Agenzia per la
Cybersicurezza Nazionale

LINEE GUIDA FUNZIONI CRITTOGRAFICHE

Introduzione alla Crittografia e alle Linee Guida

MAGGIO 2026

Versione	Data di pubblicazione	Note
1.0	**11/07/2024**	**Prima pubblicazione**
1.1	**12/05/2026**	**Aggiunta sezione “Scopo del documento”, ulteriori modifiche minori**

Sommario

pag.

Indice delle figure

pag.

Scopo del documento

Questo documento costituisce parte della serie "Linee Guida Funzioni Crittografiche" e fornisce un'introduzione su nozioni e definizioni alla base della crittografia, sulle utilità degli strumenti crittografici in base alle necessità applicative e sulle Linee Guida in generale. Per le raccomandazioni nei contesti specifici, si rimanda agli altri documenti della serie.

Ogni documento tiene in considerazione le minacce presenti al giorno della sua pubblicazione. Data la diversa natura dei sistemi informativi di destinazione, non è possibile garantire che queste raccomandazioni possano essere utilizzate senza adattamenti specifici. In qualsiasi caso, la pertinenza dell'attuazione delle soluzioni proposte deve essere sottoposta, preventivamente, a valutazione e validazione da parte dei responsabili della sicurezza dei sistemi informativi di destinazione.

I contenuti delle Linee Guida sono indirizzati a sviluppatori, produttori di dispositivi e fornitori di servizi digitali, al fine di promuovere l'utilizzo di primitive crittografiche e relativi parametri di configurazione sicuri fin dalla fase di progettazione di prodotti, reti, applicazioni e servizi. Questi documenti sono altresì rivolti a security manager, referenti e responsabili della cybersicurezza di soggetti pubblici e privati affinché verifichino che i sistemi utilizzati dalla propria organizzazione siano conformi alle raccomandazioni fornite.

La legge 28 giugno 2024, n. 90 relativa a "Disposizioni in materia di rafforzamento della cybersicurezza nazionale e di reati informatici", all'articolo 9, stabilisce a tal fine che *«le strutture di cui all'articolo 8 della presente legge nonché quelle che svolgono analoghe funzioni per i soggetti di cui all'articolo 1, comma 2-bis, del decreto-legge 21 settembre 2019, n. 105, convertito, con modificazioni, dalla legge 18 novembre 2019, n. 133, e al decreto legislativo 18 maggio 2018, n. 65, verificano che i programmi e le applicazioni informatiche e di comunicazione elettronica in uso, che utilizzano soluzioni crittografiche, rispettino le linee guida sulla crittografia nonché quelle sulla conservazione delle password adottate dall'Agenzia per la cybersicurezza nazionale e dal Garante per la protezione dei dati personali e non comportino vulnerabilità note, atte a rendere disponibili e intellegibili a terzi i dati cifrati»*.

Il documento è stato curato dal Centro Nazionale di Crittografia istituito presso ACN.

Lista dei simboli matematici utilizzati

$\{0,1\}^*$	Insieme di stringhe binarie di lunghezza arbitraria	$O(n)$	Notazione O-grande
Gen	Funzione di generazione delle chiavi	$\mathcal{K}$	Spazio delle chiavi
Enc	Funzione di cifratura	$\mathcal{C}$	Spazio dei testi cifrati
Dec	Funzione di decifratura	$\mathcal{P}$	Spazio dei testi in chiaro

1 Introduzione

La crittografia nasce storicamente dalla necessità di assicurare la **confidenzialità** dei dati trasmessi, cioè la situazione in cui si vuole mantenere la segretezza di una comunicazione tra due o più utenti, in un contesto popolato anche da altre entità, potenzialmente malevole. Questa necessità si presentò già in tempi molto antichi, principalmente nell'ambito militare, in cui la confidenzialità assicurava che ordini e altre informazioni cruciali non venissero intercettate dagli avversari. La maggior parte dei cifrari storici è di tipo **alfabetico**, basati quindi sull'utilizzo delle lettere dell'alfabeto. Dalla metà del XX secolo, con la nascita della moderna teoria dell'informazione, è risultato necessario adottare la **codifica binaria** dei dati, rivelatasi poi più efficiente e ancora attualmente utilizzata nei moderni sistemi di cifratura.

A seguito della nascita dei sistemi di comunicazione digitali e della loro rapida diffusione, la crittografia ha assunto sempre più importanza per la protezione dei dati e servizi. Al giorno d'oggi, in una società in cui la maggior parte dei dati sensibili sono digitali e non più conservati materialmente, essa svolge diverse funzioni fondamentali per tenere al sicuro le informazioni. Ad esempio, i sistemi crittografici proteggono i dati scambiati su **internet** tra un utente e un server web, difendono i dati delle carte di pagamento durante le **transazioni** online o tramite terminale POS, proteggono le **password** utilizzate dagli utenti per l'accesso a vari servizi informatici, permettono di apporre **firme digitali** non falsificabili a documenti e messaggi. Inoltre, la crittografia è alla base del corretto funzionamento di nuove tecnologie, come la **blockchain** e i nuovi strumenti di **identità digitale** (IT Wallet, EUDI Wallet, Carta d'identità elettronica, ...). Per evitare che dei malintenzionati possano appropriarsi di dati protetti sfruttando eventuali debolezze, è di fondamentale importanza che i sistemi crittografici adoperati dai fornitori di tali servizi informatici siano sicuri e al passo con gli avanzamenti della ricerca scientifica.

L'**Agenzia per la Cybersicurezza Nazionale**, quale autorità nazionale in materia di cybersicurezza, ha il compito di assumere iniziative idonee a valorizzare la crittografia, come la pubblicazione dei documenti della serie **Linee Guida Funzioni Crittografiche**.

Al fine di incentivare l'uso di sistemi crittografici sicuri e validati dalla comunità scientifica, tali documenti forniscono **raccomandazioni** su funzioni e parametri da adottare in base alle applicazioni specifiche, in linea con quanto previsto in ambito europeo. Questi documenti saranno aggiornati periodicamente per essere al passo con lo sviluppo delle nuove tecniche crittografiche e le scoperte scientifiche.

In questo documento si illustrano alcuni concetti alla base della crittografia per facilitare la lettura dei documenti della serie, più tecnici e specifici, dedicati alle singole **funzioni crittografiche**. In particolare, nel capitolo 2 si introducono alcuni concetti generali di crittografia, nel capitolo 3 si definisce cosa è la crittoanalisi e infine, nel capitolo 4, si introduce la serie delle Linee Guida, fornendo delle indicazioni su come utilizzare al meglio questi documenti.

2 I concetti della crittografia

La crittografia, seppur nata da applicazioni pratiche, è una disciplina con profonde basi teoriche che sono state sviluppate e identificate principalmente nell'ultimo secolo. Di seguito si introducono le definizioni più importanti e i concetti necessari per una comprensione più profonda dei sistemi crittografici e della loro sicurezza.

2.1 Le parole chiave della crittografia

In generale, un **crittosistema** permette di eseguire la **cifratura** di un **messaggio in chiaro** (o plaintext) utilizzando una o più **chiavi** crittografiche (key), per ottenere un **messaggio cifrato** (ciphertext). Questa operazione deve essere invertibile, in modo che utilizzando le chiavi, sia possibile effettuare la **decifratura** del messaggio cifrato e riottenere il messaggio in chiaro di partenza. Inoltre, affinché il cifrario possa dirsi **corretto**, per tutti i messaggi e le chiavi possibili, deve valere che la decifratura di un messaggio cifrato restituisca unicamente il messaggio stesso. Solo in tal caso l'operazione di decifratura non crea ambiguità e permette una comunicazione senza errori.
L'algoritmo di cifratura viene detto **deterministico** nel caso in cui, partendo da input uguali, si ottengono sempre output uguali. Se invece viene coinvolta una componente casuale all'interno della procedura, l'algoritmo si dice **probabilistico** e quindi, a partire dagli stessi input, si ottengono output cifrati differentemente a ogni esecuzione.
Quando si descrive uno schema crittografico si fa riferimento a un mittente del messaggio e a un destinatario, che assumono classicamente i nomi di Alice e Bob.
In questo scenario, è spesso presente anche una terza figura che intercetta le comunicazioni che vengono inviate nel canale di comunicazione, che prende il nome di Eve (dall'inglese eavesdropper, spione). Due esempi di schemi crittografici sono rappresentati in Figura 1, nella quale Alice e Bob, colorati in blu, eseguono una comunicazione mentre Eve, rappresentata in rosso, cerca di ricavarne informazioni. Un attaccante capace di intercettare e modificare i messaggi scambiati viene generalmente chiamato Mallory.

Dal punto di vista matematico, l'insieme di tutti i possibili messaggi in chiaro, delle chiavi e dei messaggi cifrati sono solitamente indicati rispettivamente con $\mathcal{P}$, $\mathcal{K}$ e $\mathcal{C}$. Un algoritmo $Gen : \{0,\ 1\}^* \rightarrow \mathcal{K}$ genera la chiave k utilizzata per la cifratura. Fissata la chiave, è possibile definire l'algoritmo di **cifratura** $Enc_k : \mathcal{P} \rightarrow \mathcal{C}$ e il corrispondente algoritmo inverso $Dec_k : \mathcal{C} \rightarrow \mathcal{P}$, che rappresenta la **decifratura**. In linguaggio formale, la verifica di **correttezza** si traduce nel seguente: per ogni messaggio in chiaro m e per ogni chiave k, si deve verificare che $Dec_k\left(Enc_k\left(m\right)\right) = m$.

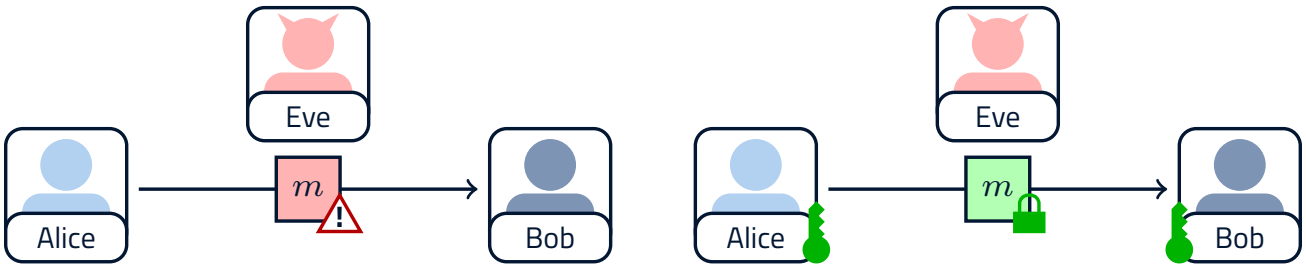

Figura 1 - Comunicazione in presenza di un utente malevolo: a sinistra, l'assenza di cifratura permette a Eve di intercettare il messaggio m e di poterne leggere il contenuto; a destra, il messaggio m è protetto tramite la cifratura con la chiave verde in possesso solo di Alice e Bob, mentre Eve non è in grado di leggere il contenuto del messaggio (crittografia simmetrica)

2.2 Gli obiettivi della crittografia

Con il passare degli anni e il rapido sviluppo della tecnologia, si è passati dalla sola necessità di dover assicurare la **confidenzialità** di un messaggio, cioè mantenerlo segreto a chiunque non sia autorizzato a leggerne il contenuto, a dover garantire anche altre importanti proprietà:

- **integrità**, ossia la protezione di un dato, trasmesso o salvato, da modifiche accidentali o illecite. Ad esempio, questa proprietà è garantita dalle **funzioni di hash**;
- **autenticazione**, che significa confermare ai destinatari l'identità del mittente. Come si intuisce dal nome, un esempio è dato dai **codici di autenticazione del messaggio** (MAC);
- **non ripudio**, cioè non consentire all'utilizzatore di negare la paternità del dato. Ad esempio, questa proprietà è garantita dagli schemi di **firma digitale**.

2.3 Crittografia simmetrica e crittografia a chiave pubblica

Fino agli anni '70, l'unica forma di crittografia conosciuta era quella cosiddetta **simmetrica**, che raccoglie i sistemi crittografici che basano la loro sicurezza sull'utilizzo di una singola **chiave segreta** condivisa tra gli interlocutori. In questo caso, quindi, gli utenti coinvolti nella comunicazione utilizzano la stessa chiave sia per cifrare che per decifrare il messaggio, impedendo allo stesso tempo che qualsiasi utente esterno, non in possesso della chiave, possa intuirne il contenuto (come illustrato a destra in Figura 1).
I cifrari simmetrici possono essere suddivisi in due sottocategorie: i **cifrari a blocchi** e i **cifrari a flusso**, trattati nei documenti dedicati [1, 2]. La differenza sostanziale risiede nella gestione dei dati ricevuti in input dagli algoritmi: mentre nel primo caso questi vengono cifrati (e decifrati) dopo essere stati suddivisi in blocchi di lunghezza fissa, nel secondo il messaggio viene cifrato byte a byte (o bit a bit). I vantaggi principali di questi ultimi sono la velocità e i bassi requisiti di hardware, mentre i cifrari a blocchi assicurano una maggior sicurezza e sono generalmente più versatili. In entrambi i casi, lo scambio della chiave segreta è di fondamentale importanza per assicurare la confidenzialità della conversazione. La soluzione teorica più diretta consiste nell'instaurare un **canale sicuro**, cioè immune a intromissioni indesiderate, ma ciò risulta molto complesso da realizzare nella pratica.
Per risolvere questo problema, nel 1970, James Henry Ellis, ingegnere e crittografo presso l'Agenzia di sicurezza del Regno Unito, propose per la prima volta l'idea di una chiave non segreta in una relazione classificata [3].
I primi cifrari basati su chiave pubblica furono sviluppati da due suoi colleghi, Clifford Cocks, che nel 1973 ideò un crittosistema equivalente a quello che divenne poi noto con il nome di RSA, e Malcolm Williamson, che nel 1974 inventò il sistema pubblicato due anni dopo con il nome Diffie-Hellman. Tuttavia, queste informazioni restarono classificate fino al 1997.
La prima pubblicazione non segretata relativa alla **crittografia a chiave pubblica** o **asimmetrica** risale al 1976, quando Whitfield Diffie e Martin Hellman svilupparono, ispirandosi alle idee di Ralph Merkle ma indipendentemente da Williamson, il primo crittosistema a chiave pubblica [4]. Nell'anno successivo, Ron Rivest, Adi Shamir e Leonard Adleman pubblicarono un sistema di cifratura a chiave

pubblica, ancora oggi ampiamente diffuso, che prese il nome dalle iniziali dei loro cognomi: **RSA** [5].
Un crittosistema asimmetrico sfrutta una **funzione one-way**, ossia una funzione semplice da calcolare su qualsiasi input, ma che richiede tempi improponibili per recuperare l'input di partenza a partire da un output casuale, a meno che non si conosca un dato segreto. Dal punto di vista crittografico, supponendo che Alice voglia inviare un messaggio a Bob, tale proprietà si concretizza tramite l'utilizzo di una coppia di chiavi generata da Bob. Questa è costituita da una **chiave pubblica**, che può essere nota a chiunque, e da una **chiave privata**, che rappresenta il dato **segreto**. Alice è in grado di cifrare un messaggio usando la chiave pubblica di Bob e applicando la funzione one-way, mentre solo Bob, l'unico a conoscenza della chiave privata, è in grado di invertire il processo e decifrare il messaggio. Uno schema di questo tipo, rappresentato in Figura 2, prende il nome di **cifratura a chiave pubblica** o **PKE** (Public-Key Encryption).
Le funzioni matematiche utilizzate sono solitamente basate su un problema matematico complesso, per esempio la **fattorizzazione** dei numeri interi nel caso di RSA o il problema del **logaritmo discreto** su campi finiti o su curve ellittiche nel caso dello schema di Diffie-Hellman.
Il principale vantaggio della crittografia a chiave pubblica è che non necessita di un canale sicuro per concordare una chiave segreta prima di iniziare la comunicazione. Tuttavia, il calcolo delle funzioni one-way è molto meno efficiente rispetto ai metodi di crittografia simmetrica, sia in termini di tempo, che in termini di prestazioni. Di conseguenza, una delle applicazioni principali della crittografia a chiave pubblica è la condivisione della chiave segreta tra mittente e destinatario per l'utilizzo di un crittosistema simmetrico. In base alle specifiche, questi schemi prendono il nome di **scambio di chiave** o **KEX** (Key Exchange) oppure **meccanismi di incapsulamento della chiave** o **KEM** (Key Encapsulation Mechanisms). Solitamente, questi vengono utilizzati prima della cifratura con sistemi simmetrici e in tal caso la combinazione ottenuta viene chiamata **crittosistema ibrido**.

Dal punto di vista matematico, uno schema di cifratura a chiave pubblica prevede i seguenti passi:

- Bob genera la sua coppia di chiavi tramite $Gen(\cdot) = pk, sk$ (dove $\cdot$ rappresenta una lista di parametri), pubblica pk e tiene segreta sk;
- Alice utilizza la chiave pubblica pk di Bob per cifrare il messaggio m ottenendo il cifrato $c = Encpk(m)$ che invia a Bob tramite un canale insicuro;
- Bob utilizza la sua chiave segreta sk per decifrare il cifrato c ricostruendo $m = Dec_{sk}(c)$.

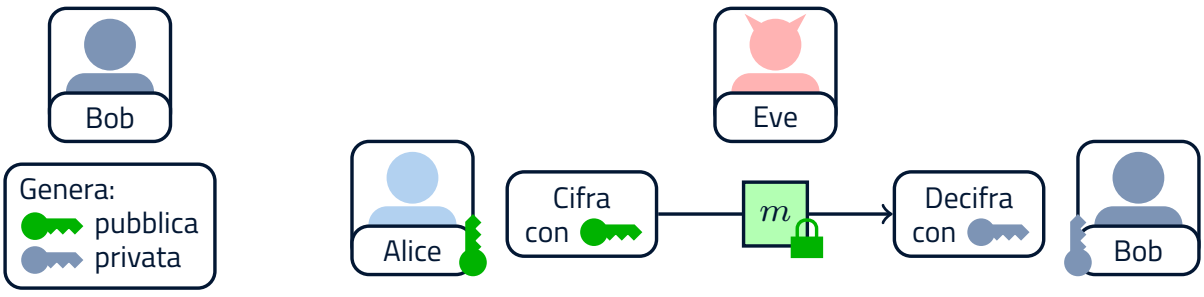

Figura 2 - Schema di cifratura a chiave pubblica: a sinistra, la fase di generazione delle chiavi; a destra, le fasi di cifratura, invio del cifrato tramite canale insicuro e decifratura

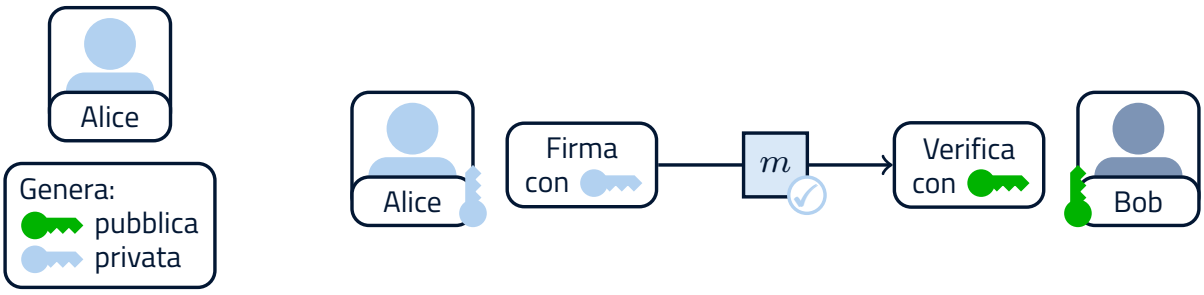

Figura 3 - Schema a chiave pubblica per la firma digitale: a sinistra, la fase di generazione delle chiavi; a destra, le fasi di firma, invio di messaggio con firma tramite canale insicuro e verifica

Anche gli schemi di **firma digitale** sono crittosistemi a chiave pubblica ampiamente utilizzati. In questo contesto, Alice, dopo aver generato una coppia di chiavi pubblica e privata, può firmare un messaggio utilizzando la sua chiave privata. Invia quindi a Bob sia firma che messaggio, eventualmente applicando a quest'ultimo un algoritmo di cifratura per proteggerne la confidenzialità. Bob può, infine, utilizzare la chiave pubblica di Alice per controllare la legittimità della firma. Il processo è rappresentato in Figura 3. Si noti come chiunque in possesso della chiave pubblica di Alice possa verificare la firma del messaggio e che, data l'unicità della corrispondenza tra chiave privata e chiave pubblica, Alice non possa negare di aver firmato il messaggio (non ripudio).

Dal punto di vista matematico, uno schema per la firma digitale prevede i seguenti passi:

- Alice genera la sua coppia di chiavi tramite $Gen(\cdot) = pk, sk$ (dove $\cdot$ rappresenta una lista di parametri), pubblica pk e tiene segreta sk;
- Alice utilizza la sua chiave segreta sk per firmare il messaggio m ottenendo la firma $\sigma = Sig_{sk}(m)$ che invia con m a Bob, anche tramite un canale insicuro;
- Bob utilizza la chiave pubblica pk di Alice per verificare la validità della firma σ sul messaggio m tramite $Ver_{pk}(m, \sigma)$. Questo algoritmo restituisce "accettata" se la firma è valida, "rifiutata" altrimenti.

2.4 Complessità

Un crittosistema deve essere **sicuro**, cioè non deve presentare delle debolezze che permettano a un attaccante di **romperlo**, cioè ottenere o modificare illecitamente le informazioni da esso protette. Questo non vuol dire necessariamente che non esistono metodi per ricavare le informazioni protette, ma che tali procedure richiedono **tempi impraticabili** dal punto di vista computazionale. Per esempio, si consideri un cifrario con una chiave di lunghezza fissata: in questo caso, un attaccante potrebbe procedere per tentativi e provare tutte le chiavi di tale lunghezza, fino a riuscire a decifrare un dato messaggio. Questo tipo di attacco, detto **attacco a forza bruta** (brute-force), può essere eseguito su qualsiasi crittosistema in quanto le specifiche delle soluzioni utilizzate vengono sempre considerate note a priori. Tuttavia, se un crittosistema utilizza parametri adeguati, questo attacco non inficia assolutamente la sua sicurezza, in quanto il tempo necessario per provare tutti i possibili input sarebbe eccessivamente elevato. Quando si parla di sicurezza di un crittosistema, si possono distinguere diverse tipologie:

- **sicurezza incondizionata** o **perfetta**: il crittosistema è inattaccabile, anche con una potenza computazionale infinita;
- **sicurezza computazionale**: non si conosce un metodo in grado di rompere il crittosistema in tempi ragionevoli.

Si parla inoltre di **sicurezza dimostrabile** quando si può dimostrare che rompere il crittosistema è equivalente a risolvere un problema riconosciuto come computazionalmente difficile. Questo concetto può essere riferito tanto alla sicurezza incondizionata, quanto a quella computazionale.

L'unico crittosistema conosciuto con sicurezza incondizionata è il **cifrario di Vernam** o **one-time pad**, che prevede l'utilizzo di una chiave segreta della stessa lunghezza del messaggio e diversa per ogni cifratura. Nella pratica, questo cifrario è particolarmente inefficiente: per ogni messaggio andrebbe precedentemente scambiata una nuova chiave segreta tramite un canale sicuro, il che sarebbe equivalente a trasmettere su tale canale direttamente il testo in chiaro. Di conseguenza, nella crittografia moderna si utilizzano crittosistemi la cui sicurezza è dimostrabile o, più spesso, solo computazionale. È quindi importante definire cosa si intende con termini di uso comune come "difficile" o "in tempi ragionevoli", introducendo il concetto di complessità.

La **complessità computazionale** di un algoritmo viene misurata in termini di numero di operazioni elementari necessarie per l'esecuzione e indicata tramite la notazione $O(\cdot)$, detta **O-grande** o **simbolo di Landau** [6], dal nome di uno dei primi matematici a utilizzare questo simbolo. Questa notazione indica generalmente il tasso di crescita di una data funzione ma, nel caso particolare della complessità di un algoritmo, rappresenta l'ordine di grandezza del numero di operazioni richieste dall'algoritmo per terminare. Per quanto riguarda il calcolo della complessità di un algoritmo di attacco, si deve sempre considerare il **caso peggiore** in cui si riesce a trovare la chiave. Per esempio, nell'attacco a forza bruta, il caso peggiore si verifica quando l'attaccante prova tutte le chiavi fino all'ultima, che risulta quella corretta. Se la lunghezza della chiave è n, allora questo attacco risulta avere complessità $O\left(2^n\right)$, corrispondente al numero di chiavi da provare. Lo studio del caso peggiore fornisce una stima attendibile della realizzabilità di un attacco, che tuttavia non corrisponde a quella dei casi reali che invece sono modellati come istanze casuali. A tale scopo, sarebbe più interessante poter calcolare la complessità dell'algoritmo nel **caso medio**, ma stime di questo tipo sono molto complesse da ricavare.

Gli algoritmi eseguibili in tempo **polinomiale**, cioè con complessità $O(n^c)$ con n parametro (ad esempio, la lunghezza della chiave) e c costante, sono quelli più rapidi da eseguire e da trattare. Al contrario, gli algoritmi con complessità $O(c^n)$ terminano in tempo **esponenziale**, per cui il tempo necessario per ottenere un risultato cresce vertiginosamente al crescere del parametro n.

Un prerequisito per qualsiasi crittosistema è che gli algoritmi che la compongono abbiano complessità polinomiale quando si è a conoscenza della chiave, mentre gli attacchi per il recupero della chiave risultino essere tutti esponenziali. Chiaramente, l'attacco a forza bruta è il più semplice da effettuare, ma ha complessità esponenziale e quindi non consente fattivamente il recupero della chiave. Si congettura che non esistano degli algoritmi polinomiali in grado di risolvere i problemi della fattorizzazione e del logaritmo discreto, ma tale enunciato non è stato ancora dimostrato matematicamente. Per questo motivo, la sicurezza degli schemi a chiave pubblica come RSA e Diffie-Hellman è computazionale ma non dimostrabile.

L'efficienza di un algoritmo non si misura solamente con la sua complessità computazionale. Sia che si tratti di crittosistemi o di attacchi, è fondamentale tenere in considerazione anche la **memoria** richiesta dagli algoritmi adottati, per evitare di sovraccaricare le componenti hardware o i canali di comunicazione. In contesti particolari, come nei protocolli crittografici, risulta rilevante anche valutare il numero di **interazioni** tra le parti coinvolte per ridurre il numero di scambi da effettuare attraverso il canale di comunicazione.

2.5 Principi di Shannon

Secondo la teoria della comunicazione di Claude Shannon [7], i principi fondamentali alla base di un cifrario simmetrico sicuro sono due:

- **confusione**: la relazione tra chiave e testo cifrato deve essere più complicata possibile, in modo che sia computazionalmente impossibile risalire facilmente alla chiave da un testo cifrato intercettato;
- **diffusione**: il cifrario deve massimizzare la distribuzione delle informazioni contenute all'interno del testo in chiaro in tutto il testo cifrato, alterando totalmente le statistiche del messaggio di partenza.

Queste proprietà continuano a essere alla base della progettazione dei moderni **cifrari a blocchi**, in cui le S-box si occupano di garantire adeguata confusione utilizzando funzioni non lineari, mentre le P-box garantiscono la diffusione tramite funzioni lineari che riordinano i bit del cifrato in modo da massimizzare l'effetto delle altre componenti. Per maggiori informazioni, si rimanda al documento dedicato ai cifrari a blocchi [1].

2.6 La minaccia quantistica

Negli ultimi decenni, prima solo teoricamente e poi più concretamente, si è definito il concetto di **computer quantistico** [8], cioè una macchina i cui processi e operazioni si basano sulla meccanica quantistica. Le capacità di queste nuove tecnologie sono ancora scarse per ragioni costruttive, ma il loro potenziale è molto alto e si prevede che saranno in grado di surclassare i computer classici su alcune particolari classi di problemi. In aggiunta, sono stati ideati nuovi algoritmi implementabili solo su computer quantistici che metterebbero a rischio alcuni problemi alla base della crittografia moderna. In particolare, l'algoritmo di **Shor** [9] sarebbe in grado di risolvere efficientemente sia il problema della fattorizzazione che quello del logaritmo discreto, rompendo tutti i sistemi moderni basati su questi due problemi.

Le soluzioni trovate ad oggi dalla comunità scientifica si dividono in due filoni principali: la crittografia **post-quantum** e la crittografia **quantistica**.

La crittografia post-quantum comprende tutti i sistemi crittografici costruiti per essere in grado di resistere sia agli attacchi classici che a quelli quantistici. La spinta principale in questa direzione viene dal National Institute of Standards and Technology (NIST) statunitense, che nel 2016 ha avviato una competizione per trovare degli standard post-quantum per lo scambio di chiave e per la firma digitale [10]. Nel 2022, il NIST ha annunciato i primi quattro vincitori della competizione, uno schema per lo scambio di chiave e tre schemi per la firma digitale [11], ma la gara è rimasta aperta per trovare nuovi schemi e diversificare le tipologie degli standard.

La crittografia quantistica, invece, sfrutta i principi intrinseci della meccanica quantistica, come il principio di indeterminazione di Heisenberg, per garantire la sicurezza di protocolli crittografici dedicati. L'applicazione crittografica più conosciuta e diffusa dei sistemi quantistici riguarda i metodi di **distribuzione quantistica della chiave** o **QKD** (Quantum Key Distribution). Questi metodi crittografici utilizzano un cavo di fibra ottica o canali satellitari per inviare dati codificati come fotoni polarizzati e la loro sicurezza è garantita dal fatto che, grazie ai principi della fisica quantistica, cercare di ottenere illecitamente i dati in fase di scambio modifica immediatamente e irreparabilmente gli stati quantistici ad essi associati e quindi ogni intrusione malevola può essere rilevata dai legittimi interlocutori.

Per maggiori informazioni su questi argomenti, si rimanda al documento informativo sulla crittografia quantistica e post-quantum [12].

3 Crittoanalisi

Quando si valuta la sicurezza di un sistema crittografico, si assume valido il **principio di Kerckhoffs** [13], ossia che le specifiche dei crittosistemi utilizzati siano note a tutti, per cui la loro sicurezza deve basarsi unicamente sulla segretezza della chiave. Con il termine **crittoanalisi** si intende il processo di analizzare un sistema crittografico al fine di trovare delle possibili debolezze all'interno della sua struttura che permettano di progettare attacchi migliori di quello a forza bruta, inficiandone così la sicurezza. La crittoanalisi di un cifrario non è utilizzata solamente per scopi malevoli, ma è anche alla base del processo di validazione di un sistema crittografico: spesso, come già detto, è difficile fornire una dimostrazione teorica della sicurezza, per cui vengono ritenuti sicuri i sistemi ampiamente studiati, ma per cui non sono stati comunque trovati attacchi efficienti. È importante evidenziare che le debolezze possono sorgere anche a causa delle modalità in cui un cifrario viene applicato o utilizzato, a prescindere dalla struttura del cifrario stesso o dal problema matematico correlato. Infatti, una scorretta implementazione potrebbe inficiare la sicurezza di un cifrario ritenuto sicuro, rendendolo vulnerabile ad attacchi da parte di malintenzionati.

3.1 L'obiettivo di un attaccante

La crittoanalisi consente a un attaccante di individuare eventuali attacchi per il **recupero della chiave segreta** (key recovery), che può poi usare in diversi modi illeciti a seconda delle funzionalità del sistema crittografico attaccato. Ad esempio, potrebbe appropriarsi di dati protetti nel caso di un metodo di cifratura simmetrico o apporre una firma non propria nel caso della firma digitale. Quando la chiave viene recuperata interamente si ottiene una **rottura totale** (total break). Questo scenario non è l'unico possibile e frequentemente è sufficiente ottenere meno informazioni per inficiare la sicurezza di un cifrario.
Infatti, se l'attaccante riuscisse a recuperare efficientemente anche solo buona parte dei bit di una chiave, potrebbe procedere con un attacco a forza bruta sui restanti bit e recuperare la chiave completa in tempi brevi.
In alternativa, un attaccante può cercare di trovare un algoritmo equivalente a quello utilizzato in fase di cifratura e decifratura, ma in grado di eseguire l'operazione senza utilizzare la chiave segreta. In questo caso si parla di **deduzione globale** (global deduction). Altri possibili scenari, più semplici da ottenere, sono la **deduzione di informazioni** (information deduction), che si riferisce al caso in cui l'attacco consente di acquisire informazioni riguardanti testi in chiaro o testi cifrati precedentemente sconosciuti, o l'**attacco di distinzione** (distinguisher attack), che permette all'attaccante di distinguere se un dato proviene dall'output di una cifratura oppure se è solamente una stringa casuale di bit [14].

Nel contesto delle firme digitali, invece, l'obiettivo dell'attaccante potrebbe essere o quello di ricavare la chiave, e quindi ricadere nel caso precedente, oppure quello di riprodurre una firma **falsificata** (forgery) per un messaggio arbitrario. I tipi possibili di falsificazione si distinguono in [13]:

- **falsificazione universale** (universal forgery): l'attaccante riesce a produrre una firma valida per qualsiasi messaggio, pur non essendo in possesso della chiave segreta;
- **falsificazione selettiva** (selective forgery): l'attaccante riesce a produrre una firma valida solo per alcuni particolari messaggi scelti dall'attaccante a priori;
- **falsificazione esistenziale** (existential forgery): l'attaccante riesce a produrre una firma valida per almeno un messaggio, sul quale l'attaccante non ha controllo.

Chiaramente la prima tipologia è la peggiore delle tre, in quanto l'attaccante ha pieno controllo e può spacciarsi per il firmatario. La seconda e la terza sono via via meno problematiche, ma risultano comunque vulnerabilità importanti per il sistema di firma che quindi deve essere considerato insicuro.

3.2 Scenari di attacco

In base alle informazioni che l'attaccante possiede per effettuare la crittoanalisi è possibile distinguere diversi modelli di attacco [13]:

- **attacco con solo testo cifrato** o **COA** (Ciphertext-Only Attack): l'attaccante possiede solo alcuni testi cifrati, ad esempio nel caso in cui egli fosse in grado di intercettare i messaggi cifrati trasmessi nel canale di comunicazione;
- **attacco con testo in chiaro** o **KPA** (Known-Plaintext Attack): l'attaccante conosce alcune coppie di testo in chiaro e il corrispondente testo cifrato. Questa situazione può verificarsi facilmente nella pratica: molto spesso i messaggi iniziano con intestazioni che si ripetono e si concludono con firme che possono essere note;
- **attacco con testo in chiaro scelto** o **CPA** (Chosen-Plaintext Attack): l'attaccante può temporaneamente scegliere dei testi in chiaro e conoscere i corrispondenti testi cifrati. Un esempio diretto è dato dai cifrari a chiave pubblica, in quanto un attaccante può sempre applicare l'algoritmo di cifratura ottenendo le suddette coppie;
- **attacco con testo cifrato scelto** o **CCA** (Chosen-Ciphertext Attack): l'attaccante può temporaneamente scegliere dei testi cifrati e conoscere i corrispondenti testi in chiaro. In questo caso, si può supporre che l'attaccante abbia a disposizione l'algoritmo di decifratura per un tempo limitato.

Il primo modello è evidentemente il più comune e meno rischioso, in quanto l'attaccante possiede la minore quantità di informazioni, mentre gli altri casi descrivono situazioni più pericolose per via dell'aumentare della quantità e della qualità dei dati su cui basare l'attacco. Inoltre, nei primi due casi, l'attaccante è un mero osservatore dei messaggi che vengono scambiati e assume quindi un ruolo **passivo**. Al contrario, negli ultimi due scenari, l'attaccante applica direttamente cifratura o decifratura con chiave fissata, che utilizza come funzioni a "scatola chiusa" o **black box**, assumendo quindi un ruolo **attivo**. Infine, è interessante notare come sia possibile vedere un'istanza di un KPA come una di CPA nella quale l'attaccante sceglie come testi in chiaro quelli intercettati con il primo attacco. Per questo motivo, la resistenza a CPA è una proprietà più forte della resistenza a KPA.

I crittosistemi attualmente utilizzati prevedono la resistenza a questi tipi di attacchi, considerando sempre lo scenario più favorevole per l'attaccante, ossia quello in cui ha a disposizione la maggior quantità di informazioni.

4 Le Linee Guida Funzioni Crittografiche

Questo documento funge da introduzione per la serie **Linee Guida Funzioni Crittografiche** prodotta dalla Divisione Scrutinio Tecnologico e Crittografia del Servizio Certificazione e Vigilanza dell'Agenzia per la Cybersicurezza Nazionale.

L'obiettivo principale di questi documenti è aumentare il grado di consapevolezza dei produttori e fornitori di servizi digitali per gli aspetti di cybersicurezza, incentivando l'utilizzo di soluzioni crittografiche sicure e al passo con i tempi. Il rapido avanzamento delle tecnologie disponibili e i progressi della ricerca scientifica generano continuamente nuove sfide per la crittografia moderna, per cui crittosistemi attualmente ampiamente utilizzati potrebbero essere dismessi perché vulnerabili a nuovi attacchi o semplicemente perché divenuti deboli a seguito della produzione di computer più potenti e dotati di tecnologie più moderne. In molti contesti, tuttavia, la sostituzione di un crittosistema o la necessità di un aumento delle dimensioni dei parametri da adoperare non sono recepiti prontamente dai fornitori dei servizi informatici, per cui crittosistemi oramai meno sicuri continuano ad essere utilizzati anche per lungo tempo, prima di essere sostituiti da altri più moderni. Questo scenario può portare a gravi conseguenze, come furti di identità o di denaro, oppure, più in generale, a una compromissione della sicurezza dei dati.

Le linee guida forniscono raccomandazioni sui crittosistemi migliori da utilizzare secondo le valutazioni dell'Agenzia, che tengono in considerazione gli ultimi sviluppi della ricerca scientifica e le soluzioni adottate a livello internazionale. Tali raccomandazioni, ove possibile, sono complete dei parametri minimi consigliati per la loro esecuzione.

Tutti i documenti forniscono una panoramica iniziale informativa sulla funzione crittografica oggetto dell'analisi, seguita da una spiegazione dei crittosistemi di riferimento più utilizzati in cui sono segnalati i riferimenti bibliografici da consultare per una corretta trattazione e implementazione.

Nel testo sono stati inseriti:

- riquadri **informativi** per ulteriori approfondimenti più tecnici o teorici;
- riquadri di **warning** per gli aspetti imprescindibili e su cui porre maggiore attenzione;
- riquadri relativi alle possibili vulnerabilità dovute alla **minaccia quantistica**, un aspetto fondamentale da tenere in considerazione nell'attuale panorama della ricerca crittografica;
- una **tabella conclusiva** che indica i crittosistemi raccomandati e una lista di parametri minimi per la loro implementazione.

I documenti sono suddivisi per argomento per facilitarne l'utilizzo ai lettori e per semplificarne l'aggiornamento.

Bibliografia

[1] ACN. *Cifrari a Blocchi e Modalità di Funzionamento*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[2] ACN. *Cifrari a Flusso*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[3] J. H. Ellis. *The possibility of secure non-secret digital encryption*. 3006. UK Communications Electronics Security Group, 1970. URL: https://nsarchive.gwu.edu/document/21971-document-02.

[4] W. Diffie e M. E. Hellman. «New directions in cryptography». In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[5] R. L. Rivest, A. Shamir e L. Adleman. «A method for obtaining digital signatures and public-key cryptosystems». In: *Communications of the ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342.

[6] E. Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. Leipzig B.G. Teubner, 1909. DOI: 10.1007/BF01742852.

[7] C. E. Shannon. «Communication theory of secrecy systems». In: *The Bell system technical journal* 28.4 (1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[8] R. P. Feynman. «Simulating Physics with Computers». In: *International Journal of Theoretical Physics* 21.6 (1982), pp. 467–488. DOI: 10.1007/BF02650179.

[9] P. W. Shor. «Algorithms for quantum computation: discrete logarithms and factoring». In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[10] NIST. *PQC Call for Proposals*. 2016. URL: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/call-for-proposals.

[11] NIST. *PQC Standardization Process: Announcing Four Candidates to be Standardized, Plus Fourth Round Candidates*. 2022. URL: https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4.

[12] ACN. *Crittografia post-quantum e quantistica - Preparazione alla minaccia quantistica*. Documenti informativi, 2024. URL: https://www.acn.gov.it/portale/crittografia.

[13] D. R. Stinson e M. Paterson. *Cryptography: Theory and Practice (4th edition)*. Chapman e Hall/CRC, 2017. DOI: 10.1201/9781315282497.

[14] L. R. Knudsen. «Contemporary Block Ciphers». In: *Lectures on Data Security: Modern Cryptology in Theory and Practice*. Springer, 1999, pp. 105–126. DOI: 10.1007/3-540-48969-X_5.