Università degli studi di Modena e Reggio Emilia

Dipartimento di Ingegneria Enzo Ferrari

Crittografia Applicata

Anno Accademico 2023/24

Indice

Capitolo 1

Introduzione

Crittologia: l'arte delle scritture segrete, può essere divisa in 3 macro argomenti:

1. **crittografia**: come *trasformare* messaggi per proteggerne il contenuto (l'informazione).
2. **steganografia**: come *nascondere* messaggi per evitare che venga individuato (ex. **least significant bit steganography**).
3. **crittoanalisi**: come *analizzare* messaggi e rivelarne l'informazione.

1.1 Crittografia Classica

La sicurezza della crittografia classica si basa unicamente sulla **segretezza** del **metodo** (noto solo al *sender* e al *receiver*), considerava come una tipologia di attacco quello **passivo** (*read only*). Basandosi su questi concetti il suo utilizzo in applicazioni reali è molto limitato (nel senso moderno), considerando la comunicazione in termini di scambio di informazioni in linguaggio naturale.
Alcuni esempi: **scytale** (*transposition cipher*), **caesar cipher** (*shift cipher*) e **vigenere cipher**.

1.2 Crittografia Moderna

1.2.1 Encryption only

La crittografia moderna si basa su due principi:

1. **Kerckhoffs principles**:
   - gli algoritmi devono essere pubblici.
   - la sicurezza del metodo si deve basare sulla **segretezza** della **chiave**.
   - uno schema deve essere "praticamente", se non "matematicamente" indecifrabile
2. **Shannon principles**:
   - **confusione**: ogni bit del crittogramma deve dipendere da più bit della chiave, oscurando, però, la correlazione tra le due.
   - **diffusione**: se viene cambiato un singolo bit del testo in chiaro, allora almeno la metà dei bit del crittogramma devono cambiare, e viceversa.

Nella crittografia moderna lo spazio delle chiavi deve essere sufficentemente ampio per evitare una ricerca esaustiva su di esso, in più, nessuna informazione (nè del *plaintext*, né della *key*) deve poter essere estrapolata dal *ciphertext*. Viene detto che il *ciphertext* deve essere **indistinguibile** da una sequenza di bit *random*.

1.2.2 Extended and applied settings

Gli avversari (i crittoanalisti) non sono più unicamente **passivi**, ma bisogna modellare delle tipologia di avversari che siano capaci anche di **interagire** con i nostri sistemi e **manipolare** dei messaggi.
Per ognuna di queste modellazioni è necessario **provare la sicurezza** dei sistemi andando a **definire** delle attività (tramite la comprensione e modellare cosa è "sicuro") e **costruendo** attività (progettandole e provandone la veridicità).

Bisogna definire le **primitive**, gli **schemi**, i **protocolli** e le *applicazioni* che vengono utilizzati, andandoli ad analizzare separatamente e completa.
⇒ può essere necessario costruire schemi e protocolli modellati su misura per applicazioni reali inerenti ad un certo caso d'uso: **Applied Cryptography**.

1.3 Crittografia Applicata

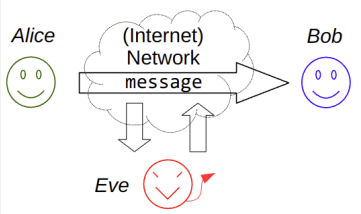
È un layer di astrazione che può essere (quasi) direttamente mappato all'interno di una soluzione per un caso d'uso reale (quindi tecniche "pratiche"). Siccome analizziamo **soluzioni pratiche** bisogna gestire possibili errori dovuti ad **implementazioni** o ***deployment*** errati. I protcolli sicuri assumono che un attaccante tenti di accedere alle informazioni in transito (violazione della **confidenzialità**) e cerchi di impersonificare un mittente (violazione dell'**autenticazione**).
Una delle sicurezze che deve garantire un protocollo sicuro è la **confidenzialità**.

Quando si studi/analizza/progetta un protocollo crittografico è necessario identificare:

- **system model**: descrive lo scenario ("idealmente") di utilizzo, andando a definire: gli attori **legittimi**, la **tipologia di protocollo** utilizzata, le **informazioni** possedute dagli attori legittimi, e altre informazioni sullo scenario applicativo.

  Esempio

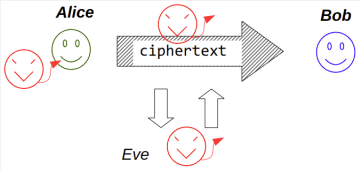
  Un protocollo sicuro che ha come **scopo** proteggere le informazioni scambiati tra due attori: **Alice** e **Bob** quando un possibile attaccante **Eve** può accedere direttamente all'informazione tramite il canale fisico.

- **threat modelling** (modellazione della tipologia di attaccante): abbiamo identificato degli attori legittimi, ma quanto sono affidabili? Modelliamo il protocollo sulla base dell'attaccante.
  1. che operazioni può effettuare sui dati: solo lettura, modifica, inserimento o eliminazione dei dati.
  2. qual è la superficie di attacco e cosa può provare a fare: ha accesso ad alcune funzionalità (cifrazione/decifrazione), che tipologia conoscenza (*white/gray/black box*), quanti tentavi si hanno: adattivo o meno.

  Esempio

  Cosa può fare **Eve** per compromettere la comunicazione: leggere, manipolare le informazioni in transito, compromissione di un attore legittimo.

- **security guarantees**: quale aspetto di sicurezza vogliamo garantire: **confidenzialità**, **integrità** (autenticazione), **disponibilità**, **non ripudio** (è anche presente il concetto di ***forward security***).
- **cryptogrphy settings**: le due classi principali sono: crittografia **simmetrica** (le funzioni di *encrypt* e *decrypt* utilizzano lo stesso **segreto**) e crittografia **asimmetrica** (sono presenti due differenti **chiavi**, uno utilizzabile durante la funzione di *encrypt* - *public* - e l'altro utilizzato durante la funzione di *decrypt* - *secret*).
- **security assumptions of a proposed scheme**

Alcuni **protocolli**:

1. ***Secure key exchange protocol*** (scambio sicuro di chiavi): Alice e Bob non hanno nessuna **chiave**, ne vogliono ottenere una **sicura** e **condivisa** comunicando su un canale sincrono e non sicuro.
2. ***Secure storage***: gli algoritmi di crittografia possono aggiungere protezione su dati conservati in canali protetti, l'avversario ha avuto accesso ai dati dopo aver sconfitto le difese iniziali, negli scenari di storage possiamo andare ad analizzare due tipologie di dati: "*data at rest*" (è lo standard e la *best-practice*) oppure "*data in use*" (continua ricerca soprattuto per i dispositivi mobili).
3. ***(Identity) Authentication***: (***challenge-response protocols***) ovvero dimostrare il possesso di un segreto senza mandarlo.
4. ***Password Protection***: gli schemi crittografici possono "proteggere" le password in caso di *data breaches*, non solo usando l'hash (anche se aggiunto il *salt*), ma anche tecniche per prevenire il ***brute-forcing*** attraverso **ASICs** (**Application-specific integrated circuit**).

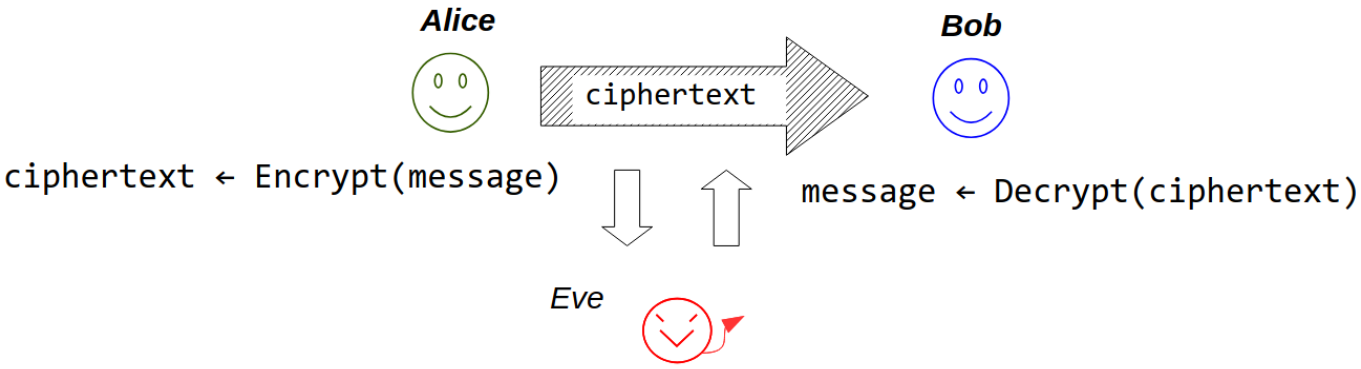
Per riassumere quanto visto fino ad adesso, possiamo dire che la **crittografia applicata** è l'insieme di molti ***system models*** (comunicazioni sincroni, messaggi di gruppi asincroni, *disk encryption*, ...), molti ***security guarantees*** (*information security*), integrità e autenticazione, molti **type of attackers** (passivi e attivi oppure online e offline). La **crittografia applicata** mira a dimostrare la sicurezza sfruttando "strati inferiori":

- la sicurezza dei **protocolli** viene ridotta alla sicurezza sugli schemi.
- la sicurezza degli **schemi** viene ridotta alla sicurezza delle primitive.
- la sicurezza delle **primitive** viene ridotta alla robusteza di problemi matematici.

Per progettare un **protocollo di comunicazione sicuro** è necessario utilizzare un approccio modulare basato su uno *stack* di altri "oggetti", un approccio tipico per lo *stack* è quello che ogni *layer* ha un determinato compito: il protocollo o lo schema è modificato su certi ***cryptographic settings*** e la sicurezza viene provato contro uno specifico tipo di avversario (***security models***).

Capitolo 2

Crittografia Simmetrica



Le garanzie di sicurezze che si cercano di mantenere sono:

- **confidenzialità**: Eve non può accedere a nessuna della informazioni sul messaggio.
- **autenticazione**: Bob può verificare se il messaggio non è stato inviato da Alice, viene anche chiamata ***data orgin authenticity*** nel contesto della comunicazioni e implica anche la protezione contro modifiche illegittime (**integrità**).