Crittografia
(SSRI Crema)

Appunti, Esercizi e Risposte ai Temi di Esame
Raccolta focalizzata per studenti Online.

Andrea Draghetti

Cifratura Classica

Si descriva la grandezza dello spazio delle chiavi dei seguenti cifrari

- Cesare: 25 possibili chiavi (non 26 perché se sposto di 26 ritorna alla posizione iniziale e quindi non sposto nulla)
- Sostituzione senza chiave: 26! possibili chiavi (permutazione dei 26 caratteri)
- Sostituzione con chiave: Ipotizzando una chiave di x lettere senza ripetizioni e un alfabeto di 26 lettere 26!/(26-x) Esempio. Con x=3 le chiavi possibili sono 26*25*24 che e' uguale a 26!/(26-3)!
- Vigenere: 26^t possibili chiavi (dove t = lunghezza della chiave)
- Affine: 26*12 possibili chiavi (b = [0,26], a = [1,12] – i valori di b sono le 25 lettere dell'alfabeto e i valori di a sono i numeri primi con 26 minori di 26)
- Parole della chiave lunghe 10 lettere: 26! / 16 ! possibili chiavi(perché il primo carattere viene scelto tra tutti i 26, il secondo carattere tra i 25 poichè è gia stato scelto il primo, ecc.. fino al decimo carattere della chiave, che viene scelto tra i 17 restanti e tutto diviso le 16 possibilità di caratteri mancanti)

Si elenchino e si descrivano brevemente le tipologie di attacco note.

- Known Ciphertext Attack: in cui l'avversario conosce solo il testo cifrato – esempio: se un testo è stato cifrato con il cifrario a sostituzione, un possibile attacco può essere l'analisi della frequenza dei caratteri, in cui è possibile risalire al testo in chiaro (anche il cifrario di Vigenere è vulnerabile a questo attacco)
- Known Plaintext Attack: in cui l'avversario conosce sia il testo in chiaro sia il testo cifrato – esempio: se un testo è stato cifrato con il cifrario di Hill, l'avversario può intercettare n^2 coppie di caratteri in chiaro e cifrati e impostare un sistema lineare che può (di solito) essere risolto facilmente
- Chosen Plaintext Attack: in cui l'avversario può ottenere la cifratura di un testo in chiaro a sua scelta – esempio: crittografia a chiave pubblica, dove la chiave di cifratura è pubblica e l'attaccante può cifrare qualsiasi testo in chiaro egli voglia (x es. RSA)
- Chosen Ciphertext Attack: in cui l'avversario può ottenere la decifratura di un testo cifrato a sua scelta – esempio: RSA, sfruttando le proprietà dell'omomorfismo e la cifratura dell'algoritmo
- Chosen Text Attack: in cui l'avversario può ottenere la cifratura e la decifratura di coppie di testi in chiaro/cifrato

Discutere le nozioni di algoritmo di cifratura incondizionatamente sicuro e computazionalmente sicuro

Un algoritmo di cifratura è incondizionatamente sicuro se, indipendentemente dal tempo e dalla risorse disponibili è impossibile decrittografare il testo cifrato. Si definisce invece computazionalmente sicuro quando il tempo necessario per violare la crittografia è molto ampio e comunque superiore alla vita utile delle informazioni contenute.

L'unico cifrario a godere di una sicurezza incondizionata (a prescindere dal tempo e dallo spazio non si potrà decifrare il messaggio) è One-Time Pad, tutti gli altri avranno una sicurezza computazionale, cioè data dalla difficoltà di trovare in un tempo utile la decifratura del messaggio.

Descrivere un cifrario incondizionatamente sicuro

Esiste un solo sistema di questo tipo, ed è il cifrario One-Time pad: si utilizza una chiave casuale di lunghezza pari alla lunghezza del testo in chiaro, in modo da non doverla ripetere. La chiave deve essere utilizzata per crittografare e decrittografare un solo messaggio e poi essere scartata e cambiata. Questo schema è considerato inviolabile perché:

- produce un output casuale che non ha più alcuna relazione statistica con il testo in chiaro;
- la chiave è di lunghezza pari al messaggio e viene cambiata di volta in volta.

Appunto per questi motivi però tale cifrario è utilizzabile sono in teoria e non in pratica poiché necessita di una grande quantità di chiavi casuali e vi è inoltre il problema di distribuzione delle chiavi poiché sia il mittente sia il destinatario devono essere a conoscenza della chiave.

Si classifichino rispetto alla sicurezza (incondizionata/computazionale) i seguenti cifrari e si discutano sinteticamente i possibili attacchi:
i. Vigenere
ii. AES
iii. One-Time Pad
iv. RSA

1. Vigenere, è un cifrario polialfabetico che resiste all'analisi delle frequenze, tuttavia tramite l'utilizzo dell'indice di coincidenza (IC) e dell'indice di mutua coincidenza (MIC) è possibile trovare sia la lunghezza che la composizione della chiave;
2. AES è un cifrario a blocchi iterato, l'unico attacco conosciuto è di tipo "side-channel", cioè bisogna attaccare lo stesso sistema sul quale l'algoritmo è eseguito ed ottenere delle informazioni di cache timing. In questo modo, tracciando i cicli di macchina, è possibile determinare la chiave AES.

3. One-Time Pad è un cifrario perfetto in cui la lunghezza della chiave è uguale a quella del testo in chiaro, per questo è impossibile decifrare il messaggio, ma necessità la memorizzazione di un repertorio di chiavi;
4. RSA è un cifrario asimmetrico in cui la sicurezza è basata sia sulla generazione delle chiavi che sulla cifratura, essa dipende dal problema di fattorizzare numeri grandi di cui al momento non si è riuscito a trovare una soluzione che porti ad un risultato in un tempo utile.

Dire a quali tipi di attacchi resistono i seguenti cifrari, motivando la risposta con un esempio:

a. Cifrario a sostituzione
b. Cifrario Playfair
c. Hill
d. DES
e. RSA

1. Cifrari a sostituzione, non resistono ad alcuni tipo di attacco.
2. Cifrario Playfair ha una maggiore sicurezza rispetto alla cifratura a sostituzione, ma anch'esso non resiste ad alcun tipo di attacco. Tramite Known Ciphertext ed analisi delle frequenze è possibile decifrare il testo.
3. Cifrario Hill è un cifrario multilettera e resiste ad un attacco Known Ciphertext, ma non Know Plaintext.
4. DES resiste all'attacco Known Ciphertext, ma non all'attacco Chosen Plaintext Attack di crittoanalisi lineare: usando le approssimazioni lineari per le S-Box e coppie di testi noti chiari/cifrati è possibile creare dei legami fra i bit della chiave, risalendo nella struttura a blocchi e creando e risolvendo equazioni lineare per i bit della chiave
5. RSA resiste agli attacchi Known Ciphertext Attack, Known Plaintext Attack e al Chosen Plaintext Attack, ma non al Chosen Ciphertext Attack. In questo caso l'avversario può ottenere la decifratura di un testo cifrato a sua scelta e, basandosi su alcune proprietà di isomorfismo, è possibile decifrare il messaggio cifrato.

Mostra come Shift Cipher, Semplice sostituzione e Vigenere possono essere rotti subito da un chosen plaintext attack e per i tre cifrari determina il minimo numero di caratteri plaintext necessario per concludere l'attacco.

Per uno Shift Cipher (monoalfabetico) come il cifrario di Cesare può bastare la richiesta di cifratura di un carattere per conoscere la chiave, cioè lo scostamento dell'indice dell'alfabeto utilizzato rispetto all'indice iniziale
es. Invio all'oracolo il plaintext "a" e ottengo "h" per cui determino che la chiave è 7

Per un cifrario a sostituzione di caratteri considerato che la sostituzione è casuale direi che serve richiedere all'oracolo la cifratura degli n-1 caratteri dell'alfabeto
es invio la stringa che contiene gli n-1 caratteri abcdefghjkilmnopqrstuvwxy e ottengo poiuytrewqasdfghjklzxcvbn da cui deduco anche il 26simo carattere z=m

Per Vigenère considerato che Ci = Mi + Ki(mod t) mod 26 e che Mi = Ci - Ki(mod t) mod 26 facendo un paio di operazioni sull'equazione ==> Ki = Ci - Mi mod 26
Quindi se invio un testo di lunghezza almeno pari alla lunghezza della chiave (che magari abbiamo calcolato con IC) otterremo conoscendo ogni Ci e Mi la chiave stessa.

Discutere la resistenza dei cifrari Shift, Hill e Vigenere a known ciphertext e chosen plaintext attack, illustrando con un esempio pratico ognuno degli attacchi

Per lo Shift (scorrimento) e Known Ciphertext la strategia migliore è la ricerca esaustiva, poiché vi sono solo 26 possibili chiavi. Se il messaggio è sufficientemente lungo un altro possibile attacco consiste nel contare la frequenza delle varie lettere. La lettera "e" appare più frequentemente nei testi in inglese, bisognerà quindi cercare nel testo cifrato quale lettera si ripete maggiormente e probabilmente essa sarà una "e" in chiaro. Trovata la prima corrispondenza è possibile determinare lo scorrimento e quindi decifrare le restanti lettere. Per il Chosen Plaintext può bastare la richiesta di cifratura di un carattere per conosce lo scorrimento necessario alla decifratura, ad esempio se richiedo la cifratura di "a" e ottengo "h" lo scorrimento/chiave sarà 7.

Hill risulta resistente ad un Known Ciphertext ma soccombe facilmente ad un attacco Chosen Plaintext Attack, se non si conosce n (intero generante la matrice) si provano diverse possibilità finché non se ne trova una che va bene.

Per Vigenere e Known Ciphertext per molto tempo si è pensato che il metodo fosse sicuro contro questo attacco ma invece studiando il testo cifrato e determinato IC e MIC è possibile trovare la lunghezza della chiave e successivamente la relativa chiave. Per il Chosen Plaintext Attack posso invece richiedere la cifratura del testo in chiaro "AAAAAA…" ottenendo quindi la chiave di cifratura.

Cifratura simmetrica

Discutere la seguente affermazione: "In un generico algoritmo di cifratura a sostituzione a blocchi di n bit, le dimensioni della chiave sono n * 2^n"

Sapendo che: i cifrari a blocchi operano su blocchi di n bit in input per produrre blocchi di n bit in output; con blocchi di n bit di testo in chiaro ho 2n possibili valori del blocco; ogni blocco di testo in chiaro deve produrre un blocco cifrato univoco e che questa mappatura 1 a 1 viene fatta dalla chiave, si può quindi dedurre che servono 2n mappature e che ciascuna mappatura è composta da una sequenza di n bit. Quindi il valore totale della chiave è n*2^n La chiave di un generico algoritmo di cifratura a blocchi si esprime come una tabella di 2n righe e n colonne (grandezza della tabella n*2^n)

ESEMPIO: se ho un blocco di 3 bit, ho i seguenti valori: 000, 001, 010, 011, 100, 101, 110, 111 = 24 = 8 valori. Una possibile cifratura potrebbe essere questa:

000 -> 100
001 -> 101
010 -> 000
011 -> 110
100 -> 011
101 -> 111
110 -> 001
111 -> 010

La chiave, guardando questa tabella, è la colonna di destra. Devo infatti procedere così: il mio blocco ha valore 010, quindi prendo la 010-esima voce della tabella, e ho la sua cifratura. Si vede quindi che la colonna di destra è composta da 8 voci da 3 bit l'una: per l'appunto n*2^n

Si indichino le caratteristiche del cifrario, specificando quanti possibili cifrari diversi è possibile ottenere cambiando il mapping.

Nei cifrari a blocchi con blocchi di n bit di testo in chiaro sono possibili 2^n input. Affinché la trasformazione sia reversibile o non singolare, ogni blocco di testo in chiaro deve produrre un blocco di testo cifrato univoco.
Con blocchi di n bit di testo in chiaro, si hanno 2^n possibili stati di input, mappati in 2^n possibili output (n bit di testo cifrato) con (2^n)! possibili trasformazioni.

Si considerino DES e AES e si calcoli lo spazio necessario per memorizzare le funzioni da loro realizzate in forma tabellare, immaginando di costruire un repertorio del codice.

Per "spazio necessario per memorizzare le funzioni da loro realizzate in forma tabellare" si intende la lunghezza della chiave e risulta essere n*2^n. Quindi sapendo che DES utilizza blocchi di 64bit la grandezza della tabella sarà 64*2^64 , mentre AES prevede 128bit, quindi la grandezza della tabella sarà 128*2^128

Si consideri un cifrario a blocchi che opera su blocchi di n=4 bit.
a. Quanti diversi cifrari esistono? (motivare la risposta)
b. Se si scrive il cifrario ottenuto in forma tabellare, qual è la dimensione della tabella in bit? Generalizzare la risposta al caso di parole formate da n bit
c. Se il cifrario opera su un blocco di 4 bit m=(m1,m2,m3,m4) e con una chiave k=(k1,k2,k3,k4) per restituire il blocco cifrato c=(m1⊕k1, m2⊕k2, m3⊕k3, m4⊕k4), dove ⊕ è lo XOR. Quante sostituzioni sono possibili con questo cifrario?
d. Cifrare il messaggio M1=0110ll0011 con chiave k=(1,1,0,0), e decifrare il cifrato ottenuto

1. 2*n!
2. 2*(2^n * n) perché la tabella ha 2 colonne, una per l'input e una per l'output.
3. Il numero di sostituzioni possibili corrisponde al numero di chiavi che sono 2^4
4. C= 1010 ll 1111, mettendo C nuovamente in XOR con la chiave K viene ritornato in M

Funzioni Hash

Si descriva l'attacco a compleanno e le sue implicazioni per le funzioni hash

Il paradosso del compleanno recita "quante persone devo scegliere, a caso, affinché io abbia la probabilità maggiore del 50% che due compiano gli anni lo stesso giorno?" La risposta è che basta scegliere 23 persone. In una comunicazione firmata attraverso l'hash l'attaccante può generare un certo numeri di variazioni del messaggio originale M. Poi deve generare un certo numero di messaggi fraudolenti (l'opposto del msg originale), in modo che l'hash della variazione fraudolenta sia uguale all'hash di una delle variazioni del messaggio originale. Infine l'attaccante dovrà far firmare al legittimo mittente una delle variazioni del messaggio originale, ma poi inviare al destinatario il messaggio fraudolento. Se ha lo stesso hash, significa che la firma sarà identica.

Si descrivano le proprietà delle funzioni hash

Le funzioni hash sono delle funzioni che ricevono in input una stringa di lunghezza arbitraria *m* e restituiscono come output una stringa di lunghezza fissa *n*.
Uno degli aspetti principali è il seguente: deve poter essere facile e veloce da computare e molto difficile da invertire.

Le proprietà delle funzioni hash sono 3:
- Pre-image resistant: è impossibile risalire da un valore hash al messaggio che l'ha generato.
- Second pre-image resistance: se ho un messaggio M ed il corrispondente valore hash h(M), non sono in grado di calcolare in qualche modo un messaggio Z tale che h(Z)=h(M).
- Collision resistance: non sono in grado di calcolare ex-novo due messaggi M e Z che diano lo stesso hash.

Discutere le proprietà di sicurezza delle funzioni hash

Le funzioni hash sono delle funzioni che ricevono in input una stringa di lunghezza arbitraria m e restituiscono come output una stringa di lunghezza fissa n.
Uno degli aspetti principali è il seguente: deve poter essere facile e veloce da computare e molto difficile da invertire.
Ne possiamo quindi dedurre che deve rispettare le seguenti 3 proprietà:
1. Pre-image resistant : dato y = hash(m) deve poter essere difficile risalire al suo messaggio originale m.
2. Second pre-image resistant : dato y = hash(m) deve essere computazionalmente difficile individuare un m' tale per cui hash(m) = hash(m')
3. Collision resistant: data una coppia di messaggi m ed m' deve risultare h(m) <> h(m') ovvero difficile trovare 2 diversi messaggio con lo stesso hash.

Le funzioni hash sono soggette ad attacchi del paradosso del compleanno.
Se voglio avere funzioni hash "sicure" devo considerare output di almeno 2^160 bit in quanto posso esprimere la probabilità di avere collisione (e epsilon) = 0,5 come sqrt(n)
Avere un probabilità di avere successo nell'attacco pari a 0,5 (quindi generare una collisione = sqrt(n) dove n = num bit output di h
Quindi con 2^40 bit input mi basterà analizzare 2^20 elementi circa.
Con 2^160 bit input dovrò analizzare 2^80 circa elementi circa.
Date le prestazioni dell'attuale tecnologia sappiamo che per stare "al sicuro" dobbiamo avere uno spazio delle chiavi pari a circa 2^80 bit per rendere il calcolo computazionalmente difficile.

Dimostrare che la funzione hash H(x) = 5x + 11 mod 19 non ha la proprietà di resistenza debole alle collisioni, mostrando come sia facile trovare una collisione per H(4).

La sicurezza debole e' definita cosi': dato un messaggio M e' difficile trovare un messaggio M' tale per cui h(M)=h(M') con M diverso da M'

Quindi, procedo a calcolare H(4) = 5*4+11 mod 19 = 12

Ora procedo a risolvere la funzione in x
5x + 11 = 12 mod 19
5x = 1 mod 19
x = 4 mod 19

Per trovare collisioni e' sufficiente prendere un valore dell'insieme 4 in base 19 = { ...., -15 ,4,23,42 , ... }

Quindi per esempio, abbiamo una collisione con M'=23
Infatti H(23) = 23*4+11 mod 19 = 12

Secret Sharing

Descrivere le fasi di distribuzione e ricostruzione del segreto per uno schema (n,n)

Lo schema (n,n) divide il segreto tra n partecipanti, e per ricostruirlo sono necessarie le share di tutti gli n partecipanti.

Un dealer vuole condividere un segreto (S) con n persone, a condizione che siano necessarie le share (a) di tutte le n persone per ricostruire il segreto (S). Ovvero n-1 persone non sono in grado di ricostruire il segreto (S). Il dealer sceglie un numero primo (Z), il segreto (S) e poi vengono fissati n-1 share causali. Viene infine calcolato l'ultimo share con la formula S - a1 - a2 - an mod Z. Ottenuti tutti gli share il dealer li può distribuire in modo sicuro.

Per la ricostruzione ci vogliono le tutte le share, poi basterà calcolare a1+a2+an mod Z per ottenere il segreto (S).

Descrivere le fasi di distribuzione e ricostruzione del segreto per uno schema (k,n)

Lo schema (k,n) invece divide il segreto sempre tra n partecipanti, ma bastano k < n partecipanti per ricostruirlo.

Un dealer vuole condividere un segreto (S) con n persone, a condizione che siano necessarie le share (k) di un numero prestabilito di persone per ricostruire il segreto (S). k-1 share non saranno in grado di ricostruire il segreto (S).
Il dealer sceglie un numero primo (Z), il segreto (S) e poi vengono fisati k-1 valori a casuali.
Successivamente si calcolerà gli share per tutte le n persone e distribuite secondo la formula: yi = S + a1 * i + a2 *i^2 + … mod Z

Per la ricostruzione si può costruire un sistema di equazioni utilizzando eventualmente una matrice.

Descrivere l'applicazione dello schema (3,4) considerando Z11 e il segreto s=9

(3,4) = calcolo e distribuisco 4 share ma sono sufficienti 3 share per ottenere il segreto.

Scelgo a caso k-1 valori di a (a=3):
a1 = 4 - a2 = 3

Calcolo lo share secondo la formula: yi = S + a1 * i + a2 *i^2 + … mod Z

y1 = 9 + 4*1 + 3*1^2 mod 11
y2 = 9 + 4*2 + 3*2^2 mod 11
y3 = 9 + 4*3 + 3*3^2 mod 11
y4 = 9 + 4*4 + 3*4^2 mod 11

Ricostruisco imponendo un sistema di equazioni con più incognite, scegliendo 3 share:

$$\begin{cases} y1 = S + a1 * 1 + a2 * 1^2 \\ y2 = S + a1 * 2 + a2 * 2^2 \\ y3 = S + a1 * 3 + a2 * 3^2 \end{cases} \quad \text{mod } 11$$

Descrivere l'applicazione dello schema (2,3) considerando Z13 e il segreto s=7

(2,3) = calcolo e distribuisco 3 share ma sono sufficienti 2 share per ottenere il segreto.

Scelgo a caso k-1 valori di a (a=2):
a1 = 4

Calcolo gli share secondo la formula: yi = S + a1 * i + a2 *i^2 + … mod Z

y1 = 7 + 4*1 mod 13
y2 = 7 + 4*2 mod 13
y3 = 7 + 4*3 mod 13

Distribuiscono quindi lo share.

Descrivere l'applicazione dello schema (3,3) considerando Z13 e il segreto s=7

(3,3) = calcolo e distribuisco 3 share, tutti e 3 sono fondamentale per ottenere il segreto.

n = 3, Z (ovvero numero primo) = 13, s (segreto) = 7

Scelgo n-1 valori casuali, si possono ripetere:
a1 = 2 a2 = 1

Calcolo a3 con la formula S - a1 -a2 mod Z > 7-2-1 mod 13 = 4
Le share da distribuire sono: a1 = 2 - a2 = 1 - a3 = 4
Per ricostruire il segreto s uso la formula s = a1+a2+a3 mod Z = 2+1+4 mod 13 = 7

Descrivere l'applicazione dello schema (3,3) considerando Z11 e il segreto s=7

(3,3) = calcolo e distribuisco 3 share, tutti e 3 sono fondamentale per ottenere il segreto.

Scelgo n-1 valori casuali, a1 = 2 - a2 = 1
Calcolo a3 > S - a1 - a2 mod Z > 7-2-1 mod 11 = 4
Ricostruisco: s = a1 + a2 + a3 mod Z = 2+1+4 mod 11 = 7

Descrivere l'applicazione dello schema (4,4) considerando Z11 e il segreto s=6

(4,4) = calcolo e distribuisco 3 share, tutti e 3 sono fondamentale per ottenere il segreto.

Scelgo n-1 valori casuali, a1 = 2 - a2 = 1 - a3= 4
Calcolo a4 = S - a1 - a2 - a3 mod Z = 6-2-1-4 mod 11 = -1 mod 11 = 10
Ricostruisco: s = a1 + a2 + a3 + a4 mod Z = 2+1+4+10 mod 11 = 17 mod 11 = 6

Crittosistema DES

Discutere le modalità di cifratura di DES, discutendone vantaggi e svantaggi.

La prima modalità operativa di DES è l' ELECTRONIC CODEBOOK CHAINING: è il modello più semplice in cui ciascun blocco in chiaro viene codificato in modo indipendente, sempre con la stessa chiave. Vantaggi: è il metodo più semplice e veloce ed eventuali errori non si propagano. Svantaggi: usando sempre la stessa chiave, uno stesso blocco in chiaro viene cifrato allo stesso modo e quindi per messaggi lunghi non è sicuro perché soggetto ad attacchi di sostituzione.

La seconda modalità è il CIPHER BLOCK CHAINING: vi è dipendenza tra i blocchi perché l'input si ottiene come XOR tra il blocco in chiaro corrente e il blocco cifrato precedente. Vantaggi: a differenza dell'ECB, non si generano blocchi uguali in output grazie alla loro dipendenza; se si cambia un solo bit del messaggio originario, cambia di conseguenza tutto il rimanente messaggio; questo vincolo di dipendenza evita attacchi di sostituzioni dei blocchi. Svantaggi: la dipendenza dei blocchi genera però un rallentamento del sistema e la propagazione di errori.

La terza modalità è il CIPHER FEEDBACK con s bit. Non utilizza blocchi di 64bit ma lavora con segmenti di s bit < 64. Anche in questo caso vi è dipendenza tra i blocchi poiché la cifratura del blocco precedente viene utilizzata per cifrare il blocco successivo. Vantaggi: il valore di s può essere scelto a piacimento e il valore della cifratura viene prodotto subito. Svantaggi: per valori di s molto piccoli, il sistema diventa più oneroso e vengono propagati gli errori.

La quarta modalità è l'OUTPUT FEEDBACK. La struttura è simile al CFB, l'unica differenza è che la cifratura del blocco corrente si esegue con l'output di DES del blocco precedente (e non il blocco cifrato). Vantaggi: Non si propagano gli errori di trasmissione dei bit. Svantaggi: E' più vulnerabile del CFB a modifica di flusso.

La quinta e ultima modalità è il COUNTER. Si impiega un contatore delle dimensioni del blocco in chiaro, e viene incrementato per ogni blocco. Vantaggi: non vi è dipendenza tra i blocchi e quindi si possono eseguire in parallelo le cifrature di diversi blocchi. Se si conosce prima il contatore, si può pre-calcolare l'output del DES ed eseguire poi solo un'operazione di XOR; si ha un accesso causale, la sicurezza è dimostrabile ed è semplice in quanto richiede solo l'algoritmo di crittografia.

Discutere il funzionamento e la sicurezza di DES doppio

Il DES doppio prevede che il messaggio in chiaro venga cifrato due volte (con l'algoritmo di DES) con due chiavi diverse in sequenza (una per ogni passo di cifratura). In questo modo la lunghezza della chiave raddoppia passando da 56 a 56*2 = 112 bit, con un notevole incremento dello spazio delle chiavi. Dato in input un blocco x di 64bit del testo in chiaro e due chiavi k1 e k2 a 56bit, cifriamo x con la chiave k1 ed otteniamo un blocco A di 64bit; cifriamo poi A con la chiave k2 ed otteniamo il corrispondente testo cifrato Y:

$Y = DESk2(DESk1(x))$

La decifratura è analoga alla cifratura, ma richiede l'applicazione delle chiavi in ordine inverso:

$X = DESk1\text{-}1(DESk2\text{-}1(y))$

Anche se può sembrare un sistema apparentemente più resistente, il DES doppio non aumenta la sicurezza del DES a singola cifratura, anzi esiste un algoritmo che è in grado di rompere tale cifrario dimostrando che è equivalente alla cifratura dello stesso messaggio con una sola chiave, ed è l'attacco Meet in the Middle. Quindi utilizzando questo attacco, anche se il DES doppio utilizza una chiave di 112bit può essere rotto con uno sforzo dell'ordine di 256, come per il DES singolo. Per questo motivo il DES doppio non viene utilizzato in pratica.

Descrivere l'attacco Meet in the Middle a DES doppio

Si tratta di un attacco known plaintext, ovvero l'attaccante conosce il testo in chiaro ed anche il suo cifrato corrispondente: x e z in questo schema.
Si procede così:
- dato x, calcolo tutte le possibili crittografie usando tutte le 256 chiavi possibili, e le salvo in una tabella
- dato z, calcolo tutte le possibili decrittografie e le salvo in una tabella
- confronto le 2 tabelle per vedere le voci che coincidono
- scopro la password usando un'altra coppia di testo in chiaro / testo cifrato

Quello che si fa in pratica è provare a cifrare in tutti i modi possibili il testo in chiaro x di cui dispongo. Cifrare in tutti i modi possibili vuol dire provare a cifrare x con tutte le 256 chiavi. Poi, devo provare a decifrare z, anche lui con tutte le 256 chiavi possibili. In pratica, cerco di arrivare ad y sia da sinistra che da destra.

Siccome la chiave di 56 bit è ben minore di 64 * 264 bit, allora ci saranno chiavi diverse che, applicate allo stesso blocco, porteranno allo stesso risultato, sia in cifratura che in decifratura: ecco perché ci sarà più di una coincidenza, e occorre una seconda coppia di testo in chiaro/testo cifrato.

Si discutano le caratteristiche della crittoanalisi differenziale o lineare per DES

Crittoanalisi lineare: recupera la chiave a partire da 2^43 coppie di testi in chiaro noti;
Crittoanalisi differenziale: recupera la chiave a partire da 2^47 coppie di testi scelti.

Discutere il vantaggio di utilizzare strutture Feistel in DES

I vantaggi di utilizzo delle strutture Feistel sono:
- che la cifratura e la decifratura sono processi invertibili, in quanto la decifratura usa lo stesso algoritmo per la cifratura con l'unica differenza in cui le sottochiavi vengono applicate nell'ordine inverso. Questo semplifica enormemente l'implementazione;
- l'alternanza di sostituzioni, permutazioni ed espansioni (grazie alla funzione di Feistel) che forniscono la cosiddetta diffusione e confusione, cioè rispettivamente l'espansione della struttura statistica del testo in chiaro (cioè assegnare più lettere del testo in chiaro ad una lettera di testo cifrato – es. bigrammi) e la complicazione della relazione esistente fra testo cifrato e la chiave.

Crittosistema AES

Descrivere le principali caratteristiche del cifrario AES e le operazioni di cifratura e di decifratura

AES è un algoritmo di cifratura a blocchi utilizzato ormai come standard in sostituzione di DES e non implementa una rete di Festitel, ma usa uno schema ben diverso. Inoltre, si basa per i suoi calcoli interni su di un tipo particolare di campo, il campo finito 2^8. Le operazioni aritmetiche vengono compiute all'interno di questo campo.

Le sue caratteristiche sono:

Dimensioni blocco: 128, 192 o 256 bit
Dimensioni chiave: 128, 192 o 256 bit
Numero di round (fasi): 10, 12, 14 (chiave lunga, più round)
Schedulazione della chiave: 44, 52 o 60 sottochiavi a 32bit

Per la cifratura e decifratura vengono eseguiti diversi round composti da 4 stadi differenti (SubBytes, ShiftRows, MixColumns, AddRoundKey). La fase di decifratura prevede un ordine delle operazioni differente con uno scambio a due a due degli stadi.

Discutere la possibilità di usare AES come cifrario a flussi

AES è un cifrario a blocchi, non a flussi, questo li permette di avere un più alto livello di sicurezza a discapito della velocità di esecuzione. RC4 può essere un cifrario alternativo basato sullo schema a flussi.

Si discuta la resistenza o meno di AES agli attacchi

Nel AES-128 la lunghezza della chiave è di 4 word(128 bit),la dimensione del blocco è di 4 word e il numero di rounds è 10; con 128 bit ci sono 2^128 chiavi possibili.

Nel AES-192 la lunghezza della chiave è di 6 word(192 bit), la dimensione del blocco è di 4 word e il numero di rounds è 12; con 192 bit ci sono 2192 chiavi possibili.

Nel AES-256 la lunghezza della chiave è di 8 word(256 bit), la dimensione del blocco è di 4 word e il numero di rounds è 14; con 256 bit ci sono 2256 chiavi possibili.

Una macchina che prova 255 chiavi al secondo impiega 149.000 miliardi di anni per rompere AES. Si pensa che AES resisterà per i prossimi 20 anni.

Diffie Hellmann

Descrivere le fasi di generazione dei parametri per lo schema di Diffide-Hellmann

L'idea è la seguente:
1) Alice sceglie un numero qualsiasi, che mantiene segreto. Chiameremo questo numero NM1.
2) Bob sceglie un altro numero qualsiasi, che, a sua volta, mantiene segreto. Chiameremo questo numero NP1.
3) Successivamente, sia Alice che Bob applicano ai loro rispettivi numeri una funzione dei tipo f(x) = ax mod p, essendo p un numero primo conosciuto.
Alice ottiene da detta operazione un nuovo numero, NM2, che questa volta invia a Bob.
Bob ottiene da tale operazione un nuovo numero NP2, che invia a Alice.
4) Alice risolve un'equazione del tipo e ottiene come risultato un nuovo numero, CM.
5) Bob risolve un'equazione del tipo e ottiene come risultato un nuovo numero, CP.
Anche se sembra sorprendente, CM e CP saranno uguali.

Sia q=13 il numero primo comune e sia il generatore g=2
* **Dimostrare che 2 è una radice primitiva di Z13**
* **Se Alice ha chiave pubblica 4 qual è la sua chiave privata?**
* **Se Bob ha chiave privata 5 qual è la sua chiave pubblica?**
* **Qual è la chiave segreta condivisa?**

Trovo la fattorizzazione di q-1 -> q = 13-1 -> q = 6 *2

g^(q-1/2) != 1 mod 13 -> 2^6 != 1 mod 13 -> 12 != 1 -> Dimostrato
g^(q-1/6) != 1 mod 13 -> 2^2 != 1 mod 13 -> 6 != 1 -> Dimostrato

Trovo la chiava privata di Alice (pubblica= 4):
g^x = 4 mod 13 -> 2^x = 4 mod 13 -> x = 2

Trovo la chiave pubblica di Bob (privata=5)
g^5 = y mod 13 -> 2^5 = y mod 13 -> y = 6

Trovo la chiave segreta condivisa (Privata Alice= 2, Privata Bob=5)

g ^(2*5) mod 13 -> 2^(2*5) mod 13 -> 2^10 mod 13 -> 10

Mostrare un possibile attacco del tipo "man in the middle", fornendo un esempio numerico

Se un attaccante riesce ad impossessarsi del canale sul quale Alice e Bob comunicano, può giocare un bello scherzetto ad entrambi. Si presuppone che il Cattivo sia in grado di intercettare tutti i messaggi provenienti da Alice o Bob, e reinviarli senza che questi si accorgano di nulla.

Come dicevamo all'inizio, i valori p e g sono pubblici. Quindi, quando Alice crede di comunicare con Bob, in realtà comunica con Cattivo, e quello che succede è che crea un segreto tra lei e Cattivo! Alice crede di condividerlo con Bob, invece lo condivide con Cattivo.
Allo stesso tempo, Cattivo si spaccia per Alice e comunica a Bob un valore gc, e in pratica concorda con Bob un'altra chiave.
Quello che succede è che Alice manda un messaggio a Cattivo, con la chiave concordata con Cattivo, mentre lei è invece convinta di mandare un messaggio a Bob con la chiave concordata con Bob. Idem per quest'ultimo.
Cattivo prende il messaggio da Alice, lo guarda tranquillamente, e lo reimpacchetta con la chiave di Bob, il quale è convinto di ricevere roba da Alice.

RSA

Discutere le fasi di generazione dei parametri in RSA e le eventuali difficoltà nella loro generazione.

Bob scegli due primi p e q grandi e destini e li moltiplica per formare il numero n = p*q, sceglie inoltre un esponente di cifratura e tale che MCD(e,(p-1)(q-1)) = 1. Invia quindi ad Alice la chiave pubblica formata dalla coppia n,e e tiene segreti i valori di p e q, ovvero la chiave privata. Alice scriverà il suo messaggio come un numero m, se è più grande di n spezzerà il messaggio in blocchi minori di n. Quindi Alice calcolerà c = m^2 mod n, e invia c a Bob.
Conoscendo p e q Bob può calcolare (p-1)(q-1) e quindi può trovare l'esponente di decifrazione d con de = 1 mod((p-1)(q-1)), successivamente procede alla decifratura con m = c^d mod n.

Le difficoltà nella generazione dei parametri sono le seguenti: trovare p e q primi questo viene fatto generando numeri casuali dispari (per raddoppiare le probabilità) e poi facendo un test di primalità. un'altra difficoltò è trovare l'inverso per "e" in modϕ(n). Per ottenere l'inverso si utilizza l'algoritmo di euclide esteso. Un'altro vincolo è sul messaggio M da cifrare: m < n.

Discutere i principali attacchi alla cifratura asimmetrica RSA

Sfruttando la tecnica di Chosen Ciphertext Attack, ovvero l'attaccante può farsi decifrare un testo a sua scelta, l'attaccate si fa cifrare il messaggio C*x^e ed è quindi in grado di ottenere la decifratura di M. Infatti decifrando C*x^e otterrei che M = M*x mod d, quindi divido per x e ottengo ottengo M a partire dalla sua cifratura.

Sfruttando la tecnica di Common Modulus Attack, vi sono due persone la cui chiave contiene lo stesso n e ad entrambi viene inviato lo stesso messaggio M. Allora l'attaccante è in grado di risalire ad M senza avere la chiave, ma avendo solo i due messaggi cifrati C1 e C2. È in grado tramite l'algoritmo di Euclide esteso di calcolare due numeri r e s che abbiano la seguente proprietà: e1*r + e2*s = 1 mod n dove e1 e e2 sono le due chiavi pubbliche.

Sfruttando la tecnica di Low Exponent Attack quando diverse chiavi pubbliche hanno lo stesso valore di e, e hanno le rispettive n prime tra loro. Occorre inoltre che lo stesso messaggio M venga inviato ai vari utenti, sfruttando quindi il teorema del resto che un sistema che coinvolge tutti i messaggi M è possibile calcolare il valore decifrato di M.

Discutere i principali attacchi sull'implementazione di RSA

Sono due attacchi particolari, che riguardano le implementazioni di RSA in hardware.

Il primo attacco è detto timing attack: se un attaccante è in grado di osservare quanto tempo ci mette un calcolatore di velocità nota a produrre i risultati quando calcola i vari valori di RSA, è in grado di risalire alla chiave privata.
Lo stesso avviene per il power attack: se l'attaccante è in grado di monitorare il consumo di corrente di un processore che calcola RSA (ricordando che per calcoli complessi l'uso di corrente sale), allora può risalire alla chiave privata.

Per proteggersi da ciò basta introdurre dei calcoli casuali qua e là per sbarellare le osservazioni.

Si discuta l'utilizzo "naive" di RSA e la sua insicurezza nel caso known plaintext (quando ad esempio i messaggi da cifrare siano pochi).

Se i messaggi sono pochi allora un avversario può enumerarli e cifrarli tutti con la chiave pubblica. A questo punto quando l'avversario vede un ciphertext C gli basta confrontare quest'ultimo con tutte le cifrature che si è calcolato in precedenza per dare in output il plaintext di C.

Si consideri una doppia cifratura usando un modulo comune N e due chiavi pubbliche e1 ed e2. Un messaggio M è cifrato usando il cifrario RSA con chiave pubblica e1, ed il risultato cifrato ancora con la chiave pubblica e2. Tale sistema aumenta la sicurezza della cifratura? Analizzare il sistema e dare una spiegazione

Cifrare due volte un messaggio con RSA non garantisce maggiore sicurezza. Prendiamo il messaggio M, la prima cifratura con RSA restituirebbe C=M^e1. Poi cifriamo C utilizzando e2 quindi ottenendo F=C^e2=M^(e2*e1). Quindi F altro non è che la cifratura di M con la public key e1*e2, di conseguenza non c'è alcun incremento di sicurezza.

Discutere i metodi per ottimizzare le computazioni nella fase di cifratura

Nella cifratura o decifratura si esegue il calcolo x^y mod Z che è molto oneroso, poiché sia x che y sono due primi grandi e distinti. Pertanto la rappresentazione binaria di y è:

y = yi*2^i + …+ yn*2^n

È quindi possibile sfruttare l'ottimizzazione left-to-right o right-to-left, nella prima il numero di operazioni che devo fare è lo stesso numero dei bit che compongono y. Quindi, se y è un numero di 512 bit, allora eseguo 512 operazioni, e non 2^512! Con right-to-left il numero di operazioni che eseguo è polinomiale nella lunghezza in bit di y.

Crittosistema Blowfish

Descrivere le principali caratteristiche del cifrario Blowfish le operazioni di cifratura e di decifratura

Il cifrario di Blowfish è un cifrario a blocchi che usa la rete di Feistel e sfrutta S-Box non predeterminate: esse vengono calcolate a partire dalla chiave. Infatti per ogni nuova chiave, l'algoritmo Blowfish viene eseguito 521 volte per riempire l'array delle sottochiave e l'array della S-Box. Per calcolare le sottochiavi e le S-Box, si inizializzano gli array con i numeri decimali di n.

Caratteristiche:
Blocco = 64bit
Chiave = da 32 a 448 bit
Sottochiavi = 18
Fasi = 16

Nella fase di cifratura un blocco viene XOrato con la sottochiave, e passato alla funzione F. La funzione F fa uso della S-Box e il risultato viene a sua volta XOrato con il successivo blocco. In questo modo, tutti e due i blocchi vengono modificati in ogni fase, mentre nella rete di Feistel normale un blocco finiva direttamente nella fase successiva.

La decifratura è identica, ma le chiavi sono schedulate in ordine inverso. In particolare, le chiavi vanno da 18 a 3, per ognuna delle 16 fasi "standard".

Algoritmo di EL Gamal

Descrivere i parametri dell'algoritmo

El Gamal è un algoritmo di cifratura asimmetrica che si basa sull'intrattabilità dell'algoritmo discreto. Facendo un esempio se Alice vuole mandare un messaggio M a Bob esso sceglie un numero primo p grande e una radice primitiva di α (la chiave privata). Si assuma che m sia un intero tale che 0 <= M < p. Se M è grande, lo si spezzi in blocchi più piccoli. Bob sceglierà inoltre un intero g segreto e calcolerà la chiave pubblica β = g^α mod p. L'informazione (p, g, β) verrà resa pubblica ed è la chiave pubblica di Bob.

Pertanto avremo:

- p è un numero primo
- g è un generatore di Zp
- α è un numero casuale, α < p - 1
- β = g^α mod p

Chiave privata: p, g, α.
Chiave pubblica: p, g, β.

Alice procederà nel modo seguente:

- Ottiene la chiave pubblica di Bob (p, g, β);
- Sceglie a caso un intero k segreto e calcola Y1 = g^k mod p
- Calcola Y2 = M * β^k mod p (dove M è il messaggio)
- Manda a Bob la coppia Y1 e Y2

Bob infine decifra calcolando:

Z = Y1^Privata mod p
M = Z^-1 * Y2 mod p

Fare cenni sulla possibilità di applicare El-Gamal su curve ellittiche

L'operazione di somma nelle curve ellittiche è l'analogo della moltiplicazione modulare.

Utilizzando i concetti il crittosistema di el gamal, la chiave privata è formata dal numero x che è stato scelto, i 2 parametri pubblici di dominio, ossia i punti data dalla curve scelta Ep(a,b) e il punto g generatore. La chiave pubblica sarà data dal punto particolare PA ottenuto con x * G.

La cifratura del messaggio PM avviene in maniera analoga a el gamal su campi Zp. Supponiamo che il messaggio sia codificato come un punto particolare PM da cifrare. Spponendo che bob vuole inviare tale punto ad alice, andrà a prendere i parametridella chiave pubblica e calcolare un intero casuale K, cifrare il punto Pm in 2 passaggi successvi: Prima calcola il punto C1 ottenuto con K*G e successivamente calcolare il secondo punto C2 trovato come somma di Pm + K * Pa .

Firma DSS

Descrivere le fasi di generazione dei parametri della firma DSS

DSS sta per digital signature standard, ed è uno standard per la firma digitale. DSA invece è l'algoritmo usato da DSS per realizzare la firma digitale. Si basa sull'intrattabilità dei logaritmi discreti e sulla funzione hash SHA. Le firme digitali devono comportarsi esattamente come le firme normali. Hanno le seguenti caratteristiche:

* devono poter essere prodotte facilmente dal legittimo firmatario
* nessuno deve essere in grado di falsificarle
* chiunque invece può verificarle

I parametri sono

p = un numero primo
q = un numero primo, tale che q l (p-1), ovvero q deve essere divisore di p-1
α = il numero di ordine q in Zp* (sotto vediamo che cos'è)
s = un numero casuale, s < q
β = αs mod p

Chiave privata: p, q, α, s
Chiave pubblica: p, q, α, β

Firma DSS cosa succede se viene firmato lo stesso messaggio in due diverse occasioni? Qual è la differenza con RSA?

Nel processo di firma DSS vi è l'utilizzo di un numero random r, quindi se il messaggio venisse firmato due volte con DSS, otterremmo una firma differente. Entrambi valide ma differenti.

Nel processo di firma di RSA non ci sono numeri casuali e quindi due firme fatte sullo stesso messaggio saranno identiche.

Quanti tentativi approssimativamente devono essere fatti per trovare dei primi validi p e q da utilizzare nella firma?

Innanzitutto nell'algoritmo viene scelto prima a caso q di 160 bit, e poi viene scelto a caso p tale che p è primo e maggiore uguale di 2^511.
Quindi il numero di tentativi è legato alla probabilità che scegliendo un elemento a caso <2^160 esso sia primo e dispari unito alla probabilità che scegliendo un elemento a caso esso sia primo e maggiore di 2^511.
Sia Q la probabilità di trovare un primo q come descritto sopra:-Sia ora pi(x) la funzione che mi restituisci la quantità di numeri primi minori uguali ad x. La probabilità di pescare un primo q come serve a noi è dunque pi(2^160)/2^160.

Per p invece le cose cambiano un po'.
Sia P la probabilità di trovare un primo p come descritto sopra : -per calcolare la probabilità di beccare un primo maggiore di 2^511 basta fare: (pi(2^512)-pi(2^511))/(2^512-2^511).
Quindi il numero di tentativi che devo fare per beccare un p ed un q come descritti è 1/Q+1/P.

Descrivere le fasi di generazione e verifica della firma DSS

Per firmare il messaggio M, Alice, che possiede la chiave privata, deve eseguire i seguenti passi:

genera un numero r casuale, con r in [1, q - 1]
γ = (αr mod p) mod q
δ = (SHA(M) + s*γ) * r-1 mod q
invia il messaggio M assieme a γ e δ

Notiamo che r-1 esiste, in mod q, perché q è primo e quindi il MCD(r, q) = 1.
Verifica

Quando Bob riceve il messaggio con γ e δ, deve eseguire i seguenti passi per verificare:

e' = (SHA(M) * δ'^-1^) mod q
e'' = (γ * δ'^-1) mod q

Poi, deve verificare la seguente uguaglianza:

$\alpha e' * \beta e'' = \gamma$

Se l'uguaglianza è verificata, la firma è autentica. Altrimenti, la firma è falsa.

Cifrario Affine

Si descriva il funzionamento dei cifrari affini.

I cifrari affine sono un caso particolare di cifrario a sostituzione, la sostituzione è data da una funzione detta affine.

ci = E(pi) = (k1 pi + k2) mod 26

La chiave è quindi data da da k1 e k2,

La decrittazione invece seguirà questa formula:

pi = D(ci) = (ci - k2) * k1^-1 mod 26

Scegliendo come chiave la coppia (3,5), si scriva la funzione di decifratura, con la stessa chiave si cifri e decifri la parola "vacanze".

Se k1 = 3 e k2 = 5

La funzione di cifratura è: (3p+5) mod 26 dove p è la lettera in chiaro
La funzione di decifratura è: 3^-1(c-5) mod 26 = 9 (c-5) mod 26

Testo in chiaro VACANZE

V = 21
Cifro la V, ovvero 21: (3*21 + 5) mod 26 = 68 mod 26 = 16 = Q

Decifro la Q, ovvero 16: 9(16+5) mod 26 = 99 mod 26 = 21

E proseguo per ogni lettera del testo in chiaro

Funzioni MAC

Descrivere caratteristiche e applicazioni delle funzioni MAC.

In crittografia un Message Authentication Code (MAC) è un piccolo blocco di dati utilizzato per l'autenticazione di un messaggio digitale e per verificarne l'integrità da parte del destinatario, generato secondo un meccanismo di crittografia simmetrica: un algoritmo MAC accetta in ingresso una chiave segreta ed un messaggio da autenticare di lunghezza arbitraria, e restituisce un MAC (alle volte chiamato anche tag). In ricezione il destinatario opererà in maniera identica sul messaggio pervenuto in chiaro ricalcolando il MAC con lo stesso algoritmo e la stessa chiave: se i due MAC coincidono si ha autenticazione e integrità del messaggio inviato.

Descrivere l'approccio HMAC e fornire una costruzione con una funzione hash nota

HMAC è una tipologia di codice per l'autenticazione di messaggi (Message Autentication Code - MAC) basata su funzione di hash utilizzata in diverse applicazioni legate alla sicurezza informatica. Tramite 68

HMAC è infatti possibile garantire sia l'integrità che l'autenticità di un messaggio. HMAC utilizza infatti una combinazione del messagio originale e una chiave segreta per la generazione del codice. Il vantaggio di HMAC è il non essere legata a nessuna funzione di hash particolare, questo per rendere possibile una sostituzione della funzione nel caso fosse scoperta debole.

Il messaggio viene suddiviso in blocchi di lunghezza pari a j bit. Seleziono una chiave segreta K, se questa risulta essere più lunga di j bit a questa applico la funzione H. Quello che si ottiene è detta K', la chiave di HMAC.

Il miglior attacco conosciuto è basato sul paradosso del compleanno. Per attaccare HMAC sono necessarie molte coppie M,HMAC: l'avversario non può calcolarle perché non conosce K, e deve quindi osservare un flusso di messaggi generati con la stessa chiave.

Vigenere

Si descrivano le caratteristiche del cifrario di Vigenere

Il cifrario di Vigenere è un cifrario a sostituzione polialfabetica. Dato un testo in chiaro M ed una chiave di lunghezza n, si divide il testo in chiaro in blocchi di n caratteri, pari alla lunghezza della chiave; per crittografare un messaggio, la chiave deve essere lunga quanto il messaggio stesso e poiché di solito la chiave è di lunghezza minore del messaggio, viene ripetuta. Grazie alla ripetizione della chiave, il vantaggio di questo sistema di cifratura è che due o più lettere uguali nel messaggio in chiaro potranno essere cifrate in modo differente. Esempio:

Stringa: CIFRARIODIVIGENERE
Verme: INFORMATICALABINF
Codice: KVKFRDIHLKVTGFVRWS

La crittoanalisi mediante frequenza delle lettere risulta quindi limitata. Per poter crittografare/decrittografare il messaggio in chiaro/cifrato veniva utilizzata per semplicità una matrice di 26x26 (le 26 lettere dell'alfabeto) in cui si ricercava la coppia "carattere del testo in chiaro – carattere della chiave". Ora si possono utilizzare le seguenti formule:

Crittografia : $Ci = (Mi + K) \mod t$
Decrittografia: $Mi = (Ci – K) \mod t$

dove t = lunghezza della chiave = 26, Mi = carattere del testo in chiaro Ci = carattere cifrato (a = 0, b = 1 …. Z = 25), K = chiave

Cosa significa nel caso di Vigenere adottare una doppia cifratura (cioè l'utilizzo in cascata di due chiavi k e k' di cifratura)? Si illustri il ragionamento con un esempio.

Una doppia cifratura prevede l'utilizzo di due parole chiavi k per cifrare la nostra parola in chiaro. Si avrà quindi uno struttura come la seguente:

Parola: ESAMEFACILE
k: ABCD
k': QWER

ESAMEFACILE
ABCDABCDABC
QWERQWERQWE

Per cifrare E: 4+0+16 mod 26 = 20 => U

Attacchi al cifrario di Vigenere

La sicurezza dell'algoritmo dipende dal fatto che non sono note né la parola chiave ne la sua lunghezza. L'analisi delle frequenze non sarà utile per rompere il cifrario, un attacco di tipo Known Plaintext Attack ha successo esclusivamente se è noto un numero sufficiente di caratteri, poiché la chiave si ottiene semplicemente sottraendo il testo in chiaro ad testo cifrato in modulo 26. Un attacco di tipo Chosen Plaintext Attack che sfrutta il testo in chiaro "aaaaa…" ci fornirà immediatamente la chiave, mentre al contrario un attacco Chosen Cipthertext Attack con lo stesso metodo ci fornirà l'opposto della chiave. Nell'ultima situazione di Known Ciphertext Attack per molto tempo si è pensato che l'algoritmo fosse sicuro a questo attacco ma una recente crittoanalisi ci permette di ottenere facilmente la chiave. Il metodo consiste nel trovare la lunghezza della chiave grazie all'indice di coincidenza e successivamente potrò determinare la chiave grazie all'indice di mutua coincidenza.

Descrivere il significato e l'utilizzo degli indici di coincidenza e mutua coincidenza nella crittoanalisi del cifrario di Vigenere.

INDICE DI COINCIDENZA (IC): è utilizzato per calcolare la probabilità che due lettere prese a caso in una stringa x1x2…xn siano uguali. Il suo utilizzo nella crittoanalisi serve per determinare la lunghezza della chiave. Grazie a questo indice è anche possibile capire in che lingua è stato scritto il messaggio in chiaro.

INDICE DI MUTUA COINCIDENZA (MIC): è la probabilità che due lettere prese a caso in una stringa x1x2…xn e in un'altra stringa y1y2…yn siano uguali. Il suo utilizzo nella crittoanalisi serve per determinare il valore della chiave

Con password “REBUS” si cifri il testo “ESAMEFACILE”

ESAMEFACILE
REBUSREBUSR

E = 4 / R = 17
4+17 mod 26 = 21 = v
E si prosegue lettera per lettera

VWBGWWEDCDV

Si calcoli IC della parola “caccia”

Formula: [∑ da i=0 a i=25 fi*(fi – 1)]/[n*(n – 1)]

frequenza f parola caccia : a = 2 c = 3 i = 1

n = lunghezza della parola caccia = 6

f0 = a = 2, (f1 = b = 0), f2 = c = 3, …. f8 = i = 1 …. (f25 = z = 0)

quindi:

IC = [2 (2 – 1) + 3 (3 – 1) + 1 (1 – 1)]/[6 (6 – 1)] = 4/15

Si calcoli MIC delle parole “birba” e “babbo"

Formula: [∑ da i=0 a i=25 (fi*fi')]/n*n'

frequenza f parola birba: a = 1 b = 2 i = 1 r = 1

frequenza f' parola babbo: a’ = 1 b’ = 3 o’ = 1

n = lunghezza della parola birba = 5

n’ = lunghezza della parola babbo = 5

f0 = a, f1 = b, …. f8 = i …. f14 = o, f17 = r, …

MIC (birba, babbo)= [(1*1) + (2*3) + (1*0) + (1*0) + (1*0)]/(5*5) = 7/25

Certificati digitali

Si illustri la funzione di un certificato digitale e si chiarisca qual è il ruolo di una autorità di certificazione

Un certificato digitale è un documento elettronico che attesta l'associazione univoca tra una chiave pubblica e l'identità di un soggetto (una persona, una società, un computer, ecc). Un Certificato Digitale, usualmente in formato X509, contiene il periodo di validità della chiave pubblica, un identificativo della chiave, informazioni sull'utente (lingua, localizzazione, ecc) e lo stato della chiave (valida o non valida).

La revoca del certificato è una parte molto importante e si può verificare per vari motivi, anche su richiesta del proprietario, come la compromissione del certificato privato, cambio di informazioni, perdita di informazioni di sicurezza (pin/token) o l'utilità del certificato è cessata e quindi non serve più.

L'autorità di certificazione (CA), anche essa autenticata, è una terza parte fidata la cui firma garantisce il legame tra chiave ed identità. L'autorità di certificazione e' un mezzo (tra altri) per la distribuzione sicura delle chiavi pubbliche.

Si faccia un esempio di come Alice e Bob possano usare un certificato rilasciato da una stessa CA e si faccia un esempio di utilizzo di una certification path

Due interlocutori Alice e Bob, che sfruttano la stessa CA, prima di iniziare una comunicazione dovranno richiedere all'Autorità di Certificazione la Certificazione Digitale della propria chiave pubblica, una volta ottenuta i due utenti si scambieranno il certificato digitale e potranno iniziare a interloquire.

Due interlocutori Alice e Bob, che sfruttano due distinte CA, dovranno seguire il percorso di certificazione dei diversi CA coinvolti per poter richiedere la certificazione della propria chiave pubblica. Ogni CA è collegata alla successiva tramite una gerarchia e si garantiscono mutua fiducia l'una con l'altra.

Playfair

Discutere la natura e i possibili attacchi al cifrario di Playfair

Il cifrario di Playfair è un cifrario a sostituzione basato sui bigrammi. Fa uso di una matrice 5x5 di 25 caratteri diversi, si inserirà all'inizio della matrice la chiave scelta (senza ripetere eventuali doppie) e poi in ordine alfabetico si inseriranno le rimanenti lettere raggruppando la i e la j. Il testo in chiaro dovrà diviso in gruppi di due lettere, se vi è una lettera doppia in uno dei gruppi si inserisce una x e si ripete il raggruppamento. Infine se necessario si aggiunge una x alla fine per completare l'ultimo gruppo. Ora si procederà ad usare la matrice per cifrare ogni gruppo di due lettere rispettando un preciso schema.

- Se le due lettere non solo sulla stessa riga o sulla stessa colonna, si sostituisce ogni lettera con la lettera che si trova sulla medesima riga e sulla colonna dell'altra lettera.
- Se le due lettere sono sulla stessa riga, si sostituisce ogni lettera con la lettera immediatamente alla sua destra.
- Se le due lettere sono sulla stessa colonna, si sostituisce ogni lettera con la lettera immediatamente sotto.

Un possibile attacco a questo cifrario è il Known Ciphertext Attack, poiché conoscendo il testo cifrato è possibile utilizzare metodi statistici per risalire al messaggio in chiaro: utilizzando l'analisi delle frequenze dei bigrammi più comuni nella lingua. Un'altra debolezza è che ogni lettera del testo in chiaro ha solo cinque possibili lettere corrispondenti nel testo cifrato. Infine a meno che la chiave non sia molto lunga le ultime righe della matrice sono prevedibili.

Discutere i vantaggi rispetto alla cifratura monoalfabetica e le debolezze del sistema Playfair

Il vantaggio rispetto alla cifratura monoalfabetica è che con Playfair si hanno 676 possibili digrammi (26x26), a differenza delle sole 26 lettere della monoalfabetica; dunque l'identificazione dei singoli bigrammi è più difficile. Nonostante ciò, la debolezza di questo sistema è che la struttura del testo rimane identica, e quindi un attacco statistico come l'analisi delle frequenze dei digrammi più comuni nella lingua è valido contro Playfair

Utilizzare la parola chiave "viva le vacanze" per cifrare il testo in chiaro "giornali" e decifrare il testo cifrato "hehinadi"

Tabella usata:

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

giornali > GI OR NA LI > PN PS ZI EA
hehinadi > HE HI NA DI > MA GA ZI NE

Hill

Discutere la natura e i possibili attacchi al cifrario di Hill

Il cifrario di Hill è un cifrario a sostituzione polialfabetica basato sull'algebra lineare. Nella fase di cifratura ogni lettera è codificata in un numero, un blocco di n lettere è quindi uno spazio vettoriale di dimensione n, e moltiplicato per una matrice n x n, modulo 26. La matrice è la chiave del cifrario, quindi casuale.
Il cifrario di Hill è vulnerabile ad un attacco known-plaintext attack (KPA) in quanto è completamente lineare. Un attaccante con n coppie di caratteri in chiaro e cifrati può impostare un sistema lineare che può essere risolto facilmente. Può capitare di incorrere in un sistema indeterminato, ma basterà aggiungere altre coppie di testo cifrato/chiaro per riuscire a risolverlo.

Descrivere il cifrario di Hill, discutendo vantaggi e svantaggi

Il cifrario di Hill è un cifrario a blocchi inventato nel 1929 da Lester Hill, il calcolo è abbastanza semplice ed inoltre è abbastanza resistente agli attacchi basati sulle frequenze. In altre parole è resistente ad attacchi che si basano solo sulla conoscenza del testo cifrato. Al contrario il suo principale svantaggio è che soccombe facilmente ad un attacco di testo in chiaro noto, infine le matrici chiave devono essere invertibili, quindi per esempio non vanno bene prese a caso.

Si consideri l'usuale corrispondenza tra l'alfabeto ed i numeri da 0 a 25

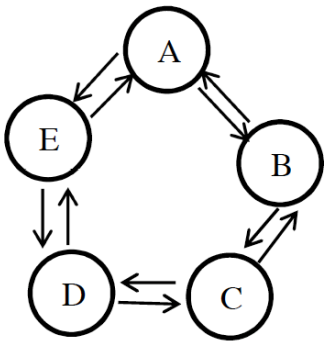
|k11 k12|
|k21 k22|
la chiave usata ed "m1 m2 m3 m4" il plaintext

Esprimere matematicamente il corrispondente testo cifrato "c1 c2 c3 c4"

c1=m1*k11 + m2*k21
c2=m1*k12 + m2*k22
c3=m3*k11 + m4*k21
c4=m3*k12 + m4*k22

Sistemi di Cifratura

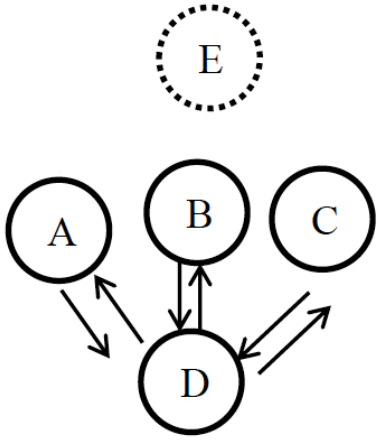
Si consideri una rete i cui nodi sono collegati come in figura:



- **Quante chiavi sono necessarie per fare in modo che A, B,C, D ed E possano comunicare con sicurezza in modo bidirezionale usando un sistema di cifratura simmetrico come indicato dalle frecce?**
- **Se si rimpiazza il sistema simmetrico con uno asimmetrico, quante chiavi sono necessarie?**
- **Se si aggiunge un nuovo canale di comunicazione fra A e D (immagina una doppia freccia fra A e D), quante chiavi sono necessarie in entrambi i casi?**
- **Se A vuole comunicare con D con sicurezza incondizionata, quale sistema deve utilizzare? In questo caso cosa si può dire sulla chiave?**

1. Le connessioni dirette tra un nodo ed un altro sono da considerarsi sicure, pertanto servono le seguenti chiavi (sia K(AB) la chiave segreta tra A e B): K(AC), K(AD), K(BD), K(BE), K(CE) per un totale di 5 chiavi segrete condivise;
2. In un sistema asimmetrico serve una coppia di chiavi (Kpub e Kpriv) per ogni utente, per un totale di 5 coppie di chiavi
3. Se si assume che A e D sfrutteranno solo il nuovo canale di comunicazione per il sistema simmetrico servono ora 4 chiavi ovvero: K(AC), K(BD), K(BE), K(CE) mentre per il sistema asimmetrico nulla cambia.
4. A e D dovrebbero usare un cifrario one-time-pad con la chiave delle stessa dimensione del messaggio da scambiare.

Si consideri una rete i cui nodi A,B,C e D sono collegati come in figura:



- **Quante chiavi sono necessarie per fare in modo che A, B e C possano comunicare con sicurezza con D in modo bidirezionale usando un sistema di cifratura simmetrico?**
- **Se si rimpiazza il sistema simmetrico con uno asimmetrico, quante chiavi sono necessarie?**
- **Se si aggiunge un nuovo nodo E in comunicazione con A, B e C, quante chiavi sono necessarie in entrambi i casi (simmetrico e asimmetrico)?**
- **Se B vuole comunicare con D con sicurezza perfetta, quale sistema deve utilizzare? In questo caso cosa si può dire sulla chiave?**
- **Nel caso di una rete con n nodi tutti comunicanti fra di loro, quante chiavi sono necessarie nei due casi?**

1. Basterà una sola chiave utilizzata da tutti i nodi, infatti il nodo B ad esempio non si trova su un path di comunicazione che va da A verso D, quindi non ha modo di intercettarne il traffico.
2. Basterà una coppia di chiave privata/pubblica
3. Per il caso simmetrico basterà aggiungere una sola chiave, ovvero una per permettere la comunicazione tra E e D in modo tale che i nodi A, B e C non capiscano cosa D comunica ad E e viceversa, nonostante possono osservare la comunicazione che avviene tra i due. Per la comunicazione tra da E verso A, da E verso B, e da E verso C può essere riutilizzata la chiave che serviva alla comunicazione tra A verso D, B verso D e C verso D. Per il caso asimmetrico vale lo stesso discorso, cioè 2 coppie di chiavi (pubblica, privata) sono sufficienti.
4. B e D Dovrebbero utilizzare one-time pad al fine di ottenere sicurezza perfetta. In questo caso quindi la lunghezza della chiave sarebbe pari alla lunghezza del messaggio che vogliono cifrare. Inoltre B e D devono utilizzare una chiave diversa per ogni cifratura che vogliono calcolare. Di conseguenza questi due nodi dovrebbero conservare un repertorio di chiavi la cui dimensione dipende dal numero di messaggi che vogliono cifrare e di conseguenza inviare.
5. Perché tutti possano comunicare con tutti utilizzando un qualunque path di comunicazione in sicurezza, per il caso simmetrico, ogni coppia di nodi deve avere una chiave privata. Dunque sono necessarie n*(n-1) chiavi. Cioè 12 chiavi. Per il caso asimmetrico invece basta che ogni nodo abbia la sua coppia (chiave provata, chiave pubblica). Quindi in tutto bastano 4 coppie di chiavi.

Feistel

Descrivere vantaggi e svantaggi dell'utilizzo delle strutture di Feistel.

Nella struttura di feistel, Il testo in chiaro viene diviso in un blocco sinistro e in un blocco destro. La parte destra diventa la parte sinistra del livello successivo. Per la parte sinistra, invece, viene eseguito uno XOR con una funzione generata con la sottochiave.
I vantaggi sono che:
Nella Cifratura basta implementare un solo round, e lo stesso codice può essere usato per ogni round.
Nella Decifratura si usa lo stesso algoritmo per la cifratura, ovvero che lo stesso codice può essere usato sia per cifrare che decifrare, inoltre vengono usate le sottochiavi in ordine inverso.

Gli svantaggi sono che:
Dimensioni di blocchi grandi migliorano la sicurezza ma riducono la velocità
Dimensioni delle chiavi grandi migliorano la sicurezza ma riducono la velocità

Principali Formule

Diffie Hellman
Verifica radice primitiva
Fattorizzo q: q-1 e trovo due multipli
g^(q-1/PrimoMultiplo) != 1 mod Z
g^(q-1/SecondoMultiplo) != 1 mod Z
Chiava Privata
g^x = Pubblica mod Z
Chiave Pubblica
g^Privata = x mod Z

Vigenere
LetteraInChiaro + LetteraChiave mod 26

Secret Sharing
Schema (n,n)
Scelgo n-1 valori casuali e poi calcolo il recente
a = S - n valori mod Z
Ho quindi n share disponibili, per ricostruire
S = somma share mod Z
Schema (n,x)
Scelgo n-1 valori casuali e li sfrutto per calcolare n share
Y1 = S + a1 * 1 + a2 * 1 ^2 mod 13
Y2 = S + a1 * 2 + a2 * 2 ^2 mod 13
Per ricostruire si fa un sistema di equazioni con gli share.

El Gamal
Chiave Pubblica
g^Privata mod p

Cifrare (con k < p-1)
Y1 = g^k mod p
Y2 = M * Pubblica^k mod p

Decifrare
Z = Y1^Privata mod p
M = Z^-1 * Y2 mod p

Hill
Cifrare
C = Elementi Matrice mod 26
Decifrare
Inverto diagonale principale del sistema e colato l'inverso additivo
In modulo dei restati. Quel numero che in mod 26 mi da 0.

Affine
Funzione Cifratura
K1*p+K2 mod 26 (p lettera in chiaro)
Funzione Decifratura
K1^-1(c-K2) mod 26 (c lettera cifrata)

IC
NrLettera (NrLettera -1) + … / TotLettere(TottLettere-1)

MIC
(NrLettera1 * NrLettera2) + … / NrLettere1* NrLettere2