Kvantresistent kryptering

Referat. De asymmetriska krypteringsalgoritmer som används allmänt idag för säker kommunikation skulle enkelt kunna forceras genom användning av Shors algoritm i en tillräckligt kraftfull kvantdator. Dagens kraftfullaste kvantdatorer klarar dock inte av att forcera någon riktig kryptering. I denna avhandling sammanställs information om krypteringsalgoritmer som kan köras på vanlig klassisk hårdvara och som antas kunna motstå attacker utförda med hjälp av en eventuell framtida kraftfull kvantdator, samt vilket nuvarande läget är angående implementationer av dessa algoritmer. En målsättning är att ta reda på vilket hot kvantdatorer eventuellt skulle kunna utgöra mot datasäkerhet och personlig integritet i framtiden. Kvantkryptering, som utnyttjar kvantmekaniska egenskaper genom specialiserad hårdvara för att skydda information, behandlas inte i denna avhandling.

Nyckelord: kvantdator, asymmetrisk kryptering, primtal, krypto-system, personlig integritet

Åbo Akademi
Fakulteten för naturvetenskaper och teknik
Skribent: Jens Lassus
Handledare: Jan Westerholm
2019

Innehållsförteckning

1. Inledning 1

1.1 Symmetrisk kryptering 2

1.2 Asymmetrisk kryptering 2

1.3 Envägsfunktioner 3

1.4 Shors algoritm 4

1.5 Grovers algoritm 5

1.6 Kvantdatorer 5

2. Kvantresistenta algoritmer 8

2.1 Gitterbaserad kryptering 8

2.2 Multivariat kryptering 8

2.3 Hashbaserad kryptering 9

2.4 Kodbaserad kryptering 9

3. Implementationer 10

3.1 CECPQ1 10

3.2 CECPQ2 11

3.3 Microsofts implementationer 11

4. Framtid 12

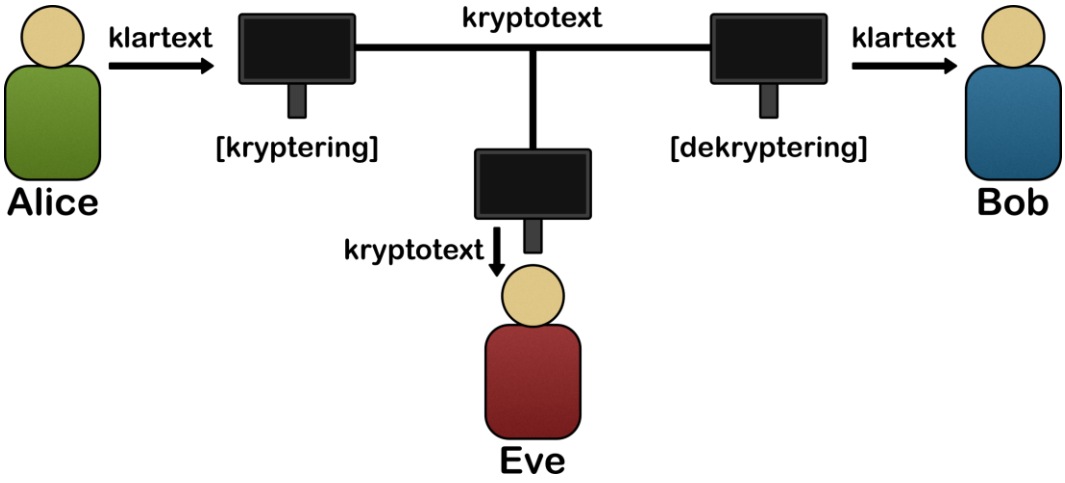
5. Diskussion 13

6. Referenser 14

1. **Inledning**

En stor del av dagens samhälle och infrastruktur är beroende av att man kan överföra information säkert över ett osäkert medium, exempelvis internet. Med säkert kan man avse att innehållet inte förvrängs, eller att ingen obehörig kan läsa innehållet. Oftast är båda dessa betydelser viktiga, men fokus kommer att ligga på hemlighållandet av informationen i denna avhandling. För att säkerställa att ingen obehörig kan läsa innehållet används *kryptering*. Därför är det mycket viktigt att denna kryptering fungerar och är säker mot attacker. Nedan följer en egen översättning av en beskrivning av kryptering [1, s. 1]. I krypteringssammanhang brukar sändaren kallas för Alice, mottagaren för Bob och tjuvlyssnaren för Eve. Scenariot illustreras i Figur 1.

> Meddelandet som ska sändas – det kan vara text, sifferdata, ett exekverbart program eller vilken annan form av information som helst – kallas *klartext*. Alice *krypterar* klartexten *m* och erhåller *kryptotexten c*. Kryptotexten *c* sänds till Bob. Bob omvandlar kryptotexten tillbaka till klartext genom *dekryptering*. För att *dekryptera* behöver Bob en bit hemlig information, en hemlig *dekrypteringsnyckel*. Tjuvlyssnaren Eve kan fortfarande fånga upp kryptotexten. Däremot borde krypteringen garantera sekretess och hindra henne från att härleda någon information om klartexten utgående från den observerade kryptotexten.



Figur 1

I detta kapitel presenteras först grunderna inom kryptering, sedan kvantalgoritmer som är av intresse i krypteringssammanhang och slutligen grundläggande information om kvantdatorer. I kapitel två förklaras funktionsprincipen för några olika typer av kvantresistenta krypteringsalgoritmer. I kapitel tre presenteras några implementationer och resultatet av tester utförda med några av dessa algoritmer.

Krypteringsalgoritmer vars uppgift är att dölja information från obehöriga kan delas upp i de två huvudkategorierna symmetrisk och asymmetrisk kryptering, som beskrivs i 1.1 och 1.2.

1.1 Symmetrisk kryptering

Symmetrisk kryptering är en form av kryptering som innebär att samma krypteringsnyckel används för både kryptering och dekryptering. Symmetrisk kryptering kallas även *secret-key encryption* på engelska, alltså ungefär *kryptering med hemlig nyckel*. Kvantdatorer verkar inte vara till någon stor nytta vid forcering av denna typ av kryptering [2, s. 6]. Därför behandlas symmetrisk kryptering endast ytligt i denna avhandling. Problemet är att man vid kommunikation inte direkt kan använda sig av symmetrisk kryptering för att undvika attacker av kvantdatorer, eftersom man behöver ett sätt att dela med sig av nyckeln, utan att någon obehörig kan fånga upp den. Detta kan göras genom användning av en nyckelutväxlings-algoritm eller med hjälp av asymmetrisk kryptering och görs ofta vid överföring av större mängder data, eftersom symmetriska krypteringsalgoritmer är mindre beräkningsintensiva än asymmetriska [1, s. 2]. För en närmare beskrivning av symmetrisk kryptering, se [1, Kap. 2].

1.2 Asymmetrisk kryptering

För att upprätta en säker kommunikationskanal över ett osäkert medium används ofta asymmetrisk kryptering, som beskrivs närmare i [1, Kap. 3]. En tidig beskrivning av en sådan algoritm publicerades 1978 och beskriver RSA-algoritmen [3] som används ännu idag. RSA står för begynnelsebokstäverna i upphovsmännens efternamn. Denna typ av kryptering använder sig av två nycklar. Ett meddelande som krypteras med en av nycklarna kan inte dekrypteras med samma nyckel, utan måste dekrypteras med den andra nyckeln. Idén är att en av nycklarna är privat och endast ägaren känner till den. Den andra nyckeln kallas publik och får sändas till vem som helst utan att det utgör en säkerhetsrisk. Fördelen med detta är att vem

som helst nu kan kryptera ett meddelande med mottagarens publika nyckel. Efter det kan endast mottagaren dekryptera meddelandet med sin privata nyckel. En annan tillämpning är signering. Om sändaren krypterar ett meddelande med sin privata nyckel kan mottagaren säkerställa sig om att sändaren verkligen skrev meddelandet genom att kontrollera om meddelandet kan dekrypteras med sändarens publika nyckel. I båda dessa tillämpningar är det viktigt att man inte utgående från en nyckel kan räkna ut den andra nyckeln. För detta används en matematisk envägsfunktion, som beskrivs i 1.3. Dessa envägsfunktioner är största orsaken till att kvantdatorer är intressanta i krypteringssammanhang och därför är det också främst de asymmetriska algoritmerna som är av intresse i denna avhandling.

1.3 Envägsfunktioner

Inom asymmetrisk kryptering, exempelvis RSA, samt i nyckelutväxlings-algoritmer, så som Diffie-Hellman, utnyttjas matematiska så kallade envägsfunktioner. De kallas envägsfunktioner eftersom de har egenskapen att de är lätta att utföra i en riktning, men svåra att utföra baklänges. Med svårt menas att det krävs en sådan mängd tid och beräkningskraft för att lösa problemet med användning av den snabbaste kända metoden, att det inte är praktiskt genomförbart. Idén är att man utgående från funktionens utdata inte ska kunna räkna ut funktionens indata. Svårighetsgraden beror på storleken av talen som används. De envägsfunktioner som används inom asymmetrisk kryptering tillhör en underkategori som på engelska kallas för *trapdoor function*. En tänkbar översättning kunde vara *falluck-funktion*. Ibland används den engelskspråkiga termen som synonym för *bakdörr*. Detta är fel, eftersom en bakdörr i detta sammanhang syftar på en avsiktlig svaghet i krypteringen, vars syfte är att låta exempelvis myndigheter avlyssna krypterad information (se [4, s. 149] för en närmare beskrivning av vad en bakdörr är). Falluck-funktioner är envägsfunktioner som går att utföra baklänges om man har tillgång till någon viss hemlig information [1, s. 34]. Falluck-funktionen som används i RSA är multiplikationen av två stora primtal. Denna är en envägsfunktion eftersom det är mycket tidskrävande att bestämma vilka två primtal som multiplicerats om endast produkten är känd. Om man däremot känner till en av faktorerna är problemet inte svårare än att dividera produkten med detta tal för att få den andra faktorn.

Vid nyckelutväxling med Diffie-Hellman utförs diskret exponentiering, eftersom beräkning av diskreta logaritmer också är ett svårt problem [källa]. Skillnaden är att denna funktion inte är en falluck-funktion, eftersom man inte känner till att det skulle finnas någon hemlig information som kan underlätta beräkningen. Detta behövs inte heller i Diffie-Hellman, eftersom endast resultatet utnyttjas. Idén bakom Diffie-Hellman-nyckelutväxling publicerades 1976 [5, s. 649].

Man har dock inte kunnat bevisa att det inte finns någon snabb algoritm för att utföra dessa operationer baklänges. Skulle en sådan upptäckas så skulle alla krypteringsalgoritmer som förlitar sig på svårigheten i dessa problem bli oanvändbara. Detta eftersom vem som helst skulle kunna köra denna baklängesalgoritm på sin dator och inom en rimlig tid kunna räkna ut privata krypteringsnycklar utgående från deras publika nycklar, eller beräkna hemliga nycklar genom att tjuvlyssna på datatrafik under en nyckelutväxling.

1.4 Shors algoritm

Peter Shor publicerade år 1994 den första versionen av Shors algoritm [6]; en uppdaterad version publicerades 1997 [7]. Denna kvantalgoritm utför både primtalsfaktorisering och beräkning av diskreta logaritmer inom polynomiell tid. Som nämndes tidigare är just dessa problem sådana som utnyttjas inom kryptering. Därför är detta är mycket intressant i ett krypteringssammanhang. Shors algoritm kan alltså enkelt lösa dessa problem och man kunde tänka sig att många krypteringsalgoritmer i och med detta är obrukbara.

Dock kan Shors algoritm, i och med att det är en kvantalgoritm, inte köras på en klassisk dator, utan kräver en tillräckligt kraftfull kvantdator. Det existerar idag inte någon kvantdator med tillräcklig beräkningskraft för att kunna forcera någon riktig kryptering. Enligt en artikel [8] publicerad 2018 finns idag kvantprocessorer med åtminstone 16 kvantsammanflätade (eng. *quantum entangled*) kvantbitar (eng. *qubit*), men enligt samma artikel skulle det krävas kvantdatorer av storleksordningen 50 kvantsammanflätade kvantbitar innan dessa eventuellt kan bli snabbare än klassiska datorer på vissa specifika beräkningar.

Den som läst om kvantdatorer i exempelvis nyhetsartiklar eller andra mindre vetenskapliga källor kunde kanske tro att Shors algoritm kan göra en uttömmande sökning av krypteringsnycklar parallellt, genom att utnyttja superposition. Detta är

inte fallet. Shors algoritm använder sig av kvantversionen av den snabba fouriertransformen för att hitta perioden hos RSA-funktionen [källa].

Även om tillräckligt kraftfulla kvantdatorer inte existerar ännu betyder existensen av Shors algoritm ändå att krypteringsalgoritmer, vars säkerhet baserar sig på svårigheten i att faktorisera primtal eller att beräkna diskreta logaritmer, inte kommer att vara säkra om utvecklingen av kvantdatorer fortsätter att framskrida. I denna avhandling ligger fokus därför på sådana krypteringsalgoritmer som kan köras på klassisk hårdvara och som antas vara säkra eller resistenta mot attacker utförda med hjälp av framtida hypotetiska kvantdatorer. På engelska brukar detta område kallas för *Post-Quantum Cryptography*. I denna avhandling används termen *kvantresistent kryptering*. Detta eftersom *kvantsäker kryptering* skulle implicera att dessa algoritmer bevisligen inte kan forceras med kvantdatorer. Kvantdatorerna är ännu under utveckling och man har inte kunnat bevisa att dessa algoritmer är säkra.

1.5 Grovers algoritm

En annan kvantalgoritm som har mycket mindre inverkan på kryptering är Grovers algoritm. Grovers algoritm är den snabbaste algoritmen för att invertera generella funktioner [2, s. 7], men den utgör inte ett lika stort hot mot kryptering som Shors algoritm. Grovers algoritm kan visserligen utnyttjas för att forcera symmetrisk kryptering, men eftersom den inte möjliggör samma exponentiella uppsnabbning som Shors algoritm kan dessa attacker enkelt blockeras genom att öka nyckelstorleken [2, s. 2].

1.6 Kvantdatorer

Detaljerna kring den bakomliggande kvantfysiken som utnyttjas i kvantdatorer ligger utanför ramarna för denna avhandling. Det är inte heller det huvudsakliga målet med denna avhandling att förklara matematiken som beskriver kvantberäkningar. Detta förklaras utförligt i *Quantum Computing Explained* [9] av David McMahon. Några grundläggande fakta om kvantdatorer presenteras nedan.

Våra vanliga klassiska datorer som finns överallt idag använder elektriska signaler för att representera och räkna med binära data. En enskild binär signal i en klassisk dator kallas *bit*. En bit kan ha två värden och dessa representeras vanligen av en

nolla eller en etta. Fysiskt representeras en bit av en *låg* eller *hög* spänningsnivå i en elektrisk ledning. Kvantdatorer skiljer sig fundamentalt från klassiska datorer genom att de använder så kallade *kvantbitar* (eng. *quantum bit* eller *qubit*) för att utföra beräkningar. Dessa kvantbitar utnyttjar något kvantmekaniskt fenomen för att representera data. Ett sådant fenomen är polariseringen hos ljus, men detta används sällan i praktiken eftersom ljusets kvantmekaniska tillstånd är svåra att bibehålla [källa]. Orsaken till att kvantmekaniska fenomen utnyttjas för beräkningar är ett fenomen som kallas *superposition*. Superposition innebär man förutom tillstånden noll och ett även kan ha tillstånd som är en kombination av dessa. Det är alltså möjligt att ha tillstånd som samtidigt är noll och ett. Vi kan aldrig se dessa tillstånd, eftersom de slumpmässigt *kollapsar* till antingen noll eller ett vid mätning. Man kunde därför tro att superposition är en illusion och att tillstånden har någon gömd variabel som med säkerhet kan säga vilket värde tillståndet kommer att ta vid mätning, men detta har motbevisats [källa].

Kvantdatorer baserade på kvantgrindar liknar konceptuellt klassiska datorer, eftersom kvantgrindar kan jämföras med logiska grindar i klassiska datorer. En skillnad är att kvantgrindar i stället för bitar hanterar kvantbitar som kan representeras av två komplexa tal som beskriver fördelningen av sannolikheterna mellan noll och ett. En annan är att de operationer som utförs av kvantgrindar måste vara reversibla. Dessutom är operationerna sina egna omvända operationer, så att om man utför en operation två gånger kommer man tillbaka till ursprungstillståndet. Genom att utnyttja matematiken som beskriver dessa tillstånd och operationer på ett smart sätt har man lyckats utveckla kvantalgoritmer som kan utföra vissa specifika beräkningar med lägre *tidskomplexitet* än vad som är möjligt med en klassisk dator. Tidskomplexiteten för en algoritm beskriver hur snabbt beräkningstiden ökar då storleken på indatat ökar. Man har lyckats bygga fungerande kvantdatorer baserade på kvantgrindar; bland annat IBM, Intel, Google och Microsoft jobbar med att utveckla kvantdatorer [källa]. Problemet är att dessa datorer har ett mycket begränsat antal kvantbitar. Antalet kvantbitar i en kvantdator begränsar storleken på problemen som kan lösas och därför kan dessa datorer ännu inte lösa problem som klassiska datorer inte klarar av. Termen *generell kvantdator* används ganska fritt idag och syftar ibland på en kvantdator baserad på

kvantgrindar, även om vissa menar att termen är tänkt att syfta på en kvantdator som kan köra både kvantalgoritmer och vanliga klassiska algoritmer [källa].

En annan beräkningsmodell för kvantdatorer är så kallad kvantglödgning (eng. *quantum annealing*). Främsta aktören inom detta området är D-Wave, som har kvantdatorer baserade på kvantglödgning med över 2000 kvantbitar [källa]. Kvantglödgning fungerar på ett helt annorlunda sätt än kvantgrindar och utnyttjas främst för optimeringsproblem. Datorer baserade på kvantglödgning verkar alltså endast kunna lösa en liten del av de problem som kvantgrindbaserade datorer kan lösa.

2. Kvantresistenta algoritmer

I detta avsnitt presenteras några olika kategorier av kvantresistenta algoritmer som kan köras på klassisk hårdvara. I de flesta fall presenteras den bakomliggande matematiken mycket ytligt, eftersom större delen av denna avhandling annars skulle bestå av förklaringar av matematiska koncept, som exempelvis algebraiska kroppar och kodningsteori.

Så kallad kvantkryptering, som utnyttjar kvantmekaniska egenskaper med hjälp av specialiserad hårdvara är inte intressant i detta sammanhang, eftersom man gärna vill kunna utnyttja dessa algoritmer över internet. Det är otänkbart att infrastrukturen för hela internet inom någon snar framtid skulle bytas ut med dyr hårdvara som har stöd för kvantkryptering, speciellt eftersom kvantkryptering fortfarande är en väldigt experimentell teknik [källa].

2.1 Gitterbaserad kryptering

Gitterbaserad kryptering (eng. *lattice-based encryption*) är en stark kandidat till kvantresistent kryptering. Författarna av [2, Kap. 5] menar att denna typ av kryptering anses vara säker mot kvantdatorer. Detta verkar stödas av det faktum att flera av kandidaterna för NIST:s standardiseringsprocess för kvantresistenta krypteringsalgoritmer är gitterbaserade [10], [11], samt att flera av Googles experimentella implementationer utnyttjar gitterbaserade algoritmer [12]–[14]. Detta behandlas närmare i kapitel 3.

Ett gitter är en mängd punkter med en periodisk struktur i ett n-dimensionellt rum. Det är inte helt självklart hur gitter kan användas för kryptering; detta upptäcktes av Ajtai [källa]. Problemet som utnyttjas handlar om att bestämma den kortaste vektorn i rummet. Detta går att göras effektivt endast om man tillåter en ganska stor felmarginal. Att bestämma den kortaste vektorn exakt ses som ett svårt problem.

Förutom att gitterbaserad kryptering anses vara säker mot kvantdatorer så har den också relativt effektiva implementationer, samt starka bevis för sin säkerhet för det värsta fallet och är ofta ganska enkel att implementera [2, s. 147].

2.2 Multivariat kryptering

Multivariat kryptering är en form av asymmetrisk kryptering där envägsfunktionen är en andragradsekvation med flera kvadratiska variabler.

2.3 Hashbaserad kryptering

Som namnet antyder utnyttjar hashbaserad kryptering hash-funktioner. En hash-funktion tar indata av godtycklig längd och producerar ett hash-värde av bestämd längd [1, s. 54]. I krypteringssammanhang är det viktigt att hash-funktionen är en envägsfunktion. Betydelsen av detta förklarades i 1.3. Ändringar i indata – också mycket små sådana – borde ha så stor inverkan på hash-värdet att det nya värdet inte verkar korrelerat till det gamla. Det borde också vara mycket svårt att hitta två olika indata som ger samma hash-värde. Dessutom ska funktionen vara snabb att beräkna, samt deterministisk så att samma indata alltid ger samma hash-värde.

Denna typ av kryptering kan inte gömma information; dess uppgift är endast att fungera som en sorts digital signatur [källa]. Därmed kan hashbaserad kryptering inte användas för att åstadkomma privat kommunikation. Däremot har det visat sig att kryptografiska hash-funktioner inte är sårbara för kvantdatorer [källa].

2.4 Kodbaserad kryptering

Kodbaserad kryptering innebär att den matematiska envägsfunktionen använder en felkorrigerande kod. Ett tidigt kryptosystem som verkar kunna hålla emot attacker av kvantdatorer är McElieces ”hidden-Goppa-code” system som presenterades redan 1978 [källa]. Systemet fungerar så att kryptotexten är ett kodord, till vilket man har tillsatt några fel. Endast den som har tillgång till den privata nyckeln kan ta bort dessa fel. Fördelarna med systemet är att det är väldigt snabbt. Nyckelstorleken begränsar däremot vad det kan användas till, eftersom den rör sig kring 100 kilobyte upp till flera megabyte [2, s. 95].

3. Implementationer

Det finns redan idag flera stora aktörer som jobbar med att implementera kvantresistenta krypteringsalgoritmer. National Institute of Standards and Technology (NIST) i USA, som tidigare definierat standarder som Advanced Encryption Standard (AES) [1, s. 19], har påbörjat en process för att samla in förslag, evaluera dem och slutligen standardisera åtminstone en kvantresistent asymmetrisk krypteringsalgoritm [15]. Ett vanligt problem med dessa algoritmer är att krypteringsnycklarna tar mycket mera utrymme, och därför vill många företag utföra tester och försöka optimera algoritmerna för att de inte ska påverka användarupplevelsen negativt.

3.1 CECPQ1

CECPQ1 är en experimentell krypteringsalgoritm som utvecklats av Google och är tänkt att kunna motstå attacker utförda med hjälp av en kraftfull kvantdator [12]. CEC står för *Combined Elliptic-Curve* och PQ står för *Post-Quantum*. CECPQ1 implementerar den kvantresistenta algoritmen *NewHope* [16] ovanpå den vanliga nyckelutväxlingen med ECDH (*Elliptic Curve Diffie-Hellman*), så även om svagheter upptäcks hos NewHope så kommer anslutningen ändå att ha samma säkerhet som normala anslutningar med ECDH. Detta tillåter Google att experimentera med CECPQ1 på sina egna webbsidor i kombination med deras egna webbläsare, Google Chrome, utan att utsätta någon för säkerhetsrisker. NewHope hör till familjen *Ring Learning-with-Errors* (*Ring LWE* eller *RLWE*), som är en typ av gitterbaserad krypteringsalgoritm. Google har inte för avsikt att CECPQ1 ska bli en standard, utan den är endast ett tillfälligt experiment. Experimentet har avslutats och resultaten diskuteras i en rapport publicerad 2016 [13]. I rapporten konstateras att inga oförutsedda hinder med att ta i bruk NewHope påträffades. Det noteras också att medianen för anslutningens fördröjning endast ökade med en millisekund. Fördröjningen hos de långsammaste 5 % och 1 % av anslutningarna ökade med 20 respektive 150 millisekunder. Man antar att detta endast beror på den ökade meddelandestorleken, eftersom NewHope inte är beräkningsintensiv. Slutligen konstaterar man att det skulle vara enkelt att ta i bruk NewHope om behovet uppstod. NewHope är en av kandidaterna som gått vidare till andra omgången i NIST:s standardiseringsprocess [10].

3.2 CECPQ2

Adam Langley, som också jobbade med Googles experimentella krypterings-algoritm CECPQ1, skriver på sin blogg [14] att Google jobbar på uppföljaren CECPQ2. Uppföljaren baserar sig precis som CECPQ1 på en kombination av ECDH och en kvantresistent krypteringsalgoritm. Denna gång har man valt HRSS [17] för den kvantresistenta delen. HRSS står för första bokstäverna i upphovsmännens efternamn. Likt NewHope, som används i CECPQ1, är även HRSS gitterbaserad och även denna är en kandidat för standardisering av NIST. Langley skriver inte hur experimentell CECPQ2 är, men att den nog på längre sikt är tänkt att ersättas av något annat. I NIST:s lista över kandidater är HRSS känd som NTRU-HRSS-KEM i den första omgången [11] och i den andra omgången hopslagen med NTRUEncrypt under namnet NTRU [10].

3.3 Microsofts implementationer

Microsoft uppger på sin webbplats [18] att man jobbar tillsammans med industrin och den akademiska världen på fyra av kandidaterna som skickats in till NIST för evaluering: FrodoKEM, SIKE, Picnic och qTESLA. Resonemanget är att dessa fyra dels har olika funktioner och att de baserar sig på olika matematiska problem, vilket kan ses som att man sprider riskerna om det senare visar sig att någon av algoritmerna eller rent av de matematiska problemen visar sig vara osäkra. Microsoft meddelar också att de har mjukvarubibliotek som implementerar dessa algoritmer och att de, precis som Google, jobbar med att integrera dessa med nuvarande internetprotokoll så att de kan utföra tester och optimera prestanda.

4. Framtid

Man kunde fråga sig varför resurser sätts på att utveckla dessa kvantresistenta krypteringsalgoritmer, då man inte ens med säkerhet kan säga att de någonsin kommer att behövas. Varför inte ta i bruk någon av dessa som redan utvecklats, ifall den dagen kommer då kraftfulla kvantdatorer förverkligas? Svaret är att krypteringsalgoritmer inte är något man utvecklar under någon vecka och sedan tar i bruk under en natt. För att man ska kunna lita på en krypteringsalgoritm måste den visa sig kunna motstå kryptoanalytikers försök att forcera den. Ju längre tid som passerat utan att några betydande svagheter har upptäckts hos algoritmen, desto säkrare kan man vara på att den största svagheten redan hittats och att algoritmen är säker att använda. Ett exempel är McElieces kryptosystem som presenterades 1978, hos vilken man ännu inte hittat någon betydande svaghet [källa]. En annan orsak är att man också vill ha tid att förbättra effektiviteten hos algoritmerna, så att de kan användas på mindre kraftfulla enheter som mobiltelefoner, utan att de kräver så mycket resurser att de påverkar batteritiden negativt.

5. Diskussion

Det kan tyckas att det inte är mycket som är säkert inom området kvantresistent kryptering. För det första vet man inte om det ens är möjligt att bygga kraftfulla kvantdatorer som kunde forcera dagens kryptering. Det kan hända att fenomen som *quantum decoherence* sätter stopp för alla sådana planer, eller att störningarna blir för stora. Det är också svårt att bevisa att nya kvantresistenta krypteringsalgoritmer som uppfinns verkligen är säkra. Det som dock är säkert är att kryptering kommer att vara mycket viktigt även i fortsättningen, då en allt större del av människors verksamhet är beroende av internet. Med tanke på vilka utmaningar man står inför inom utvecklingen av kvantdatorer och hur långt området kvantresistent kryptering ändå hunnit, verkar det troligt att kvantdatorerna inte kommer att bli ett problem för säkerheten online. Om kvantresistenta algoritmer inte redan är i allmänt bruk då de första kraftfulla kvantdatorerna realiseras, så finns det antagligen sådana algoritmer i beredskap, speciellt eftersom flera sådana redan har testats i praktiken [13]. Däremot ska det fortsatta arbetet med kvantresistenta algoritmer inte underskattas, för än så länge kan det mycket väl hända att det finns oupptäckta säkerhetsrisker i någon av dessa.

6. Referenser

[1] H. Delfs och H. Knebl, *Introduction to cryptography: principles and applications*, 2nd ed. Berlin ; New York: Springer, 2007.

[2] D. J. Bernstein, J. Buchmann, och E. Dahmén, Red., *Post-quantum cryptography*. Berlin: Springer, 2009.

[3] R. L. Rivest, A. Shamir, och L. Adleman, ”A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, *Communications of the ACM*, vol. 21, nr 2, s. 120–126, feb. 1978.

[4] C. Wysopal, C. Eng, och T. Shields, ”Static detection of application backdoors: Detecting both malicious software behavior and malicious indicators from the static analysis of executable code”, *Datenschutz und Datensicherheit - DuD*, vol. 34, nr 3, s. 149–155, mar. 2010.

[5] W. Diffie och M. Hellman, ”New directions in cryptography”, *IEEE Transactions on Information Theory*, vol. 22, nr 6, s. 644–654, nov. 1976.

[6] P. W. Shor, ”Algorithms for quantum computation: discrete logarithms and factoring”, i *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, s. 124–134.

[7] P. W. Shor, ”Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, *SIAM Journal on Computing*, vol. 26, nr 5, s. 1484–1509, okt. 1997.

[8] Y. Wang, Y. Li, Z. Yin, och B. Zeng, ”16-qubit IBM universal quantum computer can be fully entangled”, *npj Quantum Information*, vol. 4, nr 1, s. 1–46, sep. 2018.

[9] D. McMahon, *Quantum Computing Explained*. New York, United States: IEEE Computer Society Press, 2007.

[10] ”Round 2 Submissions”, *CSRC | NIST*, 30.01.2019. [Online]. Tillgänglig vid: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions. [Åtkomstdatum: 27.03.2019].

[11] ”Round 1 Submissions”, *CSRC | NIST*, 12.2017. [Online]. Tillgänglig vid: https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions. [Åtkomstdatum: 27.03.2019].

[12] M. Braithwaite, ”Experimenting with Post-Quantum Cryptography”, *Google Online Security Blog*, 07.07.2016. [Online]. Tillgänglig vid: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html. [Åtkomstdatum: 25.03.2019].

[13] A. Langley, ”CECPQ1 results”, *ImperialViolet*, 28.11.2016. [Online]. Tillgänglig vid: https://www.imperialviolet.org/2016/11/28/cecpq1.html. [Åtkomstdatum: 26.03.2019].

[14] A. Langley, ”CECPQ2”, *ImperialViolet*, 12.12.2018. [Online]. Tillgänglig vid: https://www.imperialviolet.org/2018/12/12/cecpq2.html. [Åtkomstdatum: 26.03.2019].

[15] ”Post-Quantum Cryptography”, *CSRC | NIST*, 03.01.2017. [Online]. Tillgänglig vid: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography. [Åtkomstdatum: 27.03.2019].

[16] E. Alkim, L. Ducas, T. Pöppelmann, och P. Schwabe, ”Post-quantum key exchange - a new hope”, 1092, 2015.

[17] A. Hülsing, J. Rijneveld, J. Schanck, och P. Schwabe, ”High-Speed Key Encapsulation from NTRU”, i *Cryptographic Hardware and Embedded*

Systems – CHES 2017, vol. 10529, W. Fischer och N. Homma, Red. Cham: Springer International Publishing, 2017, s. 232–252.

[18] ”Post-quantum Cryptography”, *Microsoft Research*. [Online]. Tillgänglig vid: https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/. [Åtkomstdatum: 28.03.2019].