1)

Definire e distinguere le proprietà desiderate per una buona funzione di hash *"debolmente resistente alle collisioni"* e *"fortemente resistente alle collisioni"*. Data una funzione candidata $h = h(m)$, quale delle due proprietà vi sarebbe più facile tentare di violare? Perché? *(3 punti)*

Risposta 1: I due tipi di funzioni di hash sono molto simili, semplicemente una ha una caratteristica ancora più restrittiva dell'altra.
Una funzione di hash debolmente resistente alle collisioni è una funzione che rende, conoscendo un input x che ha come risultato il suo hash y = h(x), la computazione di un altro messaggio con il medesimo hash un problema risolvibile in un tempo tendente ad infinito, ovvero che il conoscere un messaggio e il suo dato hash non mi dia alcuna informazione per trovarne un altro che collida con esso.
Invece una funzione di hash fortemente resistente alle collisioni non ha l'ipotesi della conoscenza di un messaggio e del suo hash ma afferma che è un problema computazionalmente infattibile il trovare una coppia di messaggi che collidano.
E' più facile violare la proprietà forte perché sono necessari un numero di controlli pari a $2^{n/2}$ rispetto a 2^n

2)

Descrivere lo *Schema di Lamport* per l'autenticazione di un host Alice da parte di un server Bob, precisando quali informazioni sono conosciute segretamente da A e B, e quali informazioni sono invece pubbliche o trasferite in chiaro da A o B. Descrivere lo *Small*-n *Attack* portato da Oscar a Bob e Alice. *(4 punti)*

Risposta 2: Lo Schema di Lamport permette l'autenticazione attraverso un utilizzo di un insieme finito e generato di OTP.
Il server conosce un numero causale n per ogni nome utente, e sempre ogni nome utente il corrispettivo risultato della funzione di hash applicata n volte alla password p nota solo al client.
Di conseguenza il client conosce n e la password.
Il server quindi dopo aver scelto un n casuale, inviato al client, riceve da esso nel caso voglia autenticarsi il risultato di una funzione di hash applicata n - 1 volte ad una password p che è ignota al server. Il server controlla se l'hash del messaggio ricevuto corrisponda all'hash inviato inizialmente dal client, dell'applicazione per n volte della funzione di hash sulla password.
Si continua in questo modo per n – 1 volte per garantire l'autenticazione, dopo le n – 1 volte è necessario ricominciare creando un nuovo n casuale perché altrimenti verrebbe trasmessa la password p.
Lo Small-n Attack consiste nell'impersonare inizialmente il server inviando un piccolo valore di n che chiamiamo k nell'ordine delle decine. Il client A credendo sia il vero server invia il risultato della funzione di hash applicata k – 1 volte alla password al server impersonato.
Ora colui che ha impersonato il server può autenticarsi come client A per un numero di volte pari a n – k.

3)

Si supponga di avere un sistema di autenticazione di utenti basato su biometria. Il pattern del candidato k è confrontato con il pattern memorizzato per l'utente A, misurandone la *distanza* d_{kA} secondo un'opportuna metrica. Con che criterio si decide la soglia di accettazione D, per cui il candidato è accettato come A se $d_{kA} < D$? *(2 punti)*

Risposta 3: E' necessario calcolare le funzioni di FAR (False Acceptance Rate) e di FRR (False Rejection Rate) che hanno come dominio tutti i possibili valori di soglia, la soglia d'accettazione migliore per una certa autenticazione biometrica è quella che rende le immagini delle due funzioni uguali, viene chiamato ERR (Equal Error Rate) e di conseguenza tanto più basso è tanto più il sistema d'autenticazione biometrica è preciso.

4)

Descrivere in sintesi le principali differenze tra *Transport Mode* e *Tunnel Mode* per l'opzione *Encapsulating Security Payload* (ESP) di IPsec. *(3 punti)*

Risposta 4: Le due principali differenze tra i due modi di operatività sono che nella modalità Transport Mode viene cifrato il solo payload del pacchetto incapsulato, mentre nella modalità Tunnel Mode viene cifrato l'intero pacchetto IP incapsulato insieme ad altri campi di sicurezza.
L'incapsulamento invece, ovvero l'inserimento di un pacchetto IP come payload in un altro pacchetto IP avviene nel medesimo modo per entrambi modi di operatività.

5)

Descrivere in sintesi il principio del protocollo HTTPS. Rispetto al protocollo base HTTP, quali informazioni trasmesse dall'utente che si connette a un server www vengono cifrate? Quali protocolli stabiliscono una connessione end-to-end tra i *peer* su *client* e *server*? (percorrere la pila dall'alto al basso) *(3 punti)*

Risposta 5: Il protocollo HTTPS è la combinazione del protocollo HTTP assieme a TLS/SSL per permettere l'utilizzo di contenuti ipertestuali in maniera sicura. A differenza di HTTP opera sulla porta, di default, 443.
Nel protocollo HTTPS viene cifrato : URL, contenuti della risorsa indicata dall'URL, contenuti di eventuali form, eventuali cookies scambiati da/a client server, tutto il contenuto dell'header HTTP.
Sono presenti 3 livelli di comunicazione end-to-end per HTTPS, si parte dal livello maggiore quello applicazione dove è presente HTTP e le richieste di comunicazione vengono inviate al protocollo del livello inferiore ovvero SSL/TLS, dopo che sessioni e connessioni SSL/TLS sono inizializzate si passa al livello ancora inferiore, quello di trasporto, dove verrà utilizzato TCP.

6)

Cos'è una funzione di *hash* $y = h(x)$?

Risposta 6: Una funzione di hash y = h(x) è una funzione che dato un input di qualsiasi lunghezza restituisce una stringa di lunghezza prefissata. Una funzione di hash deve inoltre garantire la proprietà di unidirezionalità, di non invertibilità e la proprietà di cercare di avere un numero di collisioni tendente a 0 dato un numero di input diversi che tenda ad infinito.

7)

Definire la proprietà di *unidirezionalità* di una funzione di *hash*, distinguendola dalla proprietà di *non invertibilità*. Fare un esempio di funzione di hash non invertibile ma non unidirezionale.

Risposta 7: La proprietà di unidirezionalità costituisce non l'impossibilità matematica di poter trovare un certo input che dia un medesimo hash, ma bensì la sua enorme difficoltà a farlo in un tempo non tendente ad infinito.
La non invertibilità è invece una restrizione matematica della funzione stessa, ottenibile per il fatto che la funziona di hash accetta in ingresso un insieme infinito di possibili input ma il suo codominio è limitato, di conseguenza essa non è una funziona invertibile.
Un esempio di funzione di hash non invertibile ma non unidirezionale è x mod n.

8)

Nella suite di protocolli *Transport Level Security (TLS)*: *(3 punti)*
quali servizi di trasporto sicuro fornisce il *TLS Record Protocol*? quali altri protocolli li utilizzano?
quali funzioni svolge *Handshake Protocol*? a chi fornisce i suoi servizi? che protocollo trasporta i suoi messaggi?

Risposta 8: TLS Record Protocol è utilizzato per la comunicazione sicura di protocolli di livello superiore come HTTP che diventa in questo modo HTTPS e fornisce due servizi principali: confidenzialità e integrità
Handshake Protocol permette la creazione di sessioni TLS, necessarie per la creazione di varie connessioni e di conseguenza per l'autenticazione client / server insieme alla negoziazione del protocollo di cifratura da scegliere.
Il trasporto dei suoi messaggi avviene attraverso TLS Record.

9)

Qual è la funzione del comando STARTTLS in SMTP? Qual è il suo effetto? E in IMAP e POP3?

Risposta 9: STARTTLS nel protocollo SMTP permette l'avvio di una sessione TLS/SSL per poter comunicare in modo sicuro, visto che per come è definito lo standard SMTP esso è un protocollo dove la comunicazione, per via testuale, avviene in chiaro. Un ulteriore effetto, a seconda dell'e-mail server, è il passaggio dalla porta di default 25 alla 587.
Nei protocolli IMAP e POP3 si ha il medesimo effetto dove si passerà rispettivamente a IMAPS con porta 993 e a POP3S con porta 995

10)

Quali sono i ruoli dell'*Autentication Server* e del *Ticket-Granting Server* in Kerberos? Cosa significa se il primo autorizza un client ma il secondo no? Citare un miglioramento o estensione introdotto in Kv5 rispetto a Kv4.*(3 punti)*

Risposta 10: Autentication Server e Ticket-Granting Server sono necessari per l'inizializzazione del processo di accesso ad una determinata risorsa. AS si occupa inizialmente, ricevendo una richiesta d'autenticazione del client il suo id e password. Esso fornirà un Ticket necessario per l'ottenimento di altri Ticket per l'accesso a determinate risorse o servizi attraverso il Ticket-Granting Server.
Se AS autorizza un client ma TGS no significa che nonostante il client sia autenticato non ha accesso ad alcuna risorsa o servizio.
Una delle interessanti migliorie in Kerberos v5 è la generalizzazione dei protocolli di cifratura e indirizzamento prima non possibile.

11)

Che differenza c'è tra i protocolli di *symmetric key agreement* e *symmetric key distribution*? In cosa consiste un *replay attack* ai protocolli di distribuzione delle chiavi? Come possono essere impediti? Fare almeno un esempio.*(3 punti)*

Risposta 11: La differenza tra Symmetric Key Agreement e Symmetric Key Distribution sta nel fatto che la prima necessita che gli host interessati si debbano calcolare le chiavi in cooperazione mentre la seconda fornisce ai vari host le chiavi già calcolate da un server.
Un Replay Attack consiste nell'invio di un messaggio cifrato da qualcuno o qualcosa di non Trusted nel tentativo di impersonare qualcuno/qualcosa.
Per impedire un Replay Attack è necessario impedire che un messaggio cifrato già inviato possa essere riutilizzato, quindi si possono per esempio utilizzare come contromisure un timestamp oppure un numero di sequenza, così da non permetterne il riutilizzo.

12)

In IPsec, cos'è una *IP Security Policy*? Qual è la sua funzione principale? A cosa si applica? *(2 punti)*

Risposta 12: Una IP Security Policy è un insieme di regole presenti nei database di SA (Security Association) e forniscono informazioni per determinare come e quando è necessario utilizzare IPSec

13)

Nella suite di protocolli *Transport Level Security (TLS)*, quali funzioni svolge *Handshake Protocol*? In particolare, specificare anche quante chiavi crea e per quali scopi. *(3 punti)*

Risposta 13: Handshake Protocol permette la creazione di sessioni TLS, necessarie per la creazione di varie connessioni e di conseguenza per l'autenticazione client / server insieme alla negoziazione del protocollo di cifratura da scegliere.
Vengono create 6 diverse chiavi, 3 per il cient 3 per il server. La prima coppia di chiavi è utilizzata per la cifratura dei dati trasmessi, quindi possono essere chiavi AES o 3DES in genere.
La seconda coppia di chiavi è per il MAC, necessario per garantire l'integrità.
La terza coppia di chiavi è necessaria per l'IV (Initialization Vector) ma necessaria solo se si è in modalità CBC (Cipher Block Chaining)

14)

Cos'è un *elemento primitivo* $\alpha \in \mathbb{Z}_p^*$? Quanti sono gli elementi primitivi di $\mathbb{Z}_{947}^*$? *(2 punti)*

Risposta 14: a è un elemento primitivo se e solo se il suo ordine è p -1, cioè il massimo, dove l'ordine di un certo valore è il più piccolo n tale per cui a^n mod p è congruente ad 1.

15)

Si consideri un certificato di Alice emesso da un'Autorità TA: $C_A = \{ A, K_A, \{h(A, K_A)\}_{K^{-1}TA} \}$. Chi possiede K_{TA}? A cosa serve K_{TA}? Che informazione serve a chi firma il certificato C_A? Se verifico che la firma del Certificato C_A è valida, posso fidarmi che chi mi ha passato il certificato sia veramente Alice? A cosa serve la *Revocation List*? (*3 punti)*

Risposta 15: K_{ta} è la chiave pubblica del CA (o TA) utilizzabile per decifrare l'hash cifrato dalla corrispettiva chiave privata K^{-1}_{ta} posseduta solo da CA (o TA). Per poter firmare il certificato C_A è necessario l'identificativo di A e la sua chiave pubblica, e, come detto prima, la chiave privata di CA (o TA).
Il certificato ha lo scopo di garantire l'identità di qualcuno attraverso la fiducia riposta nel CA (o TA), ma non garantisce che quel qualcuno sia affidabile.
La Revocation List è necessaria per identificare i certificati che nonostante non siano scaduti (perché essi hanno una scadenza temporale) sono stati revocati e quindi è necessario considerarli non più validi. Questo può accadere per vari motivi quali per esempio la compromissione della chiave privata o la compromissione del certificato di CA.

16)

Descrivere sommariamente il processo di autenticazione e cifratura di un messaggio PGP. *(3 punti)*

Risposta 16: Il processo di autenticazione e cifratura PGP tra A e B inizia con A che effettua una serie di operazioni. Calcola l'hash del messaggio da inviare, firma l'hash con RSA, genera una chiave di cifratura simmetrica da 128 bit, viene cifrata la coppia firma e messaggio con 3DES e viene cifrata la chiave di cifratura simmetrica con RSA. B, dopo aver effettuato i passaggi inversi, finisce con la verifica che la firma del messaggio sia valida garantendo quindi l'integrità e l'autenticità del messaggio.

17)

A cosa serve il Protocollo di Needham-Schroeder? Chi sono gli interlocutori del protocollo? Qual è la sua caratteristica principale e come evita i *replay attack*? *(3 punti)*

Risposta 17: Il protocollo Needham-Schroeder serve per lo scambio di chiavi tra due interlocutori A e B attraverso un server autoritario S. Può essere utilizzato sia per chiavi simmetriche che asimmetriche.
La sua caratteristica principale è l'inviare un pacchetto cifrato ad A contenente al suo interno la chiave simmetrica di Sessione ed sempre al suo interno un ulteriore pacchetto cifrato ma decifrabile solo da B, dopo che A lo invierà a B, esso scoprirà al suo interno la medesima chiave simmetrica di Sessione, quindi ora A e B possono comunicare.
I replay attack vengono evitati grazie all'utilizzo di NONCE all'interno di ogni pacchetto inviato a S.

18)

Descrivere l'*attacco dell'intruso* al Protocollo di Instaurazione della Chiave di Diffie-Hellman. Come è possibile ostacolarlo? *(2 punti)*

Risposta 18: L'attacco dell'intruso si basa sull'intercettare i due valori di a elevati alle rispettive potenze x e y e al posto di far arrivare ai destinatari questi valori vengono inviati scegliendo un certo z ad entrambi gli host un a^z mod p, così che l'intruso possa calcolare le chiavi di entrambi aggiungendo come esponente moltiplicativo z.
Si può evitare utilizzando la migliorata versione chiamata Station-to-Station che aggiunge la funzionalità d'autenticazione attraverso la firma digitale.

19)

Qual è il vantaggio di usare una modalità di concatenazione (CFB, CBC, ….) di un cifrario a blocchi, se il vettore di inizializzazione è trasmesso in chiaro e non è tenuto segreto con la chiave? *(2 punti)*

Risposta 19: Il vantaggio sta che utilizzando queste modalità ogni blocco cifrato influenza il successivo in modo tale che non è possibile avere alcuna correlazione.

20)

Enunciare il *Teorema Cinese del Resto* generalizzato a K congruenze. *(2 punti)*

Risposta 20:

Si supponga che $n_1, \ldots, n_k$ siano interi a due a due coprimi (il che significa che MCD(n_i , n_j) = 1 quando $i \neq j$). Allora, comunque si scelgano degli interi $a_1, \ldots, a_k$, esiste un intero x soluzione del sistema di congruenze

$$x \equiv a_i \pmod{n_i} \quad \text{per } i = 1, \ldots, k.$$

Inoltre, tutte le soluzioni x di questo sistema sono congruenti modulo il prodotto $n = n_1 \ldots n_k$.

Si può trovare una soluzione x come segue. Per ogni i gli interi n_i e n/n_i sono coprimi, e utilizzando l'algoritmo di Euclide esteso si possono trovare due interi r e s tali che $r\, n_i + s\, n/n_i = 1$. Ponendo $e_i = s\, n/n_i$, si ottiene

$$e_i \equiv 1 \pmod{n_i} \quad \text{e} \quad e_i \equiv 0 \pmod{n_j}$$

per $j \neq i$. Una soluzione del sistema di congruenze è quindi

$$x = \sum_{i=1}^{k} a_i e_i.$$

21)

Cosa garantisce un *certificato di identità* emesso da una CA in una PKI? Quale procedura segue un utente per verificare l'autenticità di quel certificato? *(2 punti)*

Risposta 21: Esso garantisce le informazioni su una entità o persona a cui quel certificato è associato.
Può essere verificato decifrando con la chiave pubblica di CA (o TA) l'hash cifrato con la chiave privata di CA (o TA).

22)

Descrivere le proprietà di *diffusione* e *confusione*, che secondo Claude Shannon un buon crittosistema dovrebbe avere per ostacolarne l'analisi. *(2 punti)*

Risposta 22: La diffusione costituisce la proprietà che data un certo input, in chiaro, cambiando un solo bit di esso cambierà in maniera pseudocasuale con probabilità P = 0.5 ogni bit del suo corrispettivo cifrato.
La confusione costituisce la proprietà che data una certa chiave K, cambiando un solo bit di essa cambierà in maniera pseudocasuale con probabilità P = 0.5 ogni bit del corrispettivo che avrà cifrato.

23)

Cos'è una *One-Time Password*? Come può essere generata in un *meccanismo di autenticazione a Sfida e Risposta*? Fare un esempio. *(3 punti)*

Risposta 23: Una One-Time Password è una password valida per una solo richiesta d'autenticazione o d'accesso a risorsa o servizio.
Attraverso il meccanismo Challenge and Response può essere generato una serie di n OTP con il Lamport's Hash Chain Scheme [discusso precedentemente].

24)

Fornire un esempio di *chiave monouso* (*One-Time Password*), come può essere generata e come viene utilizzata. *(2 punti)*

Risposta 24: Un esempio di chiave monouso può essere un valore numerico generato attraverso il Lamport's Hash Chain Scheme, oppure un timestamp che generalemente è espresso col numero di millisecondi passati dal 1 Gennaio 1970

25)

Quanti sono i possibili hash di lunghezza 48 bit? Descrivere un attacco del compleanno che miri a ottenere una firma valida di un documento fraudolento con hash di lunghezza 48 bit. *(3 punti)*

Risposta 25: Numero di hash = 2^{48}
Attacco del compleanno :

Consideriamo il seguente esperimento. Da un insieme di valori H scegliamo n valori uniformemente a caso, consentendo quindi anche ripetizioni degli stessi. Poniamo $p(n;H)$ la probabilità che durante l'esperimento almeno un valore sia scelto più di una volta. La probabilità può essere approssimata a

$$p(n;H) \approx 1 - e^{-(n(n-1))/2\cdot H} \approx 1 - e^{-n^2/2\cdot H},$$

Poniamo adesso $n(p;H)$ come il più piccolo numero dei valori che abbiamo scelto, tale che la probabilità che possiamo aspettarci di trovare una collisione sia almeno p. Invertendo l'espressione qui sopra, troviamo la seguente approssimazione

$$n(p;H) \approx \sqrt{2\cdot H\cdot \ln\left(\frac{1}{1-p}\right)},$$

ed assegnando la probabilità di trovare una collisione a 0,5 arriviamo a

$$n(0.5;H) \approx 1.1774\sqrt{H}.$$

Poniamo $Q(H)$ come il numero previsto di valori che dobbiamo scegliere prima di trovare la prima collisione. Il numero può essere approssimato da

$$Q(H) \approx \frac{\pi}{2}\sqrt{H}.$$

26)

Perché tutti i sistemi di firma digitale sono basati sull'applicazione dell'algoritmo di firma all'*hash* del messaggio e non direttamente al messaggio stesso? *(2 punti)*

Risposta 26: perché nonostante a livello di sicurezza non cambia nulla firmare il messaggio o firmare il suo hash, cambia invece il costo computazionale per la firma di un hash (una serie di relativamente pochi bit) rispetto a magari la firma di un messaggio di vari milioni di bit.

27)

Quali sono le tre informazioni fondamentali contenute in un *certificato di identità* in una PKI? Se il certificato è autentico, cosa garantisce? Descrivere la procedura di verifica della sua autenticità. *(2 punti)*

Risposta 27: Un certificato d'identità basato su X.509 ha come informazioni fondamentali il certificato, l'identificativo dell'algoritmo di firma del certificato e la firma del certificato.
Se il certificato è autentico garantisce solo la conferma dell'identità di qualcuno o qualcosa, non della sua affidabilità.
Può essere verificato decifrando con la chiave pubblica di CA (o TA) l'hash cifrato con la chiave privata di CA (o TA).

28)

Fare un esempio di *Linear Feedback Shift Register* utilizzato per generare una sequenza di bit pseudo-casuali. Quale può essere il periodo massimo della sequenza generata? *(2 punti)*

Risposta 28: Additivo autosincronizzante, periodo massimo è 2^{n-1}

29)

Perché sapere risolvere il Problema del Logaritmo Discreto è condizione sufficiente per risolvere il *Problema Computazionale di Diffie-Hellman*? Spiegare in cosa consiste quest'ultimo e come la soluzione di un PLD può darne la soluzione. *(2 punti)*

Risposta 29: Perché il Problema Computazionale di Diffie-Hellman si basa sul fatto di poter trovare due esponenti x e y di a^x e di a^y per poter calcolare a^{x*y} mod n. Non si sa invece se è necessario.

30)

Quali operazioni nell'algoritmo di cifratura AES hanno lo scopo di ottenere *confusione*, proprietà che un buon crittosistema dovrebbe avere secondo Claude Shannon? *(2 punti)*

Risposta 30:
La prima operazione delle 4 per ogni round, Sub-Bytes crea confusione andando ad effettuare una sostituzione attraverso la tabella di sostituzione S-Box, le cui coordinate per ogni input sono ottenibili prendendo i primi 4 bit da sinistra degli 8 in ingresso come riga espressi in base 10, i restanti come colonna espressi in base 10.

31)

Come è stata costruita la tabella S-Box nell'Algoritmo Rijndael? *(2 punti)*

Risposta 31: La tabella S-Box è costruita partendo dal blocco di 8 bit in ingresso a_i, viene calcolato il suo inverso, a_i^{-1} nel campo di Galois di 256, $GF(2^8)$, e espresso come vettore trasposto per poi subire una trasformazione lineare matriciale dove viene moltiplicato per un matrice 8x8 costante e sommato ad un vettore 8x1 costante.

32)

Descrivere sommariamente l'*algoritmo AES*. Specificare:
lunghezza delle chiavi;
ruolo e funzioni dei *round* (ingresso, uscita, *layer*);
funzioni dei singoli *layer*.

Risposta 32: L'algoritmo di cifratura a chiave simmetrica AES si base sul ripetere 4 operazioni computazionalmente semplici per almeno 10 volte, 10 perché non è conosciuto alcun attacco capace di essere più veloce di un attacco forza bruta per AES con un numero di round maggiore di 7.
Ogni round si basa appunto su 4 operazioni (layer) chiamate Sub-Bytes (Substitute Bytes) dove vengono effettuate delle sostituzioni attraverso una S-Box (tabella di sostituzioni) per creare confusione. Poi si bassa ad effettuare la rotazione di ogni riga delle 4 presenti per blocco, la prima riga viene ruotata di 0 bit, la seconda di 1 bit, la terza di 2 bit e la quarta di 3 bit.
Successivamente si passa all'operazione Mix Columns dove viene effettuata un'operazione di moltiplicazione nel campo delle matrici con una matrice costante.
Lo step finale è costituito dall'aggiunta della chiave di round che viene effettuata attraverso l'operazione di XOR tra la matrice del blocco fin ora elaborato con le precedente operazione e la matrice costituente la chiave secondo lo scheduler delle chiavi di AES per ogni round.
Le chiavi di default sono lunghe 128 bit, ma AES può funzionare anche con chiavi di 192 e 256 bit.

33)

Si considerino le funzioni di cifratura doppia $C = E_{K_2}(E_{K_1}(P))$ e sua decifratura $P = D_{K_1}(D_{K_2}(C))$, con due chiavi K_1 e K_2 ciascuna di lunghezza $n = 32$ bit, $P \in \mathbb{Z}_{256}$, $C \in \mathbb{Z}_{256}$. Si intende tentare un attacco *Meet-in-the-Middle* per trovare la coppia di chiavi K_1, K_2. *(2 punti)*
Quali informazioni è necessario conoscere per eseguire l'attacco? L'attacco ha sempre successo?
Si indichi con E il peso computazionale di una operazione di cifratura semplice $E_K(X)$, uguale al peso di una decifratura $D_K(X)$. Quanti calcoli sono necessari (in termini di E) per completare l'attacco con successo?
Quale occupazione di memoria [byte] è necessaria per completare l'attacco con successo?

Risposta 33: E' necessario conoscere una coppia testo chiaro, testo cifrato
L'attacco a meno di complessità temporali elevate ha sempre successo.
La complessità temporale dell'attacco è $O(2^{|k|+1}+1)$.
La complessità spaziale dell'attacco è $O(2^{|k|})$.

34)

Si consideri l'equazione $x^2 \equiv a \pmod{n}$, con $n = p \cdot q \cdot r \cdot s$, dove p, q, r, s sono interi primi >2. Quante soluzioni può avere al massimo questa equazione? Dare almeno un cenno di spiegazione. *(2 punti)*

Risposta 34: E' necessario utilizzare il teorema cinese del resto per potere così risolvere delle radici quadrate con modulo un numero primo.
La risposta non può essere completamente ben specificata perché per effettuare il calcolo delle radici quadrate in un'equazione congruenziale è necessario che il modulo, numero primo, sia congruente a 3 modulo 4 o almeno congruente a 1 modulo 4.

35)

Si considerino le funzioni di cifratura doppia $C = E_{K_2}(E_{K_1}(P))$ e decifratura $P = D_{K_1}(D_{K_2}(C))$ con due chiavi K_1 e K_2 ciascuna di lunghezza n bit. Nel sistema di cifratura a chiave pubblica RSA, esiste una terza chiave e_3 tale che la cifratura doppia $C = E_{e_2}(E_{e_1}(P))$ sia equivalente a una cifratura singola $C = E_{e_3}(P)$? Se la risposta è sì, specificarne il valore. Se la risposta è no, spiegare perché è impossibile. *(2 punti)*

Risposta 35: Sì, basta prendere una chiave $e_3 = e_1 * e_2$

36)

Descrivere il principio di un *cifrario a permutazione* su blocchi di *n* simboli. Si tratta di un cifrario mono- o poli-alfabetico? In cosa consiste la sua chiave? Quante sono le chiavi possibili, nel caso i simboli siano parole di 4 bit?*(2 punti)*

Risposta 36: Un cifrario a permutazione si basa su un vettore di n valori numerici i cui valori K_i sono compresi tra 1 e n. Esso si basa su effettuare una permutazione basata sui valori del vettore. Prendendo per esempio blocchi da 3 bit. I vettori hanno dimensione 3. La permutazione consiste nello scambiare il bit alla posizione corrispondente alla riga del vettore a cui mi sto riferendo con la posizione indicata dal valore contenuto nella riga del vettore stesso. Questa operazione viene effettuata per ogni blocco.
La chiave di questo cifrario è quindi un vettore che indica le posizioni delle permutazioni.
Le possibili chiavi per parole, quindi blocchi, da 4 bit sono tutte le possibili permutazioni che si possono effettuare su di esse ovvero 4! = 16.

37)

Cos'è un *application proxy*? Spiegarne il principio di funzionamento quando è impiegato come un firewall per proteggere una rete. Indicarne un vantaggio e uno svantaggio rispetto a un firewall vero e proprio. *(3 punti)*

Risposta 37: Un application proxy è un software o dispositivo di rete che si posiziona sul livello più alto della pila ISO/OSI o TCP-IP, ovvero al livello applicazione.
Il suo funzionamento è basato sull'analizzare il contenuto dell'intero pacchetto in ricezione ed è proprio questo il vantaggio rispetto ad un firewall tradizionale perché ha la possibilità di poter visionare l'intero pacchetto compreso il payload per poterne analizzarne il contenuto. Può farlo proprio perché si trova al livello più alto.
Uno svantaggio è che spesso un proxy è presente all'end-point e non, per esempio guardando la topologia della rete locale, prima che il pacchetto arrivi ai corrispettivi end-point. Potrebbe inoltre risultare più lento per via della maggior computazione richiesta.

38)

Cosa è SHA? A cosa serve? Quale risultato producono SHA-1 e SHA-2? *(3 punti)*

Risposta 38: SHA è una funzione di hash, necessaria per creare un'impronta unica per un certo input.
SHA-1 produce un hash di lunghezza pari a 160 bit, mentre SHA-2, dipendentemente dalla versione utilizzata, produce un hash la cui lunghezza può essere una delle seguenti : 224, 256, 384 e 512 bit.

39)

Descrivere le caratteristiche di un sistema Feistel.

Risposta 39: Un sistema Feistel è basato sulla divisione dell'input in due blocchi: sinistra e destra.
La cifratura avviene attraverso l'utilizzo di una funzione f che ha come argomento il blocco di destra una chiave i-esima. Il risultato di questa funzione viene messa come argomento di uno XOR con l'input di sinistra.
Questo risultato costituirà l'output di destra.
L'output di sinistra è costituito invece dall'input di destra.

40)

Si considerino le funzioni di cifratura doppia $C = E_{K_2}(E_{K_1}(P))$ e decifratura $P = D_{K_1}(D_{K_2}(C))$ con due chiavi K_1 e K_2 ciascuna di lunghezza n bit. *(5punti)*

- Può esistere una terza chiave K_3 tale che la cifratura doppia $C = E_{K_2}(E_{K_1}(P))$ sia equivalente a una cifratura singola $C = E_{K_3}(P)$? Spiegare un esempio in cui questo avviene e un esempio in cui questo non avviene.

Risposta 40: Sì, può esistere per esempio nell'algoritmo di cifratura RSA dove basta prendere l'esponente di cifratura $K_3 = K_1 * K_2$.

Un esempio in cui questo non avviene è con DES, dove infatti è spesso utilizzato, per evitare le debolezze di DES presenti da svariati anni, 3DES che è l'applicazione di 3 chiavi diverse dove ogni risultato è univoco.

E' necessario notare che perché il 3DES funzioni e che quindi non esista una chiave K che abbia lo stesso risultato della cifratura doppia C, le tre chiavi devono essere necessariamente diverse.