Capitolo 2

One-Time Pad OTP

Il One-Time Pad (OTP), noto anche come cifrario a blocchi usa e getta, è una tecnica crittografica che fornisce una segretezza perfetta quando utilizzata correttamente. Si tratta di un metodo di crittografia che utilizza una chiave casuale lunga almeno quanto il testo in chiaro. La chiave viene generata una sola volta e deve rimanere completamente segreta. Ecco come funziona il One-Time Pad:

Generazione della chiave: Generare una chiave casuale che abbia la stessa lunghezza del messaggio in chiaro. La chiave consiste in una sequenza di bit o caratteri casuali. Ogni elemento della chiave dovrebbe essere distribuito in modo indipendente e uniforme.

Crittografia: Per crittografare il testo in chiaro, eseguire un'operazione di XOR bit a bit tra il testo in chiaro e la chiave. Ad esempio, se il testo in chiaro è "CIAO" e la chiave casuale è "XKLM", si esegue l'operazione XOR tra i bit corrispondenti: 'C' XOR 'X', 'I' XOR 'K', 'A' XOR 'L', 'O' XOR 'M'. Il risultato sarà il testo cifrato.

Decrittografia: Per decrittografare il testo cifrato, eseguire nuovamente un'operazione di XOR bit a bit tra il testo cifrato e la stessa chiave utilizzata per la crittografia. Questo riporterà il testo in chiaro originale.

È fondamentale che la chiave venga generata in modo casuale e che non venga mai riutilizzata per crittografare altri messaggi. Inoltre, la chiave deve rimanere segreta e conosciuta solo dal mittente e dal destinatario. Se la chiave viene compromessa o riutilizzata, la sicurezza del sistema One-Time Pad può essere compromessa.

Il One-Time Pad offre una sicurezza perfetta perché, a causa dell'operazione di XOR con la chiave casuale, non c'è alcuna relazione statistica tra il testo in chiaro e il testo cifrato, rendendo impossibile per un crittoanalista individuare il messaggio originale anche se conosce il testo cifrato. Tuttavia, l'implementazione corretta e la gestione sicura della chiave sono fondamentali per garantire la sicurezza del sistema.

Stream cipher

Un cifrario a flusso (stream cipher) è un tipo di algoritmo crittografico che cifra i dati in modo continuo, bit per bit o byte per byte, utilizzando un flusso di caratteri generati da una chiave segreta. A differenza dei cifrari a blocco che operano su blocchi fissi di dati, i cifrari a flusso cifrano i dati uno alla volta, producendo un flusso di dati cifrati che può essere combinato con il testo in chiaro mediante un'operazione di XOR per ottenere il testo cifrato.

Ecco come funziona un cifrario a flusso:

Generazione del flusso di chiavi: Viene generato un flusso di caratteri pseudocasuali noto come "keystream" (flusso di chiavi) utilizzando una chiave segreta e un algoritmo di generazione del flusso. L'algoritmo di generazione del flusso prende la chiave come input e produce un flusso di caratteri della stessa lunghezza del testo in chiaro.

Cifratura: Il flusso di chiavi viene combinato con il testo in chiaro utilizzando l'operazione di XOR bit per bit o byte per byte. Ogni carattere del flusso di chiavi

viene combinato con il corrispondente carattere del testo in chiaro, producendo il carattere corrispondente del testo cifrato. L'operazione di XOR restituisce 0 se i bit (o byte) sono uguali e 1 altrimenti.

Decifratura: Per decifrare il testo cifrato, viene utilizzato lo stesso flusso di chiavi generato nella fase di cifratura. Il flusso di chiavi viene combinato con il testo cifrato utilizzando l'operazione di XOR, ripristinando così il testo in chiaro originale.

I cifrari a flusso sono spesso utilizzati quando è necessario cifrare grandi quantità di dati in tempo reale, poiché la cifratura e la decifratura possono essere eseguite in modo rapido ed efficiente bit per bit o byte per byte. Tuttavia, è di fondamentale importanza utilizzare una chiave segreta forte e garantire la corretta generazione e gestione del flusso di chiavi per preservare la sicurezza del sistema crittografico.

Differenze stream cipher e OTP

Le differenze tra il One-Time Pad (OTP) e un cifrario a flusso (stream cipher) sono le seguenti:

Proprietà di sicurezza: L'OTP offre una sicurezza perfetta, nota anche come sicurezza incondizionata o perfetta segretezza. Se la chiave viene generata correttamente, rimane segreta e viene utilizzata solo una volta, l'OTP è immune a qualsiasi forma di crittoanalisi. D'altra parte, i cifrari a flusso non offrono una sicurezza perfetta. La loro sicurezza si basa sulla robustezza dell'algoritmo di generazione del flusso e sulla segretezza della chiave utilizzata per generare il flusso.

Generazione delle chiavi: Nell'OTP, la chiave deve essere generata in modo completamente casuale, essere della stessa lunghezza del testo in chiaro e non essere mai riutilizzata. D'altra parte, nei cifrari a flusso, la chiave viene generata utilizzando un algoritmo di generazione del flusso che può essere basato su vari principi, come cifrari a blocchi o funzioni di hash. La chiave può essere di lunghezza variabile e può essere utilizzata per crittografare più di un messaggio.

Lunghezza della chiave: Nell'OTP, la chiave deve essere della stessa lunghezza del testo in chiaro. Questo richiede una lunga chiave per crittografare messaggi di grandi dimensioni. Nei cifrari a flusso, la chiave può essere di lunghezza variabile e può essere utilizzata per crittografare messaggi di lunghezza diversa.

Requisiti di conservazione della chiave: Nell'OTP, la chiave deve essere conservata in modo sicuro e segreto. Deve essere trasmessa in modo sicuro al destinatario e successivamente distrutta. Nei cifrari a flusso, la chiave può essere conservata e riutilizzata per crittografare più messaggi.

Efficienza: I cifrari a flusso sono generalmente più efficienti dei One-Time Pad in termini di velocità di cifratura e decifratura, poiché operano su bit o byte individuali anziché su blocchi di dati. I cifrari a flusso possono essere implementati in hardware dedicato o possono essere eseguiti in modo rapido utilizzando algoritmi di generazione del flusso ottimizzati.

In sintesi, l'OTP offre una sicurezza perfetta ma richiede una lunga chiave e un'attenta gestione della stessa, mentre i cifrari a flusso offrono una sicurezza basata sull'algoritmo di generazione del flusso e sulla segretezza della chiave, ma sono più efficienti e flessibili nella gestione delle chiavi.

PRG Pseudorandom Generator

PRG è l'acronimo di "Pseudorandom Generator" (Generatore Pseudo-Casuale). Si tratta di un algoritmo utilizzato per generare sequenze di numeri o bit che appaiono casuali, ma sono in realtà deterministici e generati da una funzione matematica.

Un generatore pseudo-casuale produce una sequenza che si avvicina alle proprietà statistiche di una sequenza casuale, ma è generata da un algoritmo a partire da un seme o una chiave iniziale. A differenza di un generatore di numeri veramente casuali (come i fenomeni naturali o il rumore termico), i generatori pseudo-casuali sono basati su algoritmi deterministici e riproducibili.

L'output di un PRG dipende sia dal seme iniziale che dallo stato interno dell'algoritmo, e può essere utilizzato per generare una sequenza continua di numeri o bit. Questa sequenza può essere utilizzata in vari contesti, come nella generazione di chiavi crittografiche, nella simulazione di eventi casuali in giochi o modelli matematici, o nella generazione di numeri casuali per applicazioni statistiche.

È importante notare che, nonostante i generatori pseudo-casuali possano generare sequenze che sembrano casuali, sono deterministici e riproducibili. Ciò significa che se si conosce il seme iniziale e lo stato interno dell'algoritmo, è possibile riprodurre l'intera sequenza generata. Pertanto, i generatori pseudo-casuali devono essere utilizzati con cautela in contesti critici per la sicurezza, come la crittografia, dove è necessario un alto livello di casualità e imprevedibilità.

Capitolo 3

Block cipher

Un cifrario a blocchi (block cipher) è un tipo di algoritmo crittografico che cifra i dati in blocchi fissi di una determinata dimensione, di solito misurati in bit. Invece di cifrare i dati uno alla volta come avviene nei cifrari a flusso, un cifrario a blocchi elabora e crittografa i dati in blocchi di dimensioni specifiche.

Ecco come funziona un cifrario a blocchi:

Divisione in blocchi: Il testo in chiaro viene suddiviso in blocchi di dimensione fissa. La dimensione dei blocchi può variare a seconda dell'algoritmo utilizzato, ma i valori comuni sono 64 bit (8 byte) o 128 bit (16 byte).

Cifratura di ogni blocco: Ogni blocco di testo in chiaro viene cifrato separatamente utilizzando una chiave segreta e un algoritmo di cifratura. L'algoritmo prende in input il blocco di testo in chiaro e la chiave segreta e produce il blocco di testo cifrato corrispondente.

Combinazione dei blocchi cifrati: I blocchi di testo cifrato vengono combinati per formare il testo cifrato completo. A seconda dell'algoritmo utilizzato, possono essere applicate diverse operazioni di combinazione, come la concatenazione o l'operazione di XOR tra i blocchi.

Decifratura: Per decifrare il testo cifrato, viene utilizzato lo stesso algoritmo di cifratura, ma con la chiave segreta invertita o con una chiave di decifratura

corrispondente. I blocchi di testo cifrato vengono decifrati uno alla volta, producendo i blocchi di testo in chiaro corrispondenti.

I cifrari a blocchi sono comunemente utilizzati per proteggere la confidenzialità dei dati nelle comunicazioni e nello stoccaggio dei dati. Alcuni esempi di cifrari a blocchi ben noti includono il DES (Data Encryption Standard), il 3DES, l'AES (Advanced Encryption Standard) e il Blowfish.

È importante utilizzare una chiave segreta forte e implementare adeguati meccanismi di gestione delle chiavi per garantire la sicurezza dei cifrari a blocchi. Inoltre, è fondamentale considerare anche altri aspetti della sicurezza, come l'integrità dei dati, l'autenticazione e la resistenza a potenziali attacchi crittografici.

Pseudo Random Function (PRPs) e Pseudo Random Permutation (PRFs)

PRP e PRF sono due concetti fondamentali nella crittografia e nella teoria delle funzioni pseudo-randomiche.

PRP (Pseudorandom Permutation) si riferisce a una funzione che appare come una permutazione casuale agli osservatori esterni. In altre parole, una PRP prende un input di lunghezza fissa e lo mappa in un output della stessa lunghezza in modo che ogni input corrisponda a un output unico e viceversa. L'importante caratteristica di una PRP è che sembra comportarsi in modo casuale e imprevedibile, anche se è deterministica. Le cifrature a blocchi, come AES e DES, sono esempi di PRP.

PRF (Pseudorandom Function) è una funzione che prende un input di lunghezza variabile e restituisce un output di lunghezza fissa. Una PRF simula casualità e comportamento imprevedibile, anche se è deterministica. A differenza delle PRP, le PRF non hanno l'obbligo di essere una permutazione e possono produrre lo stesso output per input diversi. Le PRF sono ampiamente utilizzate in diverse applicazioni crittografiche, come la generazione di chiavi, l'autenticazione e i protocolli di sicurezza.

Un aspetto importante è che una PRP può essere utilizzata come PRF, ma non tutte le PRF possono essere utilizzate come PRP. Questo perché una PRP deve essere invertibile, in modo che l'output possa essere decifrato correttamente, mentre una PRF può avere la proprietà di unidirezionalità.

Sia le PRP che le PRF svolgono un ruolo cruciale nella progettazione di algoritmi crittografici sicuri e nella costruzione di protocolli crittografici robusti.

Teorema di Luby-Rackoff

Il teorema di Luby-Rackoff è un risultato importante nel campo della crittografia che riguarda la costruzione di cifrari a blocchi sicuri a partire da cifrari a blocchi meno sicuri.

Il teorema afferma che se si ha a disposizione un cifrario a blocchi sicuro a 2n bit, è possibile costruire un cifrario a blocchi sicuro a n bit con un certo numero di iterazioni. In altre parole, utilizzando un cifrario a blocchi debole come "mattoncino" di base, si può costruire un cifrario a blocchi sicuro mediante la concatenazione di più istanze di quel cifrario.

L'idea alla base del teorema è la costruzione di una rete a sostituzione-permutazione (Substitution-Permutation Network o SPN), in cui l'input passa attraverso più round di sostituzioni e permutazioni. Ogni round utilizza il cifrario a blocchi di base e una chiave di round unica.

Il teorema di Luby-Rackoff dimostra che con un numero sufficiente di iterazioni (solitamente considerato essere almeno tre), la sicurezza complessiva del cifrario a blocchi costruito utilizzando questa tecnica diventa indistinguibile da un cifrario a blocchi casuale. Ciò significa che le proprietà di sicurezza del cifrario a blocchi di base vengono propagate nel cifrario a blocchi costruito.

Il teorema di Luby-Rackoff ha fornito una base teorica importante per la progettazione e l'analisi dei cifrari a blocchi moderni, come l'Advanced Encryption Standard (AES). Ha dimostrato che utilizzando un cifrario a blocchi relativamente semplice come "mattone" di base e una corretta configurazione delle iterazioni e delle chiavi di round, è possibile ottenere un cifrario a blocchi sicuro e resistente agli attacchi crittografici.

Data Encryption Standard (DES)

Il Data Encryption Standard (DES) è un cifrario a blocchi simmetrico che è stato ampiamente utilizzato negli anni '70 e '80 come standard per la crittografia dei dati. Fu sviluppato dal National Institute of Standards and Technology (NIST) degli Stati Uniti ed è basato sull'algoritmo Lucifer, sviluppato dalla IBM.

Ecco come funziona DES:

Dimensione dei blocchi: DES opera su blocchi di dati di 64 bit (8 byte). Questo significa che il testo in chiaro deve essere suddiviso in blocchi di 64 bit prima di essere cifrato.

Chiave di cifratura: DES utilizza una chiave di cifratura di 56 bit, anche se in realtà la lunghezza effettiva della chiave è di 64 bit, poiché ogni ottavo bit viene utilizzato come bit di controllo di parità. Questa lunghezza relativamente breve della chiave è una delle limitazioni di sicurezza di DES.

Rete di sostituzione-permutazione: DES utilizza una rete di sostituzione-permutazione (Substitution-Permutation Network) per eseguire l'operazione di cifratura. Questa rete prevede la ripetizione di una serie di sostituzioni e permutazioni su blocchi di dati in un numero fisso di round. Durante ogni round, l'input viene suddiviso in metà blocchi, e le operazioni di sostituzione e permutazione vengono eseguite sui dati.

Iterazioni e chiavi di round: DES esegue 16 round di operazioni sulla sequenza di dati in ingresso. In ogni round, viene generata una chiave di round unica a partire dalla chiave di cifratura principale, che viene utilizzata per effettuare le operazioni di sostituzione e permutazione.

Decifratura: Per decifrare i dati cifrati con DES, viene utilizzato lo stesso algoritmo di cifratura, ma con le chiavi di round utilizzate in ordine inverso. Ciò significa che DES è un cifrario a blocchi simmetrico, in cui lo stesso algoritmo e la stessa chiave possono essere utilizzati per cifrare e decifrare i dati.

Nonostante la sua popolarità nel passato, DES è stato considerato insicuro per l'uso nella crittografia moderna a causa delle dimensioni relativamente brevi della chiave. Attualmente, è consigliabile utilizzare algoritmi di cifratura più robusti e sicuri,

come l'Advanced Encryption Standard (AES), che supporta chiavi di lunghezza maggiore e offre una maggiore sicurezza crittografica.

3DES

Triple DES (3DES) è una variante del cifrario DES che è stata introdotta per aumentare la sicurezza crittografica rispetto al DES standard. Mentre il DES utilizza una chiave di cifratura di 56 bit, 3DES utilizza chiavi di cifratura di 112 o 168 bit, a seconda della modalità di utilizzo. In generale, Triple DES rappresenta un miglioramento significativo in termini di sicurezza rispetto a DES, grazie all'aumento della lunghezza della chiave. Tuttavia, è importante notare che, nonostante la sua maggiore sicurezza rispetto a DES, 3DES è considerato un algoritmo crittografico legacy e molti esperti consigliano l'uso di algoritmi più recenti come AES per garantire una protezione adeguata dei dati.

Advanced Encryption Standard (AES)

L'Advanced Encryption Standard (AES) è un algoritmo di cifratura a blocchi ampiamente utilizzato e considerato uno dei cifrari più sicuri disponibili. È diventato lo standard di crittografia per molti utilizzi, inclusi la protezione dei dati sensibili, la crittografia dei dispositivi di archiviazione e la sicurezza delle comunicazioni.

Ecco le caratteristiche principali di AES:

Dimensione dei blocchi: AES lavora con blocchi di dati di 128 bit (16 byte). Questo significa che il testo in chiaro deve essere suddiviso in blocchi di 128 bit prima di essere cifrato o decifrato.

Lunghezza della chiave: AES supporta tre lunghezze di chiave: 128 bit, 192 bit e 256 bit. La lunghezza della chiave determina il livello di sicurezza del cifrario. In generale, una chiave più lunga offre una maggiore resistenza agli attacchi crittografici, ma richiede anche un maggiore sforzo computazionale.

Struttura: AES utilizza una struttura chiamata Rijndael, che è un cifrario a blocchi a sostituzione-permutazione. Questo significa che le operazioni di sostituzione e permutazione vengono eseguite in sequenza su blocchi di dati per produrre il testo cifrato.

Round e chiavi di round: AES esegue una serie di round di operazioni sulla sequenza di dati in ingresso. Il numero di round dipende dalla lunghezza della chiave utilizzata: 10 round per chiavi di 128 bit, 12 round per chiavi di 192 bit e 14 round per chiavi di 256 bit. Durante ogni round, vengono generate chiavi di round uniche a partire dalla chiave di cifratura principale, che vengono utilizzate per effettuare le operazioni di sostituzione e permutazione sui dati.

Sicurezza: AES è stato progettato per fornire una sicurezza crittografica elevata. È stato soggetto a un'ampia analisi e valutazione da parte della comunità crittografica ed è stato adottato come standard da numerosi enti governativi e organizzazioni di tutto il mondo.

AES offre una combinazione di sicurezza, efficienza e compatibilità che lo rende uno degli algoritmi di cifratura più utilizzati. La sua robustezza e il suo ampio utilizzo lo rendono adatto per proteggere dati sensibili in vari contesti, inclusi sistemi di archiviazione, comunicazioni sicure e applicazioni di crittografia in generale.

Capitolo 4

Modes of Operation

One-Time Key

One-Time Key (Chiave monouso): Si riferisce a un sistema in cui una chiave viene utilizzata per crittografare un singolo messaggio e successivamente viene scartata. In pratica, viene generata una nuova chiave casuale per ogni messaggio. Questo garantisce una sicurezza molto elevata, poiché non esiste alcuna relazione tra le chiavi utilizzate per crittografare messaggi diversi. Un esempio di algoritmo che utilizza una chiave monouso è l'OTP (One-Time Pad), in cui ogni bit del messaggio viene combinato con un bit casuale della chiave tramite l'operazione di XOR.

- Electronic Code Book (ECB): È una modalità di utilizzo di un cifrario a blocchi in cui ogni blocco di testo in chiaro viene cifrato indipendentemente, utilizzando la stessa chiave. Questo significa che blocchi identici nel messaggio di input produrranno lo stesso blocco di testo cifrato. Questo può rivelare modelli e strutture all'interno del messaggio, che possono essere sfruttati dagli attaccanti per inferire informazioni sul testo in chiaro. Pertanto, l'ECB non è raccomandato per la cifratura di dati sensibili.
- Deterministic Counter Mode (CTR): È una modalità di utilizzo di un cifrario a blocchi che converte il cifrario in un cifrario a flusso. Un contatore viene utilizzato per generare una sequenza di numeri pseudocasuali, che vengono poi combinati con il testo in chiaro tramite l'operazione di XOR per produrre il testo cifrato. CTR offre buone proprietà di sicurezza, poiché il contatore assicura che non ci siano due blocchi di testo cifrato identici all'interno della stessa sessione di crittografia. Inoltre, CTR offre una buona parallelizzazione e può essere implementato in modo efficiente.

Many-Time Key (Chiave multipla): Si riferisce a un sistema in cui la stessa chiave viene utilizzata per crittografare più messaggi. In questo caso, è necessario considerare attentamente le modalità di utilizzo del cifrario a blocchi per garantire la sicurezza. Le modalità comunemente utilizzate per la cifratura a chiave multipla includono:

- Cipher Block Chaining (CBC): In questa modalità, ciascun blocco di testo in chiaro viene combinato con il blocco di testo cifrato precedente tramite l'operazione di XOR prima di essere cifrato. Ciò introduce dipendenze tra i blocchi e rende l'output dipendente dall'input e dallo stato precedente.
- Counter Mode (CTR): Come spiegato in precedenza, CTR utilizza un contatore per generare una sequenza di numeri pseudocasuali che vengono combinati con il testo in chiaro tramite XOR. Anche in questo caso, l'output dipende dal contatore e dall'input.

Entrambe le modalità, CBC e CTR, introducono una dipendenza tra i blocchi che contribuisce a migliorare la sicurezza rispetto all'ECB quando la stessa chiave viene utilizzata per crittografare più messaggi.

Capitolo 5

Key Exchange

Lo scambio di chiavi (Key Exchange) è un processo crittografico attraverso il quale due o più entità comunicanti stabiliscono una chiave segreta condivisa per garantire la confidenzialità e l'integrità delle comunicazioni.

L'obiettivo dello scambio di chiavi è quello di stabilire una chiave segreta condivisa senza che essa venga rivelata ad altri soggetti. Ciò permette alle entità coinvolte di comunicare in modo sicuro su un canale non sicuro.

Ci sono diversi algoritmi e protocolli utilizzati per effettuare lo scambio di chiavi, in particolare il protocollo di Diffie-Hellman

Diffie-Hellman

Il protocollo Diffie-Hellman è un algoritmo crittografico utilizzato per stabilire una chiave segreta condivisa tra due o più entità su un canale di comunicazione non sicuro. È stato inventato da Whitfield Diffie e Martin Hellman nel 1976 ed è ampiamente utilizzato per scambiare chiavi in diversi protocolli di crittografia.

Il protocollo Diffie-Hellman si basa su problemi matematici di difficile soluzione, in particolare il problema del logaritmo discreto. L'idea fondamentale del protocollo è che due parti, chiamate Alice e Bob, possono generare ciascuna una coppia di chiavi pubbliche e private e utilizzarle per calcolare una chiave segreta condivisa senza scambiare direttamente le chiavi private.

Ecco come funziona il protocollo Diffie-Hellman:

Inizializzazione: Alice e Bob si accordano su un insieme di parametri comuni: un numero primo grande chiamato modulo (p) e un numero chiamato base (g), che è un generatore di un sottoinsieme ciclico di ordine p.

Generazione delle chiavi: Alice e Bob generano una coppia di chiavi pubbliche e private ciascuno. Scelgono un numero casuale privato (a per Alice, b per Bob) compreso tra 1 e p-1.

Calcolo delle chiavi pubbliche: Alice calcola la sua chiave pubblica (A) come A = g^a mod p, mentre Bob calcola la sua chiave pubblica (B) come B = g^b mod p.

Scambio delle chiavi pubbliche: Alice e Bob si scambiano le rispettive chiavi pubbliche sui canali di comunicazione non sicuri.

Calcolo della chiave segreta: Alice calcola la chiave segreta (S) utilizzando la chiave pubblica di Bob e la sua chiave privata: S = B^a mod p. Allo stesso modo, Bob calcola la chiave segreta (S) utilizzando la chiave pubblica di Alice e la sua chiave privata: S = A^b mod p.

La chiave segreta condivisa: Alla fine del protocollo, sia Alice che Bob hanno calcolato la stessa chiave segreta (S). Questa chiave può essere utilizzata come chiave simmetrica per crittografare e decrittografare i dati utilizzando un algoritmo di cifratura a chiave simmetrica.

L'importante proprietà del protocollo Diffie-Hellman è che anche se i valori delle chiavi pubbliche vengono scambiati pubblicamente, un osservatore passivo non può dedurre la chiave segreta senza conoscere le chiavi private di Alice e Bob.

Tuttavia, è importante notare che il protocollo Diffie-Hellman non offre autenticazione delle parti coinvolte. Pertanto, è necessario utilizzarlo in combinazione con altri meccanismi, come la firma digitale, per garantire l'autenticità e l'integrità delle comunicazioni.

Capitolo 6

Introduction Number Theory

Capitolo inutile.

Capitolo 6

Asymmetric Cryptography

La crittografia asimmetrica, o crittografia a chiave pubblica, è un tipo di crittografia che coinvolge l'uso di una coppia di chiavi: una chiave pubblica e una chiave privata. Questo tipo di crittografia è basato su algoritmi matematici complessi e offre diverse funzionalità, tra cui la cifratura, la firma digitale e lo scambio di chiavi.

Ecco alcune caratteristiche e concetti chiave della crittografia asimmetrica:

Coppia di chiavi pubblica/privata: Ogni entità coinvolta nel processo crittografico ha una coppia di chiavi unica. La chiave pubblica è resa pubblicamente disponibile, mentre la chiave privata è mantenuta segreta dall'entità proprietaria.

Cifratura asimmetrica: Utilizzando la chiave pubblica di un destinatario, un mittente può cifrare un messaggio in modo che possa essere decifrato solo con la chiave privata corrispondente. In questo modo, solo il destinatario, che possiede la chiave privata, può decifrare e leggere il messaggio.

Firma digitale: Utilizzando la propria chiave privata, un mittente può firmare digitalmente un messaggio per dimostrare l'autenticità e l'integrità del mittente. La firma digitale può essere verificata utilizzando la chiave pubblica corrispondente al fine di garantire che il messaggio non sia stato alterato e che provenga da una fonte attendibile.

Scambio di chiavi: La crittografia asimmetrica è spesso utilizzata per facilitare lo scambio sicuro di chiavi segrete tra due entità. Una delle parti può crittografare la chiave segreta utilizzando la chiave pubblica dell'altra parte e inviarla in modo sicuro. Una volta ricevuta, la parte destinataria può decrittografare la chiave segreta utilizzando la propria chiave privata e utilizzarla per comunicazioni successive con la parte mittente.

Algoritmi comuni: Gli algoritmi più comuni utilizzati nella crittografia asimmetrica includono RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm), ECC (Elliptic Curve Cryptography) e altri. Ognuno di questi algoritmi ha proprietà matematiche specifiche che li rendono adatti a scopi specifici.

La crittografia asimmetrica offre una maggiore flessibilità e sicurezza rispetto alla crittografia a chiave simmetrica, ma può essere computazionalmente più intensiva. Di

solito, viene utilizzata in combinazione con la crittografia a chiave simmetrica per ottenere una combinazione di sicurezza ed efficienza.

Trapdoor permutations

Le permutazioni con trappola, o trapdoor permutations, sono un tipo di primitiva crittografica che fornisce una funzione unidirezionale con la capacità aggiuntiva di invertire il calcolo utilizzando un'informazione segreta chiamata "trappola".

In una permutazione con trappola, dato un input x, è computazionalmente facile calcolare il risultato della permutazione. Tuttavia, invertire la permutazione, ovvero trovare l'input originale dato il risultato della permutazione, è computazionalmente difficile senza la conoscenza della trappola.

La trappola è un'informazione segreta che viene utilizzata per invertire la permutazione in modo efficiente. Solo chi possiede la trappola può eseguire questa operazione, mentre per gli altri soggetti risulta computazionalmente difficile.

Le permutazioni con trappola sono fondamentali in diverse aree della crittografia, come la crittografia a chiave pubblica e la firma digitale. Ad esempio, l'algoritmo RSA utilizza una permutazione con trappola basata sulla fattorizzazione dei numeri primi. La trappola consiste nel conoscere i fattori primi del modulo utilizzato nell'algoritmo.

Queste permutazioni offrono un meccanismo per garantire la sicurezza e la riservatezza delle informazioni, consentendo a un mittente di criptare i dati in modo che solo il destinatario, che possiede la trappola corrispondente, possa decifrarli in modo efficiente.

RSA

Le permutazioni con trappola (trapdoor permutations) e l'algoritmo RSA (Rivest-Shamir-Adleman) sono collegati tra loro, in quanto l'algoritmo RSA è basato su una specifica permutazione con trappola.

L'algoritmo RSA è uno degli algoritmi crittografici asimmetrici più ampiamente utilizzati e si basa sulla difficoltà del problema della fattorizzazione dei numeri primi. L'idea chiave dell'algoritmo RSA è la generazione di una coppia di chiavi pubblica-privata utilizzando una permutazione con trappola basata sulla fattorizzazione.

Ecco come funziona l'algoritmo RSA:

Generazione delle chiavi: Prima di tutto, viene generata una coppia di chiavi pubblica-privata. Questo coinvolge la scelta di due numeri primi grandi e distinti, p e q. La chiave pubblica consiste nel modulo n, ottenuto dal prodotto di p e q, e in un esponente pubblico e. La chiave privata consiste nell'esponente privato d, calcolato utilizzando l'algoritmo di esteso Euclideano e le proprietà della funzione di permutazione con trappola.

Cifratura: Per criptare un messaggio M, il mittente utilizza la chiave pubblica (n, e) e la permutazione con trappola associata per calcolare il valore crittografato C. Questo viene fatto elevando M all'esponente e e prendendo il modulo n.

Decrittazione: Per decrittare il messaggio crittografato C, il destinatario utilizza la chiave privata d e la permutazione con trappola per calcolare il valore originale

M. Questo viene fatto elevando C all'esponente d e prendendo il modulo n.

La sicurezza dell'algoritmo RSA si basa sulla difficoltà computazionale della fattorizzazione dei numeri primi. In altre parole, trovare i fattori primi di n (il modulo) è un problema computazionalmente difficile quando i numeri primi sono sufficientemente grandi. La trappola dell'algoritmo RSA risiede nel fatto che conoscendo i fattori primi di n, l'inversione della permutazione diventa computazionalmente efficiente, consentendo la decrittazione del messaggio crittografato.

Quindi, possiamo dire che l'algoritmo RSA è un esempio di utilizzo di una permutazione con trappola per la crittografia a chiave pubblica. La permutazione con trappola associata a RSA consente di invertire il calcolo crittografico solo a chi possiede la corrispondente chiave privata, garantendo la sicurezza e la riservatezza dei dati.