UNIVERSITÀ POLITECNICA DELLE MARCHE

Dipartimento di Ingegneria dell'Informazione

TESI DI LAUREA MAGISTRALE

Thesis Title

Author:
Davide DE ZUANE

Supervisor:
Dr. Paolo SANTINI

April 8, 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

List of Figures

List of Tables

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

$\mid$	concatenazione	
a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

For/Dedicated to/To my…

Chapter 1

State of The Art

1.1 Problem

Quando è stato annunciato che il quantum-computer avrebbe rotto algoritmi come RSA, ECDSA e DES Internet aveva ormai decretato la fine della crittografia. Tuttavia c'è una sostanziale differenza tra rompere qualche crittosistema a rompere la crittografia, infatti esistono importanti classi di sistemi crittografici oltre a quelli precedentemente citati. Tra questi abbiamo:

- Hash-based cryptography
- Code-based cryptography
- Lattice-based cryptography

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

1.1.1 Efficency

Quanto deve essere la chiave per garantire lo stesso livello di sicurezza. Questa è necesssaria per i server in Internet che devono gestire centinaia di clienti ogni secondo. Google con il suo motore di ricerca già esegue il redirect su http per non rallnetare

1.1.2 Confidence

La community deve avere confidenza nei sistemi e affinchè ciò sia possbibile è richiesto tempo affincheè i crittoanalisti cerchino degli attacchi per questi sistemi. Per esempio non esiste alcun problema studiato bene come il Diffie-Hellman e la gente ha totale confidenza nei suoi confronti (da quando è stato introdotto non è mai stato rotto).

1.1.3 Usability

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 piu grandi)

1.2 Solution

Dare una panoramica delle attuali soluzioni tappabuchi che sono state realizzate. Andare per stack network, per esempio TLS PQ, IKEv2 PQ

Dire quelle che sono le possibili soluzioni, in particolare perchè si è scelto di utilizzare IKEv2 Post quantum

Parlare della libreria openquantumsafe

1.2.1 TLS-PQ

Dire che non è la cosa giusta da utilizzare in quanto non sono presenti RFC che abbiano ben definito come realizzarla. Inoltre sono presenti online numerose implementazioni custom di quest'ultima che tuttavia fanno uso sia di POst quantum crittograpy che Quantum Key Distribution

1.2.2 IKEv2-PQ

Parlare nel dettaglio della sua architettura e altro nel secondo capitolo

Spiegare che ci concentreremo su questa per quanto riguarda le comunicazioni sicure poichè è presente un RFC, c'èuna comunity ampia che si occupa di questo problema

Test strongswan tempistiche, dimnesioni post quantum Come peggiorano le prestazioni nel caso post quantum

(Confronto tra i due casi)

Interessante vedere anche cosa cambia se utilizziamo scheduler real time

- Spinqs+ no perchè è una merda - Guardare tutti gli algoritmi tranne quelli del 4° Round del NIST, l'unico da provare è bike (HQC no)

Gli schemi sono divisi in

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

Mentre per quanto riguarda i certificati vedre sia falcon che dilithium (Falcon in teoria dovrebbe essere più lento a causa di operazioni float) Sia in termini di dimensioni che di tempi

Provare solo quello che è L1 (livello di sicurezza

1.2.3 Liboqs

Dire che in generale per la transizione al post quantum fanno tutti utilizzo di questa libreria

Chapter 2

IKEv2

2.1 Pre-Quantum

Andiamo a vedere quello che era il funzionamento di IKEv2 nel caso di crittografia classica. IkEv2 funziona sul principio di *exchange* ovvero di richiesta e risposta, in Fig. **??** sono riportate le varie interazioni tra initiator e responder volte ad instaurare una Security Association.

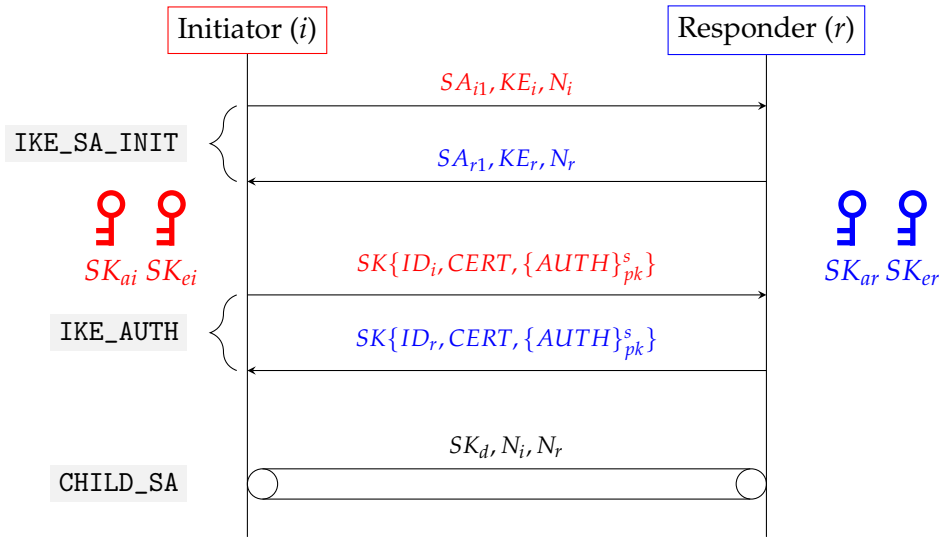

FIGURE 2.1: Rappresentazioni delle fasi di IKEv2

2.1.1 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 2.1: 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 insisme ai nonce, consentirà di ottenere quelli che sono i parametri di sicurezza della $IKESA$ al fine di instauare un canale sicuro, per approfondimenti in appendice.

2.1.2 IKE_AUTH

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

TABLE 2.2: Tabella dei parametri e delle descrizioni

Parametro	Descrizione
AUTH	Payload che deve essere firmato affinchè 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 $SK\{...\}$ La modalità di autenticazione può essere: PSK, EAP oppure mediante chiave pubblica.

2.1.3 CHILD_SA

2.2 Problemi

IKEv2 utilizza come porotocollo a livello trasporto UDP per 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.

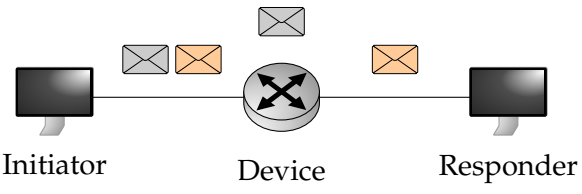

FIGURE 2.2: Drop Pacchetti

Diversi test hanno mostrato che nel caso in cui i peer si trovino in presenza di CG-NAT potrebbero non istausarsi le SA. Questo è dovuto al fatto che i device degli ISP non consentono ai frammenti IP di passare attravers di loro, ovvero scartano i pacchetti e di conseguenza bloccano le comunicazioni IKE. Questo è riportato schematicamente in Fig. 2.2. 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. Ance 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

2.2.1 `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.

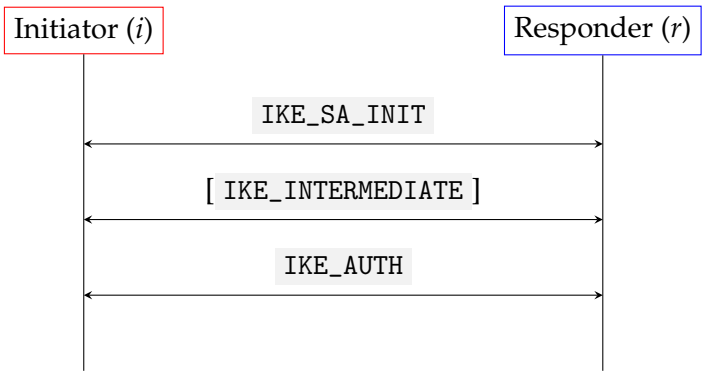

FIGURE 2.3: Scambio nuovo

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 utilizzzato per trasferire qualsiasi tipologia di dato. Quindi il principale utilizzo è quello di fare un **enforcing** delle chiavi negoziate 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_KE` 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, affinchè 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 vengno utilizzata 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 perchè introduce ritardi dovuti alla traduzione degli indirizzi

2.3 Post-Quantum

Un solo KEM con Kyber L1 usando come suite AES_GCM has vabene 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 cmposto da un ottetto che viene autenticato in base alla modalità di aunteticazione 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 aggiunge della unpredicability 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 dirrenti scambi. In particolare al termine del primo scambio viene calcolato il:

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

A partire da questo sidder vengono generati i prametri di sicurezza da utililizzare 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 utilizzata 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 utlilizzatoper negoziare gli attributi di una Secuiry Association. Dunque può contenere molteplici proposte, le quali devono essere ordinate per preferenza, ogni proposal contiene i seguenti algoritmi crittografici:

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