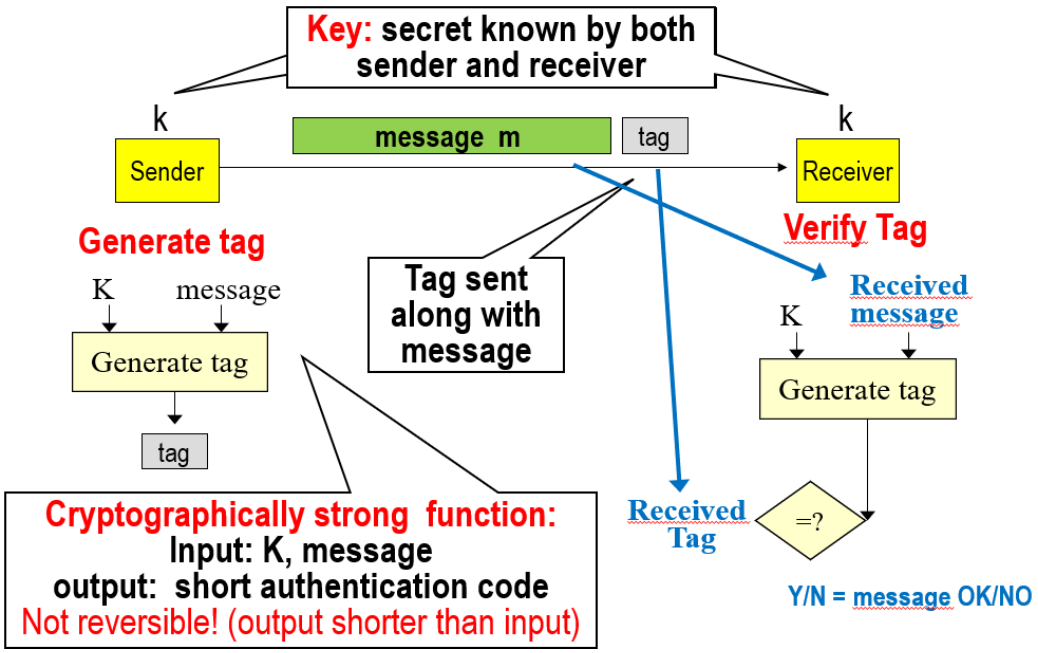
Message Authentication Codes (MAC) e Hash crittografici

Confidenzialità, Integrità e Autenticazione

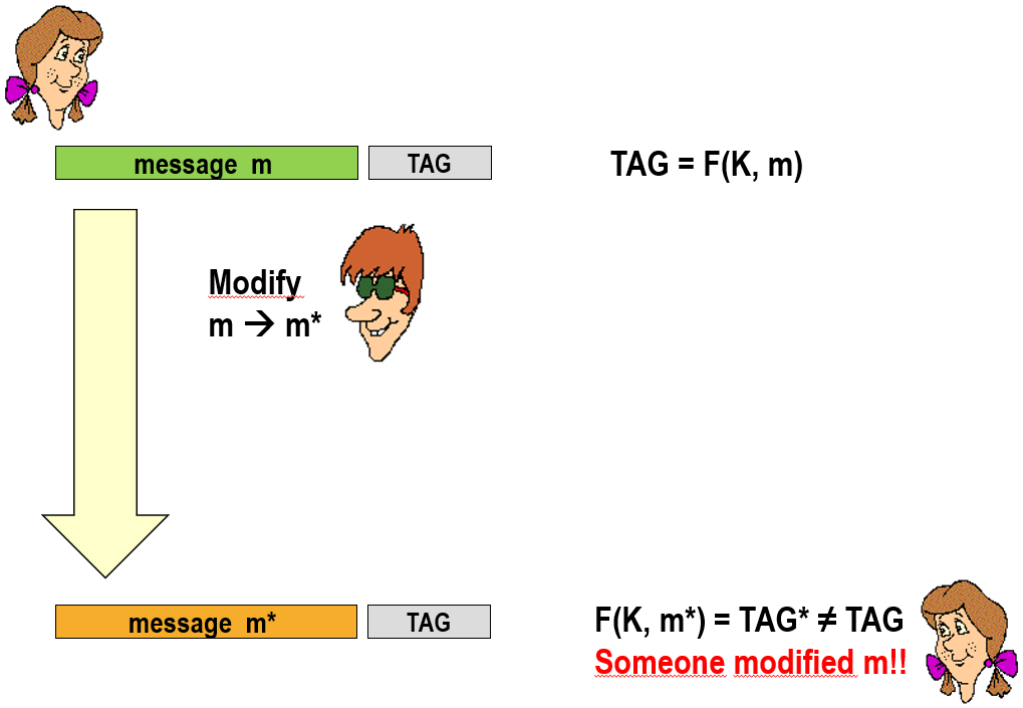
La **confidenzialità** assicura che solo il destinatario legittimo possa leggere un messaggio. Tuttavia, la **integrità** permette di verificare che il messaggio non sia stato alterato durante il transito, e la **autenticità** offre la certezza che il contenuto provenga effettivamente dall'emittente autorizzato. Per garantire integrità e autenticità occorre che mittente e destinatario condividano una chiave segreta — non basta la semplice cifratura: questa, da sola, non protegge dall'alterazione dei dati (eccezione: cifrari AEAD con autenticazione integrata).



Protezione da Attacchi Man-In-The-Middle (MITM)

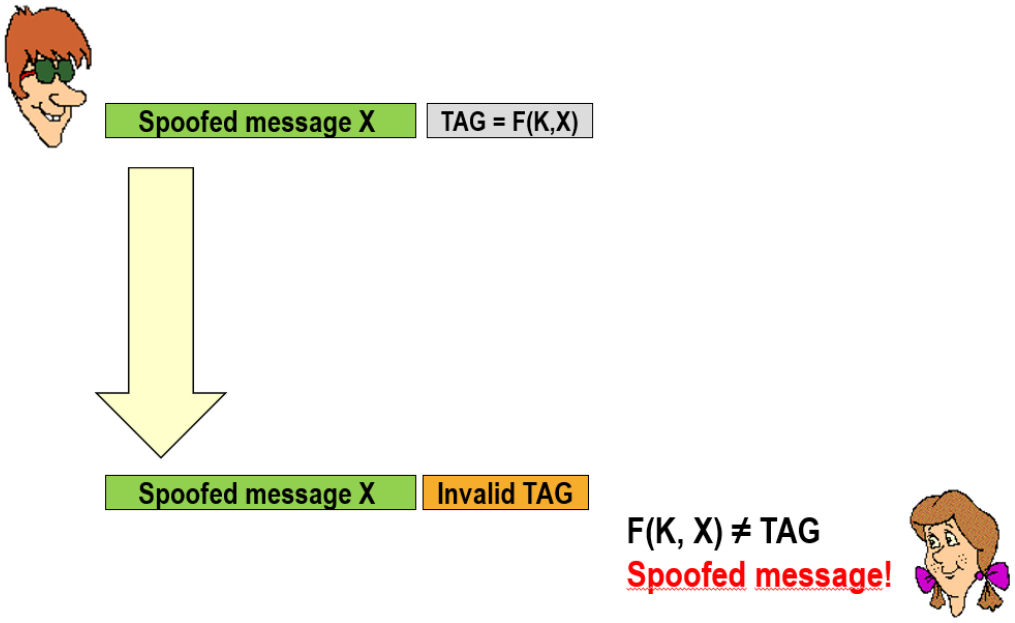
Un attacco *Man-In-The-Middle* (MITM) consiste nell'intercettazione e possibile alterazione della comunicazione tra due parti senza che esse ne siano consapevoli. Un protocollo sicuro deve proteggere la comunicazione assicurando che solo le parti legittime possano generare messaggi validi.

Questo si ottiene usando funzioni crittografiche forti come i codici di autenticazione (MAC) basati su chiavi segrete condivise. Solo chi possiede la chiave può generare un tag valido, impedendo così a un attaccante MITM di inserire o modificare messaggi senza essere scoperto.



Protezione da Spoofing

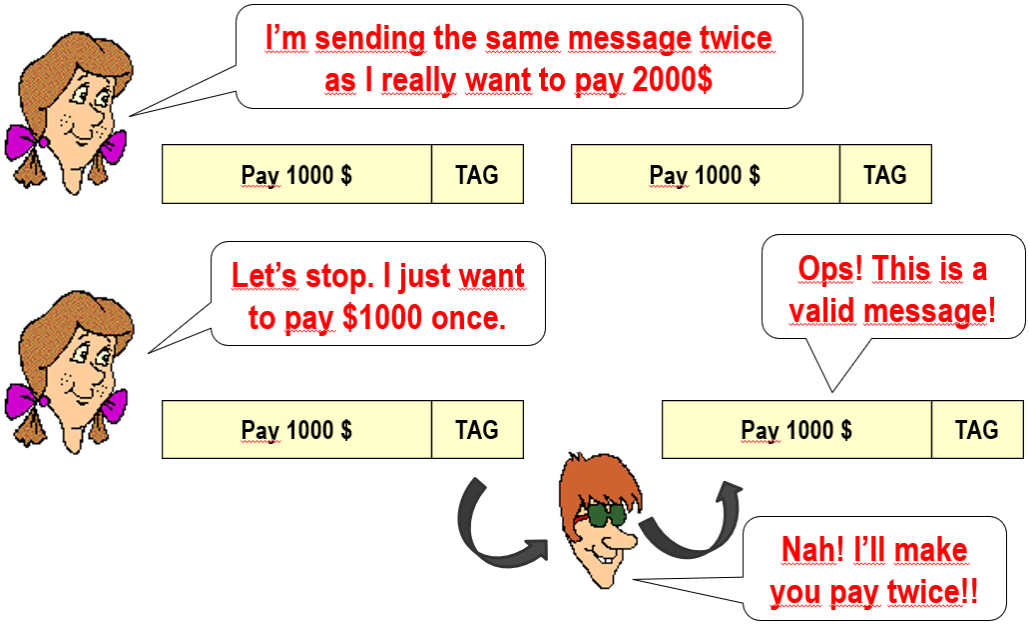
Lo spoofing è una forma di attacco in cui un aggressore invia messaggi fraudolenti fingendo di essere una parte legittima. L'autenticazione del messaggio tramite MAC protegge da questo attacco, in quanto il ricevente verifica l'autenticità del messaggio controllando il codice di autenticazione. Se il messaggio è stato falsificato o proviene da un mittente non autorizzato, il controllo fallirà.



Protezione da Replay Attack

Un attacco di replay si verifica quando un aggressore acquisisce un messaggio autentico e lo ritrasmette, spesso con l'obiettivo di provocare azioni indesiderate (ad esempio, pagamenti duplicati).

I codici di autenticazione da soli non prevengono i replay: un messaggio valido ritrasmesso ha un MAC correttamente calcolato e quindi viene accettato.



Uso dei Nonce per Prevenire i Replay Attack

Per prevenire i replay attack, si includono nei messaggi dei *nonce* (numeri usati una sola volta). I nonce sono informazioni fresche, diverse in ogni messaggio, che devono essere incorporate anche nel calcolo del MAC. Questo garantisce che ogni messaggio sia unico e che un messaggio ritrasmesso venga rifiutato perché il nonce non è più valido o è già stato usato.

Tipi comuni di nonce sono:

- **Numeri di sequenza:** incrementali, facili da gestire ma richiedono sincronizzazione tra mittente e ricevitore.
- **Numeri casuali:** creano complessità nel controllo da parte del ricevitore per verificare che siano effettivamente freschi e non riusati.
- **Timestamp:** richiedono orologi sincronizzati e affidabili, ma permettono di scartare messaggi "vecchi".

La scelta del tipo di nonce dipende dal protocollo e dall'ambiente operativo.

Ricapitolando

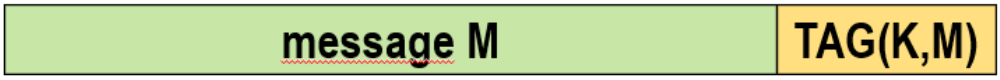
L'integrità e l'autenticità dei messaggi in presenza di un potenziale aggressore possono essere garantite tramite l'uso di MAC basati su chiavi segrete condivise. Tuttavia, per difendersi efficacemente da replay attack, è necessario introdurre nonce o altre tecniche che assicurino l'unicità dei messaggi.

Queste misure, combinate, rafforzano la sicurezza della comunicazione, impedendo:

- La modifica o falsificazione dei messaggi.
- L'inserimento di messaggi illegittimi da parte di un attaccante.
- La riutilizzazione di messaggi validi per scopi fraudolenti.

Codici di Autenticazione del Messaggio (MAC)

Un *Message Authentication Code* è un piccolo codice aggiunto al messaggio, calcolato tramite una funzione crittografica con chiave segreta. Quando il destinatario riceve il messaggio e il tag MAC, ricalcola il codice col segreto condiviso: se coincide, il messaggio è autentico. Nessuno che non conosce la chiave può forzare una variazione o produrre un nuovo MAC valido su un messaggio alterato.



Limiti e Minacce ai MAC

Un MAC robusto non permette a un attaccante di:

- Indovinare o ricavare la chiave dal (messaggio, tag) osservati.
- Cambiare tag o messaggio senza rendere il controllo fallace.
- Creare un MAC valido per un nuovo messaggio, anche conoscendo molti esempi precedenti.

Tuttavia, il MAC **non** protegge da attacchi di replay: un messaggio autentico inviato due volte sarebbe accettato due volte, quindi serve includere informazioni uniche (numero sequenziale, timestamp o nonce) per ogni messaggio.

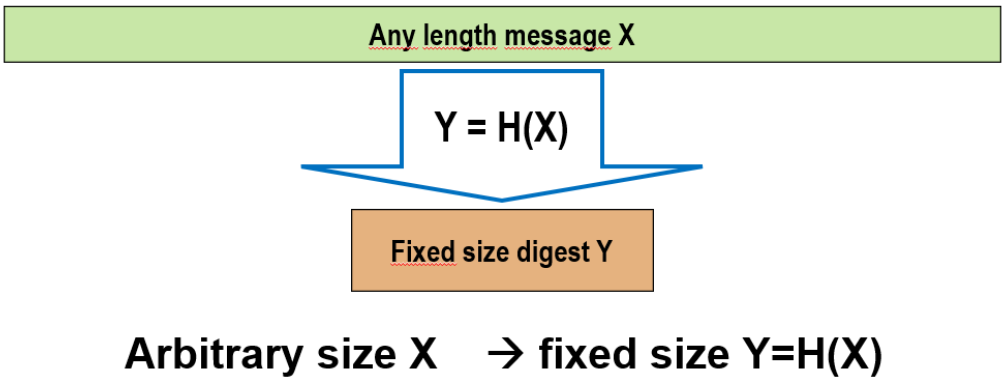
Robustezza rispetto agli attacchi

La sicurezza di una funzione MAC si valuta simulando la resistenza a tutte le tecniche di attacco: osservazione di molti messaggi/tag (known message), scelta dei messaggi da autenticare (chosen message), fino ad attacchi adattivi. L'attaccante non deve mai riuscire a generare un tag valido per un messaggio non visto prima.

La dimensione del tag è critica: se è troppo breve (ad esempio 1 byte), risulta vulnerabile a tentativi casuali (brute force). Un tag più ampio — almeno 128 bit — mitiga questi rischi.

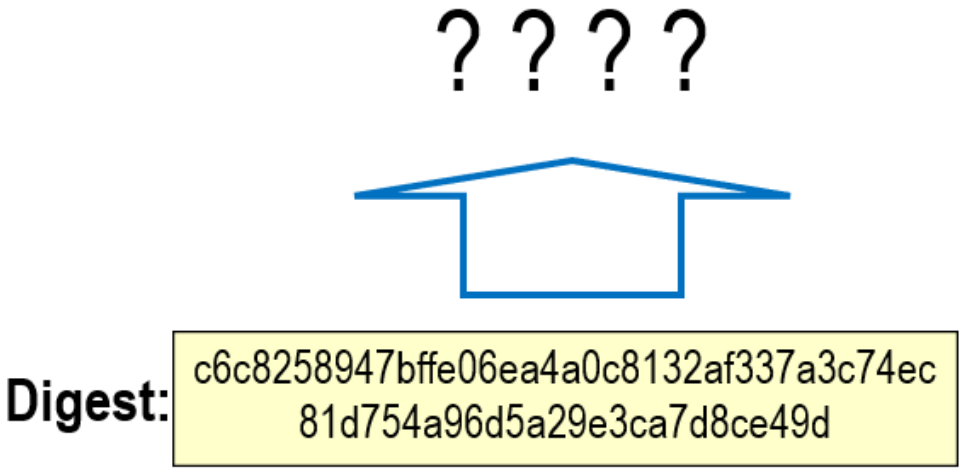
Hash crittografici: Proprietà fondamentali

Un *hash crittografico* prende un messaggio di qualunque lunghezza e lo "riassume" in una stringa di lunghezza fissa (ad es. 256 o 512 bit).

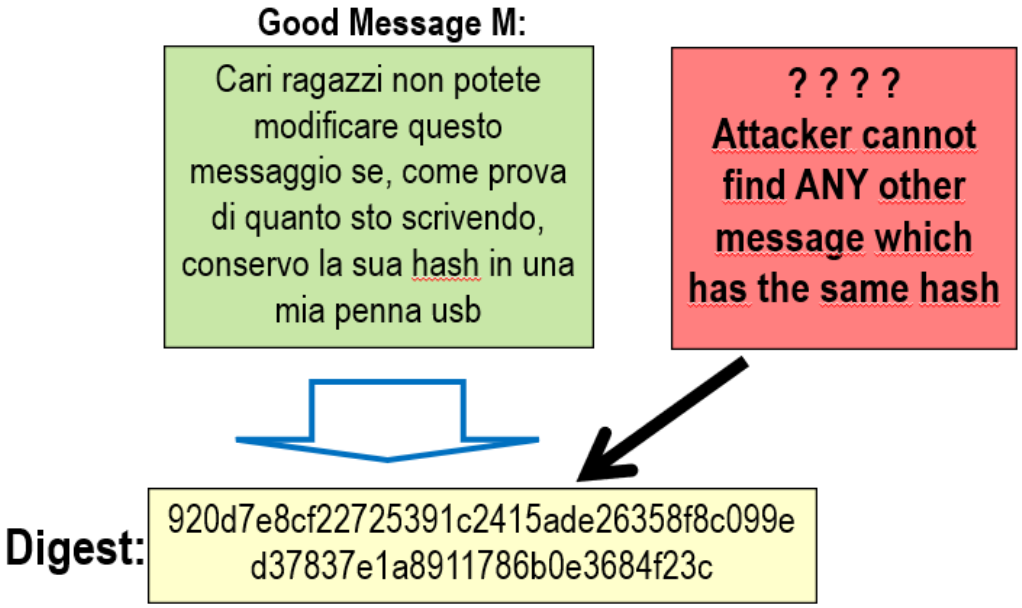


Proprietà chiave:

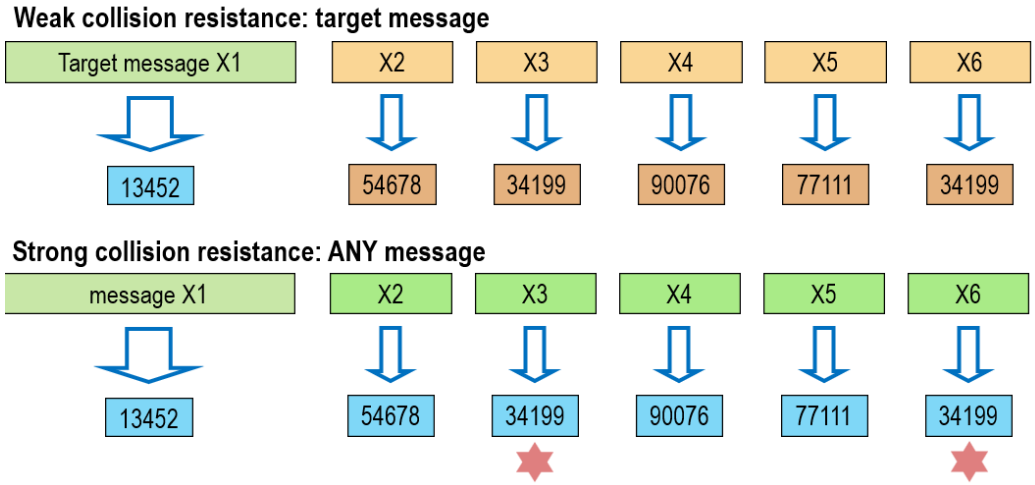
- **Preimage resistance (unicità inversa)**: Dato un hash, è computazionalmente difficile recuperare un qualsiasi messaggio che produca quell'hash.



- **Second preimage resistance**: Dato un messaggio e il suo hash, è difficile trovarne un altro con lo stesso hash.



- **Collision resistance**: È difficile trovare due messaggi diversi che abbiano lo stesso hash.



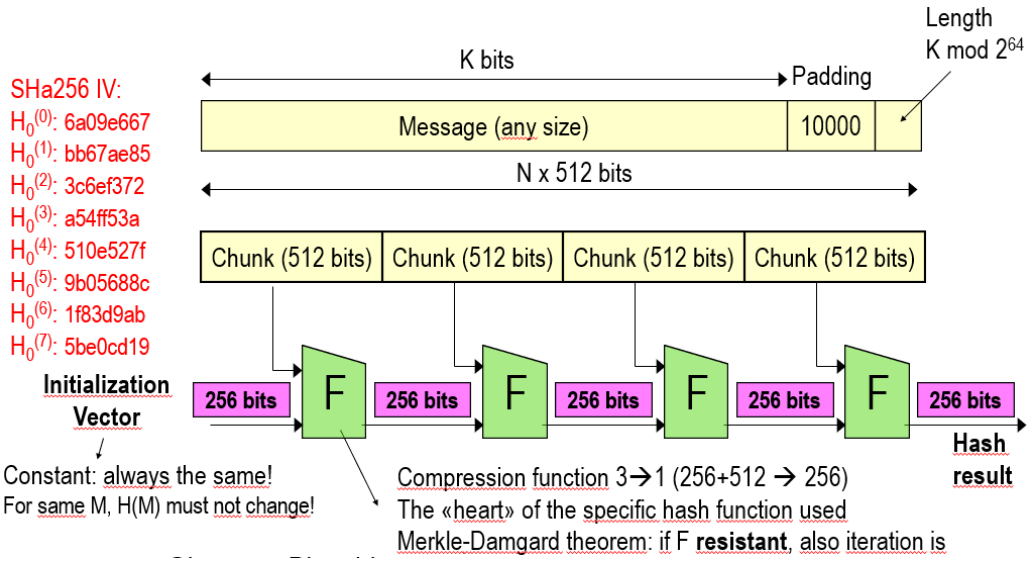
Tali proprietà sono fondamentali per non compromettere la sicurezza di MAC e firme digitali.

Hash e Paradosso del Compleanno

Il famoso *paradosso del compleanno* spiega che la probabilità di collisioni su digests hash non cresce linearmente ma molto più rapidamente. Ad esempio, con hash a 128 bit, bastano circa 2^{64} messaggi diversi per aspettarsi la prima collisione: è fondamentale quindi scegliere digest sufficientemente lunghi.

Hash e MAC: Implementazione ingenua e trappole

Una tentazione naturale è includere semplicemente la chiave all'inizio o alla fine del messaggio nella funzione hash. Tuttavia, la struttura iterativa delle hash comuni (Merkle-Damgård) permette attacchi di estensione: un attaccante può estendere un messaggio e calcolare un nuovo hash valido, anche senza conoscere la chiave, sfruttando l'hash precedente.



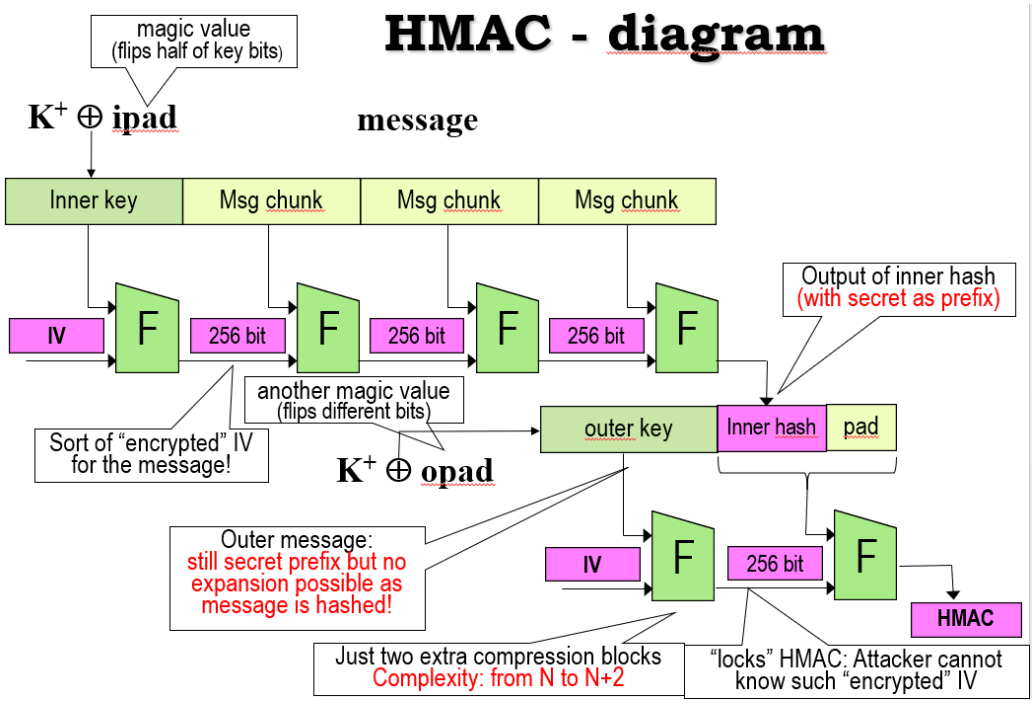
HMAC: La soluzione robusta

Per ovviare a questi problemi, nasce HMAC (Hash-based Message Authentication Code), standardizzato e formalmente dimostrato sicuro se la hash sottostante è pseudocasuale. HMAC combina la chiave e il messaggio usando strutture interne diverse (opad, ipad), evitando gli attacchi di estensione e garantendo robustezza anche con hash non collision-resistant (come MD5 o SHA-1).

Schema generale HMAC

$$HMAC(K, M) = H((K \oplus opad)||H((K \oplus ipad)||M))$$

Dove H rappresenta la funzione hash selezionata, K la chiave (allungata a blocco), $\oplus$ l'operazione XOR con costanti opad/ipad diverse, e M il messaggio.



Conclusioni e Lezioni Chiave

- Non basta la confidenzialità: l'integrità/autenticità sono elementi distinti e devono essere protetti appositamente — idealmente tramite HMAC.
- Le hash crittografiche sono strumenti potenti, ma la loro struttura interna può prestarsi ad attacchi se non usate correttamente.
- Per ogni bisogno crittografico occorre una funzione progettata ad hoc; evitare soluzioni "fatte in casa": usare sempre primitive standardizzate come HMAC.