Crittografia

Federico Matteoni

A.A. 2020/21

Indice

Introduzione

Prof.ssa: Anna Bernasconi.
Vedremo i cifrari da un punto di vista prettamente algoritmico. Vedremo anche i cifrari storici, ormai non più utilizzabili, perché hanno "aperto la strada", per poi passare ai cifrari perfetti (soluzione ideale ma con costo elevato).
Poi esamineremo i cifrari simmetrici, a chiave pubblica, curve ellittiche, firma digitale, SSL. Protocolli zero knowledge, blockchain e crittografia quantistica.

Libro di testo: Bernasconi, Ferragina, Luccio - Elementi di Crittografia.

Esame Orali nel caso di esami a distanza, scritto nel caso di esami in presenza, closed-book.

Capitolo 1

Introduzione alla Crittografia

1.1 Introduzione

Crittografia Significa "*scrittura nascosta*", si intendono tecniche matematiche per mascherare i messaggi per non renderli leggibili a terzi (**crittografia**) o tentare di svelarli quando non si è il legittimo destinatario (**crittoanalisi**). Quindi tecniche di protezione e viceversa.
Esiste per i due mondi in contrapposizione: persone che vogliono scambiarsi privatamente informazioni e gli *impiccioni* che desiderano ascoltare o intromettersi nelle conversazioni altrui (per curiosità, investigazione o altri scopi).

Due gruppi di persone Chi vuole proteggersi userà **metodi di cifratura**, gli altri useranno **metodi di crittoanalisi**

Crittografia Metodi di Cifratura

Crittoanalisi Metodi di . . . **crittologia** studio comunicazione canali non sicuri e relativi problemi

1.1.1 Lo scenario

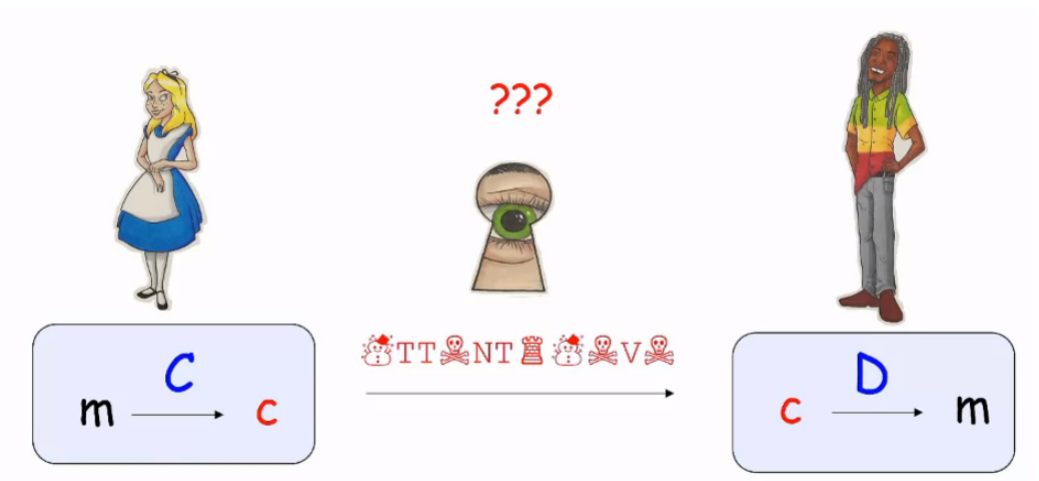
Alice vuole comunicare con Bob su un canale insicuro, quindi adottano un metodo di cifratura per spedire il messaggio in chiaro `m` sottoforma di crittogramma `c` (testo cifrato) che deve essere: incomprensibile al crittoanalista Eve (eavesdropper) in ascolto sul canale, ma facilmente decifrabile da Bob.

MSG Insieme dei messaggi in chiaro

CRITTO Insieme dei crittogrammi

$$C : MSG \to CRITTO$$
$$D : CRITTO \to MSG$$

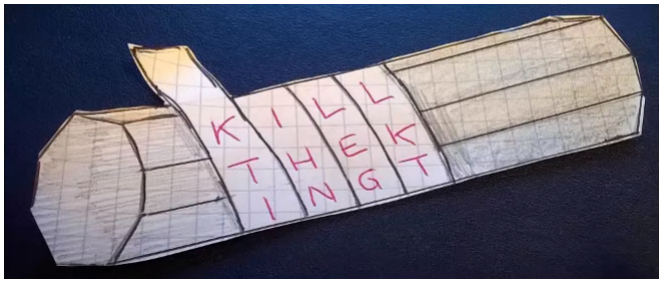
Sono operazioni da poter fare in tempo polinomiale. C e D sono una l'inversa dell'altra, ma C **deve essere iniettiva**.



1.1.2 Antichi esempi

Erodoto Nelle *Storie*, V secolo a.C.
Messaggi tatuati sulla testa, coperti dai capelli e riscoperti rasando la testa.

Scitale Spartani. Asta cilindrica in due esemplari identici. Si avvolgeva una striscia di carta attorno al cilindro e scritta. La chiave del cifrario è il diametro dello scitale.



Enea Tattico Un libro qualsiasi con un insieme di lettere segnate, o sostituire le vocali con simboli grafici.

Cifrario di Cesare Il più antico cifrario di concezione moderna. L'idea di base è che il crittogramma è ottenuto dal messaggio in chiaro `m` sostituendo ogni lettera con quella di tre posizioni più avanti nell'alfabeto.
Es. A $\to$ D, Z $\to$ C. La segretezza dipende interamente dalla conoscenza del metodo, era destinato all'uso ristretto da un piccolo gruppo di persone.

1.2 Livello di segretezza

Classificazione in base al livello di segretezza

> Cifrari per **uso ristretto**
> Le tecniche con cui si calcola e decifra il crittogramma sono tenute segrete in ogni loro aspetto. Impiegati per comunicazioni classificate (diplomatiche o militari), non adatti per uso di massa.
>
> Cifrari per **uso generale**
> Ogni codice segreto non può essere mantenuto tale per troppo a lungo. La parte segreta si limita alla chiave, nota solamente agli utenti che stanno comunicando.
> Vengono studiati dalla comunità, coinvolgendo tantissime persone. Solo la chiave deve essere segreta.
> **Il nemico conosce il sistema**.
> Quindi C e D sono note, la chiave **segreta** `k` è usata come input sia in C che in D:
> `c = C(m, k)`, `m = D(c, k)`
> Se non si conosce `k`, anche conoscendo C e D non si possono estrarre informazioni dal crittogramma.
> Tenere segreta una sola chiave è più facile che segretare l'intero metodo. Tutti possono usare C e D pubbliche con chiavi diverse, e se un crittoanalista entra in possesso di una chiave posso generarne semplicemente una nuova.

1.2.1 Chiavi segreta

Se la segretezza dipende unicamente dalla chiave bisogna proteggersi dagli attacchi a forza bruta, quindi avere un gran numero di chiavi, così da essere immuni da chi le prova tutte.
Inoltre la chiave deve essere scelta in modo casuale e non prevedibile, sennò il crittoanalista può provare le chiavi ovvie.

Attacco esauriente Il crittoanalisa potrebbe sferrare un attacco a forza bruta verificando la significatività delle sequenze `D(c, k)` $\forall$ `k`.
Se $|\text{Key}| = 10^{20}$ e con un calcolatore che impiega 10^{-6} per calcolare `D(c, k)` servirebbe in media più di un milione di anni per scoprire il messaggi provando tutte le chiavi. Però la segretezza può essere violata con altre tecniche: esistono cifrari più sicuri di altri pur con uno spazio di chiavi più piccoli.
Un cifrario complicato non è necessariamente più sicuro e **mai sottovalutare la bravura del crittoanalista**.

1.2.2 Crittoanalista

Comportamento Il comportamento di un crittoanalista può essere:

Passivo, quando si **limita ad ascoltare** la comunicazione

Attivo, quando **agisce sul canale** disturbando la comunicazione o modificando il contenuto dei messaggi.

Attacchi a un sistema crittografico Hanno l'obiettivo di forzare un sistema. Il metodo e il livello di pericolosità dipendono dalle informazioni in possesso del crittoanalista:

Cipher Text Attack: conosce una serie di crittogrammi

Known Plain-Text Attack: conosce una serie di coppie `(m, c)`

Chosen Plain-Text Attack: si procura coppie `(m, c)` relative a messaggi in chiaro da lui scelti.
Tutta la crittografia a chiave pubblica è soggetta a questo tipo di attacco (avendo la chiave pubblica, cifro dei messaggi che penso possano passare e ascolto finché non trovo nella comunicazione i crittogrammi in mio possesso).

Man in the Middle Il crittoanalista si installa sul canale di comunicazione:

Interrompe le comunicazioni dirette tra gli utenti Alice e Bob

le **sostituisce con messaggi propri**

e **convince** ciascun utente **che tali messaggi provengano leggitimamente dall'altro** utente.

Quindi il crittoanalista **Eve si finge Bob agli occhi di Alice e Alice agli occhi di Bob**.

Esiti

Successo pieno, si scopre completamente D o si ottiene la chiave

Successo limitato, si scopre solo qualche informazione ma sufficiente per comprendere il messaggio

1.2.3 Situazione attuale

Cifrari perfetti Inattaccabili, esistono ma richiedono operazioni complesse, **chiavi lunghe tanto quanto il messaggio e mai riutilizzabili**.
Shannon, 1945 (pubblicato nel 1949 per motivi di segretezza militare): `m` e `c` appaiono totalmente scorrelati, come se `c` fosse una stringa casuale di bit.
Nessuna informazione può filtrare dal crittogramma. Vedremo la teoria matematica.

One-Time Pad Anche detto blocco monouso, sicuro ma per essere usato bene richiede chiavi segrete totalmente casuali e lunghe quanto il messaggio. Come generarla e come scambiarla?

Cifrari attuali Nella crittografia di massa non si usano cifrari perfetti, ma **cifrari *dichiarati* sicuri**, inviolati dagli esperti e che usano algoritmi solo esponenziali per decrittare senza chiave. Il tempo per violare un cifrario è enorme e rende l'operazione insostenibile → impossibilità *pratica* di forzare il cifrario.

Dichiarati sicuri Non è noto se questi problemi matematici richiedano algoritmi *necessariamente* esponenziali o se sono dovuti all'incapacità nostra di trovare metodi più efficienti. Si riconduce a $P = NP$

1.2.4 Cifrari odierni

Advanced Encryption Standard AES, simmetrico a blocchi con chiavi di 128-256bit, pubblicamente noto e realizzabile su computer di ogni tipo. Il messaggio è diviso a blocchi lunghi quanto la chiave.

Le chiavi Sono stabilite dai mezzi elettronici (PC, smartphone, terminale...) e su Internet si scambia una chiave per ogni sessione.

Scambio delle chiavi La chiave va comunicata in sicurezza su un canale non ancora sicuro. Un'intercettazione nello scambio della chiave compromette il sistema.
Nel 1976 viene proposto un algoritmo per generare e scambiare una chiave segreta su un canale insicuro, senza necessità di scambiare informazioni o di incontrarsi in precedenza.
Si chiama **protocollo DH**, ancora largamente utilizzato nei protocolli crittografici su Internet.
Si scambiano pezzi di chiave tramite la rete e unendole a informazioni locali si costruisce la chiave.

Chiave pubblica Diffie ed Hellman hanno anche proposto la crittografia a chiave pubblica.

Cifrari simmetrici: stessa chiave per cifrare e decifrare, nota solo ai due utenti che comunicano. La scelgono di comune accordo e la tengono segreta.

Cifrari asimmetrici: chiavi pubbliche usate per cifrare e chiavi private per decifrare.
`c = C(m, k`$_{pub}$`)`
`m = D(m, k`$_{prv}$`)`
Si rende necessario che la C sia una one-way trapdoor: calcolare il crittogramma deve essere facile (polinomiale), ma decifrare `c` deve essere computazionalmente difficile (a meno di conoscere la trapdoor, la chiave privata).

RSA Rivest, Shamir, Adleman, 1977. Propongono un sistema a chiave pubblico facile da calcolare e difficile da invertire.

Vantaggi

Comunicazione molti a uno
Tutti possono inviare in modo sicuro allo stesso destinatario usando la sua chiave pubblica, ma solo lui può decifrarli. Un crittoanalista non può decifrare anche se conosce C, D e k_{pub}

Se n utenti vogliono comunicare servono solo $2n$ chiavi invece delle $n(n-1)/2$ necessarie nei cifrari simmetrici (una coppia per ogni coppia di utenti)

Non è richiesto nessun scambio

Svantaggi

Sono molto lenti rispetto ai cifrari simmetrici (polinomi di terzo grado)

Sono esposti ad attacchi di tipo chosen plain-text, perché conosco la chiave pubblica
Scelgo un numero qualsiasi di messaggi in chiaro, costruisce i crittogrammi relativi e ascolta sul canale confrontando i crittogrammi in transito e se trova un riscontro sa esattamente qual è il messaggio passato.

Come si usa Oggi si usa un cifrario a chiave segreta (AES) per le comunicazioni di massa, e un cifrario a chiave pubblica per scambiare le chiavi segrete relative al primo senza incontri fisici tra gli utenti.
Diventa lento solo lo scambio delle chiavi. Siamo anche al sicuro da attacchi chosen plain-text perché se la chiave è scelta bene risulta imprevedibile dal crittoanalista.

1.3 Rappresentazione matematica di oggetti

Per rappresentare gli oggetti scegliamo dei **caratteri** da un **insieme finito** detto **alfabeto**.
Un **oggetto** è **rappresentato da una sequenza ordinata di caratteri dell'alfabeto**. L'ordine dei caratteri è importante: a **oggetti diversi corrispondono sequenze diverse** e **il numero di oggetti che si possono rappresentare non ha limiti**. Significa che fissando un numero n arbitrariamente grande possiamo sempre creare un numero di oggetti $> n$, con sequenze via via più grande.

Alfabeto Γ con $|\Gamma| = s$ e N oggetti da rappresentare.
$d(s, N)$: lunghezza della sequenza più lunga che rappresenta un oggetto dell'insieme. A noi interessa la rappresentazione che minimizza $d(s, N)$, cioè $d_{min}(s, N)$
Una rappresentazione è tanto più efficiente quanto $d(s, N)$ si avvicina a $d_{min}(s, N)$

Esempio $s = 1, \Gamma = \{0\}$ l'unica possibilità è variare la lunghezza $\Rightarrow d_{min}(1, N) = N$, estremamente sfavorevole.
$s = 2, \Gamma = \{0, 1\}$, $\forall\, k \geq 1$ ho 2^k sequenze di lunghezza k. Il numero totale di sequenze lunghe da 1 a k è $2^{k+1} - 2$ (si esclude anche la sequenza nulla). Con N oggetti da rappresentare $\Rightarrow k \geq \log_2(N + 2) - 1 \Rightarrow N$ sequenze diverse tutte di $\log_2(N)$ caratteri.

Efficiente Codifica efficiente quando c'è questa riduzione logaritmica, **efficiente** quando . Sequenze della stessa lunghezza è vantaggioso perché non servono caratteri separatori. Per questo è necessario che l'alfabeto contenga almeno due caratteri.
La **notazione posizionale** è una rappresentazione efficiente indipendentemente dalla base $s \geq 2$ scelta. Un intero N è rappresentato con un numero d di cifre $\mid \log_s(N) \leq d \leq \log_s(N) + 1$

1.4 Richiamo della teoria della calcolabilità

Problemi computazionali Formulati matematicamente di cui cerchiamo una soluzione algoritmica: **decidibili** (e **trattabili** o **non trattabili**), o **non decidibili**.
Calcolabilità $\rightarrow$ **Algoritmo** e **problema non decidibile**
Complessità $\rightarrow$ **Algoritmo efficiente** e **problema intrattabile**.

Numerabilità Due insiemi A e B hanno lo stesso numero di elementi $\Leftrightarrow$ si può stabilire una **corrispondenza biunivoca** tra i loro elementi.
Questo porta alla definizione di **numerabile**: un insieme è numerabile $\Leftrightarrow$ i suoi elementi possono essere messi in **corrispondenza biunivoca con i numeri naturali**.
Numerabile significa che **possiede un'infinità numerabile di elementi**. Esempi: l'insieme dei numeri naturali N, l'insieme degli interi Z (avendo n in corrispondenza biunivoca con $2n + 1$ per $n \geq 0$ e $n \leftrightarrow 2|n|$ per $n < 0$, dando la sequenza $0, -1, 1, -2, 2 \ldots$) o anche l'insieme dei naturali pari ($2n \leftrightarrow n$)

Enumerazione delle sequenze Si vuole elencare in uno ordine ragionevole le sequenze di lunghezza finita costruite su un alfabeto finito. Le sequenze non sono in numero finito, quindi non si potrà completare l'elenco.
Lo scopo è **raggiungere qualsiasi sequenza σ arbitrariamente scelta in un numero finito di passi**. σ deve dunque trovarsi a **distanza finita** dall'inizio dell'elenco. Non va bene l'ordine del dizionario perché non saprei la posizione della prima stringa che inizia con b perché le stringhe composte da tutte a sono infinite.
Si stabilisce un ordine tra i caratteri. Si ordinano prima in lunghezza crescente e, a pari lunghezza, in ordine alfabetico.

Esempio $\Gamma = \{a, b, \ldots, z\}$, avrei
$a, b, \ldots, z$,
$aa, ab, \ldots, az, ba, bb, \ldots, bz, \ldots, zz, \ldots$
Ad una sequenza arbitraria corrisponde un numero intero, e la sequenza s arbitraria si troverà tra quelle di lunghezza $|s|$ in posizione alfabetica. Quindi ad una sequenza arbitraria $\leftrightarrow n$ che indica la posizione nell'elenco, e ad un numero naturale $n \leftrightarrow$ la sequenza che occupa l'n-esima posizione nell'elenco.

La **numerazione delle sequenze è fattibile perché sono di lunghezza finita**, anche se illimitata. Cioè per qualunque intero d scelto a priori, esistono sequenze di lunghezza maggiore di d. Per sequenze di lunghezza infinita la numerazione non è possibile

Insiemi non numerabili Insiemi non equivalenti a N come R, $(0, 1)$, l'insieme di tutte le linee del piano, insieme delle funzioni in una o più variabili... $\Rightarrow$ l'**insieme dei problemi computazionali non è numerabile**. Perché un problema computazionale è sempre visualizzabile come una funziona matematica, che associa ad ogni insieme di dati espressi da k numeri interi il corrispondente risultato espresso da j numeri interi

$$f : N^k \rightarrow N^j$$

Quindi l'insieme di queste f **non è numerabile**.

Diagonalizzazione $F = \{$ funzioni $f \mid f : N \rightarrow \{0, 1\}\}$, ogni $f \in F$ è rappresentata da una sequenza infinita

x 0 1 2 3 ... n ...

$f(x)$ 0 1 0 1 ... 0 ...

ma se è possibile è rappresentabile con una regola (f 0 se x pari 1 se x dispari)
Per assurdo, ipotizzo F numerabile. Si può assegnare ad ogni funzione un numero progressivo nella numerazione e costruire una tabella infinita con tutte le funzioni.

x	0	1	2	3	4	5	6	7	8	...
$f_0(x)$	1	0	1	0	1	0	0	0	1	...
$f_1(x)$	0	0	1	1	0	0	1	1	0	...
$f_2(x)$	1	1	0	1	0	1	0	0	1	...
$f_3(x)$	0	1	1	0	1	0	1	1	1	...
$f_4(x)$	1	1	0	0	1	0	0	0	1	...

Definisco $g(x) = \begin{cases} 0 & f_x(x) = 1 \\ 1 & f_x(x) = 0 \end{cases} \Rightarrow g$ non può corrispondere a nessuna delle f_i della tabella, perché differisce da tutte le funzioni almeno nella diagonale principale.
$g(x) \mid 0\,1\,1\,1\ldots$
Per assurdo $\exists\, j \mid g(x) = f_j(x) \Rightarrow g(j) = f_j(j)$ ma per la definizione $g(j)$ è il complemento di $f_j(j)$, quindi $g(j) \neq f_j(j)$ **contraddizione**.
Per qualunque numerazione scelta esiste sempre almeno una funzione esclusa, quindi F non è numerabile.

1.4.1 Algoritmi

Algoritmi la **formulazione di un algoritmo**, una sequenza finita di operazioni, completamente e univocamente determinate, **dipende dal modello di calcolo utilizzato**.
Qualunque modello si scelga, gli algoritmi devono essere descritti da sequenze finite di caratteri di un alfabeto finito $\Rightarrow$ sono **possibilmente infiniti ma numerabili**.

Problemi computazionali Sono **funzioni matematiche** che associano ad ogni insieme di dati il corrispondente risultato, e **non sono numerabili** come visto prima.

Problema della rappresentazione C'è una drastica perdita di potenza, perché gli algoritmi sono numerabili ma sono meno dei problemi computazionali

$$|\{Problemi\}| >> |\{Algoritmi\}|$$

$\Rightarrow$ **esistono problemi privi di un corrispondente algoritmo di calcolo**. Per esempio, il problema dell'arresto.

Lezione di Turing *Non esistono algoritmi che decidono il comportamento di altri algoritmi esaminandoli dall'esterno, cioè senza passare dalla loro simulazione.*

1.4.2 Modelli di calcolo

La teoria della calcolabilità dipende dal modello di calcolo?
Oppure
la decidibilità è una proprietà del problema?

I linguaggi di programmazione esistenti sono tutti equivalenti?
Ce ne sono di alcuni più potenti/più semplici di altri?
Ci sono algoritmi descrivibili in un linguaggio ma non in un altro?
È possibile che problemi oggi irrisolvibili possano essere risolti in futuro con altri linguaggi o altri calcolatori?
Le teorie della calcolabilità e della complessità dipendono dal modello di calcolo?

Tesi di Church-Turing Tutti i *ragionevoli* modelli di calcolo **risolvono esattamente la stessa classe di problemi**, quindi **si equivalgono nella possibilità di risolvere i problemi** pur operando con diversa efficienza.

Tesi C-H: la decidibilità è una proprietà del problema

Incrementi qualitativi sui calcolatori o sui linguaggi di programmazione servono **solo** ad abbassare i tempi di esecuzione o rendere più agevole la programmazione.