

UNIVERSITÀ POLITECNICA DELLE MARCHE

Dipartimento di Ingegneria dell'Informazione

TESI DI LAUREA MAGISTRALE

Thesis Title



Author:
Davide DE ZUANE

Supervisor:
Dr. Paolo SANTINI

September 30, 2024

?Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.?

Dave Barry

UNIVERSITÀ POLITECNICA DELLE MARCHE

Abstract

Faculty Name
Dipartimento di Ingegneria dell'Informazione

Doctor of Computer Sciences

Thesis Title

by Davide DE ZUANE

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
1.1 Post-Quantum	1
1.2 Problematiche	1
2 Fondamenti di Comunicazioni Sicure	3
2.1 Teoria	3
2.1.1 Hash Function	3
2.1.2 Schemi crittografici	3
Simmetrici	4
Asimmetrici	4
2.1.3 Confronto	4
2.1.4 Sicurezza	5
2.2 Applicazioni	5
2.2.1 IPsec	6
2.2.2 IKE_SA_INIT	6
2.2.3 IKE_AUTH	7
2.2.4 CHILD_SA	7
2.3 Problemi	7
2.3.1 IKE_INTERMEDIATE	8
2.4 Post-Quantum	9
A IKEv2 Notation	11
A.0.1 Authentication	11
A.1 Key Derivation	11
A.1.1 IKE SA	11
A.1.2 IPsec SA	12
A.2 Security Association Payload	12

List of Figures

2.1	4
2.2	Fasi di Negoziazione del Protocollo IKEv2	6
2.3	Drop Pacchetti	8
2.4	Scambio nuovo	8

List of Tables

2.1	Security Levels definiti dal NIST	5
2.2	Tabella dei parametri e delle descrizioni	7
2.3	Tabella dei parametri e delle descrizioni	7
A.1	Chiavi e loro utilizzo	11

List of Abbreviations

DH	D iffie H ellman
KE	K ey E xchange
PQ	P ost Q uantum
IKE	I nternet K ey E xchange
KEM	K ey E ncapsulation M echanism
PRF	P seudo R andom F unction
MTU	M aximum T ransmission U nit
ISP	I nternet S ervice P rovider

List of Symbols

$ $	concatenazione	
a	distance	m
P	power	W (J s ⁻¹)
ω	angular frequency	rad

For/Dedicated to/To my...

Chapter 1

Introduction

Introduzione al problema e al contesto in cui ci troviamo e le criticità presenti

Contributo apportato

Fare un'introduzione al problema e dire quale è stato il contributo.

Suddividere il contributo in:

- La parte di benchmarking, in cui andiamo a definire tutte quelle che sono le problematiche relative al post quantum
- la parte di implementazione

1.1 Post-Quantum

Il postquantum può rompere gli schemi di crittografia classici tuttavia i sono nati quelli nuovi.

1.2 Problematiche

Tuttavia il quantum computing porta ad aumentare la dimensione della chiave, in questo modo si perde di efficienza. Inoltre ha particolare effetto sui sistemi a chiave pubblica, mentre su quelli basati su chiave segreta o funzioni di hash non sono molto vulnerabili a questo. Le sfide della post-quantum cryptography sono le seguenti:

- Occorre migliorare l'efficienza
- Occorre migliorarne l'usabilità

Ovvero occorre preparare il mondo per una transizione alla crittografia post-quantum

Dire quelle che sono le problematiche del post quantum Esempi di problematiche:

- non esiste il diffie hellman post quantum
- i messaggi diventano molto più lunghi (problemi per frammentazione, certificati molto più grandi, chiavi pubbliche più grandi)

Queste hanno conseguenze importanti sui protocolli che ne fanno uso a causa dell'aumento di dimensioni delle chiavi.

Gli schemi sono divisi in

- l1:aes128 - l3:aes192 - l5:aes256

Chapter 2

Fondamenti di Comunicazioni Sicure

In questo capitolo andiamo a trattare quelle che sono le problematiche riguardanti le comunicazioni sicure. E' possibile realizzare comunicazioni sicure grazie a quelli che sono i crittosistemi, andiamo a vedere come questi vengono utilizzati nelle reti di computer per garantire la sicurezza delle comunicazioni.

2.1 Teoria

Introduciamo quelli che sono gli strumenti fondamentali per la crittografia.

2.1.1 Hash Function

2.1.2 Schemi crittografici

Uno schema di cifratura è un insieme di algoritmi e funzioni che definisce come trasformare un messaggio in chiaro (plaintext) in un messaggio cifrato (ciphertext) e viceversa, al fine di garantire la confidenzialità e la sicurezza delle comunicazioni. Formalmente possiamo rappresentarlo come una quintupla:

$$(\mathcal{P}, \mathcal{C}, \mathcal{K}, E, D)$$

- \mathcal{P} : Insieme dei messaggi in chiaro (plaintext).
- \mathcal{C} : Insieme dei messaggi cifrati (ciphertext).
- \mathcal{K} : Insieme delle chiavi utilizzate per la cifratura e decifratura, *key space*.
- $E : \mathcal{K} \times \mathcal{P} \rightarrow \mathcal{C}$: Funzione di cifratura.
- $D : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{P}$: Funzione di decifratura.

Deve esistere una relazione inversa tra le operazioni di cifratura e decifratura:

$$D(k, E(k, m)) = m \quad \forall m \in \mathcal{P}, k \in \mathcal{K} \quad (2.1)$$

Questa definizione di schema di cifratura è generale, tuttavia in base alla definizione e all'uso delle chiavi, è possibile distinguere tra schemi di cifratura simmetrica e asimmetrica.

Simmetrici

In uno schema simmetrico si utilizza la stessa chiave sia per le operazioni di cifratura che di decifratura. Significa che le due parti coinvolte nella comunicazione, ovvero che si vogliono scambiare un messaggio devono possedere la stessa chiave segreta (PSK).

Alcuni esempi di crittosistemi simmetrici sono DES, AES,...

Asimmetrici

In uno schema di cifratura asimmetrica, vengono utilizzate due chiavi distinte: una chiave pubblica e una chiave privata.

Lo spazio delle chiavi \mathcal{K} è costituito da una coppia di chiavi $(k_{\text{pub}}, k_{\text{priv}})$, dove:

- k_{pub} è la chiave pubblica (usata per cifrare)
- k_{priv} è la chiave privata (usata per decifrare)

Le due chiavi sono matematicamente legate, ma è computazionalmente difficile ottenere la chiave privata a partire da quella pubblica (questa è la base della sicurezza). Quindi le due funzioni si riscrivono come:

$$E : \mathcal{K}_{\text{pub}} \times \mathcal{P} \rightarrow \mathcal{C} \quad (2.2)$$

$$D : \mathcal{K}_{\text{priv}} \times \mathcal{C} \rightarrow \mathcal{P} \quad (2.3)$$

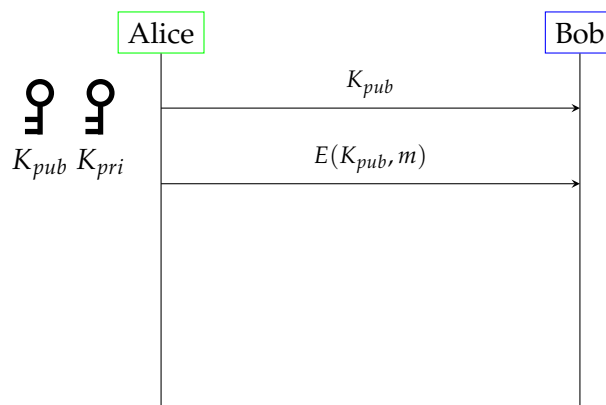


FIGURE 2.1

Successivamente andremo nel dettaglio di quelli che sono i principali utilizzi dei crittosistemi asimmetrici.

Scambio di Chiavi

Cifratura Asimmetrica

Firma Digitale

Alcuni esempi di quelli classici sono: RSA, Diffie-Hellman,...

2.1.3 Confronto

Vediamo le principali differenze tra i due
Distribuzione delle chiavi KDC e PKI.

2.1.4 Sicurezza

Il **security level** è una misura della forza che una primitiva crittografica raggiunge rispetto ad attacchi. Solitamente viene espresso come un numero di “bit di sicurezza”, dove n -bit di sicurezza significa che l’attaccante dovrebbe eseguire $2 * n$ operazioni per romperlo.

Per i cifrari simmetrici, il *livello di sicurezza* è pari alla dimensione del key-space. Mentre la sicurezza degli algoritmi asimmetrici si basa su problemi matematici che sono efficienti da calcolare in una direzione, ma inefficienti da invertire da parte dell’attaccante. Tuttavia, gli attacchi contro gli attuali sistemi a chiave pubblica sono sempre più veloci della ricerca a forza bruta dello spazio delle chiavi.

Il NIST (National Institute of Standards and Technology) ha introdotto livelli di sicurezza per gli algoritmi di cifratura asimmetrica e post-quantistica come parte della sua iniziativa per standardizzare algoritmi che resistano anche ai computer quantistici. I livelli sono definiti in *Tabella 2.1*.

Security Level	Descrizione
Livello 1	Sicurezza equivalente alla cifratura simmetrica con chiavi da 128 bit, come AES-128.
Livello 2	Sicurezza equivalente ad attacchi contro SHA-256, con complessità circa pari a 128 bit. Leggermente più sicuro del livello 1.
Livello 3	Sicurezza equivalente alla cifratura simmetrica con chiavi da 192 bit, come AES-192.
Livello 4	Sicurezza equivalente ad attacchi contro SHA-384. Leggermente più sicuro del Livello 3.
Livello 5	Sicurezza equivalente alla cifratura simmetrica con chiavi da 256 bit, come AES-256.

TABLE 2.1: Security Levels definiti dal NIST

2.2 Applicazioni

Le principali applicazioni crittografiche utilizzate oggi in contesti reali includono SSL/TLS, PGP, IPsec e tante altre. Tali applicazioni si basano su crittosistemi che combinano algoritmi di cifratura, autenticazione e gestione delle chiavi per garantire la riservatezza e l’integrità dei dati scambiati.

Nel contesto delle comunicazioni digitali, la crittografia gioca un ruolo fondamentale per garantire la sicurezza dei dati scambiati tra entità remote. Uno dei principali modelli di riferimento per la trasmissione di dati su Internet è il modello TCP/IP, che suddivide il processo di comunicazione in diversi livelli, ciascuno con funzioni specifiche.

Così come per le comunicazioni di tutti i giorni, anche per quelle che avvengono attraverso Internet vogliamo rispettare quelli che sono i requisiti di sicurezza informatica ovvero garantire:

- Confidenzialità
- Integrità

- Autenticazione

Le comunicazioni seguono il modello ISO/OSI per questo possiamo applicarla a diversi livelli. Come viene applicata la crittografia nelle comunicazioni, quindi i vari metodi con cui la sicurezza è applicata ai vari livelli dello stack network. Si parla di comunicazioni sicure ma a livello pratico queste come vengono realizzate, esiste una suite di protocolli che consente di ottenerle.

2.2.1 IPsec

Facciamo un focus sulla sicurezza a livello IP, quindi introduciamo la suite di protocolli IPsec. Le comunicazioni sicure si costruiscono sopra un concetto fondamentale noto come Security Association (SA). Le comunicazioni sono protette attraverso l'utilizzo di tre tecniche crittografiche:

- Autenticazione
- Integrità
- Cifratura

Le comunicazioni possono rendersi sicure a diversi livelli dello stack network, tuttavia farla a livello IP ha il grande vantaggio di progettare le comunicazioni per tutte le applicazioni, senza andarne a modificare ciascuna singolarmente.

Il protocollo IKEv2, definito nell'RFC 7296, è un protocollo di rete che è responsabile della negoziazione di chiavi crittografiche e dei parametri di sicurezza tra due dispositivi, solitamente chiamati peer.

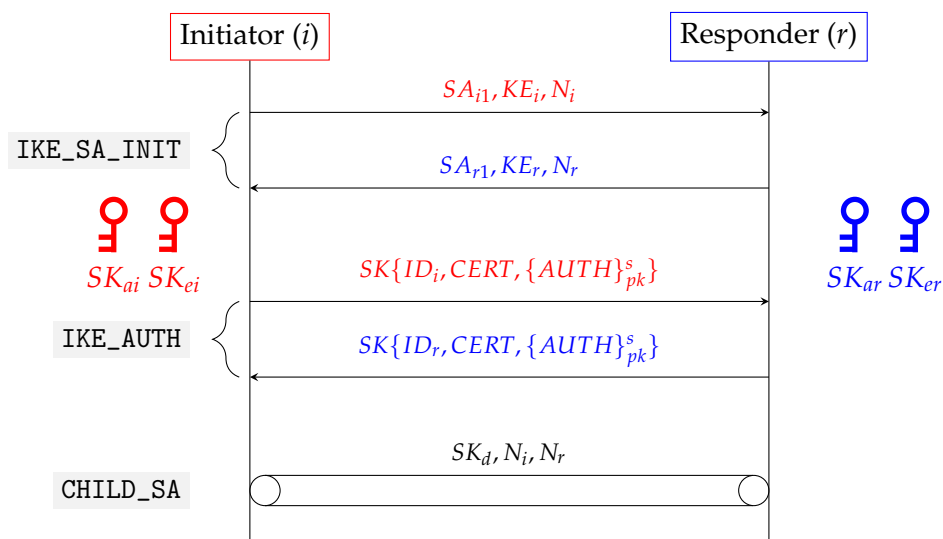


FIGURE 2.2: Fasi di Negoziazione del Protocollo IKEv2

Lo schema generale di funzionamento del protocollo al termine del quale i due hanno stabilito una SA è riportato in Fig. ??.

2.2.2 IKE_SA_INIT

Lo scopo di questa prima fase è quello di creare una **IKE SA**, che consenta di rendere sicure i successivi scambi di dati al fine di realizzare una **IPsec SA**. Dunque funge da

apripista al fine di stabilire quelli che sono i parametri di sicurezza al fine di avere una comunicazione sicura. Per questo motivo in questo scambio i peer si scambiano le seguenti informazioni:

TABLE 2.2: Tabella dei parametri e delle descrizioni

Parametro	Descrizione
SA	Security Association, vengono negoziati i parametri per la SA
KE	Key Exchange, e nel caso classico è l'esponente DH
N	Nonce

Al termine di questo scambio i due peer ottengono il *DH Shared Secret* (indicato con g^{ir}), il quale insieme ai nonce, consentirà di ottenere quelli che sono i parametri di sicurezza della *IKESA* al fine di instaurare un canale sicuro, per approfondimenti in [appendice](#).

2.2.3 IKE_AUTH

Il risultato della fase precedente è un canale sicuro su cui comunicare, in quanto è cifrato e autenticato. Si questo hanno luogo gli scambi per instaurare la IPsec SA. In questa fase i nodi si autenticano mutuamente:

TABLE 2.3: Tabella dei parametri e delle descrizioni

Parametro	Descrizione
AUTH	Payload che deve essere firmato affinché ci sia autenticazione
CERT	Si allega il certificato digitale per la chiave pubblica
CERTQ	Si fa richiesta al peer di fornire il certificato

Tutto il contenuto appena descritto è protetto mediante le chiavi segrete di quella direzione. Ciò è indicato mediante la notazione $SK\{\dots\}$. La modalità di autenticazione può essere: PSK, EAP oppure mediante chiave pubblica.

2.2.4 CHILD_SA

2.3 Problemi

IPsecv2 utilizza come protocollo a livello trasporto UDP per inoltrare i propri messaggi. La maggior parte dei messaggi che i peer si scambiano hanno dimensioni relativamente piccole e quindi che non eccedono l'MTU di un pacchetto IP, tuttavia abbiamo degli scambi che richiedono un trasferimento di dati abbastanza grandi.

Per esempio nel caso di autenticazione tramite pubkey nella fase di **IKE_AUTH** è necessario trasferire il proprio certificato che in base allo schema di firma utilizzato può arrivare anche a diversi Kbyte di dimensione. In questi casi si verifica la frammentazione a livello IP.

Diversi test hanno mostrato che nel caso in cui i peer si trovino in presenza di CGNAT potrebbero non instaurarsi le SA. Questo è dovuto al fatto che i device degli ISP non consentono ai frammenti IP di passare attraverso di loro, ovvero scartano i

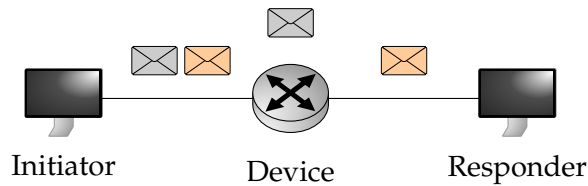


FIGURE 2.3: Drop Pacchetti

pacchetti e di conseguenza bloccano le comunicazioni IKE. Questo è riportato schematicamente in Fig. 2.3. Questo drop dei pacchetti avviene perchè esistono numerosi vettori di attacco che fanno affidamento sulla frammentazione IP, per questo motivo gli ISP operano un filtro su questa tipologia di pacchetti. Anche se in teoria uno dei requisiti del CGNAT definito dagli RFC è proprio consentire la frammentazione.

Per risolvere questa problematica e dunque consentire il passaggio dei messaggi attraverso i dispositivi di rete che non consentono il passaggio degli IP fragment attraverso di loro nell' RFC 7283 viene introdotta la *IKEv2 Message Fragmentation*. In cui la frammentazione dei messaggi è gestita direttamente da parte di chi implementa IKEv2

2.3.1 IKE_INTERMEDIATE

Per evitare che nel trasferimento di grandi dati ciò avvenga viene introdotto uno scambio aggiuntivo. Questo scambio è introdotto per quei casi in cui la dimensione dei dati da trasferire ecceda la dimensione massima che causerebbe la frammentazione IP. Questo scambio va fatto dopo la `IKE_INIT_SA` e prima della `IKE_AUTH` in questo modo è sia autenticato che cifrato tramite le chiavi negoziate dal primo scambio.

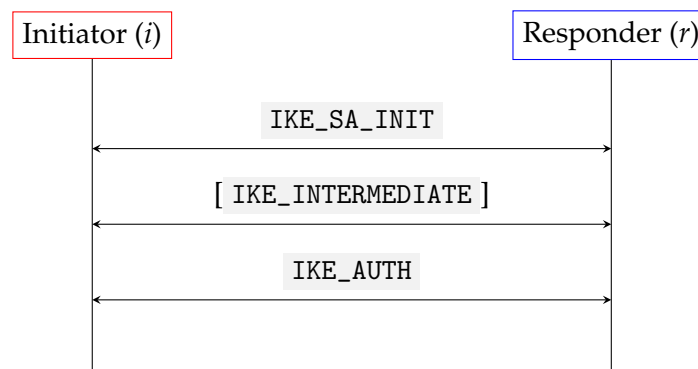


FIGURE 2.4: Scambio nuovo

Questo scambio è posizionato qui in quanto nella `IKE_SA_INIT` per motivi di sicurezza non è possibile applicare la frammentazione. Di solito i messaggi sono piccoli abbastanza da non causare la frammentazione IP, tuttavia questo potrebbe cambiare se si utilizzano scambi di chiave QC-resistant; in quanto hanno chiavi pubbliche larghe e che quindi causerebbero frammentazione IP.

Per questo viene aggiunto questo scambio che viene utilizzato per trasferire grandi quantità di dati.

L'utilizzo principale di questo scambio è quello di trasferire le chiavi pubbliche QC-resistant, tuttavia in generale può essere utilizzato per trasferire qualsiasi tipologia di dato. Quindi il principale utilizzo è quello di fare un **enforcing** delle chiavi

negoziare tramite DH al fine di renderle QC-resistant. Infatti se durante questo scambio si scambiano altre chiavi allora le coppie $\{SK_{a[i/r]}, SK_{e[i/r]}\}$ vengono aggiornate.

Permette di realizzare Multiple Key Exchange. Gli scambi di chiave aggiuntivi vengono aggiunti alla proposal tramite `PQ_KEM_1`.

Lo scambio `IKE_FOLLOWUP_KEY` è introdotto specificatamente per trasferire dati sulla chiavi addizionali da realizzare in una CHILD SA. In questo caso le chiavi aggiuntive vengono utilizzate per aggiornare il KEYMAT.

- flag `IKE_FRAGMENTATION_SUPPORT` : il peer dice di supportare la frammentazione IKEv2, affinché venga utilizzata entrambi i peer devono supportarla.
- flag `INTERMEDIATE_EXCHANGE_SUPPORT` : il peer dice di supportare gli scambi intermedi.

Una volta terminati gli scambi, per proteggere lo scambio `IKE_AUTH` e gli scambi successivi vengono utilizzate le ultime chiavi calcolate. Dato che i dati trasferiti in questi scambi aggiuntivi vanno autenticati si aggiungono all' `AUTH` payload che poi andrà.

Il supporto per lo scambio aggiuntivo viene comunicato aggiungendo all'interno dell' `IKE_SA_INIT` il flag `IKE_INT_SUP` (che sta per Intermediate Exchange Support). Se anche il responder lo supporta lo includerà nel messaggio di risposta dello scambio.

Considerazioni, L'IKE fragmentation viene introdotta a causa del NAT tuttavia nel nostro caso di satelliti non ha senso utilizzarla in quanto non credo che si utilizzi il NAT soprattutto perché introduce ritardi dovuti alla traduzione degli indirizzi.

2.4 Post-Quantum

Un solo KEM con Kyber L1 usando come suite AES_GCM ha valore come certificato dilithium L1.

Nel KEM quanti cifrano?

Cioè l'initiator manda il certificato e poi il responder cifra.

Appendix A

IKEv2 Notation

Riportare i vari approfondimenti riguardanti IKE
per esempio come vengono generate le varie chiavi e il significato delle informazioni
presenti tra i messaggi

A.0.1 Authentication

L'autenticazione dei peer avviene effettuando il sign (o calcolando il MAC) di un payload che dipende dagli scambi precedenti. In particolare questo payload è composto da un oggetto che viene autenticato in base alla modalità di autenticazione scelta:

- Nel caso di *PubKey* questo viene firmato con la chiave privata del peer e ne viene allegato il certificato della chiave pubblica
- Nel caso di *PSK* l'AUTH payload viene generato a partire dalla chiave condivisa a cui viene aggiunta la unpredictability tramite del padding e una prf

A.1 Key Derivation

A.1.1 IKE SA

Le chiavi in una IKE SA vengono derivate a partire dagli attributi dei diretti scambi. In particolare al termine del primo scambio viene calcolato il:

$$SKEYSEED = PRF(N_i | N_r, g^{ir})$$

A partire da questo seed vengono generati i parametri di sicurezza da utilizzare per la IKE SA, questi sono derivati nel seguente modo:

$$\{SK_d | SK_{ai} | SK_{ar} | SK_{ei} | SK_{er} | SK_{pi} | SK_{pr}\} = PRF + (SKEYSEED, N_i | N_r, SPI_i, SPI_r)$$

Chiave	Descrizione
SK_d	Utilizzata per generare il keymaterial per le CHILD_SA
SK_a	Chiavi per autenticare gli scambi successivi, una per direzione
SK_e	Chiavi per cifrare gli scambi successivi, una per direzione
SK_p	Chiavi utilizzate per generare l'AUTH Payload, una per direzione

TABLE A.1: Chiavi e loro utilizzo

A.1.2 IPsec SA

Nel caso di una SA questa può essere generata automaticamente dopo l'auth oppure attraverso l'apposito scambio di questo tipo il keymaterial a partire dal quale vengono derivati i parametri di sicurezza è ottenuto nel seguente modo:

$$KEYMAT = prf + (SK_d, N_i | N_r)$$

Nel caso in cui invece si utilizza lo scambio apposito il key material è ottenuto nel seguente modo

A.2 Security Association Payload

Il Security Association Payload denotato con SA è utilizzato per negoziare gli attributi di una Security Association. Dunque può contenere molteplici proposte, le quali devono essere ordinate per preferenza, ogni proposal contiene i seguenti algoritmi crittografici:

- Encryption Algorithm (ENCR)
- Pseudorandom Function (PRF)
- Integrity Algorithm (INTEG)
- Diffie-Hellman Group (KE)
- PQ KEM