Relazione di Network Security

Nome: Marco

Cognome: De Stefano

N° Matricola: 7025991

Data: Ottobre 2024

---

1 Introduzione

In questa relazione, esamineremo la crittografia tradizionale e l'approccio più recente denominato HPKE (Hybrid Public Key Encryption). Analizzeremo la struttura e la composizione di entrambi i metodi, mettendo in evidenza le debolezze dell'approccio tradizionale, e spiegando le ragioni per cui l'HPKE rappresenta un miglioramento.

2 Crittografia Tradizionale

La crittografia tradizionale si basa sull'integrazione di tre componenti principali:

- **Diffie-Hellman**
- **Crittografia asimmetrica**
- **Crittografia simmetrica**

Questi tre elementi vengono combinati per garantire uno scambio sicuro delle chiavi e la protezione dei dati. Tuttavia, l'uso del solo protocollo Diffie-Hellman presenta alcune vulnerabilità che possono compromettere la sicurezza della comunicazione. Di seguito, esaminiamo i principali rischi associati a questo approccio:

- **Attacco man in the middle (MITM):** Si verifica quando un aggressore si interpone tra due partecipanti, inducendoli a credere di comunicare direttamente tra loro. In realtà, l'aggressore riesce a manipolare la comunicazione. Nel caso di Diffie-Hellman, il processo si svolge come segue:
  1. **Alice** e **Bob** cercano di stabilire una chiave segreta tramite Diffie-Hellman.
  2. L'aggressore, **Eve**, si inserisce tra Alice e Bob, intercettando i loro messaggi.
  3. Quando Alice invia la propria chiave pubblica a Bob, Eve la intercetta e, anziché inoltrarla, invia a Bob una chiave pubblica creata da lei stessa.
  4. Eve invia ad Alice una propria chiave pubblica, facendole credere che provenga da Bob. Di conseguenza, Alice e Bob credono erroneamente di condividere la stessa chiave.
  5. Alice e Eve stabiliscono una chiave segreta, così come Bob e Eve. Tuttavia, Alice e Bob non condividono la stessa chiave.
  6. Quando Alice invia un messaggio a Bob, Eve può decrittarlo utilizzando la chiave condivisa con Alice, leggerlo e modificarlo, quindi ri-crittografarlo con la chiave condivisa con Bob prima di inoltrarlo.
- **Attacco replay:** Simile all'attacco MITM, ma qui l'aggressore intercetta un messaggio valido tra due parti per poi riprodurlo in un momento successivo, cercando di ingannare le parti coinvolte.

Per mitigare queste vulnerabilità e garantire l'autenticità delle parti che scambiano le chiavi, si combinano i tre blocchi sopra menzionati. In questo modo, si aggiunge un livello di protezione che rende più sicuro lo scambio delle chiavi e la successiva comunicazione cifrata. Nelle sezioni successive, vedremo in dettaglio come viene implementato il protocollo Diffie-Hellman e come si integra con gli altri blocchi.

2.1 Scambio Diffie-Hellman

Il protocollo Diffie-Hellman permette a due parti di calcolare una chiave simmetrica condivisa, ma non è un sistema di cifratura in sé. Infatti, non cifra messaggi, ma serve esclusivamente a generare una chiave segreta condivisa tra due partecipanti, anche se stanno comunicando su un canale pubblico e non sicuro. La sicurezza di questo protocollo si basa sulla difficoltà computazionale di calcolare il logaritmo discreto.

Vediamo ora come due partecipanti, Alice e Bob, possono calcolare una chiave segreta comune in questo modo:

1. Alice sceglie un numero primo grande N e un generatore g, che sarà pubblico e lo saprà anche Bob.
2. Alice sceglie randomicamente $a < N$ e calcola $A = g^a \cdot (modN)$ e invia A a Bob. Quindi N,g e A sono pubblici solo a è nota ad Alice.
3. Bob sceglie randomicamente $b < N$ e calcola $B = g^b \cdot (modN)$ e invia B a Alice. Quindi N,g e B sono pubblici, a è nota ad Alice e b è nota solo a Bob.
4. Quindi Alice calcola $K = B^a \cdot (modN)$ e Bob calcola $K = A^b \cdot (modN)$

Avranno tutti e due quindi la stessa chiave segreta K poichè:

$$K = B^a \cdot (modN) = A^b \cdot (modN) = g^{a \cdot b} \cdot (modN)$$

La chiave K risultante è simmetrica, ossia è la stessa per entrambi i partecipanti. Tuttavia, è importante notare che questo protocollo necessita di un meccanismo di autenticazione, come una firma digitale, per evitare attacchi come l'attacco man in the middle descritto in precedenza.

2.2 Crittografia Asimmetrica e combinazione con Diffie Hellman

Vediamo, quindi, come viene autentificato il messaggio. Sappiamo che ogni partecipante ha una chiave privata (a e b) con il quale si creano le chiavi pubbliche A e B che servono a ottenere la chiave simmetrica, poiché abbiamo detto che in un attacco man in the middle ci può essere qualcuno che si finge Bob e instaurare delle chiavi simmetriche affiché l'attaccante possa tradurre i messaggi di Alice, si deve autentificare e mandare insieme alla chiave pubblica una firma digitale. La firma digitale viene generata da Alice in questo modo:

1. **Calcolo dell'Hash:** Alice calcola un valore hash della chiave pubblica A.
2. **Creazione della firma:** Alice usa la sua **chiave privata** per firmare l'hash calcolato:

$$F_A = E_{skA}(hash(A))$$

3. **Invio della Chiave Pubblica e della Firma:** Alice invia a Bob la chiave pubblica A insieme alla firma digitale F_A.

Quando Bob riceve la chiave pubblica e la firma digitale da Alice, esegue le seguenti operazioni per verificare l'autenticità:

- Calcola l'hash della chiave pubblica A.
- Utilizza la chiave pubblica di Alice per decifrare la firma digitale e ottenere l'hash firmato *h'*.
- Confronta i due hash, se $h \neq h'$,significa che la chiave pubblica è stata alterata durante la trasmissione, e quindi Bob rifiuta la creazione della chiave condivisa.

2.3 Crittografia Simmetrica

Una volta che la chiave simmetrica è stata scambiata in modo sicuro grazie al protocollo Diffie-Hellman e all'autenticazione tramite firme digitali, questa può essere utilizzata per cifrare i dati effettivi della comunicazione. La crittografia simmetrica è particolarmente efficiente in termini di velocità, rendendola ideale per cifrare grandi quantità di dati in maniera sicura.

3 Metodo HPKE (Hyper Public Key Encryption)

HPKE (Hybrid Public Key Encryption) è un protocollo crittografico che combina la crittografia asimmetrica e simmetrica per garantire la sicurezza nelle comunicazioni.
L'idea principale di HPKE è quella di generare e condividere una chiave simmetrica in modo sicuro attraverso la crittografia asimmetrica, per poi utilizzarla per cifrare i dati in modo più efficiente. HPKE è formato da delle fasi. Inizia con Key Encapsulation Mechanism (KEM), questa fase è formata da due algoritmi:

- **Encapsulation (incapsulamento)**: questo algoritmo prende la chiave simmetrica e la cripta con la chiave pubblica del destinatario affinchè solo quest'ultimo riesca a decriptare la chiave simmetrica tramite la sua chiave privata.
- **Decapsulation (decapsulamento)**: l'algoritmo di decapsulazione prende la chiave incapsulata e la chiave privata associata alla chiave pubblica dell'incapsulamento e da li viene decriptato il messaggio incapsulato.

Il segreto condiviso è una base comune di partenza per entrambi, ma non è ancora la chiave finale usata direttamente per cifrare i messaggi. Infatti, da questo segreto comune HPKE crea una serie di chiavi derivate che sono usate per criptare e autentificare i messaggi plaintext tra mittente e destinatario. Si creano queste chiavi derivate attraverso delle funzioni KDF che vengono stabilite all'inizio della connessione tra i due terminali. Infatti entrambi devono avere stessa funzione KDF. Queste funzioni prendono come input:

- **Segreto Condiviso:** è la chiave simmetrica che hanno entrambi i partecipanti.
- **Contesto:** è un valore che rende le chiavi derivate uniche. Questo valore può includere informazioni come l'ID della sessione o altri dettagli specifici della comunicazione.

Mentre restituiscono come output:

- **Chiave di Cifratura:** utilizzata per cifrare i messaggi tra mittente e destinatario, rendendo i dati sicuri e leggibili solo a chi possiede la chiave corretta per decifrarli.
- **Chiave di Autenticazione:** assicura che i messaggi non siano stati alterati durante la trasmissione e che provengano dal mittente legittimo.

Inoltre le KDF hanno le seguenti proprietà:

- **Deterministiche:** stesso input = stesso output
- **Non Invertibilità:** non è possibile risalire facilmente all'input iniziale avendo l'output.

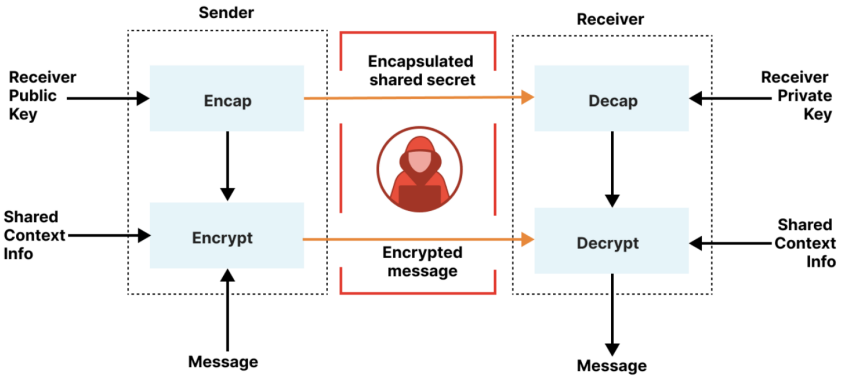

Figure 1: Funzionamento HPKE

4 HPKE e Crittografia Tradizionale

4.1 Differenze

Dopo aver illustrato le due tipologie di crittografia in questione, si vogliono analizzare le loro differenze. Iniziamo parlando dell'approccio alla gestione delle chiavi, dove in HPKE la chiave simmetrica viene creata dinamicamente dal mittente tramite un processo di **incapsulamento** e **decapsulamento**, mentre nella crittografia tradizionale si utilizza Diffie-Hellman, nel quale la chiave simmetrica viene calcolata usando **entrambe le chiavi pubbliche** degli attori.
Infatti, nella fase di setup dei due metodi si ha che:

- Diffie-Hellman Tradizionale:

  1. Entrambe le parti (ad esempio, Alice e Bob) devono generare una coppia di chiavi (privata e pubblica).
  2. Le due parti si scambiano attivamente le chiavi pubbliche.
  3. Ognuno usa la chiave pubblica dell'altro e la propria chiave privata per calcolare lo stesso segreto condiviso $g^{ab} \mod p$.

  Questo processo richiede una comunicazione sincrona e bidirezionale.

- HPKE:

  1. Solo il mittente genera una coppia di chiavi temporanee (**ephemeral key pair**) per ogni messaggio.
  2. Il mittente utilizza la sua chiave privata temporanea e la chiave pubblica del destinatario per calcolare un valore segreto usando una funzione di Diffie-Hellman.
  3. Invia la sua chiave pubblica temporanea al destinatario come parte del messaggio.
  4. Il destinatario utilizza questa chiave pubblica temporanea e la sua chiave privata per calcolare lo stesso segreto.

  Questo permette di stabilire un segreto comune senza uno scambio bidirezionale attivo delle chiavi pubbliche, come nel Diffie-Hellman tradizionale.

Inoltre, una **differenza significativa** tra il metodo tradizionale e HPKE consiste nel fatto che, nel primo, **si genera una chiave simmetrica in modo casuale e la si crittografa con la chiave pubblica del destinatario**, mentre nel secondo metodo **si deriva la chiave simmetrica da un segreto condiviso stabilito**.

4.2 Vantaggi dell'HPKE rispetto alla Crittografia Tradizionale

In questa ultima sezione, parliamo quindi dei vantaggi che ha l'HPKE rispetto alla crittografia tradizionale.

- **Uso limitato della Crittografia Asimmetrica:** La crittografia asimmetrica (come RSA o Diffie-Hellman) richiede numerose risorse computazionali rispetto a quella simmetrica. Nell'HPKE viene utilizzata solo all'inizio per creare lo *shared secret*.

- **Riduzione del numero di messaggi nel protocollo:** Nell'HPKE, la crittografia asimmetrica viene utilizzata solo all'inizio per creare lo *shared secret*. Nella crittografia tradizionale, si hanno messaggi multipli per lo scambio delle chiavi e l'autenticazione.

- **Resistenza alle intercettazioni delle chiavi:** HPKE è progettato per resistere a attacchi basati sulla compromissione delle chiavi. L'uso del KDF, che è una funzione di derivazione delle chiavi a partire dal segreto condiviso (incapsulato), previene attacchi Man-In-The-Middle (MITM). Nel caso di quest'attacco, il segreto condiviso non viene scoperto grazie all'incapsulamento (criptare con chiave pubblica di Bob). Dopo questo, i messaggi non possono essere più capiti dall'attaccante poichè le chiavi sono decise da KDF.

- **Implementazione più semplice rispetto a quella tradizionale:** HPKE standardizza i processi di "Encapsulation" e "Decapsulation", automatizzando la gestione della generazione e condivisione delle chiavi simmetriche. Questo elimina la necessità di scrivere codice personalizzato per gestire lo scambio delle chiavi in modo sicuro. La funzione di derivazione delle chiavi simmetriche (KDF) è ben documentata e ampiamente utilizzata, quindi gli sviluppatori non devono reinventare nuovi schemi per la derivazione delle chiavi.