RETI QKD E LORO INTEGRAZIONE NEGLI STANDARD DI SICUREZZA

Relatore: prof. Nicola Laurenti

Laureando: Daniele Piazza

Data di laurea: 22 luglio 2011

Anno Accademico: 2010 - 2011

Indice

1 INTRODUZIONE

1.1 SEGRETEZZA DELL'INFORMAZIONE

La necessità di proteggere le comunicazioni, in particolare in ambito militare e commerciale, ha richiesto fin dall'antichità lo sviluppo di tecniche crittografiche, inizialmente per garantire la segretezza dei contenuti e più recentemente per assicurare anche l'integrità dei messaggi e l'autenticazione delle parti. La storia della crittologia è sempre stata caratterizzata dal confronto tra i crittografi, il cui compito è quello di proteggere le informazioni, ed i crittoanalisti, impegnati a scoprire o alterare i messaggi [1].

La crittografia risponde al bisogno umano antico e moderno di comunicare un messaggio a qualcuno senza che qualche curioso, oppure una spia dell'esercito nemico, possa venire a conoscenza del suo contenuto. I personaggi di questa opera teatrale che è un protocollo crittografico sono quindi almeno due: il mittente (o trasmittente) e il ricevente (o destinatario) e vengono chiamati per comodità con i nomi di Alice (A) e Bob (B). Interviene però molto spesso, quasi per rendere più interessante la trama dell'opera, una spia, più comunemente chiamata intercettatore, in inglese eavesdropper, che per semplicità sarà per noi, come nella convenzione generale del lessico crittografico, Eve (E) [2].

La crittografia si basa su uno schema formato principalmente dai seguenti cinque elementi di cui diamo una definizione informale:

• Testo in chiaro: il messaggio originale, che si vuole trasmettere o archiviare in modo segreto;

• Algoritmo di crittografia: l'insieme di procedure di sostituzione e trasposizione applicate al testo in chiaro;

• Chiave segreta: parametro dell'algoritmo di crittografia, indipendente dal testo in chiaro, modifica l'output dell'algoritmo.

• Testo cifrato: il messaggio trasformato, prodotto dall'applicazione dell'algoritmo di crittografia con la chiave al testo in chiaro.

• Algoritmo di decrittografia: l'inverso dell'algoritmo di crittografia, consente di riprodurre il testo in chiaro a partire dalla chiave e dal testo cifrato [1].

Dunque, lo scopo della crittografia è permettere ad Alice di comunicare con Bob in maniera sicura: Eve non deve riuscire a capire cosa si comunicano trasmettitore e ricevitore. In Figura 1 è rappresentato schematicamente un classico sistema crittografico dove K_1 e K_2 sono le chiavi crittografiche (gran parte della sicurezza di un algoritmo di crittografia dipende da questi due parametri), X è il messaggio in chiaro (plain text), mentre Y è lo stesso messaggio cifrato (ciphertext). L'operazione di criptazione (encryption) è sostanzialmente il calcolo di una funzione F (algoritmo crittografico) che prende come input sia il messaggio in chiaro, sia la chiave K_1, mentre la decriptazione è l'operazione inversa (mediante la chiave K_2) [3].

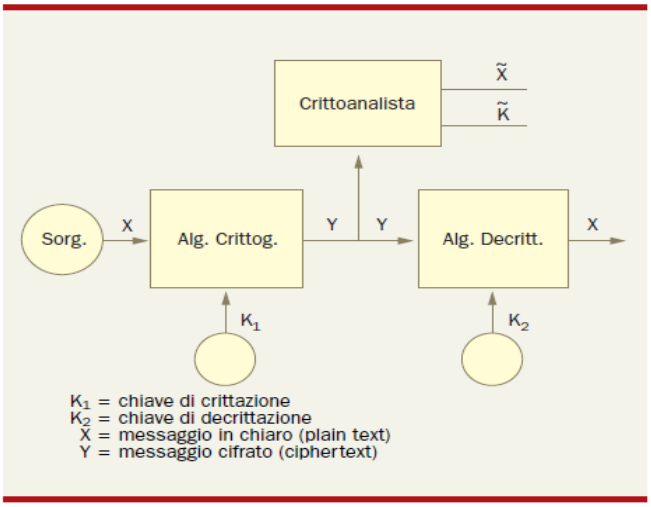

Figura 1: sistema crittografico

Dopo questa breve introduzione alla struttura e al

lessico tecnico della crittografia, vengono riportare alcune idee formalizzate dal linguista olandese Auguste Kerckhoffs nel 1883 su una rivista militare riguardanti le caratteristiche di un sistema crittografico sicuro. Lo studioso elaborò una vera e proprio lista di regole che un sistema sicuro deve avere e sono qui elencate:

1. Un sistema crittografico deve essere materialmente , se non matematicamente, indecifrabile.
2. Il sistema non deve esigere segretezza e deve poter cadere in mani estranee senza che si comprometta l'inviolabilità del messaggio.
3. Deve essere possibile scambiare, memorizzare, cambiare e modificare la chiave senza bisogno di note scritte.
4. Il sistema deve essere applicabile alla comunicazione tramite il telegrafo.
5. Il sistema deve essere portatile e non deve richiedere la presenza di un grande numero di persone.
6. Il sistema non deve richiedere la conoscenza di lunghe regole e deve essere di facile applicazione.

In sostanza i principi di Kerckhoffs affermano che un sistema crittografico dovrebbe rimanere sicuro anche se tutto il sistema, ad eccezione della chiave, sia di pubblico dominio: questo implica che è sempre possibile per un attaccante, Eve, intercettare il canale di comunicazione e quindi ottenere il testo cifrato; da notare che è considerato ammissibile che l'intercettazione sia passiva, ovvero senza interferenza, in modo che mittente e destinatario non sappiano se la comunicazione è stata ascoltata o meno da una terza parte non autorizzata [2].

Un sistema di distribuzione di chiavi è detto essere incondizionatamente sicuro se la seguente condizione è soddisfatta: per ogni attacco compiuto da Eve, il protocollo provoca l'aborto o il successo con una probabilità di $1\text{-}O(2^{-s})$, è garantito che la chiave risultante sia aleatoria e che l'informazione mutua di Eve con la chiave sia minore di $O(2^{-l})$. Qui, "s" ed "l" sono dei parametri di sicurezza che Alice e Bob possono scegliere [4].

In altre parole, uno schema crittografico è detto incondizionatamente sicuro se, indipendentemente dalla quantità di testo cifrato e dal tempo a disposizione, non è possibile de-crittografare il testo in alcun modo a meno di avere la chiave corretta: semplicemente il testo cifrato non contiene alcuna informazione che renda possibile un'analisi di qualunque tipo. Le caratteristiche matematiche di un cifrario del genere sono state enunciate da Shannon negli anni '40, in particolare servirà ricordare che l'entropia contenuta in un testo cifrato non può essere maggiore di quella presente nella chiave usata per cifrarlo, da cui si deduce che condizione necessaria (ma non sufficiente) per ottenere una assoluta sicurezza è l'utilizzo di una chiave casuale non riutilizzabile di lunghezza pari al messaggio (riprenderemo questo concetto più avanti con la descrizione del One-Time Pad) [1].

Banalmente, una chiave segreta è considerata ideale quando è identica tra Alice e Bob ed è privata verso Eve (cioè, Eve non ha informazioni su di essa) [5].

1.2 TIPI DI ATTACCO ALL'ALGORITMO DI CRITTOGRAFIA

La crittoanalisi, accennata nel precedente capitolo, può essere considerata una scienza parallela alla crittografia, infatti essa è volta a debellare le protezioni offerte dalla crittografia. Di seguito vengono descritte le tecniche più usate dai crittoanalisti per decifrare i più diffusi sistemi di crittografia.

1.2.1 Crittoanalisi statistica

Uno dei sistemi più utilizzati per la crittoanalisi di testi cifrati, specie quelli della famiglia dei metodi a sostituzione, è proprio l'analisi della frequenza con cui si ripetono alcune lettere in una data lingua. Analizzando come esempio, un testo scritto in italiano, anche di lunghezza limitata, si noterà

sicuramente che le lettere che compaiono con più frequenza, sono nell'ordine: e,i,a,o. Questa caratteristica può essere sfruttata per decrittare cifrati sconosciuti, o perlomeno come validissima informazione iniziale per la crittoanalisi. Analizzando un qualsiasi testo italiano, inglese, ecc. è possibile stabilire per conto proprio quale sono le frequenze di ogni lettera dell'alfabeto in tale lingua. Analizzando poi le frequenze di un testo cifrato, si può capire con quale lettera è stata sostituita, associando la frequenza di ogni singola lettera del testo cifrato, con quella dell'alfabeto di tale lingua (che andrà scoperta per tentativi se non conosciuta). Se ad esempio in un testo che si sa essere italiano, spicca come lettera a frequenza più alta la lettera "x", sicuramente tale lettera deve essere sostituita con la lettera "e". Operando la sostituzione di tutte le lettere del testo cifrato secondo questo schema, il testo sarà decrittato come minimo al 60%. Ciò significa che ci si troverà ad avere un testo con lettere mancanti o errate in ogni parola, ma facilmente immaginabili [6].

1.2.2 Replay-attack

Il replay-attack consiste nell'impossessarsi di una credenziale di autenticazione comunicata da un mittente ad un destinatario, e riproporla successivamente simulando l'identità dell'emittente. In genere l'azione viene compiuta da un attaccante che s'interpone tra i due lati comunicanti.

Questo attacco permette operazioni fraudolente come falsa autenticazione e/o transazioni duplicate, senza dover necessariamente decrittare la password, ma soltanto ritrasmetterla in un tempo successivo. Il replay attack può operare anche in modo asincrono quando la comunicazione originale è terminata.

Per esempio, si verifica un replay-attack quando Eve intercetta la comunicazione di Alice, che si sta autenticando con Bob, e si spaccia, agli occhi di Bob, per Alice. Quando Bob chiede a Eve (convinto di parlare con Alice) una chiave d'autenticazione, Eve invia quella di Alice, instaurando così la comunicazione.

Gli attacchi di tipo replay si evitano con l'uso di token di sessione generati pseudocasualmente: Bob invia ad Alice uno di questi token usa e getta, che Alice utilizza per criptare la propria chiave da inviare a Bob (per esempio con una funzione di hashing che calcola il *message digest* della chiave concatenata con il token). Bob effettua lo stesso calcolo e controlla che il suo risultato corrisponda con quello di Alice. Eve non può fare granché anche se ha catturato tale token di sessione, perché alla prossima comunicazione Alice e Bob si accorderanno con un altro token. Un'altra contromisura è quella di utilizzare una marca temporale e di far sì che questa sia inserita nel corpo del messaggio criptato [7].

1.2.3 Man in the middle

L'attacco dell'uomo in mezzo, meglio conosciuto come man in the middle attack, è un tipo di attacco nel quale l'attaccante è in grado di leggere, inserire o modificare a piacere, messaggi tra due parti senza che nessuna delle due sia in grado di sapere se il collegamento che li unisce reciprocamente sia stato effettivamente compromesso da una terza parte, appunto un attaccante. L'attaccante deve essere in grado di osservare, intercettare e replicare verso la destinazione prestabilita il transito dei messaggi tra le due vittime.

Supponiamo che Alice voglia comunicare con Bob, e che Eve voglia spiare la conversazione, e se possibile consegnare a Bob dei falsi messaggi. Per iniziare, Alice deve chiedere a Bob la sua chiave pubblica (la crittografia a chiave pubblica verrà vista nel paragrafo successivo). Se Bob invia la sua chiave pubblica ad Alice, ma Eve è in grado di intercettarla, può iniziare un attacco man in the middle. Eve può semplicemente inviare ad Alice una chiave pubblica della quale possiede la corrispondente chiave privata. Alice poi, credendo che questa sia la chiave pubblica di Bob, cifra i suoi messaggi con la chiave di Eve ed invia i suoi messaggi cifrati a Bob. Eve quindi li intercetta, li decifra, ne tiene una copia per sé, e li recifra (dopo averli alterati se lo desidera) usando la chiave pubblica che Bob aveva

originariamente inviato ad Alice. Quando Bob riceverà il messaggio cifrato, crederà che questo provenga direttamente da Alice. Un simile attacco è possibile, in teoria, verso qualsiasi messaggio inviato usando tecnologia a chiave pubblica, compresi pacchetti di dati trasportati su reti di computer [8].

1.2.4 Attacco forza bruta

Il metodo "forza bruta" (anche noto come ricerca esaustiva della soluzione) è un algoritmo di risoluzione di un problema che consiste nel verificare tutte le soluzioni teoricamente possibili fino a che si trova quella effettivamente corretta. Il suo principale fattore positivo è che consente teoricamente sempre di trovare la soluzione corretta, ma per contro è sempre la soluzione più lenta o dispendiosa; viene utilizzato come ultima risorsa sia in crittanalisi che in altre parti della matematica solamente in quei casi dove sia l'unico procedimento conosciuto. Questo metodo si utilizza per trovare la chiave di un sistema che impiega un cifrario per il quale non si conosca alcun attacco migliore, ed è noto appunto come attacco di forza bruta.

Quando sul sistema è possibile un attacco offline (ovvero quando l'attacco si può eseguire su una copia di lavoro locale del sistema da attaccare) si può compensare la lentezza di esecuzione con la quantità di risorse: laddove un singolo computer possa "provare" 100.000 chiavi al secondo, due computer possono provarne il doppio e così via (la velocità aumenta linearmente con le risorse utilizzate). Questa caratteristica ha nei recenti anni motivato molti attacchi "distribuiti" sfruttando solo i cicli inutilizzati di migliaia e migliaia di comuni computer. Questo ovviamente non è applicabile a sistemi informatici dove sia possibile esclusivamente un attacco online, né a sistemi che utilizzino protezioni fisiche quali lucchetti metallici: non è ovviamente possibile sveltirne l'apertura provando due o più chiavi alla volta.

1.2.5 Attacco a dizionario

Un attacco a dizionario è una tecnica per "rompere" un codice cifrato o un meccanismo di autenticazione provando a decifrare il codice o a determinare la passphrase cercando tra un gran numero di possibilità. In pratica si tenta di accedere a dati protetti da password (sia remoti, come ad esempio account su siti web o server di posta; sia locali, come documenti o archivi protetti da password) tramite una serie continuativa e sistematica di tentativi di inserimento della password, solitamente effettuati in modo automatizzato, basandosi su uno o più dizionari. In contrasto con un metodo forza bruta, dove tutte le possibili password sono ricercate in maniera esaustiva, un attacco a dizionario prova solamente quelle ritenute più probabili, tipicamente contenute in una lista (detta *dizionario*). Generalmente, questi attacchi, detti per questo "a dizionario", hanno successo perché la maggior parte delle persone ha la tendenza a scegliere password semplici da ricordare (e quindi semplici da scoprire, ad esempio il proprio nome, quello dei propri figli, date di nascita) e tendenzialmente sceglie parole prese dalla propria lingua nativa [9].

1.3 CRITTOGRAFIA A CHIAVE PRIVATA E PUBBLICA

1.3.1 Crittografia a chiave privata

La crittografia simmetrica, o crittografia a chiave privata, è una tecnica di cifratura. Uno schema di crittografia simmetrica è caratterizzato dalla proprietà che, data la chiave di cifratura "e", sia facilmente calcolabile la chiave di decifratura "d". Un caso particolare, che è quello quasi sempre utilizzato nella pratica, è l'utilizzo della stessa chiave sia per l'operazione di cifratura che quella di decifratura.

La forza della crittografia simmetrica è dunque riposta nella segretezza dell'unica chiave utilizzata dai

due interlocutori che la usano, oltre che nella grandezza dello spazio delle chiavi, nella scelta di una buona chiave e nella resistenza dell'algoritmo agli attacchi di crittoanalisi.

Generalmente gli algoritmi di crittografia simmetrica sono molto più veloci di quelli a chiave pubblica (o crittografia asimmetrica) che vedremo dopo, per questo vengono usati in tutte le operazioni di cifratura che richiedono performance alte. Oltretutto i cifrari a crittografia simmetrica permettono l'uso di chiavi lunghe N bit quanto il messaggio, ottenendo uno spazio delle chiavi di dimensioni 2^N. Alcuni esempi di cifratura simmetrica sono dati dal cifrario di Cesare, il cifrario di Vigénère, l'algoritmo DES, il nuovo standard AES e il One-Time Pad. Vediamo in dettaglio quest'ultima crittografia simmetrica [10].

1.3.2 Il One-Time Pad (OTP)

L'unico cifrario incondizionatamente sicuro attualmente conosciuto è il One-Time Pad: tale sistema richiede una chiave casuale non riutilizzabile lunga tanto quanto il messaggio da inviare, ogni bit del testo in chiaro viene messo in XOR con il corrispondente bit della chiave per formare il testo cifrato. Per le proprietà dello XOR, la decrittografia si esegue effettuando la stessa operazione bit a bit tra il testo cifrato e la stessa chiave.

In breve, per la crittografia:

$$c_i = p_i \otimes k_i$$

mentre per la decrittografia:

$$p_i = c_i \otimes k_i$$

dove

p_i = i-esima cifra binaria del testo in chiaro (plaintext)

c_i = i-esima cifra binaria del testo cifrato (ciphertext)

k_i = i-esima cifra binaria della chiave (key)

$\otimes$ = simbolo dell'operatore di OR esclusivo (XOR)

Questo schema, se la chiave è scelta come una sequenza realmente casuale e non viene mai riutilizzata, è assolutamente inviolabile in quanto produce un testo cifrato che non ha nessuna relazione statistica con il testo in chiaro che lo ha generato e non contiene quindi alcuna informazione sul plaintext stesso.

Nonostante l'assoluta inviolabilità di questo procedimento esso non viene praticamente mai adottato nella pratica a causa di un fondamentale problema: se fosse possibile distribuire in modo assolutamente sicuro chiavi di lunghezza arbitraria tra due soggetti ogni volta che questo si renda necessario, lo stesso canale "sicuro" potrebbe essere utilizzato per trasmettere direttamente il messaggio, rendendo superflua la crittografia medesima [1].

1.3.3 Crittografia a chiave pubblica

La crittografia asimmetrica, conosciuta anche come a chiave pubblica, è un tipo di crittografia dove, come si evince dal nome, ad ogni attore coinvolto è associata una coppia di chiavi:

- la chiave pubblica, che deve essere distribuita, serve a cifrare un documento destinato alla persona che possiede la relativa chiave privata.
- la chiave privata, personale e segreta, utilizzata per decodificare un documento cifrato con la chiave pubblica;

evitando così qualunque problema connesso allo scambio dell'unica chiave utile alla cifratura/decifratura presente invece nella crittografia simmetrica.

L'idea base della crittografia con coppia di chiavi diviene più chiara se si usa un'analogia postale, in cui il mittente è Alice ed il destinatario Bob, i lucchetti fanno le veci delle chiavi pubbliche e le chiavi recitano la parte delle chiavi private:

1. Alice chiede a Bob di spedirle il suo lucchetto, già aperto. La chiave dello stesso verrà però gelosamente conservata da Bob.
2. Alice riceve il lucchetto e, con esso, chiude il pacco e lo spedisce a Bob.
3. Bob riceve il pacco e può aprirlo con la chiave di cui è l'unico proprietario.

Se adesso Bob volesse mandare un altro pacco ad Alice, dovrebbe farlo chiudendolo con il lucchetto di Alice (che lei dovrebbe aver preventivamente dato a Bob) che solo lei potrebbe aprire.

Si può notare come per mettere in sicurezza il contenuto dei pacchi ci sia bisogno del lucchetto del destinatario, mentre per aprirli viene usata esclusivamente la propria chiave segreta, rendendo l'intero processo di cifratura/decifratura asimmetrico (una chiave per cifrare ed una differente per decifrare). Chiunque intercettasse il lucchetto (aperto) o il messaggio chiuso con il lucchetto non potrebbe leggerne il contenuto poiché non ha la chiave. Uno dei vantaggi della crittografia asimmetrica sta nel fatto che le chiavi pubbliche possono essere scambiate anche utilizzando un mezzo insicuro, come Internet.

Nella crittografia simmetrica invece, che basa la sicurezza del sistema sulla segretezza della chiave di codifica/decodifica utilizzata, si rende necessario utilizzare un canale sicuro per la trasmissione della chiave, poiché l'intercettazione della stessa, da parte di terzi, vanificherebbe la sicurezza del sistema stesso.

La crittografia a chiave pubblica permette a due (o più) persone di comunicare in tutta riservatezza senza usare la stessa chiave anche se non si sono mai incontrate precedentemente. La coppia di chiavi pubblica/privata viene generata attraverso un algoritmo (ad esempio RSA o Diffie-Helmann, vedere appendice) a partire da dei numeri casuali.

Oltre alla cifratura dei dati di una comunicazione la crittografia asimmetrica presenta altri possibili impieghi: firma digitale per verificare l'autenticazione del mittente e l'integrità informativa del messaggio, supporto alla fase di handshake ovvero di avvio di una sessione con crittografia simmetrica per negoziare la chiave di sessione, il protocollo e gli altri aspetti della connessione cifrata.

Un utente può firmare un messaggio utilizzando la propria chiave privata; per far ciò viene creata un'impronta (digest) del messaggio da firmare e questa viene firmata con la chiave privata ed inviata assieme al messaggio (l'impronta, generata per mezzo di un algoritmo di hash, è tale che varia sensibilmente al minimo variare del messaggio). Tutti i destinatari del messaggio possono verificare l'integrità del messaggio stesso e l'autenticazione dell'autore/mittente creando, a partire dal messaggio ricevuto e decifrato, un'impronta (utilizzando in maniera simmetrica la stessa funziona hash utilizzata dall'autore del messaggio) e confrontandola poi con quella ricevuta assieme al messaggio e cifrata con la chiave pubblica del presunto autore: se le due impronte risultano identiche il messaggio è integro, ovvero non ha subito modifiche da parte di terzi (ad esempio attraverso attacchi del tipo man in the middle) da quando l'autore a monte l'ha firmato.

In realtà il problema della sicurezza riguardante la segretezza della comunicazione non è del tutto risolto con questo tipo di crittografia in quanto passibile di attacchi di tipo man in the middle: non si può essere certi infatti che la chiave (per esempio una chiave presente sul keyserver) appartenga davvero alla persona nominata nell'intestazione della chiave stessa apportando così attacchi di tipo spoofing in assenza di un meccanismo di autenticazione tra le parti in causa. Una soluzione resta sempre il contatto fisico tra i due interlocutori, i quali, scambiandosi le chiavi pubbliche hanno una reciproca autenticazione [11].

1.3.4 Considerazioni

Gli algoritmi asimmetrici sono studiati in modo tale che la conoscenza della chiave pubblica e dell'algoritmo stesso non siano sufficienti per risalire alla chiave privata e tale meccanismo è reso possibile grazie all'uso di funzioni unidirezionali. In realtà, in molti casi, l'impossibilità di risalire alla chiave privata non è dimostrata matematicamente, ma risulta dallo stato attuale delle conoscenze in matematica e della potenza di calcolo disponibile. Per esempio è sufficiente un piccolo computer e qualche millesimo di secondo per moltiplicare due numeri primi da 150 cifre, ma occorre il lavoro di decine di migliaia di computer per un anno per trovare i fattori primi di quel numero. Un altro problema simile è quello della funzione unidirezionale esponenziale modulo *n* (aritmetica modulare) e del rispettivo problema inverso del calcolo del suo logaritmo discreto.

Attualmente, per la crittografia RSA vengono considerati "sicuri" numeri che in base 10 hanno almeno 300 cifre, il che significa chiavi di 1024 bit e oltre.

La crittografia è una scienza basata sulle probabilità: i problemi complessi vengono considerati complessi basandosi sul fatto che centinaia di anni di studio non hanno saputo risolverli in modo rapido (ricordiamoci che c'è sempre almeno un modo "non rapido" per risolvere un problema: provare a fare l'operazione diretta con tutti i numeri fino alla dimensione necessaria; questo tipo di soluzione in genere non è neanche contemplata in quanto il tempo necessario aumenta vertiginosamente con la dimensione dei numeri usati), ma nessuno dei problemi usati in crittografia ha un teorema che ne dimostra la complessità (l'unico sistema crittografico dimostrato è il One Time Pad, ma sfortunatamente è un sistema simmetrico ed estremamente scomodo da usare).

In pratica, la crittografia simmetrica è più semplice, veloce e sicura di quella asimmetrica, ma necessita che prima venga condivisa una chiave. Quindi normalmente si usa la crittografia asimmetrica per scambiarsi una chiave, con cui procedere ad una comunicazione a crittografia simmetrica [10].

La crittografia quantistica, di cui ci occuperemo nel terzo capitolo di questa tesina, potrebbe sostituire i sistemi asimmetrici, che sono materialmente inviolabili, ma non matematicamente, nello scambio delle chiavi [2].

2 LA CRITTOGRAFIA ODIERNA SU INTERNET

2.1 IPSEC

IPsec è un insieme di protocolli e algoritmi ed è una struttura flessibile che consente ai fornitori che utilizzano IPsec nei propri prodotti a selezionare gli algoritmi, le chiavi, e i metodi di autenticazione che si desidera utilizzare. IPsec fornisce due servizi di base di autenticazione e di riservatezza:

- L'autenticazione è ottenuta con l'aggiunta di un Authentication Header (AH) il quale viene dopo l'header IP di base e contiene un hash crittografico sicuro dei dati e delle informazioni di identificazione.

- La riservatezza è raggiunta attraverso l'aggiunta di un header di Encapsulating Security Payload (ESP), e l'eventuale riscrittura del carico utile in forma criptata. ESP applica concetti di crittografia che forniscono l'autenticazione, l'integrità e la riservatezza dei messaggi.

IPSEC definisce una "Security Association" (SA) come suo mezzo primitivo di protezione dei pacchetti IP. Una SA è definita da un indirizzo di destinazione del pacchetto IP e da un Security Parameter Index (SPI) da 32-bit, che funziona un po' come un numero di porta TCP o UDP consentendo più SA verso un indirizzo di destinazione unico [12].

2.1.1 IPsec Key management e ISAKMP

IKE (Internet Key Exchange) è un sistema sviluppato appositamente per IPsec per dare meccanismi di autenticazione e scambio di chiavi in una delle situazioni possibili su Internet. Esso è composto da molti elementi: ISAKMP, SKEME e una parte di Oakley.

Come abbiamo visto in precedenza, i servizi di sicurezza sono dati con l'uso di associazioni di protezione. Tali SA definiscono i parametri necessari per garantire un flusso di dati. ISAKMP (Internet da un "Security Association" and Key Management Protocol) ha come funzionalità la negoziazione, la creazione, la modifica e la soppressione di associazioni di sicurezza e dei loro attributi. ISAKMP è un framework generico indipendente dai meccanismi di negoziazione. Essa non impone alcuna condizione ai parametri SA.

ISAKMP si compone di due fasi, che consentono una netta separazione della negoziazione di SA, per un dato protocollo, dalla protezione del traffico specifico per lo stesso ISAKMP: durante la prima fase, un insieme di attributi relativi alla sicurezza è negoziata; le identità vengono autenticate e le chiavi vengono generate. Questi elementi costituiscono una prima "associazione di protezione", nota come SA-ISAKMP.

La seconda fase permette di negoziare i parametri di sicurezza relativi ad SA per conto di un meccanismo di sicurezza dato (per esempio AH o ESP). Gli scambi in questa fase sono protetti (riservatezza, autenticità ...) grazie ad SA-ISAKMP. ISAKMP è anche indipendente dal metodo di generazione e dall'autenticazione delle chiavi e dagli algoritmi di cifratura utilizzati. Essa è quindi indipendente da qualsiasi protocollo di scambio di chiave, il quale rende possibile separare chiaramente i dettagli della gestione di associazioni di protezione dai dettagli di scambio di chiave. Vari protocolli di scambio delle chiavi, che presentano caratteristiche diverse sono quindi utilizzabili con ISAKMP.

2.1.2 IKE

IKE utilizza ISAKMP per costruire un protocollo pratico. IKE include quattro modalità: il Main Mode, l'Aggressive Mode, il Quick Mode e il New Group Mode. Il Main Mode e l'Aggressive Mode

vengono utilizzati durante la fase 1; il Quick Mode è uno scambio di fase 2. Il New Group Mode non è né uno scambio di fase 1, né uno scambio di fase 2, ma può avvenire solo se SA ISAKMP è stabilito, è utilizzato per un accordo su un nuovo gruppo per scambi futuri basati sul protocollo Diffie-Hellman.

2.1.2.1 Phase 1: Main mode e Aggressive Mode

Sono utilizzati e sono negoziati durante la fase 1 i seguenti attributi da IKE: un algoritmo di crittografia, una funzione di hash, e un metodo di autenticazione e di un gruppo di Diffie-Hellman. Tre chiavi vengono generate al termine della fase 1: una per la cifratura, una per l'autenticazione ed una per la creazione di altre chiavi.

Queste chiavi dipendono dai cookie, dal numero scambiato e dai valori pubblici dell'algoritmo di Diffie-Hellman oppure dalla segretezza condivisa preliminarmente. Il loro calcolo utilizza la funzione di hash scelta per SA-ISAKMP e dipende dalla modalità di autenticazione selezionata. Il un numero di messaggi nella Main Mode è sei, mentre vi sono solo tre messaggi scambiati nell'Aggressive Mode.

In questi due casi, il metodo scelto per l'autenticazione influisce sul contenuto dei messaggi e sul metodo di generazione della chiave di sessione.

2.1.2.2 Phase 2: Quick mode

I messaggi scambiati durante la fase 2 sono protetti grazie all'autenticità e alla riservatezza degli elementi negoziati durante la fase 1. L'autenticità dei messaggi è garantita con l'aggiunta di un blocco HASH dopo l'header ISAKMP, e la riservatezza è garantita dalla cifratura dei blocchi dell'intero messaggio.

Il Quick Mode è utilizzato per la negoziazione delle SA per protocolli di sicurezza come IPsec. Ogni negoziato conduce infatti a due SA, una in ciascuna direzione della comunicazione [12].

2.2 TLS

Il protocollo TLS è stato messo a punto da Netscape, e successivamente standardizzato da IETF. Si tratta di uno standard di sicurezza delle transazioni le quali forniscono connessioni sicure tra due entità comunicanti con la sicurezza della protezione dell'integrità, l'autenticazione reciproca, e la gestione delle chiavi. Questo protocollo garantisce due servizi: una connessione crittografata punto-a-punto e l'integrità dei messaggi. Essa comprende cinque sotto-protocolli: Record Protocol, Handshake Protocol, Change Spec Protocol, Alert Protocol e Application Data Protocol.

2.2.1 Il TLS Record Protocol

Prendendo il messaggio pronto alla trasmissione, il Record protocollo frammenta i dati in blocchi più gestibili, comprime i dati (opzionale), applica un MAC, crittografa e trasmette il risultato. Per consentire a un client ed a un server di concordare i parametri di sicurezza, il protocollo TLS utilizza il protocollo di Handshake.

2.2.2 Il TLS Handshake Protocol

Il client e il server nel protocollo di Handshake si autenticano a vicenda utilizzando i certificati oppure le chiavi precondivise, creano un'istanza della negoziazione dei parametri di sicurezza e calcolano la chiave di sessione utilizzata per cifrare i dati scambiati. Questo si compone di tre fasi.

Step 1

Il client e il server nel primo passaggio negoziano i parametri della sessione protetta. Questi parametri, in particolare, contengono l'identificatore di sessione (SessionID) e l'insieme dei codici. Quest'ultimo è formato da una tripletta e trasmette il metodo di scambio della chiave che è utilizzata per scambiare la chiave di sessione, l'algoritmo di cifratura che viene distribuito per cifrare/decifrare i dati dell'applicazione, e una funzione di hash per assicurare l'integrità dei dati. Nel suo messaggio

ClientHello, il client include (Figura 2) un elenco delle terzine supportate in ordine di preferenza.

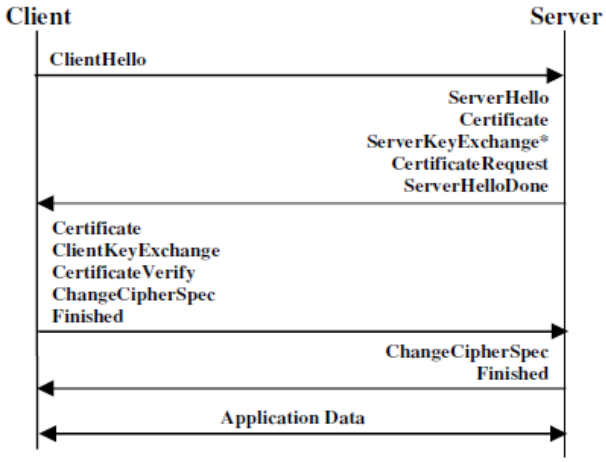

Figura 2: il TLS Handshake Protocol

Con il suo ServerHello, il server risponde trasmettendo l'insieme dei codici di cifratura selezionati o, se non è presente alcuna scelta accettabile, restituisce un avviso di "error Handshake" e chiude la connessione. Il ServerHello e il ClientHello stabiliscono i seguenti attributi: Versione di Protocollo, ID di sessione, Cipher Suite, e il metodo di compressione. Inoltre, vengono generati e scambiati due valori casuali: ServerHello.random e ClientHello.random.

Step 2

In una seconda fase il client e il server si autenticano reciprocamente. Sono state definite due modalità di autenticazione nel protocollo TLS: la sola autenticazione del server e l'autenticazione reciproca. L'autenticazione è di solito eseguita utilizzando le chiavi pre-condivise, i certificati oppure una chiave pubblica installata sia sul client che sul server e in tal caso è richiesta una infrastruttura a chiave pubblica.

Il server di autenticazione basato sul certificato invia un messaggio di richiesta di certificato (Figura 2), invitando il cliente a rispondere, a sua volta, con un certificato. Quindi, un certificato è spedito dal client al server il quale prova che il client è il legittimo titolare del certificato. Il cliente, a titolo di prova, invia il messaggio CertificateVerify, che gestisce l'hash di tutti i messaggi scambiati tra il client e il server a partire da ClientHello fino al, ma non compreso, messaggio CertificateVerify. Il server verifica che il cliente sia in possesso della chiave privata corrispondente alla chiave pubblica certificata. Nel caso in cui la convalida non riesce, il server interrompe l'handshake.

Step 3

Al fine di verificare il successo della modalità di autenticazione selezionata ed i processi di scambio della chiave, sia il client che il server si scambiano il ChangeCipherSpec ed i messaggi finiti (Figura 2). Questi ultimi danno prova, sia per il client che per il server, che hanno la stessa chiave poiché i messaggi finiti sono i primi messaggi trattati e scambiati dopo aver applicato i parametri di sicurezza negoziata. Il messaggio di chiusura TLS è calcolato con la seguente formula:

PRF (segreta _ master, finito etichetta _, Hash (messaggi di handshake _))

Qui, PRF è una funzione pseudo-casuale. Per il finished_label, usiamo la stringa "client finished", per il messaggio inviato dal client, e "server "finished" per quella inviata dal server. Hash indica un hash del messaggio di handshake. L'handshake_messages include tutti i messaggi di handshake a partire dal ClientHello fino a, ma non compreso, questo messaggio TLS di chiusura. Così, l'handshake_messages, per il messaggio di chiusura inviato dal client, sarà diverso da quello per il messaggio di chiusura

inviato dal server, poiché quello che viene inviato per secondo comprenderà il precedente. Il valore di master_secret è presentato con la formula:

master _secret = PRF (pre _master _secret , “master secret ”, ClientHello.random + ServerHello.random)

Il pre_master_secret deriva dal meccanismo di distribuzione delle chiavi (ad esempio RSA o Diffie-Hellman). Pertanto, quando l'RSA è utilizzato per lo scambio delle chiavi, un pre_master_secret è generato dal client, crittografato con la chiave pubblica del server, e inviato al server. Per decifrare il pre_master_secret, il server utilizza la sua chiave privata. Se è eseguito il calcolo convenzionale Diffie-Hellman, la chiave del negoziato è utilizzata come pre_master_secret. Nella formula precedente, il simbolo “+” rappresenta l'operatore di concatenazione [13].

3 LA CRITTOGRAFIA QUANTISTICA

3.1 LA NATURA DELLA LUCE:

Nel XVII secolo nell'ambiente scientifico era in corso un vivace dibattito sulla natura della luce: alcuni fisici, (esempio Isaac Newton) sostenevano che essa fosse di natura corpuscolare mentre altri uomini di scienza (esempio Christiaan Huygens), difendevano la teoria secondo la quale la luce è un'onda. A quel tempo entrambi questi modelli erano in grado di spiegare tutti i fenomeni conosciuti come i colori, la riflessione, la rifrazione e le ombre, ma negli anni che seguirono venne scoperto un fenomeno sperimentale, l'interferenza, che la teoria corpuscolare non era in grado di spiegare.

Successivamente Maxwell incluse il modello ondulatorio nella sua teoria sull'elettromagnetismo e spiegò con la sistemazione teorica da lui attuata anche l'interferenza, la diffrazione e la polarizzazione.

Se consideriamo la luce come un'onda in ogni punto di un raggio luminoso sono presenti un campo elettrico e un campo magnetico che oscillano perpendicolarmente l'uno rispetto all'altro e alla direzione di propagazione dell'onda.

I piani su cui vibrano i due campi, piani di vibrazione, non sono però costanti, ma variano nel tempo, mantenendo sempre le note perpendicolarità. Facendo però passare un raggio di luce in determinati materiali, all'uscita da queste sostanze, avremo un'onda luminosa il cui campo elettrico oscilla solo più su un piano lungo una stessa direzione: abbiamo ottenuto luce polarizzata.

Vi sono diversi tipi di polarizzatori, ma quelli che ci interessano sono principalmente i filtri analizzatori di cui ne sono un esempio i filtri polaroid. Questi polarizzatori filtrano la luce in un'unica direzione, assorbendo tutte le vibrazioni su piani diversi da quello della sua struttura, e permettendo soltanto a quelle che si muovono in un determinato modo di passare attraverso il filtro.

La spiegazione dei fenomeni della polarizzazione e della birifrangenza esposte brevemente, è la cosiddetta interpretazione classica. All'inizio del '900 nella fisica ci fu una rivoluzione paragonabile solamente a quella di Galilei e Newton: nacque la meccanica quantistica.

Lo stato dell'arte era il seguente: la fisica era suddivisa in due grandi filoni: uno legato alla radiazione e alle spiegazioni di Maxwell delle onde elettromagnetiche, l'altro legato alla meccanica classica, al concetto di materia e alla spiegazione atomistica della realtà. Tuttavia all'inizio di quello che è ormai il secolo scorso tre campi apparentemente non così collegati tra loro quali l'analisi degli spettri, lo studio del corpo nero e l'esperimento dell'effetto fotoelettrico condussero alla medesima conclusione: la natura è quantizzata. Ciò significa che gli scambi di energia, ad esempio, non possono avvenire in maniera continua, ma sotto forma di piccoli pacchetti detti quanti, ovvero in modo discreto. La nostra vita quotidiana offre una serie di analogie utili per comprendere il concetto: per la prima basta a pensare come possiamo definire in modo diverso la posizione di un oggetto su una salita e su una scalinata: nel primo caso esso può occupare qualsiasi punto del pendio, mentre nel secondo possiamo solo indicare quale gradino occupa; analogamente si può pensare al cambio di velocità di un'automobile: possiamo innestare la prima marcia, la seconda e così via, ma non possiamo trovarci nella "terza marcia e mezzo".

In particolar modo nell'effetto fotoelettrico si constatò che la luce poteva essere considerata un flusso di pacchetti e questo era l'unico modo per spiegare esaurientemente questo fenomeno, che sembrava scardinare l'idea della luce come onda. La teoria ondulatoria riusciva però a dare spiegazioni molto esaurienti di tutta una serie di fenomeni che non erano comprensibili se si considerava la luce come un insieme di particelle (in particolar modo la diffrazione).

Il problema era, quindi, come conciliare queste due teorie così diverse, ma entrambe indispensabili.

Nel 1924 un fisico francese, De Broglie, propose un'ipotesi completamente nuova: un dualismo onda-corpuscolo per cui sia per la luce che la materia potevano presentarsi a seconda dei casi e dei metodi di osservazione sia come onda che come insieme di corpuscoli. Questo dualismo è uno dei nuclei concettuali della meccanica quantistica come anche il cosiddetto principio di indeterminazione di Heisenberg che andremo ad illustrare nel paragrafo successivo [2].

3.2 LE BASI DELLA CRITTOGRAFIA QUANTISTICA:

3.2.1 IL PRINCIPIO DI INDETERMINAZONE DI HEISENBERG

Fino ai primi anni del XX secolo non era mai stato messo in dubbio che si potesse misurare qualsiasi grandezza fisica con il grado di accuratezza voluto, a patto che si possedesse uno strumento sufficientemente preciso. La meccanica quantistica dimostrò che anche in esperimenti in cui possediamo strumenti di misurazione ideali vi sono limiti alla precisione con cui la misura può essere effettuata. Se immaginiamo di dover attraversare la strada ci occorre conoscere le posizioni delle automobili sulla carreggiata e le loro velocità: per ricavare queste informazioni ci serviamo della vista (la luce viene riflessa dall'oggetto osservato e arriva ai nostri occhi). Se però proviamo a osservare la macchina con onde radio (di lunghezza d'onda 300m) ci accorgiamo che è impossibile localizzarla. Consideriamo analogamente un elettrone che si muova in un tubo a vuoto: vogliamo conoscerne posizione e velocità. Per misurare la posizione si deve usare una radiazione con lunghezza d'onda confrontabile con le dimensioni atomiche: un fotone di tale lunghezza d'onda ha una frequenza molto elevata e di conseguenza molta energia. Colpendo l'elettrone gli trasmette parte di quest'energia modificandone la velocità. Nel mondo quantistico infatti osservando un sistema in genere lo si perturba e si ottiene un'informazione incompleta [2].

Il principio di Heisenberg rende ragione dell'impossibilità mediante osservazione di determinare la posizione e quantità di moto di una particella elementare contemporaneamente, in quanto l'una esclude l'altra e noi, nell'osservare, restiamo totalmente estranei al mondo subatomico. Il mondo subatomico è come un mondo "a sé", una "regione cosmica" con leggi proprie che include molteplici sistemi fisici, in cui noi non possiamo entrare coi nostri strumenti per coglierne interamente la natura. Vi sono infatti due sostanziali elementi di indeterminazione concernenti le particelle elementari, che sono la dualità onda/particella e la non-località (l'entanglement di cui ne parleremo più avanti).

Il principio di indeterminazione rappresenta la chiave di volta della meccanica quantistica, confermato da oltre ottant'anni di esperienze, in quanto sancisce il sostanziale indeterminismo comportamentale delle entità appartenenti al mondo subatomico

In generale, qualunque coppia di grandezze osservabili generiche, che non siano nella relazione di essere compatibili, non si potranno misurare simultaneamente, se non a prezzo di un'indeterminazione sull'una tanto più grande quanto più piccola si riduce quella sull'altra osservabile [14].

3.2.2 TEOREMA DI NO-CLONING

Il teorema di no-cloning è il risultato della meccanica quantistica, che vieta la creazione di copie identiche di uno stato quantistico arbitrario sconosciuto. È stato affermato da Wootters, Zurek, e Dieks nel 1982, e ha implicazioni profonde nella computazione quantistica e in campi correlati.

Il teorema di no-cloning quantistico afferma che, dati i postulati della meccanica quantistica (cioè l'insieme delle ipotesi di base che rappresentano un punto di partenza nella formulazione della teoria quantistica in forma assiomatica), non è possibile duplicare esattamente (*cloning* appunto) uno stato quantistico sconosciuto a priori. È invece possibile effettuare la duplicazione senza errori se lo stato appartiene ad un insieme ortogonale di stati

conosciuto a priori: tale duplicazione fallisce se lo stato non appartiene all'insieme ortogonale. In particolare è sempre possibile duplicare uno stato conosciuto a priori.

Se fosse, infatti, possibile distinguere in modo certo stati non ortogonali, sarebbe poi possibile utilizzare una macchina specifica per la duplicazione di uno stato piuttosto che l'altro, rendendo così possibile la duplicazione di stati non ortogonali.

Nel caso classico la duplicazione di una informazione è in principio sempre possibile. L'apparente contraddizione con il caso quantistico viene risolta dal fatto che gli stati di un sistema macroscopico descrivibile classicamente appartengono sempre ad un insieme ortogonale, per il quale la duplicazione è possibile.

Anche se è impossibile fare copie perfette di uno stato quantistico sconosciuto, è possibile produrre copie imperfette. Questo può essere fatto mediante accoppiamento di un più ampio sistema ausiliario al sistema che deve essere clonato, e applicando una trasformazione unitaria al sistema combinato. Se la trasformazione unitaria è scelta correttamente, diversi componenti del sistema combinato si evolveranno in copie approssimative del sistema originale. La clonazione imperfetta, come vedremo in dettaglio nei prossimi paragrafi, può essere usata come un attacco di intercettazione sui protocolli di crittografia quantistica e la sicurezza della crittografia quantistica sta proprio in questa imperfezione [15].

3.2.3 L'ENTANGLEMENT QUANTISTICO

Gli stati di polarizzazione si possono combinare linearmente, ovvero moltiplicare per un numero e poi sommare tra loro, ottenendo come risultato un nuovo stato. La prima conseguenza di questo è la possibilità di definire un certo stato di polarizzazione (in termini di ampiezza di probabilità) in una base. Ad esempio la combinazione lineare di uno stato $|\updownarrow\rangle$ con uno $|\leftrightarrow\rangle$ fornisce come risultato uno stato $|\nearrow\rangle$, nello specifico:

$$|\nearrow\rangle=1/\sqrt{2}|\updownarrow\rangle+1/\sqrt{2}|\leftrightarrow\rangle$$

Esprimendo questo concetto in termini più formali, lo stato di polarizzazione è definito come un vettore in uno spazio di Hilbert (uno spazio vettoriale bidimensionale complesso, su cui è definita una operazione di prodotto interno). Gli elementi di una base ortonormale in questo spazio sono indicati con $|0\rangle$ e $|1\rangle$ ed un vettore normalizzato può quindi essere rappresentato come

$$|Y\rangle=a|0\rangle+b|1\rangle, \text{ con } |a|^2+|b|^2=1$$

dove a, b $\in$ **C**. Questa notazione non significa che il valore dello stato di polarizzazione assume un valore compreso tra 0 e 1 ma piuttosto che esso si trova in una sovrapposizione coerente di entrambi gli stati e che, misurando $|Y\rangle$ si ottiene un risultato non deterministico: la probabilità di ottenere il risultato $|0\rangle$ è pari ad $|a|^2$, mentre la probabilità di ottenere $|1\rangle$ è $|b|^2$. Naturalmente la somma delle probabilità individuali dei due eventi mutuamente esclusivi è 1, in accordo con il fatto che essi sono gli unici eventi che è possibile osservare.

La notazione di Dirac consente di definire completamente lo stato di una particella e la teoria prevede che per la particella stessa non sia possibile alcuna ulteriore specificazione circa lo stato. La natura genuinamente casuale dei processi quantistici è quindi implicata, oltre che dalle evidenze sperimentali, dall'assunzione che la teoria sia completa, ovvero che la specificazione del vettore di stato rappresenti l'informazione più completa (in teoria, non solo in pratica) su un dato sistema fisico e quindi che non sia possibile conoscere nulla oltre al vettore stesso. Una volta specificato il vettore di stato è possibile calcolare le probabilità che un fotone superi un test di polarizzazione, ma questa informazione è l'unica cosa che è possibile conoscere circa il processo di misura stesso. Vediamo ora in dettaglio come viene definito uno stato separabile e uno stato entangled [1].

Si considerino due sistemi non interagenti A e B a cui sono associati i rispettivi spazi di Hilbert H_A e H_B. Lo

spazio di Hilbert del sistema composto, secondo i postulati della meccanica quantistica, è il prodotto tensoriale

$$H_A \otimes H_B$$

Se il primo sistema è nello stato $|\varphi\rangle_A$ e il secondo è nello stato $|\varphi\rangle_B$ lo stato del sistema composto è

$$|\varphi\rangle_A |\varphi\rangle_B$$

stati di questo tipo vengono chiamati stati separabili.

Date due basi $|i\rangle_A$ e $|i\rangle_B$ associate alle osservabili Ω_A e Ω_B è possibile scrivere gli stati puri di cui sopra come

$$\left(\sum_i a_i |i\rangle_A\right)\left(\sum_j b_j |j\rangle_B\right)$$

per una certa scelta dei coefficienti complessi a_i e b_j. Questo non è lo stato più generale di $H_A \otimes H_B$, il quale ha la forma

$$\sum_{i,j} c_{ij} |i\rangle_A \, |j\rangle_B$$

Se questo stato non è separabile è chiamato stato entangled [16].

3.2.4 Paradosso EPR

Il punto centrale del fenomeno dell'entanglement è che la teoria quantistica prevede che l'interazione tra le due particelle avvenga istantaneamente, qualunque sia la distanza che le separa. Questa sorta di azione istantanea a distanza è stata per lungo tempo fonte di imbarazzo per gli scienziati in quanto sembra violare i principi della teoria della relatività, in particolare il cosiddetto "principio di località" per cui ciò che avviene nel luogo A non può avere alcuna relazione con quanto accade nel luogo B se A e B sono separati da una distanza l tale che $l > c\Delta t$ (eventi oltre il cono di luce) ed il "principio di causalità" secondo il quale nessuna trasformazione relativistica può capovolgere la relazione tra causa ed effetto (ovviamente la causa precede sempre l'effetto).

Il paradosso di Einstein, Podolsky e Rosen (EPR) è un esperimento ideale, proposto dagli autori nel 1935, pensato allo scopo di dimostrare che la teoria quantistica era fondamentalmente incompleta. Le leggi della meccanica quantistica stabiliscono che la funzione d'onda determina le probabilità associate all'esito di un esperimento e che la funzione d'onda stessa contiene tutta l'informazione possibile sullo stato quantistico del sistema in esame. Einstein ed altri scienziati ritenevano invece che le previsioni della meccanica quantistica fossero corrette, ma solo come risultato di distribuzioni statistiche di altre proprietà sconosciute associate alle particelle. L'esperimento si basa sulla apparente contraddizione tra i principi relativistici di località e di causalità ed il fenomeno dell'entanglement per arrivare alla conclusione che dovevano esserci delle "variabili nascoste", non previste dalla meccanica quantistica stessa.

Si consideri un sistema composto da due particelle, distanti nello spazio, che sia nello stato entangled esposto nel precedente paragrafo:

$$|\Psi,t\rangle = 1/\sqrt{2}|1,\updownarrow\rangle|2,\updownarrow\rangle + 1/\sqrt{2}|1,\leftrightarrow\rangle|2,\leftrightarrow\rangle$$

Al tempo t si sottopone il fotone 1 che si trova nella regione dello spazio A ad una misura di polarizzazione piana lungo la verticale. Si supponga che il fotone superi il test. Per quanto visto l'effetto della misura è quello di ridurre lo stato del sistema, al tempo $t+dt$ a:

$$|\Psi,t+dt\rangle = |1,\updownarrow\rangle|2,\updownarrow\rangle$$

A questo punto l'osservatore che si trova in A e che ha eseguito la misura ha la certezza che il fotone 2 supererebbe un analogo test di polarizzazione verticale, senza bisogno di compiere alcuna ulteriore azione. In altre parole immediatamente dopo la misura in A del fotone 1, il fotone 2 possiede un elemento di realtà fisica (la polarizzazione verticale) che prima non aveva. Il paradosso EPR si basa sul principio di località per affermare che l'azione sul fotone 1 nella regione A non può aver creato questo elemento di realtà sul fotone 2. Di conseguenza è possibile concludere che il fotone 2 doveva possedere la proprietà di superare con certezza un test di polarizzazione verticale anche prima ed indipendentemente dalla misura sul fotone 1 (e quindi che la teoria quantistica è incompleta) [1].

3.2.5 IL TEOREMA DI BELL

Il Paradosso Einstein-Podolsky-Rosen presume il realismo locale, ossia le nozioni intuitive che gli attributi delle particelle abbiano valori definiti indipendentemente dall'atto di osservazione, e che gli effetti fisici abbiano una velocità di propagazione finita. Bell ha dimostrato che il realismo locale impone delle restrizioni su certi fenomeni, che non sono richieste dalla meccanica quantistica. Queste restrizioni sono indicate sotto il nome di disuguaglianza (o teorema) di Bell [17].

Nella formulazione del paradosso "EPR", abbiamo visto che assume un ruolo fondamentale lo stato entanglement. La sovrapposizione che rende lo stato non fattorizzabile in due componenti separabili fa si che le particelle entangled restino sempre correlate. Il dibattito sulla realtà fisica dei sistemi quantistici e sulla validità della meccanica quantistica divento' oggetto dell'investigazione sperimentale dopo la formulazione, nel 1964, della disuguaglianza di Bell. In breve, Bell mostro' che i principi di realismo e località conducono a disuguaglianze testabili sperimentalmente in disaccordo con le predizioni della meccanica quantistica.

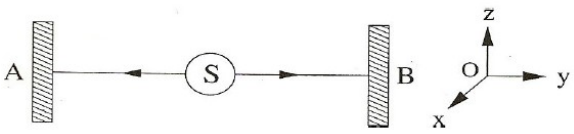

Figura 3: schema EPR idealizzato da Bohm

Di seguito riportiamo una prima formulazione della disuguaglianza. Consideriamo uno sperimentatore C, che chiamiamo Charlie, in grado di preparare due particelle e di ripetere numerose volte la stessa procedura sperimentale. Dopo la preparazione, riprendendo lo schema EPR nella formulazione di Bohm (Figura 3), Charlie invia una particella verso Alice e l'altra verso Bob. Una volta ricevuta la sua particella, Alice puo' scegliere, in maniera del tutto casuale, di effettuare uno tra due diversi processi di misura. Le misure riguardano due osservabili, che indichiamo rispettivamente con PQ e PR, corrispondenti a due grandezze fisiche dicotomiche (per esempio la componente di spin o di polarzzazione lungo una direzione con possibili valori +1 e −1). Supponiamo che Alice ottenga il valore Q se misura l'osservabile PQ ed il valore R se misura l'osservabile PR. Applicando il principio di realtà, assumiamo che Q ed R siano proprietà oggettive della particella di Alice, che si limita a rivelare tali proprietà attraverso le sue misure. Similmente, supponiamo che Bob sia in grado di misurare una delle due proprietà, PS o PT, rivelando le quantità S e T oggettivamente esistenti (elementi di realtà nel linguaggio EPR) e con possibili valori ±1. Anche Bob, come Alice, non decide precedentemente quale proprietà misurare, ma aspetta l'arrivo della particella e, solo allora, sceglie casualmente. Applicando, invece, il principio di località, imponiamo che i processi di misura di Alice e Bob avvengano allo stesso tempo ovvero siano causalmente disconnessi. Perciò, la misura effettuata da Alice non puo' disturbare il risultato di Bob e viceversa, poiché un'influenza fisica non puo' propagarsi piu' velocemente della luce. Consideriamo allora la seguente quantità:

$$QS + RS + RT - QT$$

notando molto semplicemente che:

$$QS+RS+RT-QT=(Q+R)S+(R-Q)T \quad (2.15)$$

Poichè R,Q = {+1, -1} segue che o (Q+R)S = 0 oppure (R−Q)T = 0 ma le due quantità non possono annullarsi insieme. In ogni caso, è facile mostrare dalla (2.15) che QS + RS + RT − QT = ±2. Chiamiamo ora p(q, r, s, t) la probabilità che, prima di effettuare le misure, il sistema sia nello stato con Q = q, R = r, S = s e T = t. Dunque, denotando con E(·) il valore medio di una quantità, abbiamo:

$$\begin{aligned} E(QS+RS+RT-QT) &= \sum_{qrst} p(q,r,s,t)(qs+rs+rt-ql) \\ &\leqslant \sum_{qrst} p(q,r,s,t)\,x2 = 2 \end{aligned}$$

Ma vale anche:

$$\begin{aligned} E(QS+RS+RT-QT) &= \sum_{qrst} p(q,r,s,t)\,qs + \sum_{qrst} p(q,r,s,t)rs \\ &\quad + \sum_{qrst} p(q,r,s,t)rt - \sum_{qrst} p(q,r,s,t)qt \\ &= E(QS)+E(RS)+E(RT)-E(QT) \end{aligned}$$

Comparando la (2.16) con la (2.17) otteniamo la disuguaglianza di Bell :

$$E(QS)+E(RS)+E(RT)-E(QT)\leqslant 2 \quad (2.18)$$

Dunque, ripetendo più volte l'esperimento, Alice e Bob possono determinare tutte le quantità alla sinistra della disequazione (2.18). Ad esempio, dopo aver raccolto i dati, possono osservare tutti gli esperimenti in cui Alice ha misurato PQ e Bob ha misurato PS. Moltiplicando i risultati ottengono un valore di QS per ogni esperimento e, calcolando il valore medio sul campione in esame, possono valutare E(QS) con una precisione limitata esclusivamente dal numero di esperimenti effettuati. Notiamo che la disuguaglianza di Bell espressa dalla (2.18) è un risultato assolutamente generale che potrebbe essere applicato a coppie di galassie, a coppie di stelle ed a qualunque coppia di oggetti macroscopici, una volta che siano state definite le grandezze fisiche dicotomiche da misurare. è anche in questo carattere universale che sta il fascino della disuguaglianza di Bell. Tuttavia, quando consideriamo sistemi accoppiati quantistici quali particelle entangled, applicando i principi della meccanica quantistica, la disuguaglianza di Bell risulta violata. Supponiamo infatti che, nel nostro esperimento, Charlie prepari un sistema quantistico di due particelle entangled nello stato:

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}},$$

da cui si ottengono le seguenti relazioni:

$$\langle QS\rangle = \frac{1}{\sqrt{2}}\,;\langle RS\rangle = \frac{1}{\sqrt{2}}$$
$$\langle RT\rangle = \frac{1}{\sqrt{2}}\,;\langle QT\rangle = \frac{1}{\sqrt{2}}$$

Cosi' si ottiene:

$$\langle QS\rangle+\langle RS\rangle+\langle RT\rangle-\langle QT\rangle = 2\sqrt{2} \quad (2.24)$$

Dunque la disuguaglianza di Bell (2.18) è violata: la meccanica quantistica risulta incompatibile con la naturale filosofia del realismo locale. Allora, se è valida la meccanica quantistica, almeno una delle due assunzioni originarie dell'articolo EPR (principi di realtà e località) non è corretta e va rigettata. L'importanza dell'opera di John Bell sta nell'aver ricondotto ed espresso l'incompatibilità tra la

meccanica quantistica. ed il realismo locale, introdotta ed evidenziata da EPR, attraverso una disuguaglianza verificabile sperimentalmente. Prima della formulazione della disuguaglianza di Bell (1964), fu anche proposto che la meccanica quantistica, incompleta secondo EPR, fosse completata intoducendo certe variabili nascoste. L'idea fu che la misura è in realtà un processo deterministico che appare invece probabilistico, semplicemente perché alcuni gradi di libertà (le variabili nascoste) non sono noti con precisione [19].

Esposte in maniera riassuntiva le teorie fisiche poste alla base della crittografia quantistica, passiamo ora alla descrizione pratica di come queste vengono impiegate nella costruzione dei protocolli di crittografia quantistica e di come si riesca ad ottenere, alla fine di tutto, una chiave sicura.

Gli schemi QKD possono essere classificati principalmente in due tipi: schemi di preparazione e misura e schemi basati sull'entanglement quantistico. Nei primi, il trasmettitore, Alice, prepara i segnali quantistici (utilizzando, ad esempio, una sorgente laser), secondo le sue basi e i valori dei bit e li invia attraverso un canale quantistico al ricevitore, Bob, che le misure in ricezione. Nel secondo tipo di schema, un fonte entanglement emette coppie di segnali entangled, che sono poi misurati in certe basi scelte da Alice e Bob separatamente. C'è una differenza importante in termini di sicurezza fra i segnali emessi nei due casi. Nel caso degli schemi di preparazione e misura, il segnale emesso da Alice (ad esempio, una sorgente debole coerente di stato) è base dipendente, il che significa che lo stato coerente del segnale corrispondente ad una base che è meccanicamente quantistica diversa da quella delle altre basi.

In questa tesina vengono presentati un protocollo basato sullo schema di preparazione e misura, il BB84, e uno basato sull'entanglement quantistico, l'E91 [5].

3.3 PROTOCOLLO BB84

Attualmente, il più diffuso protocollo, sia dal punto di vista teorico che dal punto di vista sperimentale e realizzativo, è il protocollo BB84; ideato da C. Bennett e G. Brassard nel 1984 **[mettere un riferimento all'articolo]**(da cui appunto il nome del protocollo BB84) questo protocollo è stato il primo proposto ed implementato fisicamente a livello di physical layer [4].

Nel seguito vengono illustrati i punti principali di questo protocollo.

Ricordiamo che la crittografia quantistica non è utilizzata per trasmettere dati di informazione, ma viene utilizzata solo per produrre e distribuire una chiave $K=\{0,1\}^N$ con la quale, attraverso un qualsiasi algoritmo di cifratura, cifrare e decifrare un messaggio che può poi essere trasmesso su una comunicazione standard. Lo schema usato dal protocollo BB84 utilizza la trasmissione di singoli fotoni polarizzati come caratteristica fisica per la trasmissione dei bit della chiave *K*. Le polarizzazioni dei fotoni sono quattro e sono raggruppate in due diverse basi non ortogonali tra di loro. Le due basi non ortogonali sono generalmente descritte come segue:

- La polarizzazione orizzontale (0°) e quella verticale (+90°) formano la base ⊕ e indichiamo gli stati della base con la classica notazione $|0\rangle$ e $|1\rangle$.
- Le polarizzazioni diagonale (+45°) e (+135°) formano la base ⊗. I due stati di base differenti sono $|+\rangle$ e $|-\rangle$ con

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

I bit di informazione, preso da un generatore di numeri casuali, sono associati alla base come mostrato nella Tabella 1.

Bit	⊕	⊗
0	$\|0\rangle = a_{00}$	$\|+\rangle = a_{10}$
1	$\|1\rangle = a_{01}$	$\|-\rangle = a_{11}$

Tabella 1: associazione tra bit di informazione e le basi nel protocollo BB84

Il protocollo BB84 procede nei seguenti passi:

1) Trasmissione Quantistica (prima fase)
   a) Alice possiede una stringa casuale di bit $d \in \{0,1\}^n$, e una stringa casuale di n basi $b \in \{\oplus,\otimes\}^n$, con n>*N*. *N* è la lunghezza della chiave finale.
   b) Un fotone in uno stato quantico a_{ij} viene preparato da Alice per ogni b_i in b e d_j in d come nella
      Tabella 1, e lo invia a Bob sul canale quantistico.
   c) In base alla scelta aleatoria della base, ⊕ oppure ⊗, Bob misura ogni aij ricevuto. Le misure di Bob producono una stringa $d' \in \{0,1\}$n, mentre le sue scelte delle basi formano una sequenza di basi $b' \in \{\oplus,\otimes\}^n$.
2) Discussione Pubblica (seconda fase)
   a) Per ogni bit d_i in d
      i) Attraverso il canale classico, Alice invia il valore di b_i a Bob
      ii) In risposta ad Alice, Bob afferma se ha usato la stessa base per la misurazione.
      I valori di d_i e d_i' vengono scartati se $b_i \neq b_i'$.
   b) Alice forma un sottoinsieme aleatorio dei bit rimanenti in d e comunica i loro valori a Bob sul canale classico (su internet per esempio). Se il risultato delle misure di Bob per uno di questi bit non corrisponde ai valori diffusi, l'intercettazione (Eve) viene rilevata e la comunicazione viene interrotta.
   c) La chiave segreta comune $K=\{0,1\}^N$ è la stringa di bit rimanenti in d una volta che i bit divulgati nel passaggio 2b) sono atati cancellati.

Al fine di comprendere il protocollo BB84 è importante introdurre come viene misurato un qubit nel campo della fisica quantistica, se abbiamo un qubit come $|qubit\rangle = a|b\rangle + c|d\rangle$ la misura di questo stato nelle basi $\{|b\rangle, |d\rangle\}$ produce lo stato $|b\rangle$ con probabilità $|a|^2$ e lo stato con probabilità $|c|^2$, per le probabilità vale ovviamente la relazione $|a|^2+|c|^2=1$ ($|a|^2$ è il modulo quadro dell'ampieza di a). Quindi, misurando con la base non corretta questa produce un risultato aleatorio, come previsto dalla teoria quantistica. Così, se Bob sceglie la base ⊗ per misurare un fotone nello stato $|1\rangle$, il risultato sarà 0 o 1 con uguale probabilità poichè $|1\rangle = 1/2(|+\rangle - |-\rangle)$; se invece viene scelta la base ⊕, il risultato sarebbe con certezza 1 poichè $|1\rangle = 1|1\rangle + 0|0\rangle$.

L'individuazione di una possibile spia (Eve) viene svolta nel punto 2b). Laddove le basi di Alice e Bob sono identiche (cioè $b_i=b_i'$), l'idea è che, i bit corrispondenti deve corrispondere (cioè $d_i=d_i'$). In caso contrario, vi è un disturbo esterno oppure c'è del rumore nel canale quantistico. Poichè al ricevitore non vi è modo di distinguere tra disturbi provocati dal canale e quelli dovuti ad Eve, si ipotizza che tutti i disturbi siano causati da Eve [13].

3.4 PROTOCOLLO E91

Un altro protocollo di QKD è stato proposto da Arthur Ekert nel 1991, esso si basa sul paradosso EPR e sul fenomeno dell'entanglement.

Lo schema proposto da A. Ekert utilizza coppie di fotoni entangled. Questi possono essere creati da Alice, da Bob, o da una terza fonte separata, che potrebbe essere anche Eve. I fotoni sono distribuiti in modo che ad Alice e a Bob giunga un fotone per ogni coppia di fotoni entangled e l'associazione base-polarizzazione-bit è la stessa del protocollo BB84. Alice e Bob, attraverso filtri polarizzatori e been splitter polarizzatori (PBS), effettuano misure della polarizzazione in basi aleatorie e indipendenti tra di loro.

Vi sono due varianti del protocollo E91. Nella prima variante, strettamente derivata da BB84, vengono utilizzate le polarizzazioni rettilinea e diagonale, dopo la trasmissione la sorgente annuncia pubblicamente le basi usate ed Alice e Bob scartano immediatamente i bit in cui le misurazioni sono state fatte nella base incompatibile (come avviene per il protocollo BB84). Lo schema si basa principalmente sulla proprietà di perfetta correlazione presente in due particelle entangled. Come abbiamo visto nei capitoli precedenti, gli stati entangled sono perfettamente correlati: se Alice e Bob misurano entrambi le polarizzazioni delle loro particelle, questi ottengono sempre la stessa risposta con il 100% di probabilità. Lo stesso vale se Alice e Bob misurano qualsiasi altra coppia di polarizzazioni complementare (ortogonali). Tuttavia, i singoli risultati sono completamente aleatori ed è impossibile per Alice o per Bob prevedere se l'uno o l'altra userà una polarizzazione verticale oppure una orizzontale. Ogni tentativo di l'intercettazione da parte di Eve, dunque, distrugge queste correlazioni producendo una diversa risposta da parte delle misure sui fotoni compiute da Alice e da Bob, le polarizzazioni diventano scorrelate e il grado di scorrelazione da o meno la presenza di Eve. A patto che la sorgente sia affidabile, ovvero non sia in mano ad Eve, tale protocollo è altrettanto sicuro del BB84.

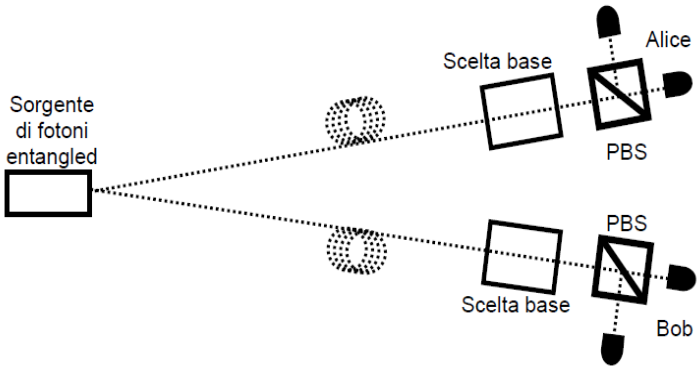

Figura 4: rappresentazione schematica della distribuzione dei fotoni entangled

La seconda variante del protocollo si appoggia invece alla disuguaglianza di Bell: i fotoni entangled vengono polarizzati in modo aleatorio in tre basi non ortogonali differenti (rettilinea, diagonale e circolare). In questo caso i risultati delle misurazioni fatte lungo le basi non coincidenti non vengono scartati a priori ma servono per fare un test basato sulla disuguaglianza di Bell, ad esempio per controllare che i due fotoni ricevuti siano realmente entangled. Un eavesdropper non può ottenere informazioni da un singolo fotone in transito, semplicemente perché esso non contiene alcuna informazione: l'informazione stessa "diviene in essere" solo dopo che essi sono stati misurati dai legittimi destinatari [18].

3.5 INFORMATION RECONCILIATION

Nel sistema QKD, abbiamo visto che si possono verificare degli errori nella chiave iniziale, detta sifted key, causati del rumore di sistema o dovuti alla presenza di Eve durante la trasmissione sul canale quantistico. Di conseguenza un sistema di criptografia quantistica deve includere delle procedure, raggruppate in una fase chiamata information reconciliation, il cui obbiettivo è quello di rimuovere questi errori attraverso lo scambio di messaggi su un canale pubblico autenticato. La sola condizione che deve essere soddisfatta dal canale pubblico è che l'informazione pubblica non possa essere modificata da Eve [Practical Error-Correction Procedures in Quantum Cryptography]. Le informazioni trasmesse sul canale pubblico sono aperte anche all'intercettatore (Eve) e questo diminuisce la privacy della chiave. Quindi, la quantità di informazione rivelata è una misura di efficienza per i diversi protocolli di riconciliazione.

3.5.1 LIMITI TEORICI DI EFFICIENZA

Ipotizziamo che le chiavi iniziali stabilite da Alice e Bob attraverso il canale quantistico siano A e B di lunghezza N e che vi sono degli errori tra A e B causati da disturbi del sistema o da intercettazioni. Sia infine ε il tasso di errore. Dunque l'entropia condizionata tra A e B è

$$\mathrm{H}(A|B) = NH(\epsilon)$$

Dove $H(\cdot)$ rappresenta l'entropia definita da Shannon, in questo caso di una sorgente binaria

$$H(x) = -x\log x - (1-x)\log(1-x)$$

Quindi, in media Bob ha bisogno di ottenere almeno $NH(\varepsilon)$ bit di informazioni da Alice per completare la riconciliazione e questo è la minima quantità di informazioni necessaria per la riconciliazione. Per un certo protocollo di riconciliazione P, tutte le informazioni scambiate sul canale pubblico formano una stringa di bit E la quale costituisca una potenziale fonte di informazione per Eve in quanto puo' essere intercettata essendo trasmessa sul canale pubblico. Per semplicità, supponiamo che alla fine Bob riesca a conciliare la sua chiave B in A. Dunque l'efficienza del protocollo P può essere definita come

$$\eta_P = 1 - \frac{I(A|E)}{N}$$

dove $I(A \mid E)$ è l'informazione che Eve può ottenere su una data stringa E. Di conseguenza, il limite teorico di efficienza per un protocollo di riconciliazione è

$$\eta_{Th} = 1 - H(\epsilon)$$

Per maggiori informazioni sui limiti teorici vedere [20].

Fatta questa breve introduzione agli aspetti teorici riguardanti l'efficienza di un protocollo di riconciliazione, vediamo ora in dettaglio due protocolli di riconciliazione: il protocollo a cascata e il protocollo BCH.

3.5.2 PROTOCOLLO A CASCATA

Il protocollo a Cascata è basato sul BBBSS [**mettere il riferimento al documento**. L'idea di base di entrambi i protocolli è quella di trovare gli errori dividendo la stringa di in blocchi di bit e di confrontare i loro bit di parità. Ma il protocollo a Cascata, a differenza del BBBSS, tiene cinto in un registro del modo in cui i blocchi vengono fatti e dei loro rispettivi bit di parità migliorando in questo modo l'efficienza. Prima di descrivere il protocollo a Cascata, spieghiamo prima il processo binario che verrà poi utilizzato nel protocollo Cascata.

Il processo binario consiste nelle seguenti tre fasi: quando le stringhe A e B hanno un numero dispari

di errori, (1) Alice dividere A in due metà, e invia a Bob la parità della sua prima metà, (2) Bob dividere in B allo stesso modo e confronta il bit di parità con quello di Alice per determinare se il bloco formato dalla prima metà contiene numero dispari di errori, in ultima (3) si applicano questi due passi ripetutamente fino a quando non viene trovato un errore.

Descritto brevemente il processo binario diamo ora una descrizione operativa del protocollo a Cascata: il protocollo procede in diversi passi e il numero di passaggi è determinato da Alice e Bob prima dell'esecuzione e la scelta è legata dal tasso di errore ε (che potrebbe essere determinato empiricamente sacrificando parte dei bit della sifted key). Siano $A = A_1, A_2, \ldots, A_N$, e $B = B_1, B_2, \ldots, B_N$ (con $A_i, B_i \in \{0, 1\}$) essere le stringhe di Alice e Bob rispettivamente. Nel passo 1, Alice e Bob scelgono k_1 e dividere la loro stringa in blocchi di k_1 bit. I bit la cui posizione è in $K^1_v = \{l \mid (v - 1)k_1 < l \leq vk_1\}$ dal blocco v nel passaggio 1. Alice invia la parità di tutti i suoi blocchi a Bob. utilizzando un processo chiamato BINARIO che sarà spiegato nel seguito, Bob corregge un errore in cui ogni blocco di parità differisce da quello di blocco corrispondente di Alice. a questo punto, tutti i blocchi di Bob hanno un numero pari di errori (possibilmente a zero).

Ad ogni passo i (>1), Alice e Bob scelgono k_i e una funzione aleatoria $f_i : [1..n] \rightarrow [1..\lceil n/k_i \rceil]$. I bit la cui posizione è in $K^i_j = \{l \mid f_i(l) = j\}$ formano il blocco j al passo i. Alice invia a Bob

$$a_j = \bigoplus_{l \in K^i_j} A_l$$

per ogni j ($1 \leq j \leq \lceil n/k_i \rceil$). Bob calcola il suo b_j-esimo nel allo stesso modo e lo confronta con l'a_j-esimo. Per ogni $b_j \neq a_j$, Alice e Bob avviare una sezione di trace-back che viene descrita nel modo seguente.

Alice e Bob eseguono il processo binario sul blocco definito da K^i_j. Bob trova $l \in K^i_j$ tale per cui $B_l \neq A_l$ e lo corregge. Tutti i blocchi K^u_v per $1 \leq u < i$ tale che $l \in K^u_v$ avrà quindi un numero dispari di errori. Sia K l'insieme di questi blocchi. Alice e Bob possono ora scegliere il più piccolo dei blocchi in K e usare il processo binario per trovare un altro errore. Sia l' essere la posizione di questo errore nelle stringhe A e B. Dopo aver corretto $B_{l'}$, Bob può determinare l'insieme B formato dai blocchi contenenti $B_{l'}$ per ogni passaggio da 1 a i. Egli può anche determinare l'insieme K' dei blocchi con un numero dispari di errori calcolandolo come

$$\mathsf{K}' = (\mathsf{B} \cup \mathsf{K}) \setminus (\mathsf{B} \cap \mathsf{K})$$

Se $\mathsf{K}, \neq \varnothing$, allora Bob trova un'altra coppia di errori nello stessa modo. Questo processo viene ripetuto fino a che non si ha $\mathsf{K}, = \varnothing$. Dopo di questo, Alice e Bob continuano a confrontare i bit di parità al passo i. Ogni volta che un bit di parità differisce, loro iniziano una nuova sezione. Il passo i termina quando tutti i bit di parità sono stati controllati. A questo punto, tutti i blocchi dal passo 1 al passo i dovrebbero contenere un numero dispari di errori (possibilmente zero).

Le dimensioni dei blocchi e il numero di passaggi che il protocollo a Cascata sceglie variano in base al valore del tasso di errore ε. Nel protocollo a Cascata originale [26], l'autore sceglie come dimensione del blocco $k_1 = 0.73/\varepsilon$ per il primo passo, mentre nei seguenti passaggi sceglie $k_i = 2k_{i-1}$ ed esegue 4 passaggi.

3.5.3 Protocollo a Cascata migliorato

Ora esponiamo un protocollo che migliora le prestazioni del protocollo a Cascata modificandone sia la sua strategia che i suoi parametri.

3.5.3.1 MODIFICA DELLA STRATEGIA

L'idea è quella di mantenere un record di tutti i blocchi prodotti durante la procedura. Così, dal secondo passaggio, ogni volta che un nuovo errore viene trovato, risalendo nel record della storia passata dei blocchi,

l'errore può essere trovato più agevolmente. Pertanto il numero di bit scambiati sul canale pubblico si riduce, e di conseguenza l'efficienza migliora.

Infatti, nell'esecuzione del protocollo a Cascata, possono essere registrate e utilizzate maggiori informazioni. Nel protocollo a Cascata, solo i blocchi divisi all'inizio di ogni passaggio vengono registrati. Ma durante il processo binario, vengono generati tanti piccoli sotto-blocchi. Se tutti questi piccoli sotto-blocchi vengono registrati, poi la sezione di trace-back può effettuare una ricerca nei blocchi più piccoli, le informazioni scambiate saranno minori e il protocollo sarà ancora più efficiente.

Adesso diamo una spiegazione del protocollo a Cascata modificato con l'idea appena esposta. Il protocollo procede nei seguenti passaggi: nel passo 1, Alice e Bob scelgono k_1 e dividono le loro stringhe in $\lceil n/k_i \rceil$ blocchi in modo tale che, di conseguenza, ogni blocco contenga k_1 bit. Questi blocchi sono formati dai bit le cui posizioni sono in $K^1_{(v-1)k_1+1,vk_1} = \{l \mid (v-1)k_1 < l \leq vk_1\}$, con $1 \leq v \leq \lceil n/k_i \rceil$. Qui utilizziamo una sottoscrizione per rappresentare le posizioni di inizio e di fine di un blocco. A questo punto, Alice spedisce a Bob il bit di parità di ogni blocco e quando Bob trova un blocco $K^1_{(v'-1)k_1+1,v'k_1}$ il cui bit di parità risulta essere diverso da quello di Alice, entrambi danno inizio al processo binario su quel blocco. Il primo passo del processo binario è modificato come segue: viene diviso il blocco $K^1_{(v'-1)k_1+1,v'k_1}$ in due sotto-blocchi $K^1_{(v'-1)k_1+1,(v'-1)k_1+\lceil k_1/2 \rceil}$ e $K^1_{(v'-1)k_1+\lceil k_1/2 \rceil+1,v'k_1}$ e viene scambiato il bit di parità del primo sotto-blocco. Viene cancellato il blocco $K^1_{(v'-1)k_1+1,v'k_1}$ dalla memoria dei blocchi e vengono aggiunti due nuovi blocchi $K^1_{(v'-1)k_1+1,(v'-1)k_1+\lceil k_1/2 \rceil}$ e $K^1_{(v'-1)k_1+\lceil k_1/2 \rceil+1,v'k_1}$.

I passi seguenti sono simili al protocollo a Cascata originale, ma con il processo binario modificato. Quindi la memoria dei blocchi è in continuo aggiornamento durante l'esecuzione del protocollo e la dimensione dei blocchi tende ad essere via via sempre più piccola.

3.5.3.2 PARAMETRI DI OTTIMIZZAZIONE

I parametri utilizzati nel protocollo, ovviamente, influenzano la sua efficienza. Se la dimensione del blocco è molto piccola si ha che il numero dei blocchi è molto elevato e vi è la possibilità che molti di questi blocchi non contengano errori. Lo scambio dei bit di parità di blocchi che non hanno alcun errore non aiuta senza alcun dubbio la ricerca degli errori stessi, ma contribuisce solamente a diminuire l'efficienza del protocollo. D'altra parte, se la dimensione del blocco è relativamente alta, si ha che un singolo blocco potrebbe avere più di un errore. Questo, di conseguenza, richiederebbe molti passi per trovare tutti gli errori e una dimensione maggiore di un blocco causerebbe un aumento dei cicli del processo binario causando, di conseguenza, una diminuzione dell'efficienza del protocollo.

La migliore situazione si ha quando ogni blocco ha un solo errore. Di conseguenza la dimensione del blocco deve essere scelta in base al tasso di errore ε con lo scopo di massimizzare la probabilità di avere esattamente un errore in un blocco. Vediamo in dettaglio gli effetti di tale parametro sulla dimensione del blocco.

Supponiamo che il tasso di errore sia ε prima che venga eseguito un passaggio, se la dimensione del blocco è k, il numero di errori in un blocco è una variabile aleatoria binomiale X di parametri k e ε, $X \sim Bin(k,\varepsilon)$, dunque si ha

$$P(X=e)=\binom{k}{e}\epsilon^e(1-\epsilon)^{k-e}$$

La probabilità che un blocco abbia un numero dispari di errori è

$$P_{odd}=\sum_{i=1}^{\infty}P(2\mathrm{i}-1)=\frac{1-(1-2\epsilon)^k}{2}$$

In questo passaggio, vi sono N/k blocchi in totale. Dunque, il valore atteso del numero di errori può essere trovato (senza trace-back) come

$$\frac{N}{k} P_{odd}$$

La proporzione di questo numero sul totale del numero di errori è

$$\frac{\frac{N}{k} P_{odd}}{N\epsilon} = \frac{1-(1-2\epsilon)^k}{2\epsilon k}$$

Per facilitare la discussione, introduciamo due parametri: sia $k_1 = \alpha/\varepsilon$ la dimensione del blocco al primo passaggio e $k_i = \beta k_{i-1}$ la dimensione del blocco al passaggio i-esimo. Secondo l'analisi appena fatta, la Tabella 2 mostra la proporzione di errori che possono essere trovati al variare del tasso di errore ε e della dimensione del blocco (parametro α). Possiamo notare che mentre α rimane tra lo 0.5 e l'1.5 (dunque la dimensione dei blocchi varia tra $0.5/\varepsilon$ e $1.5/\varepsilon$) circa il 30% ~ 75% degli errori può essere trovato nei blocchi nuovi divisi in questo passaggio.

α		0.5	0.75	1	1.25	1.5
ε	0.01	0.6458	0.5202	0.4337	0.3680	0.3172
	0.05	0.6513	0.5294	0.4392	0.3712	0.3192
	0.1	0.6723	0.5416	0.4463	0.3754	0.3216
	0.15	0.6954	0.5546	0.4536	0.3795	0.3239
	0.2	0.7211	0.5685	0.4611	0.3836	0.3261
	0.25	0.7500	0.5833	0.4688	0.3875	0.3281

Tabella 2: la proporzione degli errori che possono essere trovati al variare di ε e α

Passiamo ora al considerare gli errori trovati nella sezione di trace-back. Al secondo passaggio, dividiamo gli errori trovati in questo passaggio in due insiemi: gli errori trovati con la procedura binaria sui blocchi nuovi divisi in questo passaggio formano l'insieme S_2, mentre gli errori trovati dal trace-back nei vecchi blocchi divisi al passaggio 1 formano l'insieme S_1. Come descritto nel protocollo a Cascata, ogni volta che viene trovato un errore l in S_2, un altro corrispondente errore l' potrebbe essere scoperto.

Altrimenti, dopo che l'errore l è stato trovato, il vecchio blocco nel passaggio 1 che contiene l'errore l dovrebbe avere un numero dispari di errori. Questo è, prima di trovare l'errore l, il blocco che ha un numero pari di errori. Questo contraddice con la condizione nel protocollo a Cascata che, alla fine di ogni sezione di trace-back al passaggio i-esimo (con $i > 1$), tutti i blocchi prima del passaggio i-esimo hanno un numero dispari di errori.

Quindi abbiamo che

$$|S_1| \geqslant |S_2|$$

E il numero totale di errori può essere trovato al passaggio 2

$$|S_1| + |S_2| \geqslant 2|S_2|$$

Secondo le analisi mostrate in modo sintetico nella Tabella 2, gli errori in $S2$ sono circa 30% ~ 75% degli errori rimasti dal passaggio 1. Di conseguenza, aggiungendo $S1$ e $S2$ insieme, più del 60% dei rimanenti errori può essere trovato nel passaggio 2 totalmente. In fatti, considerando che trovare un errore potrebbe indurre alcuni tipi di reazione a catena, e più di un errore potrebbe essere trovato, questa proposizione potrebbe essere anche più alta in pratica.

Gli esperimenti mostrati nella Tabella 3 confermano questa congettura. Per vari tassi di errore ε e di

parametri α , abbiamo comparato il restante numero di errori dopo ogni passaggio. Nell'esperimento, la lunghezza delle chiavi è di 104 bit e sono generate in modo aleatorio da un computer. Il risultato è una media fatta su 100 esperimenti [20].

$\alpha = 0.5$

ε	initial	pass 1	pass 2	pass 3	pass 4
0.01	102.41	38.08	0.44	0	0
0.05	494.61	171.12	0.44	0.04	0
0.1	1001.27	326.10	0.42	0	0
0.15	1497.90	403.06	0.34	0	0
0.2	2001.07	401.30	0.06	0	0
0.25	2498.07	623.02	0.22	0	0

$\alpha = 1.0$

ε	initial	pass 1	pass 2	pass 3	pass 4
0.01	99.02	56.0	2.44	0.02	0
0.05	501.81	282.48	1.80	0.02	0
0.1	994.37	545.18	2.08	0	0
0.15	1494.68	836.42	2.20	0	0
0.2	2008.04	1082.86	1.28	0	0
0.25	2497.39	1323.94	1.28	0	0

$\alpha = 1.5$

ε	initial	pass 1	pass 2	pass 3	pass 4
0.01	98.99	66.62	5.56	0.04	0
0.05	501.17	341.72	8.42	0.04	0
0.1	996.76	674.06	6.40	0.02	0
0.15	1499.67	1013.28	6.98	0	0
0.2	1999.09	1382.58	8.64	0	0
0.25	2498.73	1674.68	4.16	0	0

Tabella 3: numero di errori rimanenti dopo ogni passo del protocollo Cascata (dimensione della chiave N = 10000, il risultato è una media su 100 esperimenti)

3.5.4 Codici BCH

Un metodo naturale per correggere gli errori della sifted key (raw key o chiave grezza) si basa sull'utilizzo di classici codici a correzione d'errore. Ribadiamo qui che l'efficienza di tale codice non può essere valutata senza tener conto della parte quantistica del protocollo, poiché un codice classico applicato nella crittografia quantistica non solo corregge gli errori, ma cambia sostanzialmente anche la probabilità condizionata per l'individuazione di Eve sulla sifted key. In ciò che segue, si discute della correzione della chiave attraverso i codici di Bose-Chaudhuri-Hocquenghem (BCH) [24, 25] i quali costituiscono una vasta classe. Questi codici possono essere utilizzati per correggere alcuni errori sul singolo bit per blocco e questo li rende dinamici a seconda del tasso di errore Q stimato. In primo luogo, forniamo le definizioni minime richiesti per analizzare i codici BCH di correzione di t-errore.

Il campo di Galois $GF(2^n)$ è lo spazio vettoriale delle parole binarie di lunghezza n con le operazioni aritmetiche modulo 2. Un codice lineare $\mathfrak{E}$ estende un sottospazio lineare in $GF(2^n)$. Un codice viene chiamato ciclico se il fatto che $\mathbf{c} = (c_0, c_1, ..., c_{n-1})$ con $(c_i = 0, 1)$ è una parola di codice implica che $\mathbf{c'} = (c_1, ..., c_{n-1}, c_0)$ è anch'essa una parola in codice. Un vettore di $GF(2^n)$ è rappresentato in modo comodo

dal polinomio di x di grado non superiore a n-1 i cui coefficienti sono i componenti del vettore:

$$c(x)=c_0+c_1x+c_2x^2+...+c_{n-1}x^{n-1}$$

Un codice ciclico è definito da un corrispondente generatore polinomiale $g(x)$ di grado n-k (k è il numero di bit di informazione), che è un divisore del polinomio x^{n-1}-1:

$$x^{n-1}-1=g(x)h(x)$$

dove h (x) è un polinomio di controllo di parità. Un codice BCH di correzione di t-errore è costruito come segue. L'elemento primitivo α è definito un per il campo $GF(2^n)$. (Per definizione, ogni elemento del campo è una potenza dell'elemento primitivo, $\alpha=\alpha^2$ in $GF(2)$). Il generatore polinomiale $g(x)$ le cui radici formano l'insieme

$$[\alpha, \alpha^2, ..., \alpha^3, ..., \alpha^{2t}]$$

è costruito per il preimpostato t. La parola di codice di lunghezza n è definita come $n = 2^m$-1, con la scelta di un numero m. Il polinomio $f_j(x)$ di grado minimo le cui radici sono gli α^j (j = 1, 2, ..., 2t) sono trovate in $GF(2)$-$GF(2^n)$. Il generatore polinomiale $g(x)$ per un codice di lunghezza n è calcolato come il minimo comune multiplo:

$$g(x)=LCM[f_1(x), f_2(x), ..., f_{2t}(x)]$$

Viene usato per la decodifica l'algoritmo di Peterson-Gorenstein-Zierler [24, 25]. Supponiamo che la parola di codice di ingresso sia $\mathbf{c} = (c_0, c_1, ..., c_{n-1})$, l'uscita

$$\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}$$

contiene un vettore degli errori $\mathbf{e} = (e_{i_1}, ..., e_{i_v})$ rappresentato come $e(x) = e_{i_1}x^{i_1} + ... + e_{i_v}x^{i_v}$ (e_{i_l}= 0,1), e il numero v di errore sul singolo bit non è conoscuto nel caso generale. Un ciclo di decodifica è costituito da tre fasi: il calcolo di una sindrome di errore $S(x)$, la costruzione di un localizzatore di errore polinomiale $\Lambda(x)$, e il calcolo delle radici di $\Lambda(x)$, cioè la determinazione degli e_j e la correzione degli errori. Supponiamo che il vettore di codice sia $C(\alpha^j) = 0$ per diversi α^j:

$$y(\alpha^j)=e(\alpha^j)$$

introducendo

$$Y_l=e_{i_l}, \quad X_l=\alpha^{i_l}$$

per comodità, una sindrome di errore è definita componente per componenti:

$$S_1=Y_1X_1+...+Y_vX_v$$

$$.....$$

$$S_{2t}=Y_1X_1^{2t}+...+Y_vX_v^{2t}$$

Dalla definizione della sindrome di errore, esiste un'unica soluzione per questo sistema di equazioni (nonlineare nel caso generale). Il localizzatore di errore polinomiale è definito come

$$\Lambda(x)=\Lambda_vx^v+...+\Lambda_xx+1$$

con radici X_l^{-1} (l = 1, 2, ..., v), cioè

$$\Lambda(x)=(1-xX_1)...(1-xX_v)$$

Le sue radici si possono trovare se i coefficienti Λ_l sono noti. Se le S_l sono note, allora i Λ_l posso essere trovati risolvendo il seguente sistema lineare di equazioni. Da cui:

$$Y_l(X_l^{j+v}+\Lambda_lX_l^{j+v-1}+...+\Lambda_vX_l^j)=0 \quad (87)$$

la somma di (87) su tutti gli l

$$\sum_{l=1}^{v} Y_l(X_l^{j+v}+\Lambda_l X_l^{j+v-1}+...+\Lambda_v X_l^{j})=0$$

che può essere riscritta come un sistema lineare di equazioni per Λ_l impostando $j = 1, ..., v$ e usando (84):

$$\Lambda_1 S_{j-v-1}+\Lambda_2 S_{j+v-2}+...+\Lambda_v S_j=-S_{j+v}$$

$$\begin{pmatrix} S_1 ..... S_v \\ ..... \\ ..... \\ S_v ..... S_{2v-1} \end{pmatrix}\begin{pmatrix} -S_{v+1} \\ . \\ . \\ -S_{2v} \end{pmatrix}=\begin{pmatrix} -\Lambda_v \\ . \\ . \\ \Lambda_1 \end{pmatrix} \quad (90)$$

Il sistema di equazioni (90) è non degenere se v è uguale al (ancora sconosciuto) numero effettivo di errori sul singolo bit altrimenti è degenere. Di conseguenza, se il determinante calcolato per $v = t$ è zero, allora v viene ridotta di un'unità e il calcolo viene ripetuto fino a che non viene ottenuto un determinante diverso da zero per v uguale al numero di errori sul singolo bit. Poi, il sistema (90) è risolto per trovare un unico insieme di coefficienti Λ_l, e le radici del polinomio (85) sono determinate dal successivamente calcolando il suo valore per ogni elemento del campo. Da cui:

$$X_l^{-1}=\alpha^{-i_l}$$

la posizione i_l dell'errore sul singolo bit e_{i_l} da correggere si trova come l'esponente in

$$X_l \cdot \alpha^{i_l}=1$$

La procedura di correzione degli errori per mezzo dei codici BCH è costituita dai seguenti passaggi.

1. Una stima per il tasso di errore Q si ottiene da un confronto pubblico di circa la metà della sequenza trasmessa. Per la lunghezza della parola di codice preimpostata $n = 2^m$-1, il numero medio di errori sul singolo bit per parola di codice è stimato come $t = [Q \cdot n]$. La sequenza è partizionata in blocchi di dimensione n. Il generatore polinomiale $g(x)$ corrispondente a n e t è dunque costruito.

2. Alice genera parole di codice. In ogni blocco, i primi k bit trasportano informazioni, e i rimanenti $n – k$ sono utilizzati come bit di controllo. Viene annunciata l'inversione di un necessario numero di bit, la quale non cambia l'informazione private nota ad Eve poiché Alice genera parole di codice per il calcolo dei bit di controllo come funzioni di quelle informazioni. Per il codice (n, k) con generatore polinomiale $g(x)$ nel campo $GF(2^m)$, il polinomio che rappresenta n-2^m-1 bit di codifica è il resto della divisione di $i(x) \cdot x^{n-k}$

$$g(x)-c(x)=-R_{g(x)}[x^{n-k} i(x)]$$

dove

$$i(x)=i_0+i_1 x+...+i_{k-1} x^{k-1}, \quad i_l=0,1$$

è il polinomio di informazione. I bit i_l ($l = 0, ..., k - 1$) sono i primi bit del blocco nella stringa di bit inviata da Alice.

3. Bob decodifica la parola di codice ricevuta e, quindi, Alice e Bob rimuovono i bit di controllo per ogni parola di codice.

Abbiamo usato questi codici BCH per correggere gli errori nella sifed key con l'esecuzione di un ciclo. La Figura 5 mostra la corrispondente lunghezze della chiave riconciliata tracciata come funzione dei tassi di errore stimati Q. Si noti che il tasso di errore rimane finito solo dopo un solo ciclo di esecuzione; se il numero di errori sul singolo bit in una parola di codice supera il numero di errori che possono essere corretti da un codice particolare si ha come risultato che vengono generati nuovi errori

(parole errate vengono ottenute dalla decodifica). Tuttavia, non è giustificata l'esecuzione di un altro ciclo, questo perché la lunghezza della stringa viene moltiplicata per k/n dopo ogni ciclo e la chiave finale risulterebbe troppo corta.

Quando la lunghezza della parola di codice è di 127 o 255 e $Q = 6\%$, non viene lasciato nessun errore dopo l'esecuzione di un singolo ciclo di correzione degli errori (vedi Figura 6). La lunghezza della chiave corretta è di circa il 20% della lunghezza della sifted key, cioè, simile a quella ottenuta con l'esecuzione del protocollo a Cascata per $Q = 6\%$. Quando la parola di codice è di lunghezza 15, 31, o 63 bit la chiave riconciliata (cioè quella corretta) generata in un singolo ciclo contiene un numero significativo di errori sul singolo bit; anche se la sua lunghezza relativa può ammontare al 35%, ma l'efficienza raggiunta eseguendo un altro ciclo è inferiore rispetto delle procedure appena descritte.

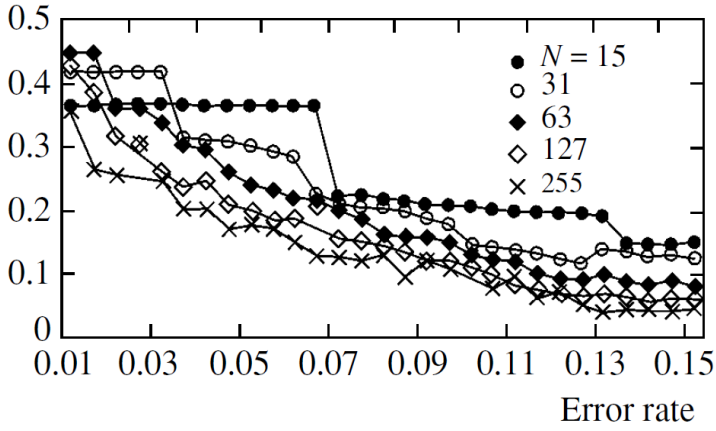

Figura 5: relativa lunghezza della chiave riconciliata

L'efficienza di esecuzione della correzione di errore "a ciclo unico" per parole di codice da 127 (o 255) bit è paragonabile a quella del protocollo a Cascata. Tuttavia, è importante che la probabilità condizionata, per l'individuazione di Eve sulle stringhe di bit condivise da Alice e Bob, rimanga invariante dopo la correzione degli errori nel protocollo a Cascata con la rimozione dell'errore, questo poiché essa viene determinata dall'errore $\varepsilon(Q)$ di misura ottima sugli stati non ortogonali $|e_0\rangle$ e $|e_1\rangle$ (vedi (20) e (21)), mentre cambia in modo sostanziale dopo che viene applicato un codice BCH. La probabilità condizionata per l'individuazione di Eve sulla chiave riconciliata determina il rapporto di compressione di hashing in una chiave segreta finale. Pertanto, il fatto che la lunghezza della chiave riconciliata è simile a quella del protocollo a Cascata non implica necessariamente che la lunghezza della chiave finale sarà la stessa.

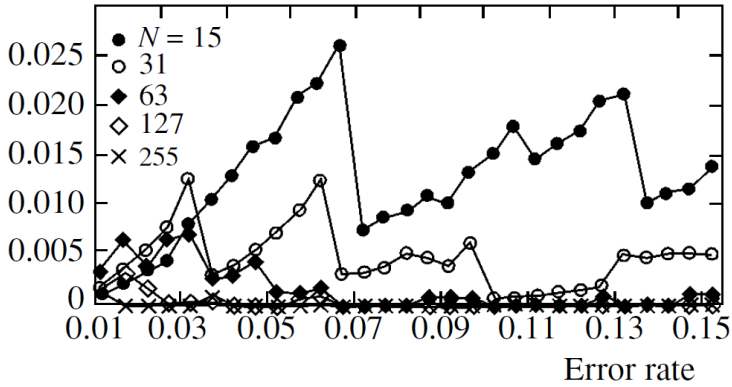

Figura 6: relativo numero di errori rimanenti

La Figura 7 mostra il numero di bit trasmessi attraverso il canale pubblico per bit della chiave riconciliata riportata in funzione del tasso di errore di Bob Q. È chiaro che questo numero è una frazione di un bit, come nel protocollo a Cascata [21].

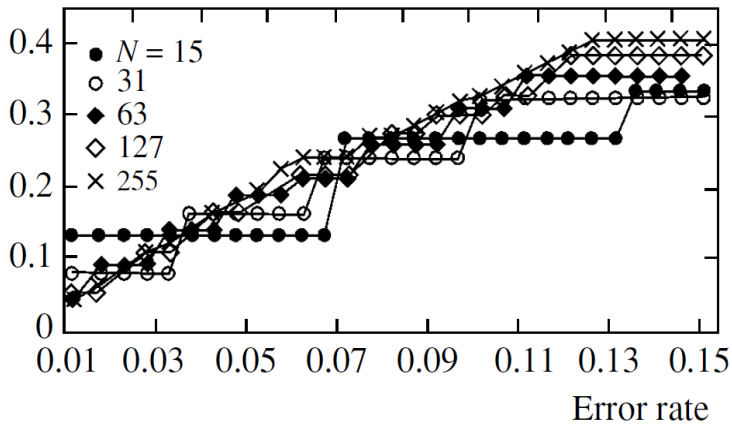

Figura 7: numero relativo di bit trasmessi

3.5.5 CALCOLO DEL RATE DI ERRORE DI EVE DOPO LA CORREZIONE DELL'ERRORE CON I CODICI BCH

prima di tutto, valutiamo la probabilità condizionata per la misura dell'individuazione di Eve. Prima della correzione, il rate di errore di Eve è

$$E(Q)=1-(1-Q)(1-\epsilon(Q))+Q\epsilon(Q)$$

(vedere (26)), mentre il rate di errore di Bob è Q. Mentre la conoscenza di Bob della chiave riconciliata è virtualmente perfetta, il rate di errore di Eve è determinato come segue.

poiché i parametri del codice a correzione di errore utilizzati attualmente (lunghezza delle parole di codice, il numero di informazioni e dei bit di controllo, la partizione in blocchi, etc) sono ipotizzati essere noti, Eve può tentare di correggere nella sua stringa di uscita con l'uso delle regole di decodifica. è ben noto che la probabilità di decodificare gli errori è

$$P_e=\frac{1}{M}\sum_{i=1}^{M} Pr(w_E^i\neq w^i)$$

dove w^i_E è la parola di codice generata da Eve dalla parola di codice w^i spedita da Alice [24, 25]. Per il codice $[n,k]$, la probabilità di errore è

$$P_e(p)=1-\sum_{j=0}^{n}\alpha_j p^j(1-p)^{n-j}$$

dove α_j è il numero di coset leaders (???) con istanza di Hamming j e p è il rate di errore iniziale per bit. Cerchiamo il rate di errore per bit $P_{symb}(p)$, il quale è noto essere inferiormente limitato (sottostimato in favore di Eve):

$$P_{symb}(p)\geqslant\frac{P_e(p)}{k}$$

Il calcolo immediato di $P_{symb}(p)$ richiede una ricerca esaustiva, la particolarmente difficile da compiere per codici con lunghe parole di codice. Comunque, diamo delle stime. Per ogni codice $[n, k]$,

$$P_e(p)\geqslant[C_n^{t+1}-\alpha_{t+1}]p^{t+1}(1-p)^{n-t-1}+\sum_{i=t+2}^{n}C_n^i p^i(1-p)^{n-i}$$

dove

$$\alpha_{t+1}=2^{n-k}-1-\sum_{i=1}^{t}C_n^i\geqslant 0 \quad (95)$$

e t è il maggiore intero per il quale la (95) vale.

Il rate di errore di Eve per bit della chiave riconciliata condivisa da Alice e Bob è data dalle equazioni (93) e (94) con

$$p\rightarrow E(Q)=1-(1-Q)(1-\epsilon(Q))+Q\epsilon(Q)$$

Il rate di errore di Bob dopo un ciclo di correzione di errore è dato da (93) e (94) con p al posto di Q.

Il rate di errore di Eve e di Bob dopo un ciclo di correzione degli errori svolto con il codice BCH [63, 39] “accordato” con Q = 6% sono mostrati in Figura 8.

L'entropia di Eve $R(P_e(E(Q)))$ mostrata nelle Figure 8a'-8c' determina la relativa lunghezza della chiave finale. Secondo la Figura 8, la lunghezza della chiave finale non supera l'8% della lunghezza della relativa chiave riconciliata. Quando $\alpha = \pi/16$, l'entropia di Renyi tende a zero per $Q \approx 1\%$. In dettaglio, $R(P_e(E(Q))) \approx 0.02$ (vedere Figura 8c'); cioè, la lunghezza della chiave finale non è maggiore del 2% della lunghezza della chiave riconciliata.

La Figura 9 mostra delle curve analoghe per i rate di errore di Eve e Bob valutati per vari angoli di

overlap tra gli stati portanti dopo un ciclo di correzione di errore eseguito da un codice BCH [127, 78]. Nel casi di un significante overlap ($\alpha = \pi/5$), i rate di errore sono quasi costanti dopo la correzione di errore. Quando gli stati portanti preparati sono per lo più ortogonali, il rate di errore di Eve decresce fortemente con un incremento di Q. L'entropia di Eve $R(P_e(E(Q)))$ mostrata nelle Figure 9a'-9c' dimostra che la lunghezza della chiave finale non supera il 4% della lunghezza della rispettiva chiave riconciliata. Notiamo che la correzione di errore svolta da un codice BCH [63, 39] lascia l'8% sebbene la relativa lunghezza della chiave riconciliata è maggiore.

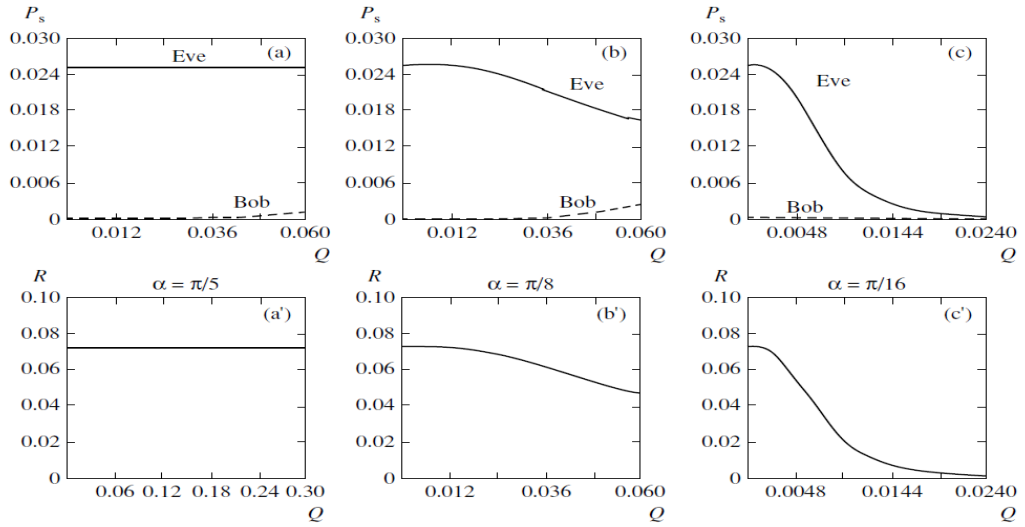

Figura 8

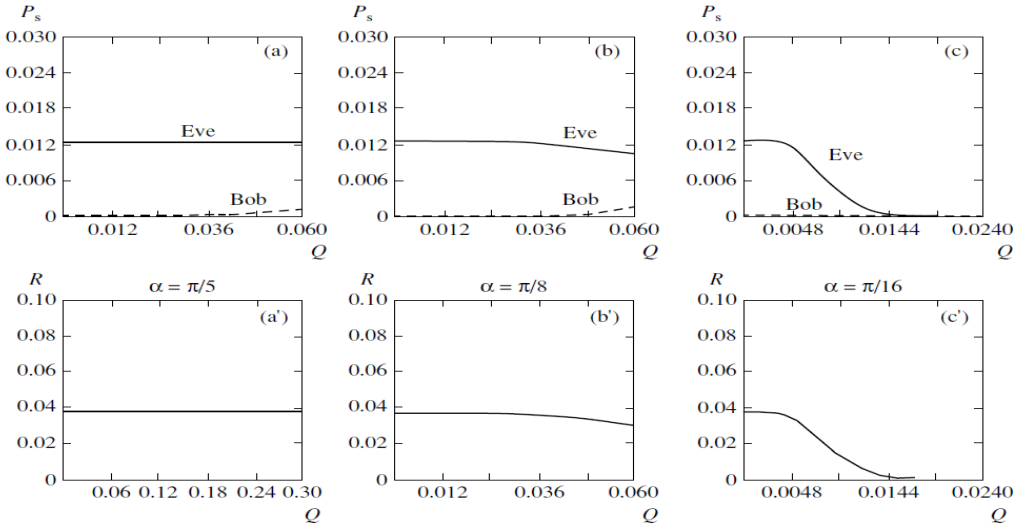

Figura 9

Questi risultati illustrano il fatto che un efficiente codice a correzione di errore può essere usato da una spia tanto bene quanto il legittimo partner: la probabilità condizionata per l'individuazione di Eve sulla chiave riconciliata aumenta con l'efficienza della correzione di errore, e la lunghezza della chiave segreta finale decresce di conseguenza.

Quindi, l'efficienza della correzione di errore di un codice valutata senza prendere in considerazione la parte quantistica del protocollo non può essere usata per quantificare la sua efficienza per quanto

riguarda la lunghezza della chiave segreta finale [21].

3.6 PRIVACY AMPLIFICATION E AUTENTICAZIONE

privacy amplification:

doc 2 pag 102,

doc 12 pag 40,

C. H. Bennet, G. Brassard, and J. M. Robert, “How to reduce your enemy's information”, in Advances in Cryptology-Procedings of Crypto'85, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 1986, vol. 218, p. 468.

C. H. Bennet, G. Brassard, and J. M. Robert, “Privacy amplification by public discussion”, SIAM J. Comput., vol 17, pp. 210-229, 1988.

C. H. Bennet, G. Brassard, and U. M. Maurer, “Generalized privacy amplification, “IEEE Trans. Inf. Theory, vol 41, no. 6, pt. 2, pp 1915-1923, nov. 1995.

autenticazione doc 8 pag. 29, doc 15 pag 1736 cap 2, (3, 4, 5 questi molto difficili)

3.7 SICUREZZA DI QKD

4 INTEGRAZIONE DI QKD NEI PROTOCOLLI ODIERNI

4.1 QKD IN IPSEC

Il protocollo IPsec utilizza la criptografia classica per le comunicazioni sicure, in questo paragrafo prooponiamo l'uso della crittografia quantistica per rimpiazzare i protocolli di criptografia classici usati per la distribuzione simmetrica.

In questa tesina, viene proposta una soluzione di QKD per IPsec chiamata SeQKEIP (Secure Quantum Key Exchange Internet Protocol) che non è basata su IKE, ma su ISAKMP. Usando questo metodo, evitiamo il problema della compatibilità tra IKE e QKD (qua mettere un riferimento a Elliott2003 con problema)

L'idea è di incollare al tradizionale IPsec (???). In fatti, ISAKMP non impone alcuna condizione ai meccanismi di negoziazione o ai parametri delle SA. Per usare la crittografia quantistica con IPsec abbiamo semplicemente definito le due fasi descritte nel capitolo 2. Procediamo dunque alla creazione della SeQKEIP. La SeQKEIP, come IKE, usa i meccanismi di ISAKMP e prende vantaggio dalla crittografia quantistica per costruire un pratico protocollo.

SeQKEIP lavora in modo simile ad IKE. Essa include 3 fasi: la fase 1 per la negoziazione della ISAKMP SA, la fase 2 per la negoziazione della SA e noi aggiungiamo una fase 0 nella quale Alice e Bob condivideranno la prima chiave segreta. Vi sono solamente tre modi in SeQKEIP: Quantum Mode, Main Mode e Quick Mode. Il Quantum Mode è lo scambio della chiave quantistica nella fase 0. Main mode viene usato durante la fase 1 e il Quick Mode è uno scambio nella fase 2. Sia il Main Mode che il Quick Mode sono molto simile a quelli in IKE.

4.1.1 Fase 0: scambio della fase – Quantum Mode

Questa fase è l'inizio dello scambio sicuro e viene implementata usando la criptografia quantistica. Alla fine di questi scambi, sia il mittente che il destinatario condividono un chiave segreta. questa chiave costituisce l'informazione segreta pre-condivisa nel meccanismo di IKE.

4.1.2 Fase 1: Negoziazione di ISAKMP SA – Main Mode

Durante questa fase, vengono negoziati l'algoritmo di criptografia e la funzione di hash. Solo questi due parametri discussi nella fase 1 costituiscono gli attributi di SeQKEIP. Il metodo di autenticazione è l'informazione segreta pre-condivisa (la chiave segreta scambiata con il metodo QKD). Contrariamente ad IKE, SeQKEIP non definisce alcun gruppo di vallori Diffie-Helman e non ha necessità di usare alcuna firma o certificato digitale. Nessuna chiave crittografica viene generata in questa fase, La prima chiave scambiata viene usata per criptare i pacchetti e per autenticare gli utenti.

Dopo le fasi 0 e 1, sia il mittente che il destinatario avranno le seguenti informazioni:

Chiave segreta condivisa	Questa chiave viene generata durante la fase 0 con il meccanismo della QKD. La chiave segreta viene usata per autenticare gli utenti e per criptare i pacchetti
Algoritmi di criptografia	L'algoritmo di criptografia viene applicato alla fase 2 (negoziazione dei parametri di SA). L'algoritmo potrebbe essere 3DES, DES, AES. Ma, se vogliamo avere il massimo della sicurezza, dobbiamo usare la funzione di One-Time-Pad (OTP)
Funzioni di Hash	La funzione di hash darà l'opportunità al mittente e al destinatario di verificare l'integrità del messaggio e l'autenticazione

Notiamo che la fase 0 e la fase 1 sono totalmente indipendenti e potrebbero essere svolte allo stesso tempo. Abbiamo bisogno della chiave segreta solo dalla fase 2.

4.1.3 Fase 2: negoziazione di SA – Quick Mode

Come in IKE, i messaggi scambiati nella fase 2 sono protetti nell'autenticazione e nella confidenzialità dai parametri negoziati dalla fase 1 e dalla fase 0. L'autenticazione è garantita dall'aggiunta del blocco di hash dopo l'header di ISAKMP e la confidenzialità è assicurata dalla criptografia dell'intero blocco del messaggio. L'obbiettivo di questa fase è di negoziare le SA. Cioè negoziare i parametri di "IPsec". I parametri di SA sono (mettere riferimento Mason2002): indirizzo di dstinazione, indici di parametro di sicurezza (SPI: Security Parameter Index), i meccanismi di sicurezza (AH o ESP) e di criptografia e la funzione di Hash, la chiave di sessione e un attributo aggiunto come il tempo di vita di SA.

Per SeQKEIP, per estendere la sicurezza, possiamo usare la funzione di criptografia One-Time-Pad. La prima chiave scambiata in questo caso, avrà la lunghezza del messaggio. Non abbiamo bisogno, quindi, di alcun algoritmo di criptografia per SA. Abbiamo invece bisogno di una funzione di Hash per verificare l'integrita del dato. L'IPsec potrebbe essere modificato in grado di usare il OTP.

All'inizio (Figura 10), la fase 0 e la fase 1 iniziano (le frecce indicate con 1 e 2 in Figura 10). Dopo queste due fasi, vengono fissati i parametri del protocollo. In (3), sarà usata la chiave scambiata grazie alla criptografia quantistica. Questa chiave sarà usata sia come chiave di sessione (4) o come funzione di OTP (4').

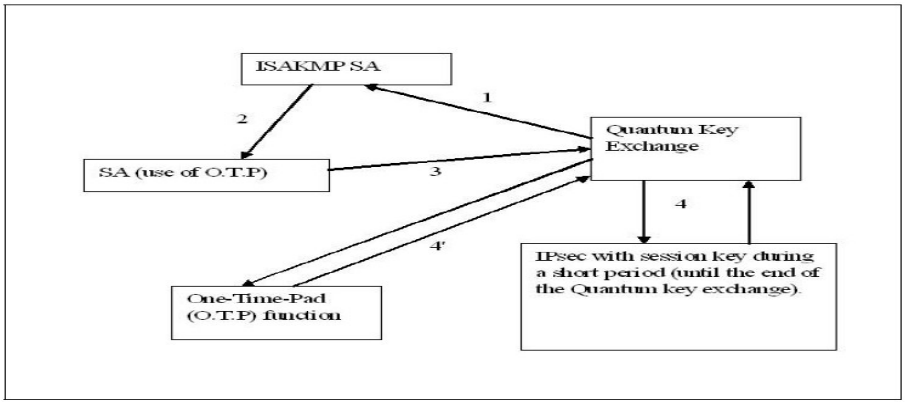

Figura 10: funzionamento di IPsec con la criptografia quantistica

In (4), usiamo un algoritmo di criptografia simmetrico tradizionale per scambiare i dati. I pacchetti IPsec sono gli stessi usati nei protocolli odierni dove non vi è la QKD. La durata del temo di vita della chiave di sessione è veramente corta ed è uguale al tempo impiegato per lo scambio della chiave segreta con l'uso della criptografia quantistica. Questa soluzione è una soluzione di transizione per il (4').

Nel (4'), usiamo in modo totale i concetti della criptografia quantistica. L'idea è di mutare completamente per una funzione di sicurezza incondizionata, cioè lo scambio della chiave quantistica e la funzione OTP. Dopo aver fissato i parametri di SA, la lunghezza delle chiavi di sessione sarà della dimensione dei pacchetti dato in IPsec. Quindi, è possibile usare il OTP facendo semplicemente un XOR del messaggio con la chiave e spedire il risultato verso il destinatario.

Per usare il OTP, abbiamo bisogno di scambiare una chiave per ogni pacchetto. Il punto debole di questa soluzione risiede nel tempo necessario allo scambio della chiave. Il rate di bit totale è fortemente indebolito da questo problema il quale però, dato che la tecnologia della criptografia quantistica si sta sempre di più evolvendo, verrà risolto molto probabilmente [12].

4.2 QKD IN TLS

4.2.1 Processo Di distribuzione Della Chiave In TLS

Il protocollo TLS Record usa il protocollo di Handshake TLS per generare i parametri di sicurezza. Questo compito è portato a termine facendo uso del processo di distribuzione della chiave. Nella descrizione del TLS Handshake [8] la distribuzione della chiave è limitata all'uso dei protocolli Diffie-Hellman e RSA. Il problema è che entrambi non sono incondizionatamente sicuri; la loro sicurezza è computazionale e dipende dalla potenza computazionale o dal tempo.

Usando la QKD, tentiamo di compiere una sicurezza incondizionata poiché la QKD è provata essere incondizionatamente sicura in modo scientifico. Questo significa che la sicurezza in questo caso è indipendente dalla potenza di calcolo in possesso di una spia e di conseguenza la sicurezza non è minacciata da alcuna tecnologia. Per questa ragione proponiamo di integrare la QKD nel protocollo TLS nello scambio delle chiavi invece dei protocollo Diffie-Helman e RSA.

4.2.2 Necessità Richieste Per QKD-TLS

Alcune richieste devono essere soddisfatte per integrare la QKD nel protocollo TLS:

a) Un canale ottico: la QKD usa i fotoni per codificare le informazioni per sfruttare le leggi della fisica quantistica. Oggigiorno, vi sono due mezzi di trasporto per i fotoni: la fibra ottica e lo spazio libero [35]. Ma alcune recenti ricerche si stanno concentrando anche nell'uso di atomi e di elettroni come particelle quantistiche [36-37] e probabilmente vi sarà un nuovo tipo di canale quantistico in futuro. Per il nostro lavoro, scegliamo di utilizzare le fibre ottiche poiché sono le più usate nei sistemi quantistici. Questo è dovuto al fatto che le fibre ottiche introducono meno rumore rispetto allo spazio libero.

b) Modem ottico: il modem può giocare il ruolo di rivelatori e di emettitori. Gli scopi del modem ottico sono di rilevare e di spedire fotoni. Il modem deve includere un rilevatore di fotoni ed un emettitore di fotoni e anche un polarizzatore per codificare il dato usando differenti valori di polarizzazione come gli stati quantistici. È impiegato per fornire chiavi quantistiche, ma può anche essere utilizzato per

lo scambio di dati a seconda del metodo di codifica delle informazioni. Il modem è molto

importante perché può funzionare sia sul canale classico sia su quello quantistico. Vi sono

molte tecniche utilizzate per elaborare tale modem [38-39].

c) Protocollo di QKD: per generare una chiave, è necessario implementare un protocollo di QKD tra

i due modem ottici. La chiave una volta generata, è memorizzata in una memoria flash al fine di essere

utilizzata nella fase di cifratura dei dati. Abbiamo scelto nel nostro lavoro il protocollo BB84, poiché in primo luogo lo scambio

è stato dimostrato essere assolutamente sicuro e in secondo luogo è facile da implementare

4.2.3 Un componente aggiunto per il protocollo TLS: QKD Configuration Protocol

Per facilitare l'attuazione del nostro schema di TLS (compreso il servizio di QKD) si

aggiungere al protocollo TLS un nuovo componente che svolge il ruolo di configurazione di sottorete di QKD.

Il componente aggiuntivo porta il nome di Configuration Protocol QKD. Quindi nella nostra soluzione, il protocollo TLS ha cinque componenti: Handshake Protocol, Change Spec Protocol, Alert

Protocol, Application Data Protocol e QKD Configuration Protocol. Abbiamo proposto un formato di messaggio per il QKD Configuration Protocol. Il formato contiene un campo importante della lunghezza della chiave che saranno generate dal meccanismo di QKD. Altri campi sono mostrati nella Figura 11.

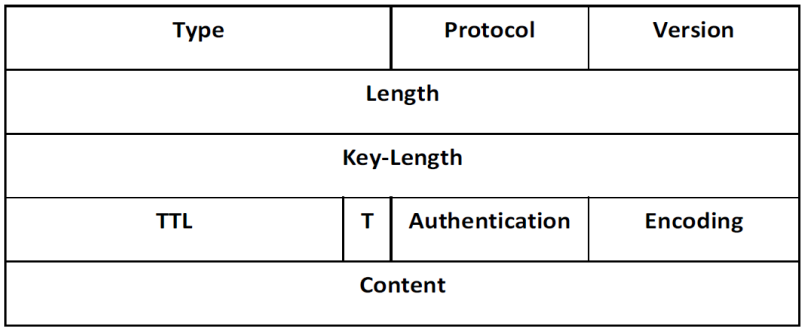

Figura 11: il messaggio formato dal QKD Configuration Protocol

La descrizione dei campi del formato del messaggio è la seguente:

- *Type* (1 byte): indica il tipo di protocollo di crittografia quantistica utilizzato. Per esempio i protocolli basati sul principio di indeterminazione di Heisenberg come il protocollo BB84 o i protocolli basatI sulla disuguaglianza di Bell come il protocollo E91 [40].
- *Protocol* (1 byte): indica il protocollo di chiave quantistica utilizzato (ad esempio BB84, B92 [41], o E91).
- *Version* (1 byte): consente l'utilizzo di più di una versione dello stesso protocollo.
- *Length* (4 byte): indica la lunghezza del messaggio in byte.
- *Key-length* (4byte): questo campo fornisce la lunghezza della chiave fornita dall'esecuzione del protocollo a chiave quantistica. La sua lunghezza è compresa tra 1 e 4 byte. La lunghezza è così grande in modo tale da utilizzare il One Time Pad per raggiungere la sicurezza incondizionata.
- *TTL field* (2 byte meno un bit): indica un periodo di tempo (in secondi) o il numero di messaggi quando una chiave potrebbe essere utilizzato in operazioni di cifratura. Se il tempo è scaduto o il massimo dei messaggi è raggiunto, il meccanismo di QKD inizia a generare una nuova chiave.
- *T field* (un bit): questo campo indica se usiamo il numero di messaggi o la quantità di tempo. Quando il suo valore è "1", il TTL field mostra una quantità di tempo e quando il suo valore è "0", il TTL field corrisponde al numero di messaggi.
- *Authentication* (1byte): mostra se il messaggio è autenticato o meno.
- *Encoding* (1byte): questo campo specifica certe tecniche di codifica se è utilizzata per crittografare il contenuto del messaggio archiviato.
- *Content* (la sua lunghezza non è fissa): questo campo mostra i dati associati a questo messaggio

4.2.4 Il TLS handshake Protocol modificato: Quantum TLS Handshake Protocol

Nel QKD-TLS Protocol, abbiamo aggiunto alcune modifiche al TLS Handshake Protocol. Il nostro obiettivo principale è quello di generare i parametri di sicurezza dal meccanismo di QKD e per rimuovere tutte le strutture basate sul PKI (Public Key Infrastructure). Prima di tutto, supponiamo che il client e il server condividono un segreto indicato con S. In secondo luogo, abbiamo sostituito nel

TLS Handshake Protocol la procedura del classico processo di scambio della chiave (ad esempio RSA o Diffie-Hellman) con il meccanismo di QKD utilizzando il protocollo BB84. Diamo al TLS handshake protocol modificato il nuovo nome di Quantum TLS Handshake Protocol. La Figura 11 riassume come diversi messaggi vengono scambiati tra il client e il server durante il Quantum TLS Handshake Protocol. Poiché il protocollo BB84 è vulnerabile al "man in the middle" [31], verifichiamo se è un intercettatore è individuato una volta che l'esecuzione del protocollo BB84 è finita, calcolando il TLS finito in entrambi i lati del client e del server. Questo viene fatto utilizzando il segreto condiviso S e la chiave K derivata dal protocollo BB84. Durante il Quantum TLS Handshake Protocol e quando il server riceve il ClientHello, si invia al client una serie di fotoni polarizzati. Il numero di fotoni da trasmettere dipende dalla lunghezza della chiave desiderata, vengono usati gli algoritmi di correzione degli errori e la privacy amplification. Per ogni fotone da inviare, il server sceglie a caso uno stato a_{ij}. I passaggi successivi sono esattamente gli stessi descritti nei capitoli precedenti.

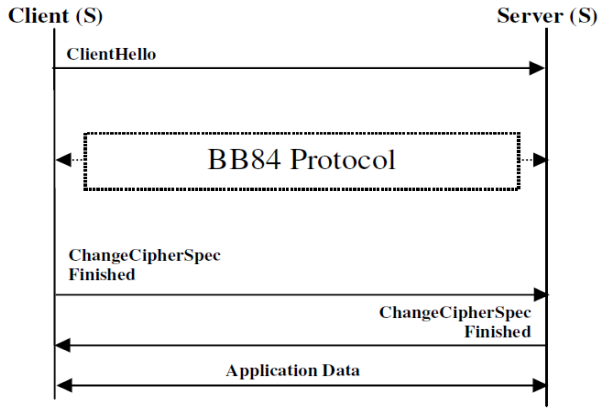

Figura 12: messaggi scambiati nel Quantum TLS Handshake Protocol

4.2.5 QKD-TLS in Operation Mode

Il nostro obiettivo è quello di utilizzare il meccanismo di QKD nel processo di autenticazione e nei dati criptati. In primo luogo, si usa la chiave generata dal protocollo BB84 con il segreto S nell'espressione di pre_master_key presentata nella formula di calcolo del master_secret la quale viene utilizzata in calcolo del TLS finito e così abbiamo controllato la mutua autenticazione del cliente e del server. In secondo luogo, sfruttiamo la chiave fornita dal protocollo BB84 per generare la chiave materiale per i dati criptati nel protocollo TLS. Quindi, la QKD viene sfruttata nella procedura di autenticazione e nella crittografia dei dati tra il client e il server. La Figura 13 dà il nostro schema di QKD-TLS protocollo.

Nel nostro schema, il TLS Protocol verifica due cambiamenti. Integriamo un nuovo componente chiamato QKD Configuration Protocol e abbiamo fatto alcuni cambiamenti nell'originale TLS Handshake Protocol (Quantum TLS Handshake Protocol) per includere il servizio di QKD. Nella modalità di funzionamento, quando TLS Record riceve dal Application Layer i dati, il QKD Configuration Protocol viene scambiato tra il client e il server per concordare la lunghezza della chiave desiderata, i campi TTL e T e altri campi come quelli mostrati in Figura 11. Una volta il QKD Configuration Protocol viene eseguito, comincia una sessione di Quantum TLS Handshake (Figura 12). Quindi il client e il server avviano il protocollo BB84 per derivare una chiave K la qui sicurezza è garantita dalle leggi della fisica quantistica.

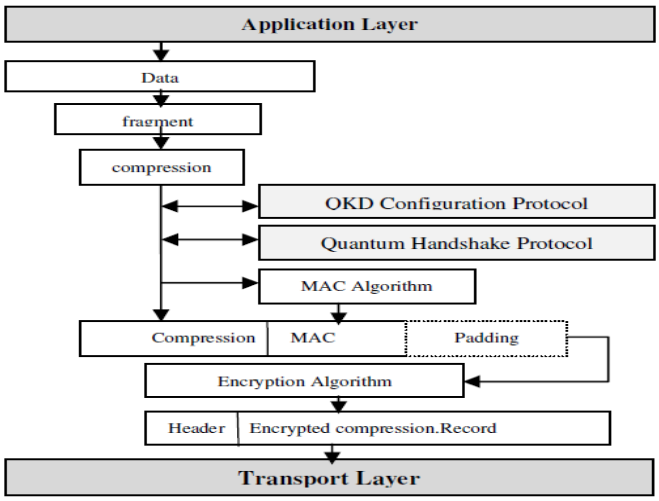

Figura 13: protocollo QKD-TLS in Operation Mode

Come accennato prima, il protocollo BB84 è vulnerabile all'attacco "man in the middle", per verificare se l'autenticazione reciproca è stabilita in maniera corretta, il client e il server devono calcolare il messaggio di TLS finito usando il segreto condiviso S e la chiave generata dal processo di QKD, $K = \{0,1\}^N$, proponiamo:

$$pre\_master\_secret = K + S$$

Il TLS finito è calcolato come descritto nel paragrafo 3 per l'espressione: (vedere se coincide il paragrafo 3)

$$PRF(master\_secret,\ finished\_label,\ hash\ (handshake\_messages))$$

Notiamo che il calcolo del TLS finito usa la chiave generata dal QKD perché abbiamo per il nostro protocollo QKD-TLS:

$$master\_secret = PRF(pre\_master\_secret,\ \text{mastersecret},\ ClientHello.random)$$

È molto importante ricordare che in tutti i messaggi pubblici scambiati durante l'esecuzioni del protocollo BB84 sono parte del valore del handshake_messages. Una volta che il server riceve il messaggio di TLS finito dal client, egli calcola il suo proprio TLS finito e verifica se è lo stesso di quello del cliente o no, se sì, allora il cliente è autenticato. La stessa operazione viene effettuata dal cliente quando riceve il TLS finito del server. Concludiamo allora che il meccanismo di QKD è sfruttato nel controllo dell'autenticazione reciproca tra il client e il server.

Il Record Protocol ha bisogno di un algoritmo per generare le chiavi richieste dai parametri di sicurezza previsti dall'Handshake Protocol. La chiave K (invece di pre_master_key come nell'originale TLS Protocol) è divisa in una sequenza di byte sicuri: una parte per il MAC del client, un'altra per il MAC del server, un'altra per criptografia del client e un'altra per la criptografia del server [8].

Nella successiva connessione tra il client e il server cambiamo il segreto condiviso da S con K: S = K e così ogni chiave generata dal protocollo BB84 svolgerà il ruolo di S nelle prossime connessioni. La variazione di S ad ogni connessione migliora notevolmente la sicurezza in quanto questo rende molto difficile il compito da parte di utenti malintenzionati a scoprire S.

4.2.6 Esempio di utilizzo QKD-TLS protocollo

In questa sezione, presentiamo un esempio di implementazione QKD-TLS. Consideriamo due LAN reti connesse tramite due modem ottici come illustrato in Figura 14. Per migliorare la sicurezza di una

connessione TLS tra A e B impiegando la crittografia quantistica, devono essere svolte sei fasi:

Fase 1: quando il TLS Record Protocol nel punto A riceve i dati dal livello applicazione, egli chiama il suo Application Protocol Data. Il dato poi è frammentato e per ogni frammento potrebbe essere fatta una compressione.

Fase 2: il TLS Record Protocol utilizza il QKD Configuration Protocol in modo da lasciare che i punti A e B concordino i parametri illustrati nel formato QKD Configuration Protocol (Figura 11). I campi più interessanti sono i seguenti: Protocol, Key-length, TTL field e T field. Scegliamo il protocollo BB84 per questo esempio e ipotizziamo che non esisto meccanismi di autenticazione e di crittografia. Proponiamo le scelti scelte: keylength = 40 bye, TTL = 400 messaggi, T = 0. Dobbiamo scegliere TTL=1 se si prevede di utilizzare il OTP per raggiungere la sicurezza incondizionata.

Fase 3: il Quantum Handshake Protocol è utilizzato dal TLS Record Protocol per ottenere i

parametri di sicurezza. Il Quantum Handshake Protocol ha inizio e durante il processo di QKD, viene implementato il protocollo BB84 tra i due modem. La chiave K generata viene memorizzata in un memoria flash da utilizzare successivamente nella crittografia dal Record Protocol.

Fase 4: Il TLS Record Protocol riceve la chiave K fornite dal servizio di QKD e costruisce i suoi parametri di sicurezza. Questi parametri sono utilizzati per generare le chiavi per cifrare i dati e per garantire l'integrità (MAC) come illustrato nella Figura 13. Inoltre, in questa fase A e B verificano reciprocamente l'autenticazione utilizzando TLS finito con i due segreti S e K.

Fase 5: Una volta che l'intero record è cifrato, viene aggiunto un header al blocco cifrato e l'intero pacchetto è passato al Transport Layer.

Fase 6: Cambiamo il valore di S con K (S = K) e il nuovo segreto condiviso tra il client e il server è K [13].

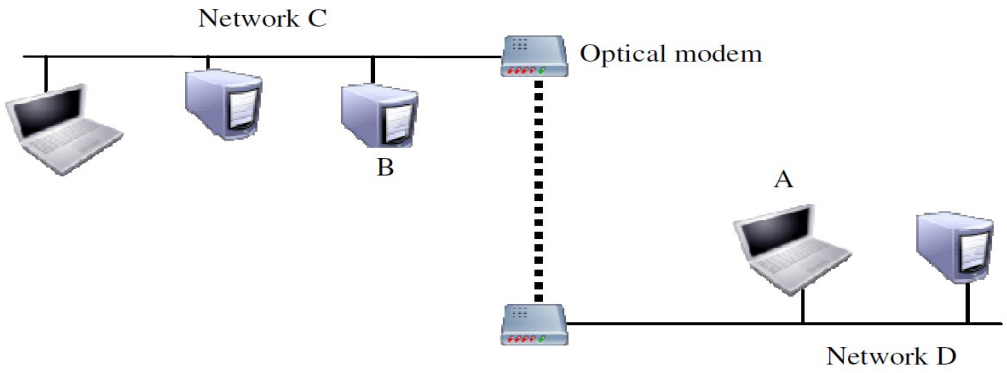

Figura 14: un esempio di implementazione di QKD-TLS

5 RETI QKD

5.1 ATTRIBUTI DI UNA RETE QKD

Abbiamo visto nei capitoli precedenti che la QKD offre una tecnica per creare e distribuire delle chiavi su un casuale condiviso tra due dispositivi distinti, con una probabilità molto bassa di intercettazione. Più in generale, si possono votare QKD successo contro una serie di obiettivi importanti per la distribuzione delle chiavi, come schierati nel paragrafi successivi (da rivedere questa frase tradotta)

5.1.1 Protezione delle chiavi

La QKD offre vantaggi significativi in questo senso e in effetti questa è la ragione principale di interesse per la QKD. I sistemi a chiave pubblica soffrono di una persistente incertezza legata al fatto che questi sistemi, un giorno, potrebbero essere rotti in futuro, con conseguente perdita di capacità di comunicare in modo sicuro. I classici sistemi a chiave segreta soffrono di problemi piuttosto diversi i quali sono principalmente le minacce interne e gli oneri logistici legati alla distribuzione materiale delle chiavi. Assumendo che le tecniche QKD siano correttamente incorporate in un sistema globale e sicuro, queste possono fornire la distribuzione automatica delle chiavi e offrire una sicurezza superiore ai sistemi diffusi oggigiorno.

5.1.2 autenticazione

Quando si consegna una chiavi segrete a qualcuno, è molto importante non darla alla persona sbagliata! La QKD non fornisce di per sé l'autenticazione. Attuali strategie per l'autenticazione nei sistemi QKD comprendono l'installazione di chiavi segrete nei dispositivi per essere utilizzate in schemi di autenticazione basati sulle funzioni di hash o in sistemi ibridi di QKD e chiave pubblica. Nessuno dei due approcci è del tutto attraente. Chiavi segrete preinstallate richiedono un certo modo di distribuzione prima che QKD si inizi, per esempio tramite corriere umano, che ovviamente può essere costoso e logisticamente difficile. Inoltre, questo schema sembra aperto ad attacchi di tipo “denial of service” nei quali un avversario forza un sistema QKD per esaurimento delle sue chiavi, a quel punto non l'autenticazione non può più avvenire. Schemi ibridi formati da chiavi QKD e pubblica, d'altra parte, ereditano le vulnerabilità dei sistemi a chiave pubblica e cioè la rottura degli algoritmi tramite computer quantici o progressi inaspettati nel campo matematica.

5.1.3 Consegna sufficientemente rapida delle chiavi

Un sistemi di distribuzione delle chiavi deve consegnare le chiavi abbastanza velocemente in modo che i dispositivi di crittografia che utilizzano queste chiavi non debbano rallentare la loro trasmissione dei dati. Questa è ovviamente una sfida tra la velocità con la quale le chiavi vengono generate e la velocità con la quale esse vengono consumate per l'attività di criptografia. Sistemi di QKD odierni raggiungono l'ordine di 1000 bit s^{-1} throughput per chiave, nella migliore delle ipotesi, e spesso vengono eseguiti a tassi molto più bassi. Questo è inaccettabile in quanto basso se uno usa queste chiavi in un certo modo, per esempio con il one-time pad ad alta velocità dei flussi di traffico. Tuttavia può essere accettabile se le chiavi vengono usate come input per meno sicuro algoritmi come l'Advanced Encryption Standard (AES). Nel complesso, tuttavia, sarebbe utile aumentare i rate di consegna QKD di almeno diversi ordini di grandezza.

5.1.4 ROBUSTEZZA

Questa è una proprietà fondamentale per tutti i sistemi non-stop, ma non è sempre stata presa in considerazione dalla comunità della QKD. Dato che la chiave è essenziale per le comunicazioni sicure, è estremamente importante che il flusso di tali bit non sia interrotto sia da atti casuali che intenzionali compiuti da un avversario (cioè con la negazione del servizio). Qui QKD ha fornito un fragile servizio fino ad oggi poiché le tecniche QKD sono implicitamente state impiegate lungo un singolo collegamento punto-a-punto. Quindi se questo singolo link viene distrutto, per esempio dalle intercettazioni o addirittura da taglio della fibra, tutto il flusso per la distribuzione delle chiavi cesserebbe. Noi sosteniamo che una rete a maglia QKD è intrinsecamente molto più robusta di un singolo collegamento punto-a-punto in quanto offre percorsi multipli per la distribuzione delle chiavi. Se un collegamento è interrotto o è soggetto ad intercettazioni, la rete può indirizzare automaticamente il flusso in modo tale da aggirare il link guasto. Mentre questa è una tecnica utile per qualsiasi forma di rete di comunicazione, lo è particolarmente per la QKD, dove l'informazione quantistica scorre tradizionalmente lungo un unico percorso senza alcuna forma di backup o di fail-over.

5.1.5 INDIPENDENZA DALLA DISTANZA E DALLA POSIZIONE

Idealmente, qualsiasi entità sarebbe in grado di scambiare chiavi con qualsiasi altra entità del mondo. L'architettura di sicurezza di Internet offre questa funzionalità: qualsiasi computer di Internet può formare un'associazione di protezione con qualsiasi altro computer. Questa funzione è priva nella QKD la quale richiede che le due entità abbiano un percorso diretto tra di loro e senza vincoli per i fotoni, e che può operare solo per poche decine di chilometri attraverso la fibra.

5.1.6 RESISTENZA ALLE ANALISI DI TRAFFICO

Gli avversari possono essere in grado di effettuare delle analisi del traffico su un sistema di distribuzione delle chiavi al fine di comprendere la relazione esistente tra le entità comunicanti. Per esempio, un flusso pesante di chiavi tra due punti potrebbe indicare che un grande volume di informazioni riservate sta, o sarà, fluendo tra i due estremi. Può essere auspicabile rendere tali analisi il più difficile possibile.

Lo scopo di questo capitolo è di rafforzare le prestazioni della QKD in queste aree più deboli. In alcuni casi, questo comporta l'introduzione di nuove tecnologie QKD, ad esempio, l'introduzione di un nuova fonte di fotoni entangled per ottenere una più rapida consegna delle chiavi. In altri casi, ci affidiamo a una migliore architettura del sistema per raggiungere questi obiettivi affrontando l'Indipendenza dalla distanza e dalla posizione introducendo una rete i cui nodi non sono sotto il controllo di Eve [22].

5.2 ARCHITETTURA DI SISTEMA DI UNA RETE DI FIDUCIA

Abbiamo visto fino ad ora in dettaglio sono il collegamento punto-a-punto nelle tecniche QKD, ma questo soffre ancora di difetti notevoli. In primo luogo, è geograficamente limitato dalla distanza a cui un singolo collegamento può essere impiegato. L'attenuazione in fibra pone dei limiti sui collegamenti terrestri a 50 km o meno nelle applicazioni pratiche. Il collegamento nello spazio libero, per esempio con l'uso di satelliti, può consentire collegamenti WAN o anche transcontinentale, ma ancora non consente una copertura veramente globale. In secondo luogo, i collegamenti isolati punto-a-punto sono soggetti a semplici attacchi di denial-of-service, come le intercettazioni o il taglio della fibra. In terzo luogo, in pratica può essere proibitivo economicamente stabilire a coppie, dedicati collegamenti punto-a-punto tra tutti gli enclave privati che desiderano comunicare tra di loro.

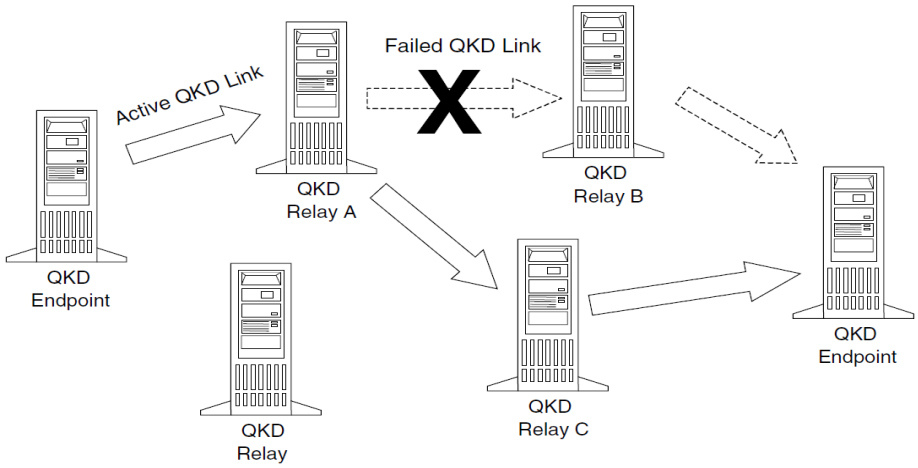

Figura 15: rete QKD con nodi sicuri

In misura sorprendentemente, questi inconvenienti possono essere attenuati attraverso l'organizzazione di una serie di collegamenti QKD in una rete quantistica. La Figura 15 mostra una rete QKD in forma altamente schematica. In contrasto con il collegamento punto-a-punto descritto prima, però, gli endpoint QKD sono ora collegati tramite una rete di relè o router QKD.

Come si vede, questa forma di rete QKD è composta da una collezione di collegamenti punto-a-punto QKD. Così il nodo più a sinistra della rete QKD scambia una chiave con il Relay A, il quale invece scambia chiavi con alcuni o tutti i suoi vicini, cioè Relay B e/o C, ecc. Quando un dato collegamento punto-a-punto QKD all'interno della maglia dei Relay cade – per via di un taglio di fibra o di un livello troppo alto di intercettazioni o di rumore - questo link viene abbandonato e ne viene usato un altro al suo posto. Così la rete QKD generale può essere progettata per essere resistente anche a fronte di intercettazioni o attacchi del tipo denial-of-service.

Tali reti QKD possono essere costruite in diversi modi. In una variante, i relay QKD possono trasportare solo materiale riguardante le chiavi, ma non messaggi di traffico. Così, dopo che i vari relay hanno stabilito a delle coppie di chiavi lungo il collegamento end-to-end, possono impiegare queste coppie di chiavi per trasportare in modo sicuro un chiave “hop by hop” da un capo all'altro, magari cifrate e decifrate tramite il OTP per ogni coppia di chiave tra un relè e l'altro. Con questo approccio, la chiave corretta per il collegamento end-to-end appare in chiaro solo all'interno delle memorie dei relay, ma è sempre cifrata quando passa attraverso un link. Tale progetto può essere definito come una “rete a trasporto di chiave”.

In un'altra variante, i relay QKD possono trasportare sia materiale riguardante le chiavi che messaggi di traffico. La Figura 16 illustra questa seconda variante, in cui i relay sono integrati ai router di internet e che, a coppie con meccanismi di QKD, forniscono collegamento cifrati tra i router. In sostanza, ogni datagramma IP di messaggi di traffico viene cifrato ogni volta che transita per ogni collegamento. Al relay di destinazione viene decifrato, mantenuto in chiaro nella memoria del relay, e poi criptato di nuovo con un secondo insieme di chiavi e inviato al relay successivo. Questa operazione procede, hop by hop, fino a quando il datagramma è finalmente pervenuto alla destinatario finale. Notiamo che questa rete si differenzia dalla definizione standard di Internet con l'interposizione di una serie di tunnel cifrati (“link virtuali”) tra router cooperanti.

Tali reti QKD apportano significativi vantaggi che mitigano in modo notevole gli svantaggi del collegamento punto-a-punto elencati all'inizio di questa sezione. In primo luogo, consentono di estendere la portata geografica di una rete di comunicazione protetta dalla crittografia quantistica, dato

che le wide-area network possono essere create da una serie di collegamenti punto-a-punto con relay attivi. Questi collegamenti possono ulteriormente essere mezzi di trasmissione eterogenei, cioè alcuni possono essere tramite fibra, mentre altri attraverso lo spazio libero. Così, in teoria, una tale rete potrebbe fornire piena copertura globale. In secondo luogo, diminuire la possibilità che un avversario possa disabilitare il processo di distribuzione della chiave, sia per causate da intercettazioni o semplicemente dal taglio della fibra. Una rete QKD può essere progettato con la ridondanza desiderata semplicemente con l'aggiunta di più collegamenti e relay alla rete. In terzo luogo, le reti QKD possono ridurre notevolmente i costi di interconnessione su larga scala delle enclavi private riducendo il numero di collegamenti richiesto, cioè (N × N - 1) / 2, ad appena N collegamenti nel caso di una semplice topologia a stella per la rete di distribuzione delle chiavi.

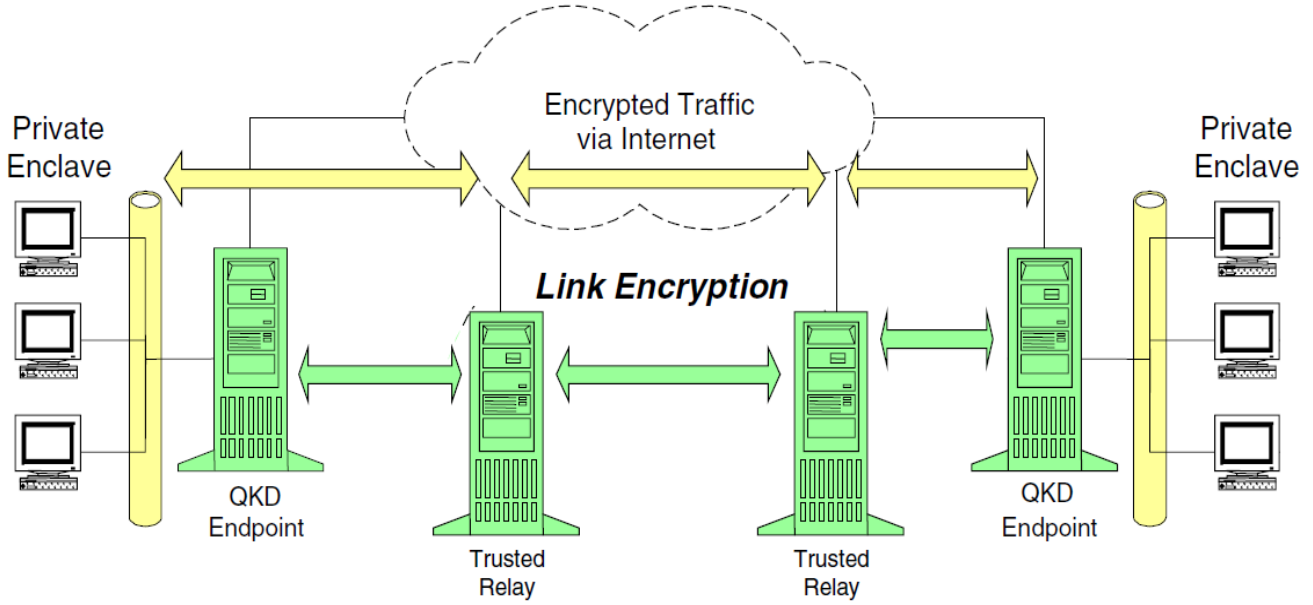

Figura 16: rete QKD con nodi sicuri e collegamenti cifrati

Tali reti QKD, comunque, sono da non considerare delle panacee. La loro debolezza principale è che i relay devono essere *attendibili*. Cioè, dato che la chiave e - direttamente o indirettamente - il traffico di messaggi sono disponibili in chiaro nelle memorie dei relay, a questi relay deve essere impedito di cadere nelle mani di un avversario. In pratica, avrebbero bisogno di essere in luoghi protetti fisicamente e forse custoditi se il traffico è veramente importante. Anche se questa è una condizione abbastanza onerosa, notiamo che solo i dispositivi realy devono essere protetti, fibre o collegamenti attraverso lo spazio libero tra loro non hanno bisogno di tale protezione. Quindi, solo un numero finito di piccole aree richiedono elevati livelli di sicurezza fisica. Un correlato, ma forse più sottile, svantaggio è che tutti gli utenti nel sistema devono fidarsi della rete (e degli operatori della rete), con tutte le chiavi per il loro traffico di messaggio.

Concludendo, la sicurezza sul singolo collegamento è stata ampiamente dimostrata durante i precedenti capitoli di questa tesina, tuttavia non siamo in possesso di prove simili per le reti QKD. Dimostrazioni dettagliate ed esplicite saranno necessarie prima di dichiarare queste reti “sicure” [22].

5.3 RETI QKD ESISTENTI

Negli ultimi dieci anni, le reti QKD sono state ampiamente studiate in vari ambiti di ricerca. La "DARPA Quantum Network", come parte di un progetto sostenuto dalla "US Defense Advanced Research Projects Agency" (DARPA appunto), apri la strada alla diffusione della QKD nel campo delle reti [6]. La rete DARPA Quantum Network è costituita da 10 nodi collegati tra loro mediante una rete a commutazione ottica attiva. Il progetto European FP6 "Secure Communication using Quantum Cryptography" (SECOQC) ha integrato una serie di diversi sistemi di QKD in un'unica rete sviluppando una interfaccia multipiattaforma [7]. Dal progetto SECOQC, un gruppo specifico industriale per la QKD, l'European Telecommunications Standards Institute (ETSI), è stato lanciato per offrire una sede per la creazione di standard QKD universalmente accettati [8]. Il funzionamento a lungo termine della QKD, inoltre, è stato testato nel campo sperimentale con la rete "Swiss Quantum" a Ginevra [9], con la rete "Durban" in Sud Africa, sviluppata dal progetto "Durban-Quantum City" [10], e con la "Cambridge Network" [11] nel Massachusetts. Implementazioni di rete trasparente di QKD sono state dimostrate, ad esempio la rete riconfigurabile dinamicamente in un banco di prova della "Advanced Technology Demonstration Network" (ATDNet) nella zona di Washington DC compiuta dalla "Telcordia Technologies" [12], la rete ottica passiva a Madrid, composta da un anello centrale e della rete di accesso realizzata dalla "Universidad Politécnica de Madrid" e dalla "Telefónica Investigación y Desarrollo" [13] e la rete gerarchica costituita da 5 nodi collegati da una dorsale quantistica (chiamata QBB, Quantum BackBone) multiplexata a divisione di lunghezza d'onda e da sottoreti collegate da nodi di fiducia, a Wuhu, nella regione dell'Anhui (a ovest di Shanghai), da un gruppo dall'"University of Science and Technology of China" [14] fino ad arrivare alla "Tokyo QKD Network" ultimata lo scorso anno [23].

In questa tesina si è voluto descrivere la rete quantistica presente a Vienna e quella a Tokyo.

5.3.1 VIENNA

Presentiamo ora la rete QKD progettata e realizzata dal progetto europeo SECOQC (*Se*cure *CO*mmunication based on *Q*uantum *C*ryptography) a Vienna, unificando gli sforzi di 41 organizzazioni di ricerca ed industriali. In questo paragrafo daremo una descrizione molto descrittiva della rete e della velocità con la quale vengono generate le chiavi.

La rete QKD presente a Vienna ha una struttura basata sulla sicurezza dei nodi, cioè i nodi devono essere attendibili e non essere sotto il controllo di una potenziale spia.

La funzionalità della rete SECOQC è stata dimostrata pubblicamente durante la sua conferenza d'inaugurazione l'8 ottobre del 2008. La dimostrazione coinvolse l'uso del One-Time Pad per cifrare la comunicazione telefonica, dell'AES per una videoconferenza con tutti i nodi distribuiti e una serie di esperimenti di rerouting, mettendo in evidenza i meccanismi di base della funzionalità della rete SECOQC. La rete è composta da 6 nodi collegati da 8 collegamenti di rete punto-a-punto di sei tipi differenti:

- tre sistemi Plug-and-Play construiti da idQuantique
- un sistema Coherent One-Way dalla Optique GAP con la partecipazione di idQuantique e l'Austrian Research Centers
- un sistema CV dal Centre National de la Recherche Scientifique e dal THALES Research and Technology con la partecipazione dell'Universite Libre de Bruxelles
- un One-Way Weak Pulse System dal Toshiba Research of the United Kingdom
- un Entangled Photons System dall'University of Vienna e dall'Austrian Research Centers

- Inoltre, due nodi situati in edifici adiacenti sono stati collegati da un collegamento a spazio libero realizzato dalla Maximillians Ludwig University di Monaco di Baviera (linea di vista di 81 m)

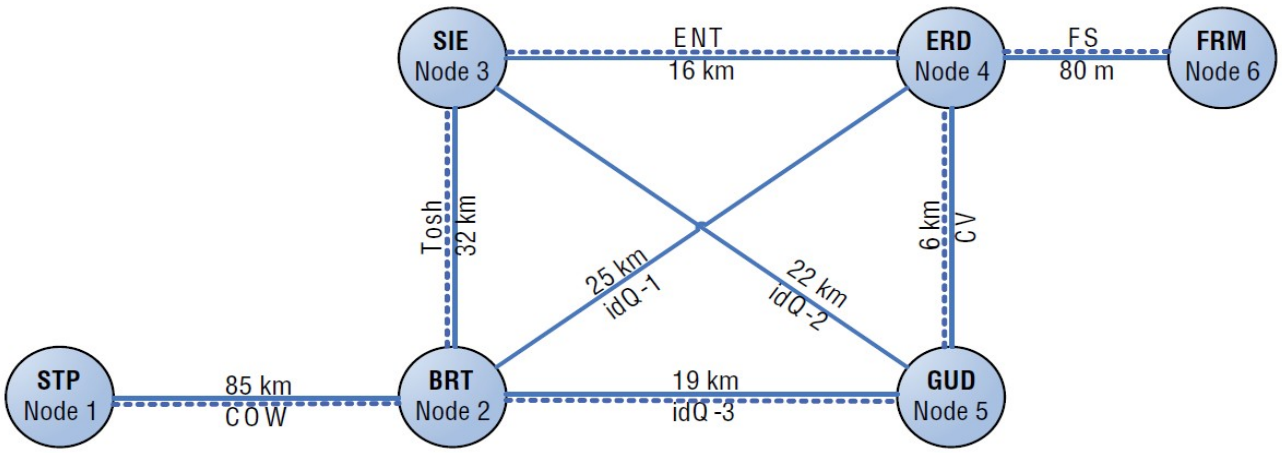

Figura 17: topologia di rete della SECOQC QKD. Le linee sontinue rappresentano i canali a comunicazione quantistica, le linee tratteggiate indicano i canali a comunicazione classica

Figura 18: mappa satellitare con le posizioni dei nodi della rete

I collegamenti QKD sono stati integrati in una rete composta da sei nodi. La distanza media tra i nodi è compresa tra 20 e 30 chilometri e il collegamento più lungo è di 83 km (vedere le Figure 17, 18 e 19).

La rete SECOQC ha introdotto un ulteriore vincolo che le chiavi segrete iniziali (necessarie per l'autenticazione) sono condivise tra i nodi adiacenti (cioè i nodi collegati direttamente da un link QKD) e non tra una qualsiasi coppia arbitraria. Questo vincolo assicura che il numero iniziale di segreti da condividere sia proporzionale (per le wide-area networks) con il numero dei nodi della rete e non con il suo quadrato. Questo a sua volta semplifica largamente l'inizializzazione di una rete QKD e l'introduzione di altri nodi durante il funzionamento.

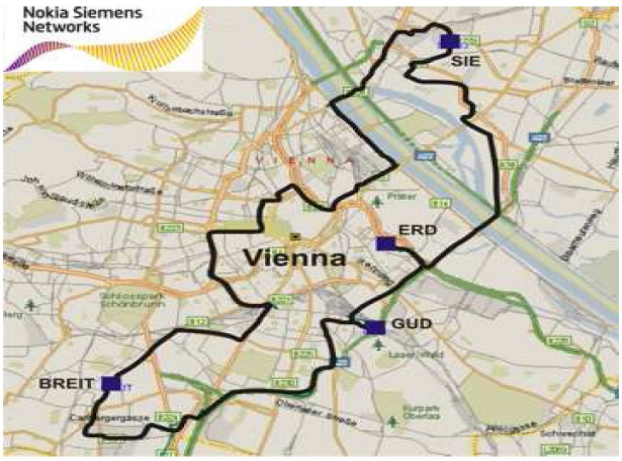

Figura 19: mappa della città di Vienna con la rete ad anello in fibra e i nodi SIE, ERD, GUD e BREIT

Una caratteristica essenziale della SECOQC è la sua architettura di rete e un fattore importante di questa architettura è il progetto del nodo. Come descritto nel paragrafo precedente, il nodo contiene essenzialmente l'insieme dei circuiti che gestiscono le chiavi generate sul collegamento QKD e che garantiscono i servizi di crittografia (cifratura e autenticazione) per il trasporto di informazioni segrete. Allo stesso tempo, ogni dispositivo QKD è dotato di un meccanismo di comunicazione classica dispositivo-a-dispositivo, di servizi per la gestione delle chiavi (le chiavi di autenticazione iniziale e le successive) e per la crittografia (autenticazione) in modo da avere la capacità di produrre una chiave.

Per superare questa ridondanza, SECOQC ha presentato il seguente approccio: i dispositivi QKD sono spogliati delle funzionalità di stand-alone, hanno accesso solamente al canale quantistico e la classica comunicazione con un nodo dedicato, chiamato nodo modulo (progettato e realizzato dall'Austrian Research Centers). Il nodo modulo gestisce tutto il materiale della chiave del dispositivo QKD sottostante e fornisce un canale classica autenticato per il dispositivo. In questo senso, l'unico obiettivo del dispositivo QKD è quello di comunicare attraverso il canale quantistico e distillare e mettere una chiave al nodo, utilizzando le strutture di comunicazione di questi ultimi. Il nodo, a sua volta gestisce la connessione punto-a-punto (compresa la comunicazione classica con i vicini, la gestione delle chiavi, e i servizi di crittografia) per essere in grado di trovare i percorsi per le destinazioni richieste e realizzare i protocolli di trasporto sicuri [4,24]. La Figura 18 mostra la posizione geografica dei nodi della rete SECOQC. I nodi ERD e FRM si trovano in diversi edifici nella zona di ERD e FRM per questo motivo non è rappresentato. Sulla mappa le diagonali tra SIE e GUD da un lato, e tra ERD e BRT, dall'altro, sono apparentemente mancanti. Infatti queste connessioni quantistiche dirette passano oltre ERD e GUD, rispettivamente, come può essere facilmente dedotto dalla Figura 17 confrontando la lunghezza dei collegamenti corrispondenti.

Ci soffermiamo brevemente sul collegamento tra ERD e FRM: sviluppato presso l'Università di Monaco di Baviera utilizza il protocollo BB84 usando impulsi laser attenuati di polarizzazione codificata con una lunghezza d'onda di 850 nm.

Path index	Nodes	Generated key
1	$s \sim t$	$p_1(s,t) = c(s,t)$
2	$s \sim u \sim t$	$p_2(s,t) = \min[c(s,u), c(u,t)]$
3	$s \sim v \sim t$	$p_3(s,t) = \min[c(s,v), c(v,t)]$
4	$s \sim u \sim v \sim t$	$p_4(s,t) = \max[0, \min[c(s,u) - c(u,t), c(u,v), c(v,t) - c(s,v)]]$
5	$s \sim v \sim u \sim t$	$p_5(s,t) = \max[0, \min[c(s,v) - c(v,t), c(u,v), c(u,t) - c(s,u)]]$

Tabella 4: percorsi non ciclici tra il nodo s e il nodo t nel grafo della rete

Il sistema può essere utilizzato durante la notte e il giorno, utilizzando i filtri eccessivi in modo da sopprimere la luce di sfondo. Il trasmettitore (Alice) utilizza diodi laser per la generazione delle sequenze casuali di WCPs di polarizzazioni differenti e del numero medio di fotoni. Nell'installazione presso la sede della Siemens di Vienna, i fotoni passano su un canale quantistico con distanza di 80m (Figura 20).

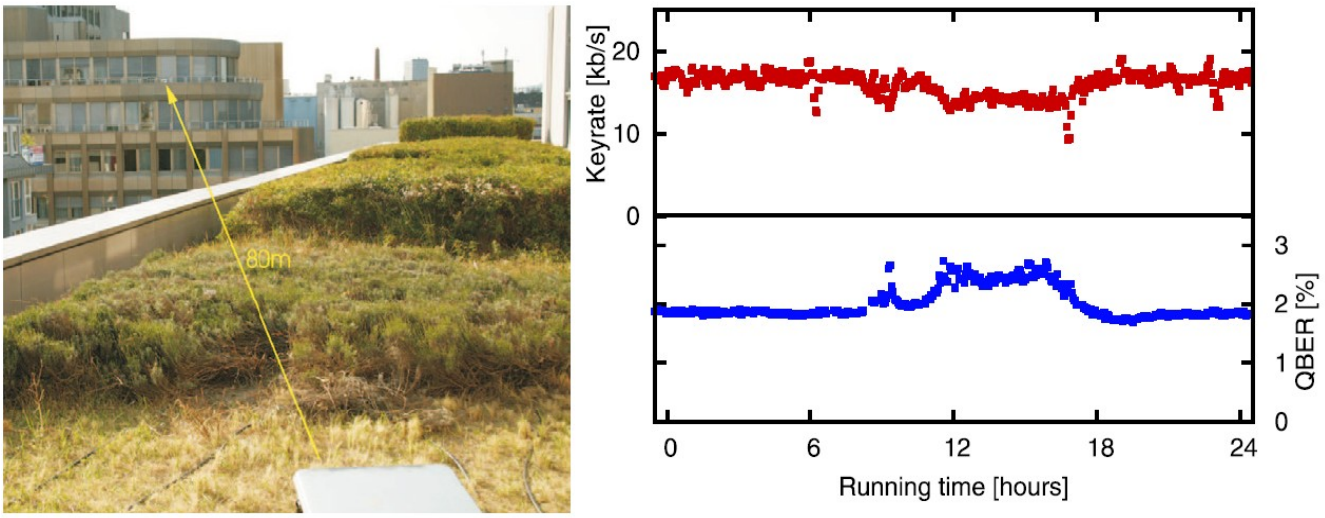

Figura 20: Sinistra: vista dal trasmettitore FS al ricevitore installato sull'edificio dopo la strada. Destra: grafico rappresentante il rate di bit sicuri con il relativo QBER

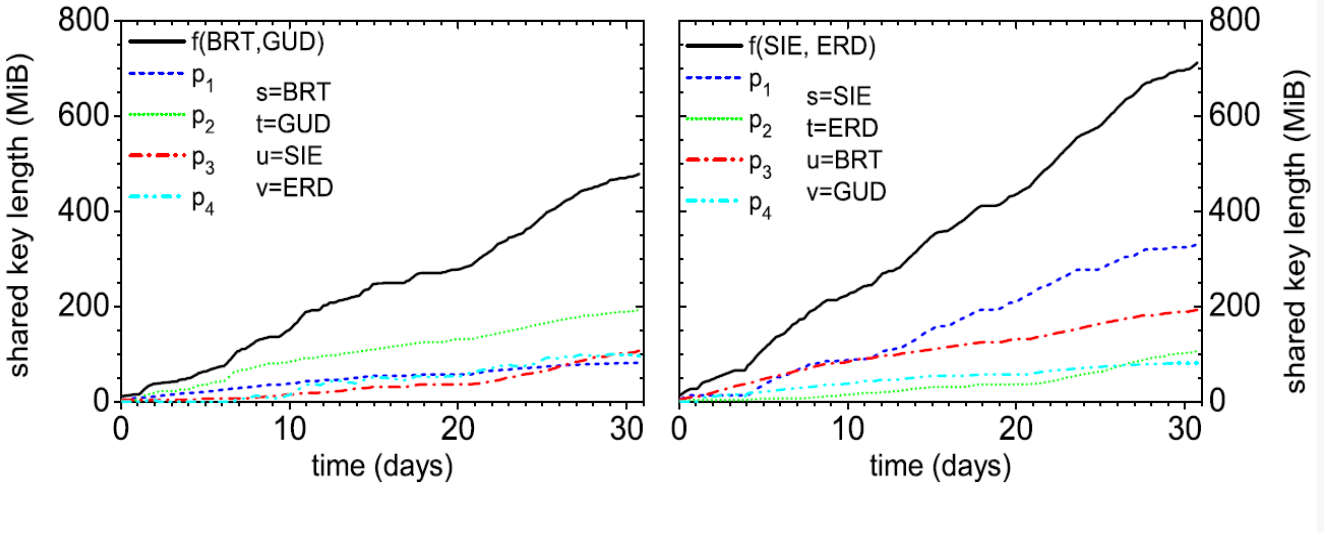

Figura 21: Sinistra: lunghezza massima della chiave condivisa tra i nodi BRT (n2) e GUD (n5) (linea continua) e la corrispondente generazione della chiave condivisa lungo percorsi individuali (linee tratteggiate), tutte in funzione del tempo. Destra: lo stesso per i nodi SIE (n3) e ERD (n4). Per il significato di s, t, u, v e p1, p2, p3, p4 vedere la tabella 4

Per concludere, diamo una descrizione della capacità di trasmissione massima tra due nodi (e come trovare ottimale (routing) delle strategie che permettono di realizzare questo massimo). Vogliamo determinare la lunghezza massima della chiave condivisa (o equivalentemente, il massimo rate di generazione della chiavi condivise) tra due nodi, se l'intera rete è dedicato solo a tale compito. Utilizzando il OTP, la lunghezza massima della chiave è uguale alla lunghezza massima di un

messaggio che può essere scambiato con informazioni teorico-sicurezza (in una delle due direzioni) tra questi due nodi. In questo senso, siamo in grado di visualizzare questa quantità come massima capacità di trasmissione segreta. In generale, si può ricorrere ai metodi della teoria dei grafi. Qui questo problema è noto come il problema del flusso massimo, a volte chiamato anche problema del flusso massimo tra s e t (dove s e t denotano la sorgente e la destinazione, rispettivamente).

La Figura 21 mostra la lunghezza massima della chiave condivisa tra i nodi BRT (n_2) e GUD (n_5) e SIE (n_3) e ERD (n_4), rispettivamente, e i contributi dei percorsi separati come indicato nella Tabella 4. Ovviamente, la capacità di trasmissione segreta punto-a-punto è migliore rispetto a quella del collegamento punto-a-punto diretto grazie allo sfruttamento delle capacità degli altri percorsi di collegamento. La ridondanza della rete garantisce una crescita quasi lineare della chiave disponibile a discapito dei blocchi temporanei dei singoli collegamenti. Nella Figura 20 la crescita della chiave condivisa è espressa in MiB (detto anche mebibyte, **me**ga **bi**nary **byte**, 1 mebibyte = 1.048.576 byte).

Per inciso, in entrambi gli esempi le capacità del collegamento ha portato alla esclusione del percorso 5 in tutto l'esperimento. Inoltre, il percorso 4 non è contribuire in modo costate alla massima capacità di trasmissione. Nella parte sinistra della Figura 20, vi sono due periodi di inattività del percorso 4, mentre nella parte destra vi è un solo periodo di tale durata di circa un giorno dall'inizio dell'esperimento.

L'analisi appena fatta non intende essere esaustiva, in qualsiasi forma, ma solo per dare ulteriori approfondimenti in considerazione al valore aggiunto fornito dalla rete rispetto al collegamento stand-alone QKD [25].

5.3.2 Tokyo

In questo paragrafo presentiamo un'altra rete QKD metropolitana presente a Tokyo, e per cui chiamata Tokyo QKD Network, i cui nodi non sono sotto il controllo di una spia e dove sono state installate le più recenti tecnologie QKD e un'interfaccia d'applicazione aggiornata. In questo paragrafo verrà presentata a grandi linee la struttura topologica della rete e verranno illustrate le due principali applicazioni di questa rete: la conferenza visiva sicura soggetta ad un attacco di una spia e l'applicazione del OTP alla telefonia mobile. Tutte queste applicazioni sono state dimostrate pubblicamente durante il meeting di inaugurazione della rete nell'ottobre dello scorso anno.

Schema del QKD Tokyo Network

La rete di Tokyo QKD è costituita da parti del banco di prova aperto della rete NTIC chiamato Giappone Giga Bit Network 2 plus (JGN2plus), come mostrato in Figura 22. La rete ha quattro punti di accesso posizionati nei quartieri di Koganei, Otemachi, Hakusan, e Hongo. Gli access point sono collegati da un fascio di fibre commerciali e includono molti connettori e punti di giunzione i quali vengono gestiti in parte in modo non interrato. La percentuale di fibre non interrate è di circa il 50% e questo causa un aumento dell'attenuazione e del rumore con le variazioni ambientali ai vari collegamenti. Il tasso di perdita è di circa 0.3dB/km in media per il collegamento tra i quartieri Koganei e Otemachi, e di 0,5 dB/km in media per gli altri due link. Viene spesso osservato il fenomeno del crosstalk tra fibre vicine il quale conduce ad un aumento delle perdite di fotoni.

Nel 2010, nove organizzazioni provenienti dal Giappone e dall'Unione Europea hanno partecipato alla operazione di Tokyo Network QKD. Sul lato giapponese vi erano, oltre alla NTIC, le tre società NEC, Mitsubishi Electric Corporation (Mitsubishi) e NTT, mentre dal lato europeo Toshiba Research Europe Ltd. (TREL) dal Regno Unito, ID Quantique (IDQ) proveniente dalla Svizzera e tre organizzazioni provenienti dall'Austria: l'Austrian Institute of Technology (AIT), l'Institute of Quantum Optics e il Quantum Information (IQOQI) dell'University of Vienna i quali formano insieme una squadra soprannominata "All Vienna". La rete ha sei collegamenti QKD e sono disposti come mostrato in

Figura 22. In modo sintetico, i collegamenti sono cosi descritti:

- Collegamento n°1: implementato dalla Mitsubishi, usa il protocollo BB84 in un collegamento di 24 km nella configurazione ad anello tra Otemachi e Hakusan.
- Collegamento n°2: implementato dalla NEC, usa anch'egli il protocollo BB84 in un collegamento di 45 km tra Koganei e Otemachi, con un rivelatore di singolo fotone a superconduttore della NTIC (SSPD).
- Collegamento n°3: implementato dalla NTT, usa il DPS-QKD e ha accettato la sfida della lunga distanza di 90 km in una configurazione a ciclo anch'egli con l'SSPD della NTIC.
- Collegamento n°4: implementato da "All Vienna", questi hanno usato il protocollo BBM92 con fibre installate nei locali della NTIC.
- Collegamento n°5: implementato dalla TREL, questo gruppo ha usato la codifica a stati del protocollo BB84 utilizzando fotodiodi a valanga auto-differenzianti raffreddati elettricamente (SD-APD) in un collegamento di 45 km.
- Collegamento n°6: implementato dalla IDQ, questa applica il suo sistema commerciale utilizzando il protocollo SARG04 nel collegamento tra i quartieri Otemachi e Hongo.

Questa logica configurazione di collegamento costituisce una rete di tipo a maglia di 6 nodi, come mostrato in Figura 23.

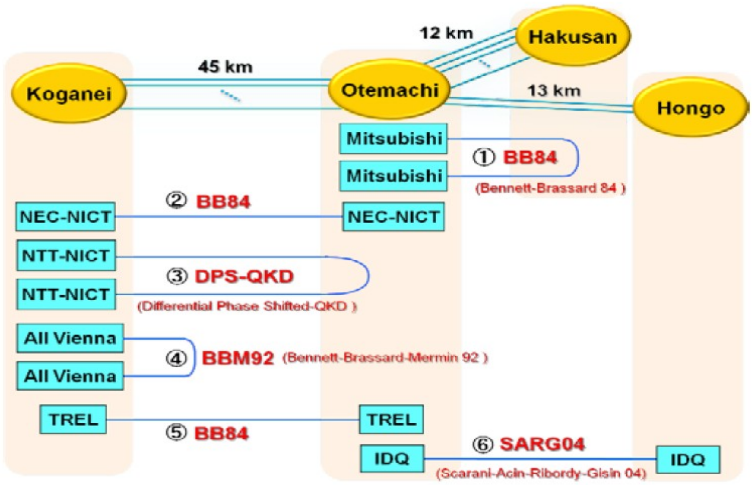

Figura 22: configurazione fisica dei collegamenti della Tokyo QKD Network

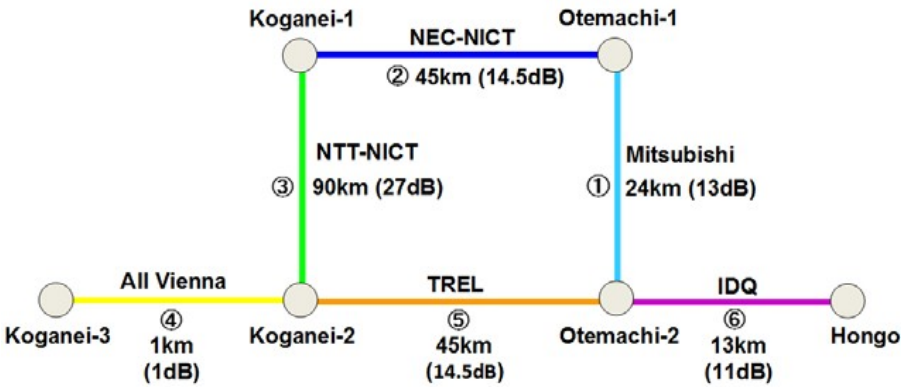

Figura 23: configurazione logica dei collegamenti con 6 nodi

Abbiamo adottato una architettura a triplo strato basata sul relay chiave attraverso i nodi di fiducia, come mostrato in Figura 24, simile alla rete SECOQC vista nel precedente paragrafo. Lo strato quantistico è costituito da collegamenti quantistici punto-a-punto, formando la Quantum BackBone (QBB).

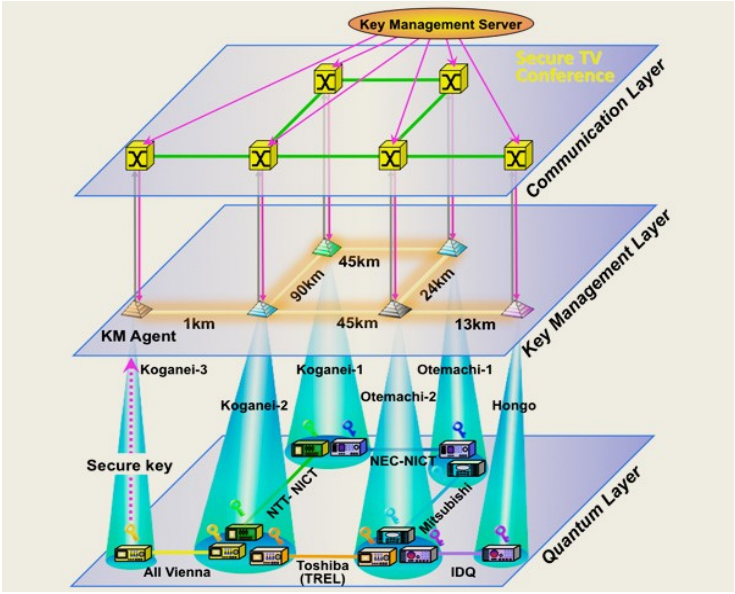

Figura 24: architettura a tre strati della Tokyo QKD Network. Essa consiste dello strato quantistico, di gestione delle chiavi e di comunicazione

Ogni link genera la chiave di sicurezza a modo suo. I protocolli di QKD, così come il formato e le dimensioni della chiave possono essere arbitrari. I dispositivi QKD posti al livello quantistico inviano la chiave allo strato centrale il quale viene detto strato di gestione delle chiavi. In questo strato, un dispositivo di gestione delle chiavi (indicato con KMA, Key Management Agent) che si trova in ogni sito, riceve la chiave tramite l'interfaccia API. Ogni KMA è in realtà un PC, fisicamente protetto, e funzionnte di conseguenza come un nodo di fiducia. Le funzioni di rete sono eseguite interamente in questo strato KM da un software. Un KMA può trasmettere una chiave sicura condivisa con un nodo a un secondo nodo cifrandola con il OTP usando un'altra chiave condivisa con il secondo nodo. Così una chiave sicura può essere condivisa tra i nodi che non sono direttamente collegati tra loro da un collegamento quantistico. Nel livello di comunicazione, la comunicazione sicura è assicurata dalle chiavi distribuite per la crittografia e la decrittografia dei dati di testo, audio o video prodotti da diverse applicazioni. Infine, un server di gestione delle chiavi (indicato con KMS, Key Management Server) viene introdotto per una gestione centralizzata del ciclo di vita della chiave e la fornitura dei percorsi sicuri. Questo differenzia la Tokyo QKD Network dalla rete SECOQC vista nel precedente paragrafo. In quest'ultima, il problema del percorso sicuro è risolto da un algoritmo di ricerca autonomo seguendo approcci standard in rete. La ragione principale per l'adozione della gestione centralizzata nella Tokyo QKD Network è che questa assume un banco di prova per un governo-commercialista di rete o per una rete di infrastrutture critiche che spesso hanno un dispaccio centrale o un server dati centrale.

L'implementazione fisica dei tre strati dell'architettura è rappresentato dallo schema elettrico di Figura 25. Le linee blu rappresentano le fibre ottiche per lo strato quantistico. Per i collegamenti n°1, 3, 5 e 6 una seconda fibra è stata utilizzata per inviare le informazioni classiche richieste per il protocollo QKD e per i segnali di sincronizzazione, secondo le specifiche originali della Tokyo QKD Network. Per i

collegamenti 2 e 4, l'informazione classica è stata inviata attraverso una seconda fibra nella rete locale (mostrato come un colore diverso in Figura 25). La stessa rete è collegata a Internet tramite un router.

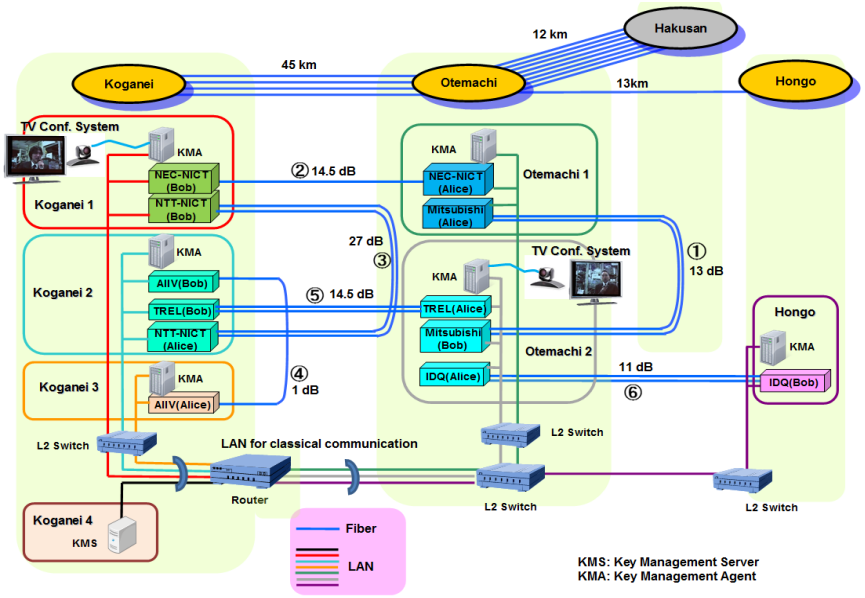

Figura 25: schema di collegamento della Tokyo QKD Network

Le prestazioni tecnologiche della QKD sono stata migliorate negli ultimi anni grazie al progresso delle moderne tecnologie, come i rivelatori di fotoni che operano a velocità più elevate con basso rumore. Nella Tokyo QKD Network, i sistemi ad alta velocità QKD sviluppati da NEC-NTIC e TREL hanno permesso una video conferenza in tempo reale in assoluta sicurezza in un'area metropolitana. Il sistema DPS-QKD sviluppato da NTT permette in tempo reale a distanza la comunicazione vocale sicura. La Mitsubishi ha combinato il suo sistema QKD con un'applicazione di telefonia sicura su smartphone. Un sistema commerciale affidabile e altamente stabile è stato dimostrato dalla IDQ. Infine, "All Vienna" ha contribuito con un impressionante sistema di nuova generazione che sfrutta l'entanglement quantistico.

La video conferenza e la telefonia mobile sicure

La dimostrazione dal vivo della video conferenza sicura, del rilevamento di intercettazioni, e della conseguente deviazione dei collegamenti QKD nella Tokyo QKD Network è stata effettuata e resa pubblica nel mese di ottobre dello scorso anno. La configurazione della video conferenza sicura è rappresentata in Figura 26. Nella video conferenza sicura, vari Polycom Video Conference System sono stati istituiti nei quartieri Koganei-1 e Otemachi-2 che sono stati collegati direttamente tramite il JGN2plus L2-VPN. Un flusso video dal vivo per questa VPN è stato crittografato con il OTP nei vari KMA usando la modalità a chiave memorizzata. Il tasso di crittografia è stato di 128 kbps e le chiavi sicure sono state fornite da uno dei due seguenti percorsi QKD: uno è tramite Koganei-2 con una distanza totale di 135 km, e l'altro è tramite Otemachi-1 con una distanza totale di 69 km, come indicato dalle linee rosse e blu, rispettivamente.

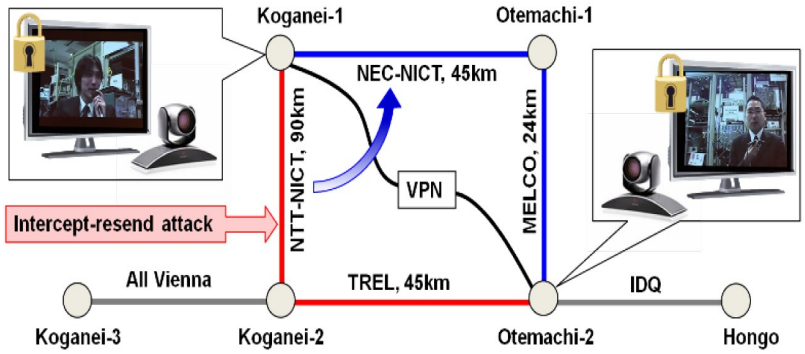

Figura 26: la configurazione di rete per la video conferenza sicura tra Koganei e Otemachi. La VPN usata per la video conferenza uò scegliere tra due diversi percorsi indicati con le linee rossa e blu

Figura 19 (a) mostra uno screenshot della schermata di gestione delle chiavi, indicando che i 90 km QKD collegamento funziona perfettamente. Il link è stato poi attaccato da intercettare un flusso di fotoni dalla fibra attraverso uno specchio riflettente e alta iniezione di un laser CW con la stessa potenza di quello sfruttato nella fibra. Il KMS rilevato questo attacco in pochi secondi a causa di un improvviso aumento della QBER, interrotto il processo di QKD nel link, e ha sollevato un allarme. Figura 19 (b) è una copia dello schermo KMS poco dopo la rilevazione dell'attacco. Il KMAs a Koganei-1 e Koganei-2 aveva del materiale conservato chiave sicura, e la conferenza TV sicuro potrebbe continuare per un po '. Il KMS immediatamente commutato la strada secondaria per poter continuare con la distribuzione delle chiavi prima che il buffer delle chiavi a corto di chiave. Così conferenza TV sicuro potrebbe andare avanti senza ostacoli e la sicurezza era garantita. Un relè chiave è stato anche testato e utilizzato con successo non solo per lo streaming video di cui sopra sicura, ma anche in fase di test vie di relè diversi tra cui i nodi Koganei-3 e Hongo con le squadre di tutti i Vienna e IDQ.

La Mitsubishi ha applicato un esca-stato BB84 sistema per il collegamento 24 km con una perdita totale di 13dB. Un diagramma schematico della configurazione è mostrato in fig. 11. In questo sistema, una sorgente laser con una larghezza di impulso di 500 ps è operato a 100MHz. Le fonti di luce classico e quantistico sono progettati utilizzando DWDM (Dense Wavelength Division Multiplexing) moduli laser DFB a lunghezze d'onda delle telecomunicazioni (segnale quantistica: 1549,32 nm; classico segnale 1550,92 nm). Sincronizzazione di tempo è stato fatto da segnali classica attraverso una seconda fibra ottica. In questo test sul campo, la proprietà di polarizzazione del canale è stato relativamente stabile.

Veloce distillazione chiave è realizzato con PC standard solo in attuazione del software, utilizzando l'algoritmo migliorato per l'amplificazione della privacy. Per la correzione degli errori, la bassa densità di parità controllo (LDPC) è adottato di ottenere una performance avvicina limite di Shannon. Per l'amplificazione della privacy, il tempo di calcolo è ridotto da O (n2) a O (nlog (n)) per la dimensione del blocco n utilizzando l'algoritmo di trasformazione rapida di Fourier per moltiplicare la matrice Toeplitz e una chiave riconciliato. La riduzione è pari a 4 ordini di grandezza per n = 106, che attualmente è noto per essere la dimensione del blocco minimo per eliminare l'effetto dimensione finita nel distillare la chiave di sicurezza. Il tasso chiave di sicurezza è stato di 2 kbps e il QBER è di circa 4,5%. Abbiamo confermato la stabilità della generazione delle chiavi. Figura 12 mostra i risultati sperimentali di funzionamento continuo. Stabile la generazione di chiavi per circa 3 giorni è stata dimostrata.

Veloce distillazione chiave è realizzato con PC standard solo in attuazione del software, utilizzando

l'algoritmo migliorato per l'amplificazione della privacy. Per la correzione degli errori, la bassa densità di parità controllo (LDPC) è adottato di ottenere una performance avvicina limite di Shannon. Per l'amplificazione della privacy, il tempo di calcolo è ridotto da O (n2) a O (nlog (n)) per la dimensione del blocco n utilizzando l'algoritmo di trasformazione rapida di Fourier per moltiplicare la matrice Toeplitz e una chiave riconciliato. La riduzione è pari a 4 ordini di grandezza per n = 106, che attualmente è noto per essere la dimensione del blocco minimo per eliminare l'effetto dimensione finita nel distillare la chiave di sicurezza. Il tasso chiave di sicurezza è stato di 2 kbps e il QBER è di circa 4,5%. Abbiamo confermato la stabilità della generazione delle chiavi. Figura 12 mostra i risultati sperimentali di funzionamento continuo. Stabile la generazione di chiavi per circa 3 giorni è stata dimostrata.

6 CONCLUSIONI

6.1 CONFRONTO CRITTOGRAFIA CLASSICA CON QKD

6.2 PROSPETTIVE PER IL FUTURO

Molti protocolli di QKD, come il senso unico protocollo BB84, hanno dimostrato di essere assolutamente sicuro, il che significa che il protocollo non può essere 'rotto' fino a quando le leggi della fisica rimangono vere. D'altra parte implementazioni reali imperfezioni sono inevitabili e sarà quindi suscettibili di canale laterale attacchi. Per garantire la sicurezza delle implementazioni mondo reale è importante per caratterizzare e definire le ipotesi sottostanti. Inoltre, i canali indesiderati dovrebbero essere indagati e messo a punto contromisure appropriate. A questo proposito, prove preliminari di Rete Tokyo QKD è necessario continuare a scoprire e caratterizzare le feritoie attuazione e incorporare contromisure appropriate. Condividere le informazioni su tali attività con i vari banchi di prova della rete QKD in tutto il mondo istituirà un denominatore comune per la standardizzazione QKD nel contesto della certificazione di sicurezza. Una forte rete QKD dovrebbe incorporare sistemi di stabilizzazione attiva e altrettanti canale laterale contromisure possibile, senza sacrificare le prestazioni.

Vorremmo affrontare altri tre numeri successivi: il primo è quello di sviluppare una tecnologia più efficiente gestione delle chiavi per permettere un'estensione scalabile per configurazioni di rete multipunto con le applicazioni multicast. Introducendo un nuovo metodo di elaborazione del nodo basata sulla rete di codifica nel livello di comunicazione può anche essere utile per ridurre il consumo di chiave di sicurezza.

Il secondo è quello di integrare le tecnologie di rete QKD in un'infrastruttura emergente delle comunicazioni ottiche, chiamata rete fotonica, dove vengono elaborati i segnali ottici nel dominio ottico senza conversione al formato elettronico. Rete quantistica alla fine dovrebbe essere fatta nelle infrastrutture di reti WDM percorso ottico, che corrisponde al livello molto basso di reti fotoniche. In base alle richieste degli utenti, un collegamento efficiente QKD sarà realizzato direttamente via croce ottica collega e ottiche riconfigurabili add-drop multiplexer. La gamma di trasparenza quantistica si allarga, le chiavi di sicuro può essere utilizzata per crittografare strettamente i segnali di controllo nel piano di controllo. In realtà, le tecnologie di controllo innovative come la generalizzata multi-protocol label switching rendere il piano di controllo più aperto agli strati superiori, altri operatori e utenti finali. Hacker può anche avere la possibilità di accedere facilmente al piano di controllo che potrebbero mettere in serio pericolo la sicurezza di tutta la rete. QKD giocherà un ruolo chiave nello sviluppo di proteggere le reti fotoniche.

L'ultimo è quello di ampliare le applicazioni QKD non solo per proteggere la riservatezza dei dati, ma anche di fornire servizi, che sono funzioni essenziali nei sistemi di sicurezza attuali, come l'autenticazione dei messaggi, l'identificazione e la firma digitale. Mentre QKD supporta facilmente l'autenticazione "assolutamente sicuro" messaggio, l'identificazione e la firma digitale richiedono ulteriori ricerche e può essere realizzato utilizzando hardware già esistenti QKD a scapito di assumere un avversario delimitata da memoria finita quantistica o supplementari (di prossima generazione) risorse quantistica a disposizione del legittimo partiti come quantum di elaborazione e di memoria quantistica. In ogni caso, se le prestazioni QKD è ulteriormente migliorata e costi ridotti, allora potenziali reti QKD (con le funzionalità di cui sopra) potrebbe diventare un'infrastruttura essenziale per assicurare la generazione di chiavi per una vasta gamma di obiettivi di crittografia. Questo potrebbe

essere lo stimolo principale per perseguire il miglioramento della tecnologia QKD e futuro della ricerca rete QKD.

7 APPENDICE

7.1 NOTAZIONE BRA-KET E SPAZIO DI HILBERT

La notazione bra-ket è una notazione standard per la descrizione di stati quantici nella teoria della meccanica quantistica, essa è composta da parentesi angolari e da barre verticali. La notazione bra-ket, detta anche notazione di Dirac, puo' anche essere usata per indicare vettori astratti e funzionali lineari in matematica. È così chiamato perché il prodotto interno di due stati è indicato da un **bra**c**ket**, $\langle\phi|\psi\rangle$, costituito da una parte sinistra, $\langle\phi|$, chiamata **bra**, e da una parte destra, $|\psi\rangle$, chiamata **ket**. La notazione è stata introdotta nel 1930 da Paul Dirac.

Nella meccanica quantistica, quasi ogni fenomeno, tra cui una gran parte della moderna fisica, viene generalmente spiegato con l'aiuto della notazione bra-ket. L'espressione $\langle\phi|\psi\rangle$ è generalmente interpretata come l'ampiezza di probabilità per lo stato ψ di collassare nello stato ϕ.

Il concetto matematico di spazio di Hilbert, introdotto dal matematico David Hilbert, generalizza la nozione di spazio euclideo. Esso estende i metodi di algebra e calcolo vettoriale dalle due dimensioni del piano bidimensionale euclideo e dalle tre dello spazio tridimensionale a spazi con un numero finito o infinito di dimensioni. Uno spazio di Hilbert è uno spazio vettoriale astratto nel quale è definito un prodotto interno che permette la misura della lunghezza e dell'angolo.

I primi spazi di Hilbert sono stati studiati da questo punto di vista nel primo decennio del 20° secolo da David Hilbert, Erhard Schmidt, e Frigyes Riesz. Sono strumenti indispensabili, oltre che per la meccanica quantistica, per le teorie sulle equazioni differenziali alle derivate parziali, per l'analisi di Fourier (che include applicazioni di elaborazione del segnale e il trasferimento di calore) e per la teoria ergodica che costituisce la matematica alla base dello studio della termodinamica.

Un elemento di uno spazio di Hilbert può essere specificato in modo univoco dalle sue coordinate rispetto a un insieme di assi coordinati (una base ortonormale), in analogia con le coordinate cartesiane del piano.

Uno spazio di Hilbert H è uno spazio di prodotto interno reale o complesso che è anche uno spazio metrico completo rispetto alla funzione distanza indotta dal prodotto interno. Dire che H è uno spazio di prodotto interno complesso significa che H è un complesso spazio vettoriale su cui vi è un prodotto interno $\langle x,y\rangle$ associando un numero complesso per ogni coppia di elementi x, y di H che soddisfa le seguenti proprietà:

$\langle y,x\rangle$ è il complesso coniugato di $\langle x,y\rangle$;

$$\langle y, x\rangle = \overline{\langle x, y\rangle}$$

$\langle x,y\rangle$ è lineare nel suo primo argomento. Per tutti i numeri complessi *a* e *b*,

$$\langle ax_1 + bx_2, y\rangle = a\langle x_1, y\rangle + b\langle x_2, y\rangle$$

il prodotto interno $\langle \bullet,\bullet\rangle$ è definito positivo:

$$\langle x, x\rangle \geqslant 0$$

dove il caso di uguaglianza si ha solamente quando $x = 0$.

segue dalle proprietà 1 e 2 che un prodotto interno complesso è antilineare nel suo secondo argomento:

$$\langle x, ay_1 + by_2\rangle = \bar{a}\langle x, y_1\rangle + \bar{b}\langle x, y_2\rangle$$

Uno spazio reale di prodotto iunterno è definito allo stesso modo, tranne che H è uno spazio vettoriale reale e il prodotto interno assume valori reali. Tale prodotto interno sarà bilineare: cioè, lineare in ogni

argomento.

La norma definita dal prodotto interno $\langle \bullet, \bullet \rangle$ è la funzione reale

$$\|x\| = \sqrt{\langle x, x \rangle}$$

e la distanza tra due punti x, y in H è definita in termini della norma da:

$$d(x, y) = \|x - y\| = \sqrt{\langle x - y, x - y \rangle}$$

Questa funzione è una funzione di distanza: è simmetrica in x e y, la distanza tra x e se stesso è zero ed è valida la disuguaglianza triangolare, il che significa che la lunghezza di un lato di un triangolo xyz non può superare la somma delle lunghezze degli altri due lati:

$$d(x, y) \leq d(x, y) + d(y, z)$$

Quest'ultima proprietà è una una conseguenza della disuguaglianza fondamentale di Cauchy-Schwarz la quale afferma che:

$$|\langle x, y \rangle| \leq \|x\| \|y\|$$

con uguaglianza se e solo se x e y sono linearmente dipendenti.

7.2 FUNZIONI DI HASH

7.3 ALGORITMI RSA E DIFFIE-HELMAN

Ringraziamenti

prof. Laurenti Nicola, prof. Kurt Lechner, prof. Marchetti Pieralberto, prof. Finesso Lorenzo, prof. Simonetto Franco, dott. Occhipinti Tommaso,

Bibliografia

[1] Filippo Corsi; “Presentazione didattica della crittografia quantistica e dei concetti collegati”; Università degli Studi di Milano; Anno Accademico 2006-2007

[2] Alice Della Penna; “La Crittografia Quantistica”; Università degli Studi di Bari; Anno Accademico 2005-2006.

[3] Tommaso Occhipinti; “La crittografia quantistica: stato dell'arte”; Notiziario Tecnico Telecom Italia; Anno 17 n.2 – Agosto 2008.

[4] Donna Dodson, Mikio Fujiwara, Philippe Grangier, Masahito Hayashi, Kentaro Imafuku, Ken-ichi Kitayama, Prem Kumar et al.; “Updating Quantum Cryptography Report ver. 1 (May 2009)”.

[5] Chi-Hang Fred Fung, Xiongfeng Ma e H. F. Chau; “Pratical issue in quantum-key-distribution postprocessing”; Phy Rev A.81.012318.

[6] http://www.apav.it/sitostudenti/sito%20giur/federica/crit_stat.htm

[7-11] www.wikipedia.org

[12] M. A. Sfaxi, S. Ghernaouti-Hélie, G. Ribordy, O. Gay; “Using Quantum Key Distribution within IPSEC to secure MAN communications”, MAN 2005 conference, 2005.

[13] Mohamed Elboukhari, Mostafa Aziz e Abdelmalek Azizi; “Improving TLS Security By Quantum Cryptography”; International Journal of Network Security & Its Applications (IJNSA), Vol.2, N.3, 2010.

[14-18] www.wikipedia.org

[19] Vito Giovanni Lucivero, “Entanglement: Teoria, Esperimenti e Applicazioni in Crittografia”; Università degli Studi di Bari; Anno Accademico 2006-2007.

[20] Hao Yan et al.; “Information Reconciliation Protocol in Quantum Key Distribution System”; International Conference on Natural Computation ICNC (2008).

[21] A. P. Makkaveev, S. N. Molotkov, D. I. Pomozov and A. V. Timofeev; “Practical Error-Correction Procedures in Quantum Cryptography”;

[22] Chip Elliot; “Building the quantum networm”; New Journal of Physics 4 (2002) 46.1-46.12.

[23] M. Sasaki et al.; “Field test of quantum key distribution in the Tokyo QKD Network”; 270.0270 Quantum optics; 270.5568 Quantum cryptography.

[24] M Peev et al.; “The SECOQC quantum key distribution network in Vienna”; New Journal of Physics 11 (2009).

[25] A. Poppe, M. Peev e O. Maurhart; “Outline Of The SECOQC Quantum-Key-Distribution Network In Vienna”; International Journal of Quantum Information Vol. 6, No. 2 (2008).

[26] G. Brassard and L. Salvail. Secret-key reconciliation by public discussion. Advances in Cryptography - Eurocrypt ’93, Lecture Notes in Computer Science, 1993:410-423, 1993.