IN QUANTO A SICUREZZA DEL PROTOCOLLO, UN CRIPTODANALISTA CHE INTERCEDE $\underline{P_A}$ E $\underline{P_B}$ DAL CANALE PUBBLICO NON È IN GUADO DI VIOLARE LO SCHEMA
- ESSENDO CHE $\underline{P_A}$ E $\underline{P_B}$ SONO STATI SCAMBIATI IN CHIARO, PUR CONOSCENDO I PARAMETRI DELLA CURVA E IL PUNTO \underline{B} , CALCOLARE \underline{S} COMPORTA LA RISOLUZIONE DEL PROBLEMA DEL LOGARITMO DISCRETO PER LE CURVE ELIPTICHE, CIOÈ:
 - CALCOLARE $\underline{m_A}$ DA \underline{i} \underline{B} E $\underline{P_A}$ // CON $P_A = m_A B$
 - CALCOLARE $\underline{m_B}$ DA \underline{i} \underline{B} E $\underline{P_B}$ // CON $P_B = m_B B$
- IL PROTOCOLLO È COMUNQUE VULNERABILE AGLI ATTACCI ATTIVI DI TIPO HAN-IN-THE-MIDDLE

Impiegando una curva ellittica $E_p(a,b)$ su un campo finito, descrivere l'algoritmo di Koblitz per trasformare un messaggio m , codificato come numero intero, in un punto di una curva ellittica prima.

- SUPPONIAMO DI VOLER TRASFORMARE UN INTERO POSITIVO $m < p$ IN UN PUNTO DI UNA CURVA ELLITTICA

PRIMA $E_p(a,b) \rightarrow P_m \in E_p(a,b)$

ABBIANO CHE:

• L'INPUT È IL MESSAGGIO (m)

• L'OUTPUT È IL PUNTO SULLA CURVA ELLITTICA $\rightarrow P_m$

// QUESTA TECNICA SERVE AD AUMENTARE IL NUMERO POSSIBILE DI CASI ATTRIBUIBILI AD (m) .

ANZICHE' USARE (m) COME ASCISSA, CHE DA LUOGO AD UN SOLO CASO POSSIBILE, SI CERCANO DI UTILIZZARE VALORI PIÙ AMPI

- FISSATO UN INTERO POSITIVO h , TALE CHE $(m+i)h < p$, APPLICHIAMO L'ALGORITMO DI KOBBLITZ:

```

for (i=0; i<h; i++) {
    x = mh + i;
    z = (x3 + ax + b) mod p;
    if (z È UN RESIDUO QUADRATICO) {
        y = √z;
        return P_m(x,y);
    }
}
return "FAILURE";

```

//1
//2
//3
//4

1- POTENZIARE VALORE DELL'ASCISSA CHE VARIA AL VARIARE DI i

2- ESTRAZIONE DELLA RADICE IN TEMPO POLINOMIALE SE P È PRIMO

3- L'ESISTENZA DELLA RADICE COMPORTA IL FATTO DI PRENDERE $P_m = (x, y)$ COME PUNTO SULLA CURVA, CHE È CORRISPONDENTE AD m

4- CASO IN CUI NON È STATO RESTITUITO UN PUNTO.

QUINDI, NON È STATO POSSIBILE TRASFORMARE IL MESSAGGIO IN UN PUNTO DELLA CURVA



- Poiché per \textcircled{h} volte, il risultato della valutazione della curva deve darci un numero che non è un residuo quadratico (che avviene con probabilità $\frac{1}{2}$), allora abbiamo:
- PROBABILITÀ DI FALLIMENTO $\simeq \left(\frac{1}{2}\right)^h$
 - PROBABILITÀ DI SUCCESSO $\simeq 1 - \left(\frac{1}{2}\right)^h$
- // PER \textcircled{h} GRANDE ci saranno tanti tentativi di mappatura sulla curva.

QUINDI, SARÀ MAGGIORE LA PROBABILITÀ DI INCAPSULARE IL MESSAGGIO IN UN PUNTO DELLA CURVA

- IL DESTINATARIO DEVE RISALIRE AD \textcircled{m} , però nel punto P_m che codifica il messaggio c'è \textcircled{x} .
- QUINDI, DOBBIATO fare $\rightarrow m = \lfloor \frac{x}{h} \rfloor$
- Poiché $x = mh + i$, per $i \in [0, h-1]$, avremo che il messaggio sarà dato da $\rightarrow m = \lfloor \frac{x}{h} \rfloor = \lfloor \frac{mh+i}{h} \rfloor$
- FACENDO ALCUNE MANIPOLAZIONI, ABBIATO $\rightarrow m = \lfloor \frac{mh+i}{h} \rfloor = \lfloor m + \frac{i}{h} \rfloor$
- Poiché $\frac{i}{h}$ NON RAGGIUNGE MAI L'UNITÀ, cioè può essere $\textcircled{0}$ oppure un numero ≤ 1 , allora si escludono
- OTTENIAMO $\rightarrow m = \lfloor m \rfloor$

KOBLITZ, SCAMBIO MESSAGGI, ELGAMAL → Curve ellittiche

Sviluppare un esempio di applicazione dell'algoritmo di Koblitz per trasformare il messaggio m corrispondente alla cifra meno significativa del proprio numero di matricola in un punto della curva ellittica $E_{23}(1,1)$. Se $m > 5$, si ponga $m = 5$. Se $m < 3$, si ponga $m = 4$. Si assegna ad h il valore 3.

- COMINCIANO A CONSIDERARE LA CIFRA MENO SIGNIFICATIVA DEL MIO NUMERO DI MATRICOLA $\rightarrow 7$

- POICHÉ $7 > 5$ SI PONE $\underline{m=5}$

- CONSIDERANDO LA CURVA $E_{23}(1,1)$ E $\underline{h=3}$, SI FA LA SIMULAZIONE DELL'ALGORITMO DI KOBLITZ:

- INIZIANO A VERIFICARE LA CONSISTENZA DI (h) , CONTROLLANDO SE VALE $\rightarrow (m+1)h < p$.

IN QUESTO CASO LA CONSISTENZA È VERIFICATA, PERCHÉ $\underline{m+1} \cdot \underline{h} < p \rightarrow (5+1) \cdot 3 < 23$

- SCRIVIAMO LA CURVA SECONDO LA FORMA NORMALE DI WEIERSTRASS $\rightarrow y^2 \equiv (x^3 + ax + b) \pmod{p}$.

IN QUESTO CASO, POICHÉ $a=1$ E $b=1$, ABBIANO $\rightarrow y^2 \equiv (x^3 + x + 1) \pmod{23}$

- TROVIAMO I RESIDUI QUADRATICI, CIOÈ QUEI PUNTI NEL CAMPO \mathbb{Z}_p CHE HANNO RADICE NEL CAMPO.

DA (1) SI CALCOLA $y^2 \pmod{p}$, CIOÈ $\rightarrow y^2 \pmod{23}$.

I VALORI ASSUNTI NEL CAMPO SONO IN $[0, p-1]$, CIOÈ IN $\rightarrow [0, 22]$

\rightarrow	y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	$y^2 \pmod{23}$	0	1	4	9	16	2	13	3	18	12	8	6	6	8	12	18	3	13	2	16	9	4	1

OSS (DA $\frac{p-1}{2}$ IN Poi, CIOÈ DA (11), C'È UNA SIMMETRIA SUI VALORI DEI RESIDUI QUADRATICI)

- I VALORI ASSUNTI NEL CAMPO SONO $\rightarrow 0, 1, 2, 3, 4, 6, 8, 9, 11, 13, 16, 18$ //SENZA RIPETIZIONI

- SI SVILUCA L'ALGORITMO PER $i \in [0, h-1]$, QUINDI, ABBIANO CHE $\rightarrow 0 \leq i \leq 2$:

$i=0 \rightarrow$ SI CALCOLA $x = mh + i \rightarrow 5 \cdot 3 + 0 = 15$.

SI VERIFICA SE ABBIANO UN RESIDUO QUADRATICO METTENDO NELLA CUBICA, $(x^3 + x + 1) \pmod{23}$, IL

VALORE \rightarrow APPENA TROVATO $\rightarrow (15^3 + 15 + 1) \pmod{23} = 10$, CHE NON È UN RESIDUO QUADRATICO

$i=1 \rightarrow$ SI CALCOLA $x = mh + i \rightarrow 5 \cdot 3 + 1 = 16$.

SI VERIFICA SE ABBIANO UN RESIDUO QUADRATICO METTENDO NELLA CUBICA, $(x^3 + x + 1) \pmod{23}$, IL

VALORE \rightarrow APPENA TROVATO $\rightarrow (16^3 + 16 + 1) \pmod{23} = 19$, CHE NON È UN RESIDUO QUADRATICO

$i=2 \rightarrow$ SI CALCOLA $x = mh + i \rightarrow 5 \cdot 3 + 2 = 17$.

SI VERIFICA SE ABBIANO UN RESIDUO QUADRATICO METTENDO NELLA CUBICA, $(x^3 + x + 1) \pmod{23}$, IL

VALORE \rightarrow APPENA TROVATO $\rightarrow (17^3 + 17 + 1) \pmod{23} = 9$, CHE È UN RESIDUO QUADRATICO SI:

$\rightarrow y=20$ E $y=3$

• Si procede con la mappatura del messaggio in un punto della curva ellittica $\rightarrow m \rightarrow P_m = (x, y)$.

Per la mappatura si considerano le \otimes dei residui quadratici trovati, con annesso (4).

In questo esercizio, possiamo mappare il messaggio in due punti diversi della curva ellittica:

• $m \rightarrow P_m = (17, 3)$

• $m \rightarrow P_m = (17, 20)$

KOBLITZ, SCAMBIO MESSAGGI, ELGAMAL → Crittografia ellittica

Impiegando una curva ellittica $E_p(a, b)$ su un campo finito, **descrivere** un algoritmo per lo scambio di messaggi cifrati su curve ellittiche, **dimostrarne** la correttezza e **discuterne** la sicurezza.

- UN ALGORITMO PER LO SCAMBIO DI MESSAGGI CIFRATI SU CURVE ELLITTICHE È IL:
 - CIFRARIO DI ELGAMAL, CHE FA USO DELLA CHIAVE PUBBLICA
- FISSATA UNA CURVA $E_p(a, b)$ SI SCEGLIE UN PUNTO B DELLA CURVA DI ORDINE m ELEVATO
- OGNI UTENTE U , CHE NUOGLÉ COMUNICARE, SI GENERA LA COPPIA (CHIAVE PUBBLICA, CHIAVE PRIVATA) COSÌ:
 - PER $K[\text{priv}]$ SI SCEGLIE UN INTERO CASUALE $\rightarrow m_0 < m$
 - PER $K[\text{pub}]$, CONSIDERANDO m_0 , SI OTTIENE FACENDO $\rightarrow P_U = m_0 B$
- SI SISUOLANO LE FASI DI CIFRATURA E DECIFRAZIONE NEL CASO IN CUI IL MITTENTE, ALICE, NUOGLÉ MANDARE UN MESSAGGIO (m) AL DESTINATARIO, BOB

FASE DI CIFRATURA

- ALICE TRASFORMA IL MESSAGGIO (m) NEL PUNTO P_m SULLA CURVA
// PER ESEMPIO, CON L'ALGORITMO DI KOBLITZ, USANDO $P_m = (x, y) \in h$
- Poi, ALICE, SCEGLIE UN INTERO CASUALE τ E SI CALCOLA I SEGUENTI PUNTI:
 - $V = \tau B$
 - $W = P_m + \tau P_{BOB}$, Dove P_{BOB} È LA CHIAVE PUBBLICA DI BOB $\rightarrow m_{BOB} B$
// $P_m = (x, y)$ È IL PUNTO SULLA CURVA
- OSS (τ SERVE A MASCHERARE IL MESSAGGIO CON UN INTERO A CASO SULLA CURVA.
QUESTO "MASCHERAMENTO" SI FA SU (W) IN MODO TALE CHE τ SIA ESTRAISIBILE SOLO DA (V))
- INFINE, ALICE, INVIA A BOB LA COPPIA $\rightarrow \langle V, W \rangle$

FASE DI DECIFRAZIONE

- BOB, DOPO ANER RICEVUTO $\langle V, W \rangle$ DA ALICE, PROVA A RI COSTRUIRE P_m CON LA SUA CHIAVE PRIVATA:
 - m_{BOB} , ESEGUENDO I SEGUENTI PASSAGGI:
 - CALCOLA $\rightarrow W - m_{BOB} V$
 - SCRIVE (W) IN MODO ESTESO $\rightarrow (P_m + \tau P_{BOB}) - m_{BOB} V$
 - SCRIVE ANCHE P_{BOB} E (V) IN MODO ESTESO $\rightarrow (P_m + \tau m_{BOB} B) - \tau m_{BOB} B$
 - DOPO ALCUNE SEMPLIFICAZIONI RIMANE $\rightarrow P_m$
 - DA P_m OTTIENE IL MESSAGGIO FACENDO COSÌ $\rightarrow m = \lfloor \frac{x}{h} \rfloor$

- L'OTTENIMENTO DEL MESSAGGIO m DALLA FASE DI DECIFRAZIONE, CI HA PERMESSO DI VERIFICARE LA CORRETTEZZA DEL CIFRARIO
- IN QUANTO A SICUREZZA DEL CIFRARIO, IL CRITTOANALISTA (EVE) POTREBBE PROVARE A DECIFRARE SECONDO DUE MODALITÀ:

1) SE TROVA r DECIFRA FACENDO $\rightarrow w - r p_{Bob}$, E SCRITTO PER ESTESO È:

$$\rightarrow (p_m - r p_{Bob}) - r p_{Bob} = p_m.$$

QUINDI, HA TOLTO LA MASCHERA CASUALE TROVANDO IL PUNTO SULLA CURVA

OSS (SE INVECE, DEVE TROVARE r DA $\textcircled{1}$ E $\textcircled{2}$ SU $v = rB$ ALLORA DEVE RISOLVERE IL PROBLEMA
DEL LOGARITMO DISCRETO SU CURVE ELIOTTICHE)

2) PROVA A TROVARE LA CHIAVE PRIVATA DEL DESTINATARIO, Bob , IN MODO DA POTER DECIFRARE

COME LUI

- PER TROVARE M_{Bob} , DA P_{Bob} E $\textcircled{2}$ SU $P_{Bob} = M_{Bob}B$, IL CRITTOANALISTA DEVE RISOLVERE IL PROBLEMA

DEL LOGARITMO DISCRETO SU CURVE ELIOTTICHE.

CIOÈ È DONATO AL FATTO CHE LA CHIAVE PRIVATA È PROTETTA

- QUESTO METODO COMPORTA, QUINDI, TEMPO ESPOENZIALE NEL SUO URIBITTO // UTILE PER LA SICUREZZA

OSS (IL NUMERO r CASUALE È DI TIPO ONE-TIME, CIOÈ SI USA UNA SOLA VOLTA E Poi SI BUTTA VIA)

HASH, FIRMA DIGITALE → Funzioni hash e autenticazione di messaggi

Spiegare che proprietà devono possedere le funzioni *hash one-way* e perché tali funzioni sono importanti nei protocolli di autenticazione e firma digitale.

- DATA UNA FUNZIONE HASH $f: X \rightarrow Y$, QUANDO VIENE APPLICATA IN CRIPTOGRAFIA DEVE RISPETTARE LE SEGUENTI PROPRIETÀ:

- 1) PER OGNI $x \in X$ È COMPUTAZIONALMENTE FACILE CALCOLARE L'IMMAGINE HASH $\rightarrow y = f(x)$
- 2) DATA UN'IMMAGINE HASH $y \in Y$, È COMPUTAZIONALMENTE DIFFICILE RISALIRE AD UN ELEMENTO DEL DOMINIO, $x \in X$, TALE CHE $\rightarrow f(x) = y$.
// CIOÈ QUEL'ELEMENTO (x) CHE ABBAIA PROPRIO QUELL'IMMAGINE
OSS (IN MANIERA ALTERNATIVA SI PUÒ SCRIVERE COME $\rightarrow x = f^{-1}(y)$)

QUESTA PROPRIETÀ È CHIAMATA ONE-WAY

- 3) È COMPUTAZIONALMENTE DIFFICILE DETERMINARE UNA COPPIA DI ELEMENTI $x_1, x_2 \in X$ TALI CHE:
 $\rightarrow f(x_1) = f(x_2)$.

CIOÈ RAPPRESENTA IL FATTO CHE ELEMENTI DIVERSI FRA LORO COLLIDONO, PERCHÉ HANNO LA STESSA IMMAGINE HASH.

QUESTA PROPRIETÀ È CHIAMATA CLAW-FREE

- L'IMPORTANZA DI QUESTE PROPRIETÀ SI PUÒ RAPPRESENTARE NEL MODO IN CUI ESSE VENGONO UTILIZZATE NEI PROTOCOLLI, CIOÈ:

- NEL PROTOCOLLO DI AUTENTICAZIONE, LA FUNZIONE È USATA NEL CALCOLO DELLA PRODUZIONE DEL MAC, CHE CONTIENE LA CHIAVE SEGRETA K LEGATA AL MESSAGGIO m .
PER RECUPERARE K SI DOVREBBE INVERTIRE LA FUNZIONE HASH, IL CHE RICHIEDEREbbe TEMPO ESPONENZIALE
- NEL PROTOCOLLO DI FIRMA DIGITALE, LA VERIFICA DELLA FIRMA PRODUCE L'IMMAGINE HASH, DALLA QUALE NON È POSSIBILE RICOSTRUIRE IL MESSAGGIO

Descrivere un protocollo di **identificazione** che utilizza il cifrario RSA.

- CONSIDERIAMO UN PROTOCOLLO DI IDENTIFICAZIONE BASATO SU RSA,
- COME PRIMA COSA PRENDIAMO $\langle r, m \rangle$ E $\langle d \rangle$ LE CHIAVI, PUBBLICA E PRIVATA, DI UN UTENTE (U) CHE VUOLE ACCEDERE AI SERVIZI DI UN SISTEMA (S)
- IL PROTOCOLLO PRESENTA I SEGUENTI PASSI:
 - IN CORRISPONDENZA AD UNA RICHIESTA DI ACCESSO PROVENIENTE DA (U) , IL SISTEMA GENERA UN NUMERO CASUALE $\rightarrow r \in \mathbb{N}$.
TALE NUMERO, LO INVIA IN CHIARO AD (U)
 - (U) DECIFRA IL MESSAGGIO (r) CALCOLANDO $\rightarrow f^d = r^d \bmod m$, CON d SUA CHIAVE PRIVATA.
Poi, LO SPEDISCE AD (S)
 - (S) VERIFICA LA CORRETTEZZA DEL VALORE CIFRANDOLO CON LA CHIAVE PUBBLICA DI (U) , $\langle r, m \rangle$, E CONTROLLANDO SE IL RISULTATO COINCIDA CON $(r) \rightarrow f^e \bmod m = r$
//DA NOTARE L'INVERSIONE DELLE OPERAZIONI DI CIFFRATURA E DECIFRAZIONE, RISPETTO ALL'IMPIEGO STANDARD DELL'RSA
 - NEL CASO IN CUI L'ULTIMO PUNTO VADA A BUON FINE, IL SISTEMA GARANTISCE CHE L'UTENTE CHE HA MA RICHIESTO L'IDENTIFICAZIONE SIA PROPRIO (U) .
Ciò avviene anche se il canale è insicuro

HASH, FIRMA DIGITALE → Funzioni hash

Descrivere un protocollo di firma digitale.

- CONSIDERIAMO UN PROTOCOLLO DI FIRMA DIGITALE, IN CUI ABBIANO UN MITTENTE \textcircled{U} CHE NUOLE INVIARE UN DOCUMENTO FIRMATO AL DESTINATARIO \textcircled{V} .

L'INVIO DEL DOCUMENTO DEVE SVOLGERSI CON RISERVATEZZA DURANTE LA COMUNICAZIONE

- CONSIDERIAMO PER QUESTO PROTOCOLLO LE FASI DEL MITTENTE E DEL DESTINATARIO

FASE DI U

- GENERA LA FIRMA PER IL MESSAGGIO \textcircled{m} FACENDO $\rightarrow f = D(m, K_u[\text{priv}])$
- CALCOLA IL CRITTOGRAMMA CON LA CHIAVE PUBBLICA DI \textcircled{V} FACENDO $\rightarrow c = C(f, K_v[\text{pub}])$

// FACENDO LA CIFRATURA DELLA FIRMA È COME SE SI METTESSE LA FIRMA AD UN DOCUMENTO E LO SI CHIUDESSE IN UNA BUSTA, LA QUALE, È PRONTA PER ESSERE SPEDITA

- INVIA A \textcircled{V} LA COPPIA $\langle \textcircled{U}, c \rangle$, DOVE \textcircled{U} È IL SUO IDENTIFICATIVO COME MITTENTE DEL MESSAGGIO

FASE DI V

- \textcircled{V} , DOPO ANER RICEVUTO LA COPPIA $\langle \textcircled{U}, c \rangle$, DECIFRA IL CRITTOGRAMMA FACENDO $\rightarrow D(c, K_v[\text{priv}]) = f$
- CIFRA IL VALORE OTTENUTO CON LA CHIAVE PUBBLICA DI \textcircled{U} $\rightarrow C(f, K_u[\text{pub}])$, E SOLITO PER ESTESO VIENE:
 $\rightarrow C(D(m, K_u[\text{priv}]), K_u[\text{pub}]) = m$, E OTTIENE IL MESSAGGIO \textcircled{m}
- DOPO LA RICOSTRUZIONE DI \textcircled{m} , SE \textcircled{m} È SIGNIFICATIVO, ALLORA \textcircled{V} ATTESTA L'IDENTITÀ DI \textcircled{U} .

LO IDENTIFICA COME MITTENTE E ACCETTA IL DOCUMENTO FIRMATO

OSS (QUESTO PROTOCOLLO È POSSIBILE POICHÉ OGNI UTENTE HA LA PROPRIA COPPIA $(K[\text{priv}], K[\text{pub}])$)

HASH, FIRMA DIGITALE → RSA, Complessità in algebra

Descrivere un protocollo di **firma digitale** che utilizza il cifrario RSA. Inoltre, data la funzione di Eulero $\Phi(n)$, spiegare in termini matematici quale implicazione avrebbe questo algoritmo sulla firma digitale.

- CONSIDERIAMO (U) E (V) , cioè MITTENTE E DESTINATARIO, CHE UTILIZZANO IL CIFRARIO RSA SUL PROTOCOLLO DI FIRMA DIGITALE, PER LA TRASMISSIONE DI UN MESSAGGIO (m)
- NEL PROTOCOLLO SI USANO LE SEGUENTI CHIAVI, UTILI NELLA CIFRATURA RSA:
 - $\langle d_u, e_u, m_u \rangle$ COME CHIAVI, PRIVATA E PUBBLICA, DI (U)
 - $\langle d_v, e_v, m_v \rangle$ COME CHIAVI, PRIVATA E PUBBLICA, DI (V)

FASE DI U

- GENERA LA FIRMA DEL MESSAGGIO (m) FACENDO $\rightarrow d_u = m^{d_u} \text{ mod } m_u$
- CIFRA (f) CON LA CHIAVE PUBBLICA DI (V) FACENDO $\rightarrow c = f^{e_v} \text{ mod } m_v$
- SPEDISCE A (V) LA COPPIA $\langle U, c \rangle$, DOVE (U) È IL SUO IDENTIFICATIVO COME MITTENTE DEL MESSAGGIO.

FASE DI V

- DOPO ANER RICEVUTO LA COPPIA $\langle U, c \rangle$, DECIFRA c COSÌ $\rightarrow c^{d_v} \text{ mod } m_v = f$
- VERIFICA LA FIRMA "DECIFRANDO" (f) , CON LA CHIAVE PUBBLICA DI $(U) \rightarrow f^{e_u} \text{ mod } m_u = m$
- SE (m) È SIGNIFICATIVO, ALLORA CONCLUDE CHE TALE MESSAGGIO È AUTENTICO
- PER LA CORRETEZZA DEL PROCEDIMENTO È NECESSARIO CHE $\rightarrow m_u \leq m_v$.

- CIOÈ COMPORTA $\rightarrow d < m$, QUINDI, (f) PUÒ ESSERE CIFRATA E SPEDITA CORRETTAMENTE A (V)
- OGNI UTENTE DOVRÀ STABILIRE CHIAVI DISTINTE PER LA FIRMA E PER LA CIFRATURA: VERRÀ FISSATO PUBBLICAMENTE UN VALORE H MOLTO GRANDE (es. $H = 1024$ bit) E VERRANNO IMPOSTE LE CONDIZIONI:
 - CHIAVI DI FIRMA $< H$
 - CHIAVI DI CIFRATURA $> H$.

TUTTO CIÒ, SERVE A RENDERE GLI ATTACCHI ESURIENTI MOLTO COSTOSI.

- AVERE LA FUNZIONE DI EULERO $\phi(m)$ CI PERMETTE DI OTTENERE LA CHIAVE PRIVATA, AD ESEMPIO,
DI $(U) \rightarrow \langle d_u \rangle$.

TALE CHIAVE VIENE APPLICATA AL MESSAGGIO, PERMETTE DI FARE LA FIRMA "FALSIFICATA" IN TEMPO POLINOMIALE $\rightarrow d' = m^{d_u} \text{ mod } m_u$ // si indica con d' poiché è UNA FIRMA FALSA

HASH, FIRMA DIGITALE → Funzioni hash e autenticazione di messaggi

Descrivere un protocollo di autenticazione che utilizza funzioni hash crittografiche e discuterne la sicurezza.

- DURANTE LO SCAMBIO DI MESSAGGI, TRA UN MITTENTE (MITT) E UN DESTINATARIO (DEST), TROVIAMO IL PROCESSO DI AUTENTICAZIONE.

- IN QUESTA PROCEDURA ABBIANO CHE DEST DEVE AUTENTICARE IL MESSAGGIO (m) , FACENDO:

- L'ACCERTAMENTO DELL'IDENTITÀ DI MITT
- LA VERIFICA DELL'INTEGRITÀ DEL MESSAGGIO (m)

- IL PROTOCOLLO DI AUTENTICAZIONE PRENDE UNA FASE DI MITT E UNA DI DEST.

PRIMA DI QUESTE DUE FASI MITT E DEST CONCORDANO UNA CHIAVE SEGRETA SIMMETTRICA $\rightarrow K$
FASE DI MITT

- PRENDE (m) E GLI ALLEGGA UN MAC (MESSAGE AUTHENTICATION CODE) $\rightarrow A(m, K)$, ALLO SCOPO DI GARANTIRE LA PROVENIENZA E L'INTEGRITÀ DEL MESSAGGIO

- SPEDISCE IN CHIARO LA COPPIA $\rightarrow \langle m, A(m, K) \rangle$

FASE DI DEST

- ENTRA IN POSSESSO DI (m)

- POICHÉ È A CONOSCENZA DI (A) E (K) , CALCOLA $\rightarrow A(m, K)$ //PER CONTO SUO

- CONFRONTA IL VALORE OTTENUTO CON QUELLO INVIAZIO DA MITT PER VERIFICARE CHE IL MAC RICEVUTO CORRISPONDA AL MESSAGGIO A CUI RISULTA ALLEGATO:

- IN CASO DI VERIFICA CON SUCCESSO, IL MESSAGGIO È AUTENTICATO
- IN CASO DI VERIFICA SENZA SUCCESSO, DEST SCARTA IL MESSAGGIO.

- LA SICUREZZA DI QUESTO TIPO DI AUTENTICAZIONE SI BASA SUL MAC, CHE È UN'IMMAGINE BREVE BREVE DEL MESSAGGIO, LA QUALE VIENE GENERATA SOLO DA UN MITTENTE NOTO AL DESTINATARIO

- IL MAC PUÒ ESSERE REALIZZATO CON FUNZIONE HASH ONE WAY, FATTA COSÌ $\rightarrow A(m, K) = h(m, K)$

- QUINDI, RISULTA COMPUTAZIONALMENTE DIFFICILE PER UN CRITTOANALISTA SCOPRIRE LA CHIAVE SEGRETA (K) .

(h) È NOTA A TUTTI, E (m) PUÒ VIAGGIARE IN CHIARO O ESSERE SCOPERTO PER ALTRA VIA.

INVECE, (K) VIAGGIA ALL'INTERNO DEL MAC, QUINDI PER TROVARLO SI DOVREBBE INVERTIRE h
//ESPOENZIALE

HASH, FIRMA DIGITALE → Firma digitale

Sia S la somma delle sei cifre decimali del proprio numero di matricola. Si ponga $M = S + 20$. Si convertano le cifre di M in binario su 4 bit, se ne calcoli lo XOR bit a bit e si riconverto il valore ottenuto in un numero decimale H che sarà utilizzato come hash di M .

Per due utenti Alice e Bob di un sistema RSA si considerino i seguenti insiemi di parametri.

Alice: $p = 5, q = 11, e = 7, d = 23$.

Bob: $p = 7, q = 13, e = 5, d = 29$.

Alice deve spedire a Bob il messaggio M cifrato e firmato in hash, impiegando le chiavi RSA e la funzione hash di cui sopra.

Eseguire esplicitamente tutte le operazioni aritmetiche eseguite da Alice e da Bob nella trasmissione e verifica del messaggio M e della firma.

- CONSIDERO LA SOMMA DELLE SEI CIFRE DEL MIO NUMERO DI MATRICOLA (SUPpongo 146892):

$$\rightarrow S = 31 // \underline{31 = 1+4+6+8+5+2}$$

$$- PONIAMO M = S + 20 \text{ E CIo} \rightarrow M = 51 // \underline{51 = 31 + 20}$$

- SI PROCEDE CON LA SIMULAZIONE DEL 3° Protocollo di FIRMA DIGITALE (in HASH).

TALE PROTOCOLLO È UTILIZZATO PER LA TRASMISSIONE E VERIFICA DI (M) , E DELLA FIRMA

- INOLTRE, CALCOLIAMO PER IL MITTENTE (Alice) E PER IL DESTINATARIO (Bob) IL VALORE DEL SEMIPRIMO (m):

$$\bullet M_A = p_A q_A, \text{cio} \rightarrow M_A = 55 // \underline{55 = 5 \cdot 11}$$

$$\bullet M_B = p_B q_B, \text{cio} \rightarrow M_B = 91 // \underline{91 = 7 \cdot 13}$$

FASE DI ALICE

• SI INIZIA A COSTRUIRE L'HASH PRENDENDO LE CIFRE DI (M) SU 4 bit E FACENDONE LO XOR bit a bit, PER CIASUNA DELLE SINGOLE CIFRE:

$$\rightarrow \underline{S = 0101} - \underline{1 = 0001}$$

$$\rightarrow \underline{0101} \oplus \underline{0001} = \underline{0100}$$

OSS LA TABELLA DI VERIFICA DELL'XOR bit a bit È LA SEGUENTE

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

• SI CONCLUDE CONVERTENDO IN BASE-10 IL RISULTATO OTTENUTO $\rightarrow (0100)_2 = (4)_{10} \rightarrow H = 4$

- Poi, Alice, genera la firma con la sua chiave privata ($d_A=3$) e l'hash ($H=4$), facendo:
 $\rightarrow f = H^{d_A} \bmod M_A$, e cioè $\rightarrow f = 4^3 \bmod 55 \rightarrow f = 9$
- Cifra (M) con la chiave pubblica di Bob ($e_B=5$), facendo $\rightarrow c = M^{e_B} \bmod M_B$, e cioè:
 $\rightarrow c = 51^5 \bmod 91 \rightarrow c = 25$
- SPEDISCE A BOB LA TRIPPA $\rightarrow < A, c, d >$, Dove A è il suo identificativo come mittente del messaggio:
 $\rightarrow < A, 25, 3 >$

FASE DI BOB

- Dopo aver ricevuto la tripla $< A, 25, 3 >$, Bob, decifra il crittogramma ($c=25$) con la sua chiave privata ($d_B=25$), facendo $\rightarrow M = c^{d_B} \bmod M_B$, e cioè $\rightarrow 25^{25} \bmod 91 = 51$
- Anche lui, si calcola l'hash prendendo le cifre di (M) su 4 bit e facendone lo xor bit a bit, per ciascuna delle singole cifre:
 $\rightarrow S = 0101 \quad A = 0001$
 $\rightarrow 0101 \oplus 0001 = 0100$
- Convertendo in base-10 il risultato ottenuto, trova l'hash $\rightarrow (0100)_2 = (4)_{10} \rightarrow H = 4$
- Poi, fa la verifica della firma con la chiave pubblica di Alice ($e_A=7$), facendo $\rightarrow f^{e_A} \bmod M_A$:
 $\rightarrow 9^7 \bmod 55 = 4$
- Per autenticare il messaggio si verifica se vale l'uguaglianza $\rightarrow H = f^{e_A} \bmod M_A$, e cioè:
 $\rightarrow 4 = 4$, e il messaggio è autenticato e viene accettato

HASH, FIRMA DIGITALE → Certificato digitale

Descrivere l'utilizzo e l'importanza del certificato digitale.

- CONSIDERANDO LE CERTIFICATION AUTHORITY (CA), QUESTE GARANTISCONO LA VALIDITÀ DELLE K_{pub} E NE REGOLANO L'USO, GESTENDO LA DISTRIBUZIONE SICURA DELLE CHIAVI TRA SUE ENTITÀ CHE VOGLIONO COMUNICARE
- LA CA AUTENTICA L'ASSOCIAZIONE <UTENTE, K_{pub}> EMETTENDO UN → CERTIFICATO DIGITALE
- QUINDI, IL CERTIFICATO DIGITALE PER UN UTENTE COMPRENDE:
 - NUMERO DI VERSIONE
 - NOME CA CHE LO HA RILASCIATO
 - SERIALE DI IDENTIFICAZIONE PER LA CA EMITTENTE
 - ALGORITMO DELLA CA PER LA CREAZIONE DELLA FIRMA ELETTRONICA
 - PERIODO DI VALIDITÀ DEL CERTIFICATO
 - NOME UTENTE A CUI IL CERTIFICATO SI RIFERISCE E ALTRE SUE INFORMAZIONI
 - INDICAZIONE PROTOCOLLO A K_{pub} USATO DA QUESTO UTENTE PER CIFRATURA E FIRMA, CIÒÈ:
 - NOME ALGORITMO, PARAMETRI, K_{pub} DI QUESTO UTENTE
 - FIRMA DELLA CA SU TUTTE LE INFORMAZIONI PRECEDENTI

FASE DI COMUNICAZIONE TRA UN UTENTE (U) CHE VOGLIE COMUNICARE CON UN UTENTE (V) :

- (U) RICHIEDE K_V[pub] ALLA CA
- CA RISPONDE INVIANDOGLI IL CERTIFICATO DIGITALE DI (V) → CERT_V
- POICHÉ (U) CONOSCE K_{CA}[pub] → CONTROLLO L'AUTENTICITÀ DEL CERTIFICATO VERIFICANDONE:
 - PERIODO DI VALIDITÀ
 - FIRMA PRODOTTA DALLA CA SU DI ESSO
- SE TUTTI I CONTROLLI VANNO A BUON FINE → K_V[pub] NEL CERTIFICATO È CORRETTA E (U) AVVIA LA COMUNICAZIONE CON (V)

OSS / UN CRITTOANALISTA SI PUÒ INTROMETTERE SOLO FALSIFICANDO LA CERTIFICAZIONE, MA SI ASSUME
che la CA sia affidata e che abbia un archivio di chiavi inattaccabile

ZERO KNOWLEDGE → Identificazione

Presentare il protocollo di identificazione di utente per canali insicuri, basato su un sistema Zero Knowledge.

- NEL PROTOCOLLO DI IDENTIFICAZIONE DI FIAT-SHAMIR, ABBIAMO UN PROVER P CHE DEMOSTRA AD UN VERIFIER V LA SUA IDENTITÀ SENZA SVELARE NESSUN'ALTRA INFORMAZIONE
 - IL PROTOCOLLO È BASATO SULLA DIFFICOLTÀ DI INVERTIRE LA FUNZIONE POTENZA NELL'ALGEBRA MODULARE, CIÒ È IL CALCOLO DELLA RADICE IN MODULO
 - IN QUESTO CASO, TRATTIAMO LA RADICE QUADRATA DELL'EQUAZIONE $\rightarrow T = S^2 \text{ mod } m$, CON M PRIMO.
È POLINOMIALMENTE FACILE CALCOLARE T DATI S ED M.
INVECE, È ESPONENZIALMENTE DIFFICILE CALCOLARE S (RADICE QUADRATA DI T) DATI T ED M
SE NON SI CONOSCE LA FATTORIZZAZIONE DI M
 - NEL PROTOCOLLO ABBIAMO CHE
 - V CONOSCE T ED S
 - P DEVE CONVINCERE V DI CONOSCERE S
 - PER IL PROTOCOLLO, ABBIAMO UNA FASE DI PREPARAZIONE DI P, IN CUI SCEGLIE:
 - UN VALORE M = PQ, CON P E Q NUMERI PRIMI
 - UN INTERO POSITIVO SCM
 - SUCCESSIVAMENTE SI CALCOLA $\rightarrow T = S^2 \text{ mod } m$, CHE INSIEME AD M, FORMANO UNA SPECIE DI "CHIAVE PUBBLICA" DA RENDERE NOTA $\rightarrow \langle m, T \rangle$
 - IL SUO SEGRETO, INVECE, È DATO DALLA TERNA $\rightarrow \langle P, Q, S \rangle$. // TIPO "CHIAVE PRIVATA"
NESSUNO DEI VALORI DELLA TERNA È RICAVABILE, IN TEMPO POLINOMIALE, DALLA COPPIA $\langle m, T \rangle$
 - IL PROTOCOLLO AVRA K PASSI, AL TERMINE DEI QUALI, IL VERIFIER V DRA STABILIRE L'IDENTITÀ DI P.
CIÒ DRAVÀ AVVENIRE CON PROBABILITÀ $\rightarrow 1 - (\frac{1}{2})^k$, E SENZA INFIERIRE SUL SEGRETO DI P
// CIÒ È SENZA SCOPRIRE I FATTORI PRIMI DI M, OPPURE LA RADICE DI T
- PROTOCOLLO // RIETTUTO PER K VOLTE
1. V CHIEDE A P DI INIZIARE UN'ITERAZIONE
 2. P GENERA UN INTERO CASUALE $\rightarrow r \in \mathbb{Z}_m$.
SI CALCOLA UN VALORE $\rightarrow U = r^2 \text{ mod } m$.
COMUNICA U A V
 - // U VIENE UTILIZZATO SOLO PER QUESTA ITERAZIONE, E VIENE Poi SCARTATO PERCHE' U VIENE RICALCOLATO

3. ① GENERA UN INTERO CASUALE $r \in \{0,1\}$. // bit casuale

COMMUNICA ② A P

4. P CALCOLA $\rightarrow z = rs^2 \text{ mod } m$.

COMMUNICA ② A U

// DALLA FORMULA PRECEDENTE, ABBIAMO CHE:

• PER $r=0 \rightarrow z=r$

• PER $r=1 \rightarrow z=rs \text{ mod } m$.

A QUESTO PUNTO P SI E' CALCOLATO SIA U CHE Z

5. U CALCOLA $\rightarrow x = z^2 \text{ mod } m$.

if ($x = Uz^2 \text{ mod } m$)

RIPETE DAL PASSO 1. // CASO SFIDA VINTA DA PARTE DI P

ELSE BLOCCA IL PROTOCOLLO SENZA IDENTIFICARE P // SFIDA PERSA E P E' UN "IMBROGLIO"

ZERO KNOWLEDGE → Identificazione

Con riferimento al protocollo di identificazione di Fiat-Shamir, spiegare come un *prover dishonesto* P possa tentare di ingannare il *Verifier* V , prevedendo i valori del bit casuale e che V genera nel corso del protocollo. Discutere la probabilità di successo di questo attacco.

- PER DISCUTERE CIÒ, SI RICORRE ALLA DIMOSTRAZIONE DI CORRETTEZZA.

Si premette che a \textcircled{P} è noto il valore di t pubblico, e non di s

- Considerando \textcircled{P} disonesto, cioè che vuole farsi identificare come un altro utente, egli tenterà di ingannare \textcircled{V} , prevedendo i valori di r che vengono generati da \textcircled{V} al 3° passo del protocollo

- Partendo dal 2° passo del protocollo, il prover \textcircled{P} procederà inviando a \textcircled{V} il valore:

- $u = r^2 \bmod m$, se prevede di ricevere $\rightarrow r=0$
- $u = \frac{r^2}{t} \bmod m$, se prevede di ricevere $\rightarrow r=1$

- Arrivato al 4° passo, in entrambi i casi, \textcircled{P} invierà a \textcircled{V} lo stesso valore $\rightarrow z=r$

- Se la previsione di \textcircled{P} su \textcircled{R} è corretta, al 5° passo, per entrambi i valori di \textcircled{R} \textcircled{V} scoprirà che:
 $\rightarrow x = ut^2 \bmod m = r^2 \bmod m$, quindi, accetterà l'iterazione

- Invece, se la previsione di \textcircled{P} è sbagliata, la relazione non sarà verificata e sarà scoperto l'inganno

- Concludiamo dicendo che, poiché \textcircled{R} è generato a caso, le previsioni di \textcircled{P} su \textcircled{R} sono corrette con probabilità $\rightarrow \frac{1}{2}$

SSL → Protocolli

Descrivere il protocollo SSL, con particolare riferimento alla fase di *Handshake*.

- IL PROTOCOLLO SECURE SOCKET LAYER (SSL) SERVE A GARANTIRE LA SICUREZZA DELLE COMUNICAZIONI SU INTERNET.
- L'SSL IMPLEMENTA LA SICUREZZA NEGLI SCAMBIO DI DATI, ATTRAVERSO UNA GESTIONE SU DUE LIVELLI:
 - ① SSL RECORD
 - ② SSL HANDSHAKE

SSL RECORD

- SI TROVA AL LIVELLO PIÙ BASSO, DIRETTAMENTE CONNESSO AL LIVELLO DI TRASPORTO
- INCAPSULA I DATI SPEDITI DAI PROTOCOLLI SUPERIORI ASSICURANDO:
 - CONFIDENZIALITÀ DELLA COMUNICAZIONE
 - INTEGRITÀ DELLA COMUNICAZIONE

SSL HANDSHAKE

- SI TROVA AL LIVELLO SUPERIORE, DIRETTAMENTE CONNESSO ALL'APPLICATIVO
- CREA UN CANALE SICURO, AFFIDABILE E AUTENTICO TRA UTENTE E SISTEMA
- SPEDISCE I MESSAGGI, INCAPSULATI IN BLOCCHI CIFRATI E AUTENTICATI, IN UN CANALE SICURO INVIANO:
 - MECCANISMI DI CIFRATURA E DECIFRAZIONE
 - RELATIVE CHIAVI
- UNA SESSIONE DI COMUNICAZIONE SSL È INNESCATA DA UNO SCAMBIO DI MESSAGGI PRELIMINARI, CHE PRENDE IL NOME DI HANDSHAKE
- IN QUESTA SESSIONE, IL SISTEMA SERVER S E L'UTENTE CLIENT U SI IDENTIFICANO A VICENDA, COOPERANDO ALLA COSTRUZIONE DI CHIAVI SEGRETE DA IMPIEGARE, Poi, NELLE COMUNICAZIONI SIMMETRICHE SUCCESSIVE. TALE COSTRUZIONE AVVERRÀ DAL UTILIZZO DI UN'INFORMAZIONE SEGRETA, IL MASTER SECRET, SU CUI UTENTE E SISTEMA SI ACCORDANO IN MODO INCREMENTALE

- SCRIVIAMO I PASSI DELL' SSL HANDSHAKE:

1. UTENTE: CLIENT HELLO

- \textcircled{S} MANDA AD \textcircled{U} UN MESSAGGIO, DETTO CLIENT HELLO, CON CUI:

- RICHIEDE LA CREAZIONE DI UNA CONNESSIONE SSL
 - SPECIFICA I CRITERI DI SICUREZZA CHE DOVRANNO ESSERE GARANTITI DURANTE LA CONNESSIONE, CIOÈ I CIFRARI E I MECCANISMI CHE \textcircled{U} PUÒ SUPPORTARE
 - INIZIA UNA SEQUENZA DI byte CASUALI
- QUESTI CIFRARI E MECCANISMI SONO RAPPRESENTATI DALLA CIPHER SUITE.

UN ESEMPIO DI ESSA È → SSL_RSA_WITH_AES_CBC_SHA1, IN CUI ABBIAMO CHE:

- RSA → È USATO PER LO SCAMBIO DELLE CHIAVI DI SESSIONE
- AES → È USATO PER LA CIFRATURA SIMMETRICA
- CBC → INDICA UN CIFRARIO A COMPOSIZIONE DI BLOCCHI
- SHA1 → È LA FUNZIONE HASH ONE-WAY PER IL MAC

2. SISTEMA: SERVER HELLO

- IL SISTEMA \textcircled{S} :

- RICEVE IL MESSAGGIO DI CLIENT HELLO
- SELEZIONA UNA CIPHER SUITE CHE ANCHE \textcircled{S} È IN GRADO DI SUPPORTARE
- INVIA AD \textcircled{U} UN MESSAGGIO, DETTO SERVER HELLO, CHE SPECIFICA LA SUA SCELTA E CHE CONTIENE UNA NUOVA SEQUENZA DI byte CASUALI

- SE \textcircled{U} NON RICEVE IL MESSAGGIO DI SERVER HELLO, ALLORA INTERROMPE LA COMUNICAZIONE

3. SISTEMA: INNIO DEL CERTIFICATO

- \textcircled{S} SI AUTENTICA CON \textcircled{U} INVIANDOGLI IL PROPRIO CERTIFICATO DIGITALE, E GLI EVENTUALI ALTRI CERTIFICATI DELLA SUA CATENA DI CERTIFICAZIONE, I QUALI VANNO DALLA SUA CA FINO ALLA CA RADICE
- SE I SERVIZI DI \textcircled{S} DEVONO ESSERE PROTETTI NEGLI ACCESSI, \textcircled{S} PUÒ RICHIEDERE A \textcircled{U} DI AUTENTICARSI INVIANDO IL SUO CERTIFICATO DIGITALE.

CIOÈ AVVIENE RARAMENTE PERCHÉ LA MAGGIOR PARTE DEGLI UTENTI NON HA UN CERTIFICATO PERSONALE.

INOLTRE, IL SISTEMA DOVRÀ ACCERTARSI DELL' IDENTITÀ DELL' UTENTE IN UN SECONDO MOMENTO

4. SISTEMA: SERVER HELLO DONE

- \textcircled{S} INVIA IL MESSAGGIO SERVER HELLO DONE CON CUI SANCISSCE LA FINE DEGLI ACCORDI SULLA CIPHER SUITE E SUI PARAMETRI CRITTOGRAFICI A ESSA ASSOCIATI

5. UTENTE: AUTENTICAZIONE DEL SISTEMA

- L'UTENTE \textcircled{U} , PER ACCERTARE L'AUTENTICITÀ DEL CERTIFICATO RICEVUTO DA \textcircled{S} , CONTROLLA CHE:
 - LA DATA CORRENTE SIA INCLUSA NEL PERIODO DI VALIDITÀ DEL CERTIFICATO
 - LA CA CHE HA FIRMATO IL CERTIFICATO SIA TRA QUELLI "FIDATE"
 - LA FIRMA APPOSTA DALLA CA SIA AUTENTICA

6. UTENTE: INVIO DEL PRE-MASTER SECRET E COSTRUZIONE DEL MASTER SECRET

- L'UTENTE \textcircled{U} :
 - COSTRUISCE UN PRE-MASTER SECRET COSTITUITO DA UNA NUOVA SEQUENZA di byte CASUALI, CIOÈ GENERA UN PRIMO VALORE SEGRETO
 - LO CIFRA CON IL CIFRARIO A CHIAVE PUBBLICA SU WI SI È ACCORDATO CON \textcircled{S}
//cioè con l'RSA, USANDO LA CHIAVE PUBBLICA PRESENTE NEL CERTIFICATO DI \textcircled{S}
 - SPEDISCE A \textcircled{S} IL RELATIVO CRITTOGRAMMA
 - POI, IL PRE-MASTER SECRET VIENE COMBINATO DA \textcircled{U} CON ALCUNE STRINGHE NOTE E CON i byte CASUALI PRESENTI:
 - NEL MESSAGGIO DI CLIENT HELLO
 - NEL MESSAGGIO DI SERVER HELLO
 - \textcircled{U} APPLICA A TUTTE QUESTE SEQUENZE DELLE FUNZIONI HASH ONE-WAY, UTILIZZANDO UNA COMBINAZIONE OPPORTUNA
 - OTTIENE COSÌ IL MASTER SECRET
- ## 7. SISTEMA: RICEZIONE DEL PRE-MASTER SECRET E COSTRUZIONE DEL MASTER SECRET
- \textcircled{S} DECIFRA IL CRITTOGRAMMA CONTENENTE IL PRE-MASTER SECRET RICEVUTO DA \textcircled{U}
 - \textcircled{S} CALCOLA IL MASTER SECRET MEDIANTE LE STESE OPERAZIONI ESEGUITE DA \textcircled{U} AL PASSO 6.
//Ciò è dovuto al fatto che dispone delle stesse informazioni.
IL MASTER SECRET CALCOLATO DA \textcircled{S} ANNIENE IN MANIERA INDIPENDENTE RISPETTO AD \textcircled{U}

8. UTENTE: INVIO DEL CERTIFICATO (OPZIONALE)

- SE ALL'UTENTE VIENE RICHIESTO UN CERTIFICATO (PASSO 3.) ED EGLI NON LO POSSIENE, ALLORA IL SISTEMA INTERROMPE L'ESECUZIONE DEL PROTOCOLLO
- ALTRIMENTI \textcircled{U} INVIA IL PROPRIO CERTIFICATO CON ALLEGATE UNA SERIE DI INFORMAZIONI FIRMATE CON LA SUA CHIAVE PRIVATA, TRA CUI:
 - IL MASTER SECRET
 - TUTTI I MESSAGGI SCAMBIAI FINO A QUEL MOMENTO, CIOÈ LA SSL-history
- \textcircled{S} CONTROLLA CHE IL CERTIFICATO DI \textcircled{U} E VERIFICA L'AUTENTICITÀ E LA CORrettezza DELLA SSL-history
- IN PRESENZA DI ANOMALIE, LA COMUNICAZIONE CON \textcircled{U} VIENE INTERROTTA

9. UTENTE/SISTEMA: MESSAGGIO FINISHED

- FINISHED, È IL PRIMO MESSAGGIO PROTETTO MEDIANTE IL MASTER SECRET E LA CIPHER SUITE, SU CUI I DUE PARTNER SI SONO ACCORDATI
- IL MESSAGGIO VIENE PRIMA COSTRUITO DA \textcircled{U} E SPEDITO A \textcircled{S} , Poi COSTRUITO DA \textcircled{S} E SPEDITO A \textcircled{U} . IN QUESTI DUE INVII LA STRUTTURA DEL MESSAGGIO È LA STESSA, MA CAMBIANO LE INFORMAZIONI IN ESSO CONTENUTE
- LA COSTRUZIONE AVVIENE IN DUE PASSI:
 - 1) ALL'INIZIO SI CONCATENANO IL MASTER SECRET, TUTTI I MESSAGGI DI HANDSHAKE SCAMBIAI FINO A QUEL MOMENTO E L'IDENTITÀ DEL MITTENTE \textcircled{U} O \textcircled{S}
 - 2) LA STRINGA OTTENUTA VIENE TRASFORMATA APPLICANDO LE FUNZIONI SHA-1 E MDS. QUINDI, SI OTTIENE UNA COPPIA DI VALORI CHE COSTITUISCE IL MESSAGGIO FINISHED
- IL MESSAGGIO È DIVERSO NELLE DUE COMUNICAZIONI PERCHÉ \textcircled{S} AGGIUNGE AI MESSAGGI DI HANDSHAKE ANCHE IL MESSAGGIO FINISHED RICEVUTO DA \textcircled{U}
- IL DESTINATARIO DELLA COPPIA, \textcircled{S} O \textcircled{U} , NON PUÒ INVERTIRE LA COMPUTAZIONE PRECEDENTE IN QUANTO GENERATA DA FUNZIONI ONE WAY, MA RICOSTRUISCE L'INGRESSO DELLE DUE FUNZIONI SHA-1 E MDS.
Poi, RICALCOLA QUESTE FUNZIONI E CONTROLLO CHE LA COPPIA COSÌ GENERATA COINCIDA CON QUELLA RICEVUTA.
- LA DEMOSTRAZIONE DEGLI ULTIMI PASSAGGI IMPLICA CHE LA COMUNICAZIONE È AVVENUTA CORRETTAMENTE

Illustrare il protocollo BB84 per lo scambio di chiavi segrete basato sulla trasmissione di foton polarizzati e spiegare perché può ritenersi sicuro.

- INNANZITUTTO, INTRODUCIAMO IL FATTO CHE UN FOTONE POSSIEDE UNA POLARIZZAZIONE, CHE RAPPRESENTA IL PIANO DI OSCILLAZIONE DEL SUO CAMPO ELETTRICO

- IL PIANO DI OSCILLAZIONE IMPLICA IL FATTO CHE, IL FOTONE, POSSA TROVARSI IN DIVERSI STATI:

- POLARIZZAZIONE VERTICALE \downarrow , TRAMITE LA FRECCIA \uparrow
- POLARIZZAZIONE ORIZZONTALE H , TRAMITE LA FRECCIA \rightarrow
- POLARIZZAZIONE " $A +45^\circ$ ", TRAMITE LA FRECCIA \nearrow
- POLARIZZAZIONE " $A -45^\circ$ ", TRAMITE LA FRECCIA \searrow

- LE POLARIZZAZIONI SONO DIVISE A DUE A DUE A COPPIA, Dette BASI DI POLARIZZAZIONE, E SONO DIVISE IN:

- ORTOGONALI $\rightarrow +$, PER \textcircled{V} E \textcircled{H}
- DIAGONALI $\rightarrow \times$, PER $\textcircled{\nearrow}$ E $\textcircled{\searrow}$

- QUESTI QUATTRO STATI SARANNO ASSOCIATI AI b.R 0/1 COSÌ:

- b.R 0 ASSOCIATO A UN FOTONE POLARIZZATO $\textcircled{V} \uparrow$
- b.R 0 ASSOCIATO A UN FOTONE " $A +45^\circ$ " \nearrow
- b.R 1 ASSOCIATO A UN FOTONE POLARIZZATO $\textcircled{H} \rightarrow$
- b.R 1 ASSOCIATO A UN FOTONE " $A -45^\circ$ " \searrow

- PER REALIZZARE IL BB84 SONO NECESSARIE TRE APPARECCHIATURE DI BASE:

1. ONE-PHOTON GUN (OPG), CHE CONSENTE DI GENERARE ED EMETTERE UN FOTONE ALLA VOLTA CON UNA POLARIZZAZIONE PRESTABILITA
2. CELLA DI POCKELS (PC), CHE CONSENTE DI IMPORRE LA POLARIZZAZIONE DI UN FOTONE AGENDO SU UN CAMPO ELETTRICO DI CONTROLLO
3. BEAM SPLITTER POLARIZZANTE (PBS), CHE DIRIGGIA I FOTONI IN INGRESSO VERSO UNA TRA DUE USCITE, A O R, RAPPRESENTANTI UN FENOMENO OTTICO DI:
 - ATTRANERAMENTO \textcircled{A}
 - RIFLESSIONE \textcircled{R}

- CONSIDEREREMO COME UTENTI ALICE E BOB CHE PARTECIPANO ALLA COMUNICAZIONE SU QUESTO PROTOCOLLO, DONDE TALE COMUNICAZIONE È DIVISA SU DUE FASI.

C'È UNA PRIMA FASE SUL CANALE STANDARD E UNA SECONDA FASE SUL CANALE QUANTISTICO

- il protocollo BB84 è organizzato nel modo seguente:

1. Alice invia una sequenza S_A sul canale quantistico.

S_A è una sequenza di bit codificati nelle polarizzazioni dei fotoni

2. Bob, utilizzando la sua parte di apparecchiatura, farà due misure interpretando S_A con le basi che ha scelto casualmente; il tutto gli farà ottenere una sequenza S_B . Considerando le basi scelte, da Alice e Bob, si nota che:

- Dove le basi coincidono, allora le sequenze sono uguali
- Dove le basi sono discordi, metà delle volte le sequenze saranno uguali e metà delle volte no // quest'ultime le scartiamo

ESITO DELLA LETTURA DI BOB SU UN FOTONE INVIAZIO DA ALICE

		bit e fotone inviato da A			
		0 ↑	0 ↗	1 →	1 ↓
basi di B	+	↑	↑→	→	↑→
	×	↗ ↓	↗	↗ ↓	↓

// LA PRESENZA DI DUE FRECCIE IN UNA CASELLA INDICA LETTURA IMPROVVISATA:

→ 50% DI PROBABILITÀ DI ASSUMERE UNO DEI DUE VALORI

3. Usando il canale standard, Bob comunica ad Alice le basi che ha scelto e Alice gli indica quali basi sono comuni alle sue

4. In assenza di interferenze sul canale quantistico da parte di un crittoanalista attivo Eve, Alice e Bob possiedono una sottosequenza identica $\rightarrow S'_A = S'_B$.

Tale sottosequenza, è formata dai bit codificati da Alice e decodificati da Bob

// ANREMO CIRCA $|S'_A| = |S'_B| \simeq \frac{|S_A|}{2}$

OSS (L'IDEA È QUELLA DI COSTRUIRE LA CHIAVE UTILIZZANDO LA SEQUENZA COMUNE, CONTROLLANDO PRIMA CHE NON SIA INTERVENUTO EVE.)

5. ALICE E BOB SACRIFICANO UNA PORZIONE S_A'' , S_B'' DELLE SEQUENZE S_A' , S_B' IN POSIZIONI PRESTABILITE, COMUNICANDOLE SUL CANALE STANDARD.

IN QUESTA SITUAZIONE SI VERIFICANO DUE CASI:

- CASO IN CUI IL CRITTOANALISTA, EVE, NON È INTERVENUTO SUL CANALE, ALLORA LE DUE SEQUENZE SONO UGUALI $\rightarrow S_A'' = S_B''$.

DI CONSEGUENZA, ALICE E BOB, USCIRANNO COME CHIAMI $S_A' - S_A''$, CHE È UGUALE A $S_B' - S_B''$

- CASO IN CUI IL CRITTOANALISTA, EVE, È INTERVENUTO SUL CANALE, ALLORA LE DUE SEQUENZE POTREBBERO NON ESSERE COINCIDENTI $\rightarrow S_A'' \neq S_B''$.

CIO' IMPONE, AD ALICE E A BOB, DI INTERROMPERE LA COMUNICAZIONE

- IN QUANTO A SICUREZZA DEL BB84, SI PUÒ AFFERMARE CHE TALE PROTOCOLLO È SICURO, PERCHÉ UN CRITTOANALISTA ATTIVO, EVE, CHE VOGLIE INTRODURRESI NELLA COMUNICAZIONE VERREBBE SCOPERTO.
- PER AGGIURARE IL CONTROLLO SI POTREBBE SUPPORRE CHE EVE, OLTRE AD INTERPRETARE I FOTONI IN ARRIVO CON LE SUO CHIAVI, LI DOPPICHI PRIMA DI LEGGERLI PER INVIALI INCORROTTI A BOB.
QUINDI, ALICE E BOB NON RILEVEREBBERO alcuna DIFFERENZA NELLE PORZIONI DI SEQUENZE S_A'' E S_B''
- TUTTO CIÒ, PERO', SI OPPONE AD UNO DEI PRINCIPI DELLA MECCANICA QUANTISTICA, CIDÈ CHE NON È POSSIBILE DOPPLICARE UN SISTEMA QUANTISTICO SENZA CAUSARNE LA DECOERENZA //SICUREZZA DEL PROTOCOLLO

BB84 → Scambio di chiavi

Considerando il protocollo BB84, darne un breve esempio di applicazione (in assenza di crittoanalista sul canale).

- Si usi la sequenza di 16 bit ottenuta trasformando ordinatamente in binario le quattro cifre decimali meno significative del proprio numero di matricola.
- Si scelgano a caso le basi per impostare, intercettare e per misurare la polarizzazione dei fotoni.
- Si indichino con precisione tutti i passi del protocollo.

- INIZIAMO CONSIDERANDO LE CIFRE DECIMALI MENO SIGNIFICATIVE DEL MIO NUMERO DI MATRICOLA:

→ 4, 3, 6, 7 // PRESE COME ESEMPIO

- CONVERTIAMO TALI CIFRE DA BASE DECIMALE IN BINARIO SU 4 b/f, visto che DOBBIANO FORMARE UNA SEQUENZA DA 16 b/f: // ④ SEQUENZE DA 4 b/f → ① SEQUENZA DA 16 b/f

• 4 = 0100

• 3 = 0011

• 6 = 0110

• 7 = 0111

- LA SEQUENZA DA SCAMBIARE È DATA DALLA CONCATENAZIONE DEUE PRECEDENTI SOTTOSEQUENZE:

→ 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1

PROTOCOLLO

- SUPPONIAMO CHE ALICE VOGLIA INVIARE A BOB, SUL CANALE QUANTISTICO, LA SEQUENZA

→ SA: 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1 // SEQUENZA DI ALICE

- SUPPONIAMO CHE ALICE SCEGLIE COME BASE PER LA POLARIZZAZIONE DEI FOTONI:

→ + x + + x x x + x + + x x + x

- SI CONSIDERANO I SEGUENTI CASI PER LA POLARIZZAZIONE DEI FOTONI, CHE SI OTTENGONO CONFRONTANDO LA BASE CON LA SEQUENZA INVIATA:

• b/f 0, BASE ORTOCONALE + → SI ASSOCIA UN FOTONE POLARIZZATO N ↑

• b/f 0, BASE DIAGONALE X → SI ASSOCIA UN FOTONE "A +45°" ↗

• b/f 1, BASE ORTOCONALE + → SI ASSOCIA UN FOTONE POLARIZZATO H →

• b/f 1, BASE DIAGONALE X → SI ASSOCIA UN FOTONE "A -45°" ↓

- SI APPLICA IL PASSAGGIO PRECEDENTE: // TRA LA SEQUENZA DI ALICE CON LA SUA BASE

→ 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1

→ + x + + x x x + x + + x x + x, FOTONI CHE ALICE PREPARA ED INVIA A BOB

- SUPPONIAMO CHE BOB SCEGLIE COME BASE PER LA MISURAZIONE DEI FOTONI:

$\rightarrow + \times + + \times \times \times + \times + + + \times \times \times +$

// PER SEMPLICITÀ HO CAMBIATO LE ULTIME DUE COMPONENTI DELLA BASE RISPETTO A QUELLA DI ALICE

- PER CONTROLLARE L'ESITO DELLA MISURA DA PARTE DI SOB SUI FOTONI RICEVUTI DA ALICE, FACCIAMO RIFERIMENTO ALLA SEGUENTE TABELLA:

bit e fotone inviato da

$$\| B = B \cup B \quad \& \quad A = A \cup C$$

basi di B	0 ↑	0 ↗	1 →	1 ↘
+	↑	↑ →	→	↑ →
×	↗ ↘	↗	↗ ↘	↘

Dove ci sono due fotoni in una casella, allora verrà scelto casualmente uno dei due fotoni.

PER LA LETTURA DI BOR.

- BOB PRENDE CIÒ CHE HA INVIATO ALICE E FA LA MISURA :

$$A: \left(\begin{array}{cccccccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ + & x & + & + & x & x & x & + & x & + & + & + & x & x & + & x \\ \downarrow & \searrow & \uparrow & \uparrow & \nearrow & \nearrow & \rightarrow & \rightarrow & \nearrow & \rightarrow & \uparrow & \nearrow & \rightarrow & \searrow & \rightarrow & \searrow \end{array} \right) // Alice$$

R: + x + + x x x + x + + + x x x + //Bob

→ ↑ ↗ ↑ ↗ ↗ ↗ → ↗ → → ↑ ↗ ↗ ↗ ↑ , LETTURA DI SOB (ESITO MISURA)

BASI CONCORDI, LETTURA CORRETTA | BASI DISCORDI, LETTURA CASUALE SU VALORI
DELLE CELLE NELLA TABELLA

- SI CONSIDERANO I SEGUENTI CASI SULLA POLARIZZAZIONE DEI FOTONI CONFRONTATI CON LA BASE, PER OTTENERE I BIT DELLA SUA SEQUENZA:

- b.r. 0 DA \rightarrow BASE ORTOCONALE + E FOTONE POLARIZZATO (1) ↑
 - b.r. 0 DA \rightarrow BASE DIAGONALE X E FOTONE "A + 45°" →
 - b.r. 1 DA \rightarrow BASE ORTOCONALE + E FOTONE POLARIZZATO (4) →
 - b.r. 1 DA \rightarrow BASE DIAGONALE X E FOTONE "A - 45°" ↘

— Si applica il passaggio precedente: //TRA i FOTONI di SOS CON LA SUA BASE

↑ + x + + x x x + x + + + x x x +

$\rightarrow 0\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0 : S_8 \rightarrow$ SEQUENZA OTTENUTA DA S_{03}

- A QUESTO PUNTO ALICE E BOB SI COSTRUISCONO UNA SOTTOSEQUENZA, CIASCUNO, FATTA DAI b.F CONCORDI AD ENTRAMBE LE SEQUENZE, CIÒ È QUELLI CHE RAPPRESENTANO LE BASI COMUNI.

VERRANNO SCARTATI I b.F DISCORDI

$$A: \left(\begin{array}{ccccccccccccc} S_A: & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ & + & x & + & + & x & x & x & + & x & + & + & + & x & x & + & x \end{array} \right)$$

$$B: \left(\begin{array}{ccccccccccccc} S_B: & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ & + & x & + & + & x & x & x & + & x & + & + & + & x & x & + & x \end{array} \right)$$

$$\rightarrow \text{SOTTOSEQUENZA } S_A' = S_B' = \underline{\underline{0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1}}$$

- A QUESTO PUNTO, ALICE E BOB SACRIFICANO UNA PORZIONE DI CODICE DELLA PROPRIA SOTTOSEQUENZA. SUPPONIAMO CHE LA PORZIONE DI CODICE DA TOLIERE SIANO I TRE b.F FINALI CHE CHIAMEREMO:

- $S_A'' = 001$, PER ALICE
- $S_B'' = 001$, PER BOB

- TALI SEQUENZE VERRANNO COMUNICATE SUL CANALE STANDARD, E POICHÉ SIANO NEL CASO DI ASSENZA DEL CRITTOANALISTA, ALLORA SI VERIFicherà $\rightarrow S_A'' = 001 = S_B''$

- DOPO AVER FATTO LA VERIFICA, ALICE E BOB STABILIRANNO LA CHIAVE DA UTILIZZARE SCARTANDO i b.F CHE SONO STATI SCAMBIAI:

- $K_{key} = S_A' - S_A'' = 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1$, ALICE
- $K_{key} = S_B' - S_B'' = 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1$, BOB

$$\cancel{\cancel{S_A' - S_A'' = S_B' - S_B''}}$$

Considerando il protocollo BB84, darne un breve esempio di applicazione (in presenza di crittoanalista sul canale).

- Si usi la sequenza di 16 bit ottenuta trasformando ordinatamente in binario le quattro cifre decimali meno significative del proprio numero di matricola.
 - Si scelgano a caso le basi per impostare, intercettare e per misurare la polarizzazione dei fotoni.
 - Si indichino con precisione tutti i passi del protocollo.
-

- INIZIAMO CONSIDERANDO LE CIFRE DECIMALI MENO SIGNIFICATIVE DEL MIO NUMERO DI MATRICOLA:

→ 4, 3, 6, 7 // PRESE COME ESEMPIO

- CONVERTIAMO TALI CIFRE DA BASE DECIMALE IN BINARIO SU 4 b/f, visto che DOBBIANO FORMARE UNA SEQUENZA DA 16 b/f: // ④ SEQUENZE DA 4 b/f → ① SEQUENZA DA 16 b/f

• 4 = 1100

• 3 = 0011

• 6 = 0110

• 7 = 0111

- LA SEQUENZA DA SCAMBIARE È DATA DALLA CONCATENAZIONE DELE PRECEDENTI SOTTOSEQUENZE:

→ 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1

PROTOCOLLO

- SUPPONIAMO CHE ALICE VOGLIA INVIARE A BOB, SUL CANALE QUANTISTICO, LA SEQUENZA

→ SA: 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1 // SEQUENZA DI ALICE

- SUPPONIAMO CHE ALICE SCEGLIE COME BASE PER LA POLARIZZAZIONE DEI FOTONI:

→ + x + + x x x + x + + x x + x

- SI CONSIDERANO I SEGUENTI CASI PER LA POLARIZZAZIONE DEI FOTONI, CHE SI OTTENGONO CONFRONTANDO LA BASE CON LA SEQUENZA INVIATA:

• b/f 0, BASE ORTOCONALE + → SI ASSOCIA UN FOTONE POLARIZZATO ① ↑

• b/f 0, BASE DIAGONALE X → SI ASSOCIA UN FOTONE "A +45°" ↗

• b/f 1, BASE ORTOCONALE + → SI ASSOCIA UN FOTONE POLARIZZATO ④ →

• b/f 1, BASE DIAGONALE X → SI ASSOCIA UN FOTONE "A -45°" ↓

- SI APPLICA IL PASSAGGIO PRECEDENTE: // TRA LA SEQUENZA DI ALICE CON LA SUA BASE

→ 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1

→ + x + + x x x + x + + x x + x, FOTONI CHE ALICE PREPARA ED INVIA A BOB

- Poiché siamo in presenza del crittanalista, supponiamo che Eve si sia intronizzato tra Alice e Bob

- Eve, sceglie come base per la misurazione dei fotoni. (di Alice che sono stati intercettati):

$\rightarrow + \times + + \times \times \times + \times + + + \times \times \times +$

// PER SEMPLICITÀ SI CAMBIANO LE ULTIME DUE COMPONENTI DELLA BASE RISPETTO A QUELLA DI ALICE

- PER CONTROLLARE L'ESITO DELLA MISURA DA PARTE DI EVE sui fotoni ricevuti da Alice, facciano

RIFERIMENTO ALLA SEGUENTE TABELLA:

		bit e fotone inviato da A				$\text{E} = \text{EVE}$ e $A = \text{ALICE}$	
		0 \uparrow	0 \nearrow	1 \rightarrow	1 \nwarrow		
basi di E	+	\uparrow	$\uparrow \rightarrow$	\rightarrow	$\uparrow \rightarrow$		
	X	$\nearrow \nwarrow$	\nearrow	$\nearrow \nwarrow$	\nwarrow		

Dove ci sono due fotoni in una casella, allora verrà scelto casualmente uno dei due fotoni.

PER LA LETTURA DI EVE.

- Eve prende ciò che ha inviato Alice e fa la misura:

$$A: \left(\begin{array}{cccccccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ + & x & + & + & x & x & x & + & x & + & + & + & x & x & + & x \end{array} \right) // \text{Alice}$$

E: $+ \times + + \times \times \times + \times + + + \times \times \times +$ //EVE

$\rightarrow \uparrow \nwarrow \uparrow \nearrow , LETTURA DI EVE (ESITO MISURA)

BASI CONCORDI, LETTURA CORRETTA | BASI DISCORDI, LETTURA CASUALE SUI VALORI
DELLA CELLE NELLA TABELLA

- Si considerano i seguenti casi sulla polarizzazione dei fotoni confrontati con la

base, per ottenere i bit della sua sequenza:

- b.f. 0 DA \rightarrow BASE ORTOCONALE \uparrow E FOTONE POLARIZZATO ① \uparrow
- b.f. 0 DA \rightarrow BASE DIAGONALE \times E FOTONE "A +45°" \nearrow
- b.f. 1 DA \rightarrow BASE ORTOCONALE \uparrow E FOTONE POLARIZZATO ④ \rightarrow
- b.f. 1 DA \rightarrow BASE DIAGONALE \times E FOTONE "A -45°" \nwarrow

- Si applica il passaggio precedente: // TRA I FOTONI DI EVE CON LA SUA BASE

$\rightarrow + \times + + \times \times \times + \times + + + \times \times \times +$
 $\rightarrow \uparrow \nwarrow \uparrow \nearrow

$\rightarrow 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0$: Sequenza ottenuta da Eve

- LE MISURE CHE EVE HA FATTO HANNO TURBATO I FOTONI
- ADESSO ABBIANO CHE BOB INTERVIENE NELLA COMUNICAZIONE, E SUPPONIAMO CHE SCEGLIE COME BASE PER LA MISURAZIONE DEI FOTONI (CHE DORREBBERO ESSERE QUELLI DI ALICE, MA SONO QUELLI DI EVE):
- $\rightarrow + \times + + \times \times \times + \times + + + \times \times \times \times$
- // PER SEMPLICITÀ SI CAMBIA L'ULTIMA COMPONENTE DELLA BASE RISPETTO A QUALEVAI DI EVE
- PER CONTROLLARE L'ESITO DELLA MISURA DA PARTE DI BOB SUI FOTONI RICEVUTI DA EVE (CHE DORREBBERO ESSERE QUELLI RICEVUTI DA ALICE), SI FA RIFERIMENTO ALLA SEGUENTE TABELLA:

		bit e fotone inviato da E				$\text{B} = \text{BOB}$ $\text{E} = \text{EVE}$	
		0 ↑	0 ↗	1 →	1 ↓		
basi di B	+	↑	↑→	→	↑→		
	×	↗ ↓	↗	↗ ↓	↓		

Dove ci sono due fotoni in una casella, allora verrà scelto casualmente uno dei due fotoni.
PER LA LETTURA DI BOB.

- BOB PRENDE CIÒ CHE AVREBBE DOVUTO INVIARE ALICE, MA CHE INVECE HA INVIATO BOB, E FA LA MISURA

$$E: \left(\begin{array}{cccccccccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ + & x & + & + & x & x & x & + & x & + & + & + & x & x & x & + \\ \uparrow & \downarrow & \uparrow & \uparrow & \nearrow \end{array} \right) // \text{EVE}$$

$$B: + \times + + \times \times \times + \times + + + \times \times \times \times // \text{BOB}$$

$\rightarrow + \times + + \times \times \times + \times + + + \times \times \times \times$, LETTURA DI BOB (ESITO MISURA)

BASI CONCORDI, LETTURA CORRETTA

| BASI DISCORDI, LETTURA CASUALE SU
| VALORI DELLE CELLE NELLA TABELLA

- SI CONSIDERANO I SEGUENTI CASI SULLA POLARIZZAZIONE DEI FOTONI CONFRONTATI CON LA BASE, PER OTTENERE I BIT DELLA SUA SEQUENZA:

- b.Γ 0 DA → BASE ORTOGONALE + E FOTONE POLARIZZATO ① ↑
- b.Γ 0 DA → BASE DIAGONALE × E FOTONE "A +45°" ↗
- b.Γ 1 DA → BASE ORTOGONALE + E FOTONE POLARIZZATO ④ →
- b.Γ 1 DA → BASE DIAGONALE × E FOTONE "A -45°" ↘

// COME A PAGINA PRECEDENTE



- Si APPLICA IL PASSAGGIO PRECEDENTE : //TRA i FOTONI DI BOB CON LA SUA BASE

$\rightarrow + x + + x x x + x + + x x x x$
 $\uparrow \searrow \uparrow \nearrow

$\rightarrow 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0: S_B \rightarrow$ SEQUENZA OTTENUTA DA BOB

- A QUESTO PUNTO, ALICE E BOB SI COSTRUISCONO UNA SOTTOSEQUENZA, CIASCUNO, FATTA DAI bit CONCORDI AD ENTRAMBE LE SEQUENZE, CIDÈ QUELLI CHE RAPPRESENTANO LE BASI COMUNI.
VERRANNO SCARTATI i bit DISCORDI

A: $(S_A: 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1 1)$ // ALICE

B: $(S_B: 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0 0)$ // BOB

\rightarrow QUA, ALICE E BOB NOTANO CHE LE DUE SOTTOSEQUENZE NON COMBACIANO PERONE:

• $S_A' = 0 1 0 0 0 0 1 1 0 1 1 0 0 1 1$

• $S_B' = 0 1 0 0 0 0 1 1 0 1 1 0 0 1 0$

\rightarrow DIFFERISCONO DELL' ULTIMO bit

- POICHÉ ABBIAMO CHE $S_A' \neq S_B'$, ALICE E BOB SI ACCORGONO DEL' INTRUSIONE (DI EVE), E INTERROMPONO LA COMUNICAZIONE

BITCOIN → Funzioni hash e firma digitale

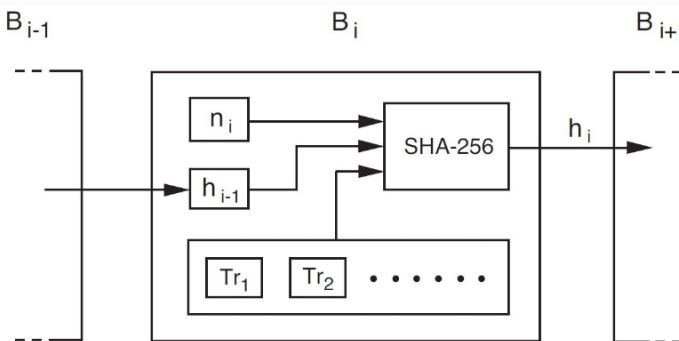
Descrivere dove intervengono le funzioni hash nell'ambito della valuta digitale bitcoin.

- CONSIDERANDO UN PROTOCOLLO DI TRANSAZIONE BITCOIN, INDICHIAMO CON A UN UTENTE CHE NUOLE INVIARE UN CERTA SOMMA S. BTC AD UN ALTRO UTENTE, CHE INDICHEREMO CON B
- L'UTENTE A GENERA UN MESSAGGIO FATTO COSÌ → $m = \text{addr}_A - x - \text{addr}_B$, DONGE:
 - addr_A → È L'INDIRIZZO BITCOIN DI A
 - addr_B → È L'INDIRIZZO BITCOIN DI B
 - x → È LA SOMMA DA TRASFERIRE
- L'UTENTE A CALCOLA L'HASH DEL MESSAGGIO → $h = \text{SHA-256}(m)$, E Poi GENERA LA FIRMA PER (m):
→ $f = D(h, K_A[\text{priv}])$
- INFINE, A DIFFONDE SULLA RETE LA COPPIA (TRANSAZIONE, FIRMA) → $\langle m, f \rangle$, CHE DOVRÀ ESSERE POI VALIDATA TRAMITE L'UTILIZZO, NUOVAMENTE DELLA FUNZIONE HASH
TALE DIFFUSIONE È CHIAMATA BROADCAST
- QUINDI, LA FUNZIONE HASH È UTILIZZATA CON I PROTOCOLLI DI TRANSAZIONE BITCOIN:
 - NELLA FASE DI FIRMA
 - NEL PROCESSO DI VALIDAZIONE

BITCOIN → Moneta elettronica

Descrivere il processo di validazione tramite *mining* delle transazioni Bitcoin.

- PER SPIEGARE IL PROCESSO DI VALIDAZIONE PRENDIAMO COME ESEMPIO UN BLOCCO DELLA BLOCKCHAIN:



• ABBIAMO UN UTENTE (U) CHE DISPONE DI UNA SERIE DI TRANSAZIONI Tr_1, Tr_2, \dots E NUOLO CERCARE DI VALIDARLE, ALLORA EGLI SI CREA UN BLOCCO CHE LE CONTIENE, DENTRO IL BLOCCO B_i :

- IN B_i , h_{i-1} È L'HASH DEL CONTENUTO DEL BLOCCO PRECEDENTE DELLA CATENA (PRESO IN INPUT)
- Poi in B_i TRONIAMO M_i , CHIAMATO **NONCE** ED È UN NUMERO INTERO
- TUTTI I VALORI PRECEDENTI FINISCONO NELLA FUNZIONE HASH **SHA-256**, CHE LI COMBINA CALCOLANDO L'IMMAGINE $\rightarrow h_i$

h_i È L'HASH DI QUESTO i -ESIMO BLOCCO CHE ANDRÀ IN INPUT AL BLOCCO B_{i+1} (SUCCESSIVO)

- CONSIDERANDO UN VALORE SOGLIA FISSATO DAL SISTEMA (BITCOIN), IL PROCESSO DI VALIDAZIONE IMPONE CHE $h_i < \text{SOGLIA FISSATA}$.

QUESTA CONDIZIONE SI VERIFICA TRONANDO IL VALORE DEL NONCE (M_i) DA DARE ALLA FUNZIONE **SHA-256**, IN MODO CHE h_i COMINCI CON T ZERI, DONDE T È IL VALORE FISSATO DAL SISTEMA.

- (M_i) : SI TRONA FACENDO UNA RICERCA ESAUSTRIVA ED È UN'OPERAZIONE DIFFICILE, MENTRE VERIFICARLO È FACILE

- I NODI CHE SI OCCUPANO DI CERCARE IL NONCE E ATTACCARE UN NUOVO BLOCCO ALLA CATENA SI CHIAMANO **MINER** ED IL LAVORO CHE FANNO È IL **MINING**

- PRIMA DELL'AGGIUNTA DI UN BLOCCO NELLA CATENA, ABBIAMO CHE OGNI MINER PREPARA UN SUO BLOCCO E CERCA DI AGGIUNGERLO

- IL MINER CHE TRONA IL NONCE LO COMUNICA IN RETE A TUTTI I NODI (BROADCAST), DOPODICHE' AVVIENE LA VALIDAZIONE E L'AGGIUNTA DEL BLOCCO ALLA CATENA

- I NODI CHE HANNO RICEVUTO IL NONCE, LO CONTROLLOANO GUARDANDO CHE LE TRANSAZIONI SIANO VALIDE. A CONTROLLO VERIFICATO, ESPRIMONO IL LORO CONSENTO E CERCANO DI CREARE NUOVI BLOCCI DA AGGIUNGERE AL NODO CHE RAPPRESENTA IL NODE

EXTRA

TLS

- Protocollo baso su SSL → Stabilisce insieme di chiavi simmetriche condivise per cifratura e autenticazione in comunicazioni → Tra client C (web browser) e server S (web site)
- L'autenticazione è reciproca tra → Client-Server
- TLS strutturato in due parti:
 - 1) Protocollo di Handshake → Scambio di chiavi per stabilire chiavi simmetriche → Fase in cui:
 - C possiede insieme di $K[\text{pub}]$ di CA → $pk_1, pk_2 \dots$
 - S possiede coppie $\langle K[\text{pub}], K[\text{priv}] \rangle = \langle pk_s, sk_s \rangle$, con certificato digitale → cert_s per pk_s (rilasciato dalla CA punto precedente)
 - 2) Protocollo Record-Layer → Utilizzo di chiavi simmetriche per autenticazione e cifratura
- Handshake descritto su tre step:
 - 1) C invia ad S → Messaggio iniziale del protocollo DH, includendo:
 - Gruppo G usato da C, cioè → Generatore Z_p^* (p primo) oppure curva ellittica
 - Generatore g oppure punto B della curva con suo ordine
 - Valore g^x oppure xB , dove → x scelto casualmente da C
 - Nonce → N_c
 - Ciphersuite utilizzabile
 - 2) S invia a C → Messaggio finale del protocollo DH per completarlo, includendo:
 - Valore g^y oppure yB , dove → y scelto casualmente da S
 - Nonce → N_s
 - Calcola $K=g^{xy}$ oppure $K=yxB$ e applica → KEY DERIVATION FUNCTION → Per estrazione da K delle chiavi:
 - K'_s, K'_c, K_s, K_c per una cifratura autenticata
 - Invio di $K[\text{pub}]=pk_s$, cert_s e firma σ → Quest'ultima ottenuta con $K[\text{priv}]=sk_s$ su tutti i messaggi di handshake inviati
 - Tutto cifrato con K'_s
 - 3) C svolge le seguenti funzioni:
 - Calcola $K=g^{xy}$ oppure $K=xyB$ e applica → KEY DERIVATION FUNCTION → Per estrazione da K delle chiavi:
 - K'_s, K'_c, K_s, K_c
 - Con K'_s recupera → pk_s, cert_s e firma σ //tutti cifrati con K'_s
 - Verifica firma σ sui messaggi Handshake con → pk_s
 - Calcola MAC dei messaggi di Handshake scambiati usando → K'_c → Invia poi a S

OSS(A fine Handshake → C e S condividono stesse chiavi di sessione → K_c, K_s per cifratura e autenticazione di comunicazione.

Poiché C verifica cert_s sa che → $K[\text{pub}]=pk_s$ è chiave corretta di C.

Se firma σ valida → C sta comunicando con S → S unico che conosce → $K[\text{priv}]=sk_s$ associata a → pk_s .

S firma tutti i messaggi scambiati per → Esecuzione protocollo DH → A C è noto che nessun valore è stato modificato.

Protocollo DH utile per non ottenimento di informazioni su K da crittonalista, e su tutte le K derivate)

- Procedimento con Record-Layer → Se Handshake OK:

- C usa K_c per cifratura messaggi da inviare a S
- S usa K_s per decifrazione messaggi da inviare a C

TLS (FORWARD SECRECY)

- La Forward Secrecy è una proprietà dei protocolli di negoziazione delle chiavi → Assicura se una chiave di cifratura a lungo termine viene compromessa → Le chiavi di sessione generate a partire da essa → Rimangono invariate
- Protocollo DH garantisce → Forward Secrecy → Valore y del server usato in Handshake → Può essere cancellato a fine protocollo → Senza y, crittoanalista non può ricostruire → Chiave di sessione K
- Con cifrari a $K[\text{pub}]$ → No Forward Secrecy, poiché → $K[\text{priv}]$ del server non può essere cancellata → Crittoanalista, se ottiene y → Decifra crittogrammi scambiati in fase Handshake e → Recupera le chiavi di sessione di client e server