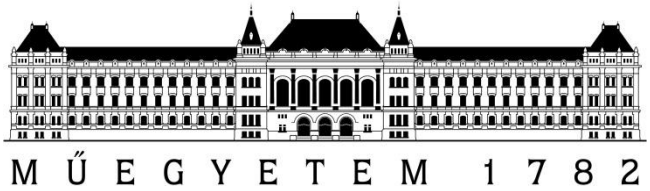


Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Hálózati Rendszerek és Szolgáltatások Tanszék

IOT eszközök kommunikációjának támadása kvantumszámítógéppel

Készítette: Török Ádám István

Témavezető: Dr. Imre Sándor

E-mail: adam.torok.00@gmail.com

2022. november 1.

1. Összefoglaló

A kvantuminformatika az informatika azon területe, amely kvantumfizikai jelenségeket alapul véve, és azokat kihasználva old meg informatikai problémákat. A kriptográfia az egyik olyan terület, ahol a kvantuminformatika elterjedése várhatóan forradalmi újításokat fog hozni, és meg fogja változtatni a jelenleg bevett, és klasszikus keretek között nem törhető protokollokat.

A terület fejlődése arra mutat, hogy a közeljövőben számos olyan algoritmus válik használhatóvá, amelyek olyan problémákra nyújtanak megoldásokat, amelyek klasszikusan nem megoldottak, vagy a gyakorlatban észszerű keretek között klasszikusan nem megoldhatóak, pl. túl sok idő az algoritmust lefuttatni, túl magas az algoritmus futtatásának a számításigénye.

Az Internet of Things (IOT) napjaink egyik dinamikusan fejlődő területe. A terület olyan hálózatokra és eszközökre vonatkozik, mint például szenzorhálózatok, ahol sok eszköz kommunikál egymással az interneten keresztül, és folyamatos információáramlás van közöttük. Az IOT-nek rengeteg alkalmazási területe van, például a közlekedésben, vagy az orvoslásban.

Az IOT eszközök sajátossága, hogy a hardverük nagy mértékben korlátozott például méretben, súlyban az alkalmazás módja miatt. Emiatt ezekben az eszközökben nagyon sok mérnöki kompromisszumot kell tenni a tervezéskor.

Sok IOT alkalmazásban nagyon fontos problémakör az adatok védelmének biztosítása, ezzel a feladattal a kriptográfia foglalkozik. Az IOT-ben lévő mérnöki kompromisszumok kiterjedhetnek az adatok védelmére használt kriptográfiai protokollokra is, ezért ezek az eszközök várhatóan hamar fognak támadási felületet kínálni az új kvantumalgoritmusoknak.

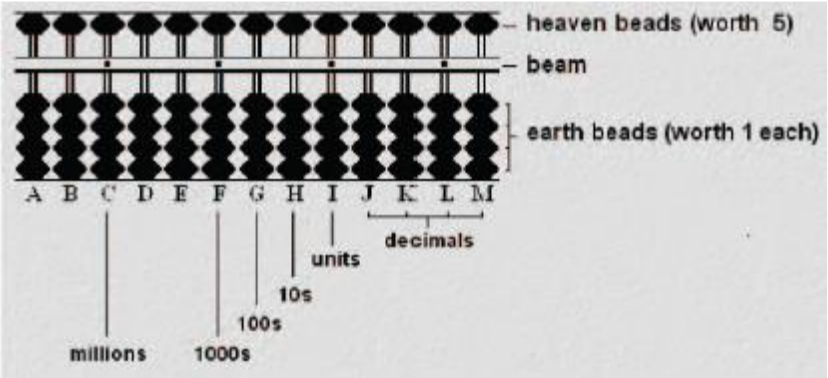
Ebben a dolgozatban az IOT eszközök kommunikációs protokolljait szeretnénk megtámadni, és elemezni a támadást, annak hatékonyságát. A támadás alapjának egy kvantumalgoritmust: a Simon-algoritmust vesszük, amely függvények periodicitásának megkeresésére alkalmas. Az algoritmust úgy módosítjuk, hogy alkalmazható legyen valós helyzetekben.

2. Tartalomjegyzék

3. Bevezetés

Az emberiség történelme során hamar felmerült a matematikai számítások elvégzése egy erre a feladatra készült eszközzel. A nagy számokkal történő számítások mindig is problémások voltak például a kereskedők életében. Az ókori görögök és rómaiak feljegyzéseiben is olvashatunk már az abacusról, több ezer évig használta az emberiség. Az eszköz képes volt arra, hogy az absztrakt matematikai számolásokat: a négy alapműveletet képes volt mechanikusan reprezentálni, így a számolások sokkal kényelmesebbé váltak nagy számkörökben is [1].

Manapság, ha számítógépekre gondolunk, akkor általában félvezetőkön és elektronikán alapuló számítógépekre gondolunk. A sebesség és használhatóság jelentősen nőtt az eszköz felmenőihez képest, de az alapötlet nagyon hasonló: valamilyen fizikai reprezentációt használunk arra, hogy könnyebben végezzük el a számításokat [16].



1. ábra Szorobán Abacus (Japán)

A számítógépek megjelenése jelentősen növelte meg az emberek számítási kapacitását. A legtöbb algoritmizálható probléma könnyen megoldhatóvá vált számítógépek segítségével, és sok új probléma jelent meg, sok probléma pedig megmaradt, de a számítási kapacitás növekedése miatt eltolódott más számkörökbe, olyanokba, amelyek a számítógépek kapacitásának jelentenek kihívást.

A kvantumszámítógép ötletét először Richard Feynmann [16] és Yuri Manin [17] tették meg.

Az ötlet egy olyan kvantumfizikai jelenségeken alapuló eszköz elkészítése, amelyet irányíthatunk matematikai számítások elvégzéséhez. Az olyan kvantummechanikai

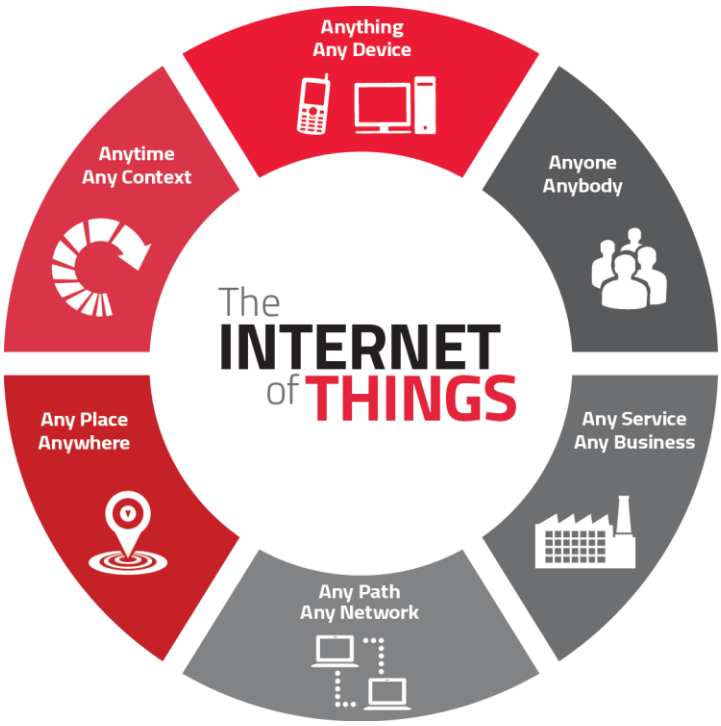
jelenségek, mint a szuperpozíció vagy összefonódás kihasználás olyan alkalmazásokkal bír, amelyeket egy klasszikus számítógéppel nem lehet megvalósítani [16].

Az IOT az internet következő lépcsőfoka, ahol az az eszközök között folyamatos kommunikáció van az interneten keresztül [2]. A környezet hardveres korlátai miatt a kriptográfiai megoldásoknál gyakran választják az elliptikus görbéket használó protokollokat [18].

Ebben a dolgozatban egy kvantuminformatikai algoritmussal foglalkozunk, amellyel megtámadjuk az elliptikus görbén alapuló protokollt. Bemutatjuk a kvantuminformatikát megalapozó posztulátumokat [3], az IOT-t [2], a Shor-algoritmust [10], az algoritmust végrehajtó kvantumáramkört, a támadás konstrukcióját. Végül elemezzük a támadást.

4. IOT bemutatása

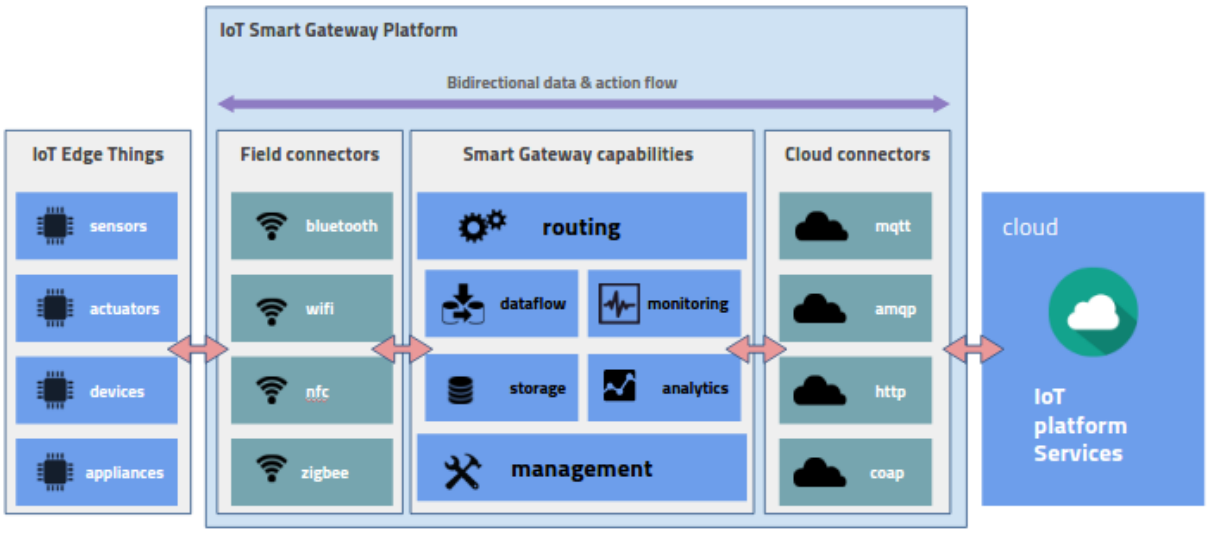
Az internet a korai számítógépek összekötésével született meg. Később a számítógépek elterjedésével, és ennek a sok eszköznek az összekapcsolásával megszületett a világháló (World Wide Web). Ezután a mobil eszközök is képessé váltak az internetre csatlakozni, ami a mobilinternethez vezetett. A következő technológiai ugrás az, hogy a hétköznapi tárgyak is képesek legyenek felcsatlakozni az internetre. Ez a jelenség vezet az Internet of Things fogalomhoz, vagy magyarul a Dolgok Internetéhez, ami a 2. ábra Dolgok Internete forrás: https://learninternetgovernance.blogspot.com/p/internet-of-things-iot.html látható [2].



2. ábra Dolgok Internete forrás:
https://learninternetgovernance.blogspot.com/p/internet-of-things-iot.html

IOT architektúra

Az IOT architektúráját feloszthatjuk az eszközök funkcióinak és fizikai megvalósulásuk szerint. A 3. ábra IOT architektúra láthatjuk az architektúra felosztását.



3. ábra IOT architektúra forrás:[4].

Edge tárgyak:

Az edge tárgyak lehetnek szenzorok, működtető/irányító eszközök, önálló készülékek, berendezések. Ezek a tárgyak teremtik meg a kapcsolatot a valóság és a hálózat között. Az edge kifejezés az Edge Computing paradigmából jön, ahol a cél az adatok feldolgozása minél közelebb a hálózat széléhez, vagyis az adatok forrásához. Az edge lehet akár okos város, okos ház, gyár, okoshálózat, fúrótorony, farm, szélerőmű, repülők, hajók, autók. Az edge egyik fontos eleme, hogy minél közelebbi legyen lennie a valós idejű feldolgozáshoz [2].

Field protokollok

Az edgen lévő tárgyaknak kommunikálni kell egymással és a Smart Gatewayjel. Ez a kommunikáció általában vezeték nélküli protokollokkal kerül implementálásra, és a leggyakoribb alkalmazások az alábbiak [2]:

Bluetooth:

Alacsony energiafelhasználásra tervezett vezeték nélküli protokoll. Jelentős az IOT alkalmazások között. Bluetooth 4.2 Standard: Frekvencia: 2.4 GHz (ISM), Távolság: 50-150m, Sávszélesség: 1 Mbps (Smart/ BLE) [5].

Zigbee:

A bluetoothhoz hasonló protokoll, de ipari alkalmazások között elterjedtebb. A Zigbee alacsony energiaigényre és alacsony költségre tervezett protokoll. 2.4 Ghz frekvencián üzemel és olyan alkalmazásokat céloz amelyek zárt területen kommunikálnak [5].

Wi-Fi:

A wi-fi elterjedt választás az IOT fejlesztők között a protokoll otthoni környezetekben nagy számú elterjedésnek köszönhetően. Nagy sávszélességet biztosít és gyors adatáramlást [5].

NFC:

Az NFC protokoll egy nagyon kis távolságra tervezett vezeték nélküli protokoll, a hatótávolság általában 4cm alatti. Biztonságos kétirányú kommunikációt tesz lehetővé és olyan praktikus alkalmazásai vannak, mint: tranzakciók lebonyolítása telefonnal, kártyával, eszközök párosítása, digitális tartalmak gyors elérése [5].

IoT Smart Gateway

Az IOT Smart Gateway az architektúra azon része, ami kommunikációt teremt az edge és a felhő között. A Smart Gateway értelmezi mind az edge protokolljait, mind a felhő protokolljait, és biztosítja a konvertálását az információnak a kettő között. A Smart Gateway rendelkezik a routolásról, menedzseli az adatfolyamot, monitorozza az adatokat és tárolásukat [7].

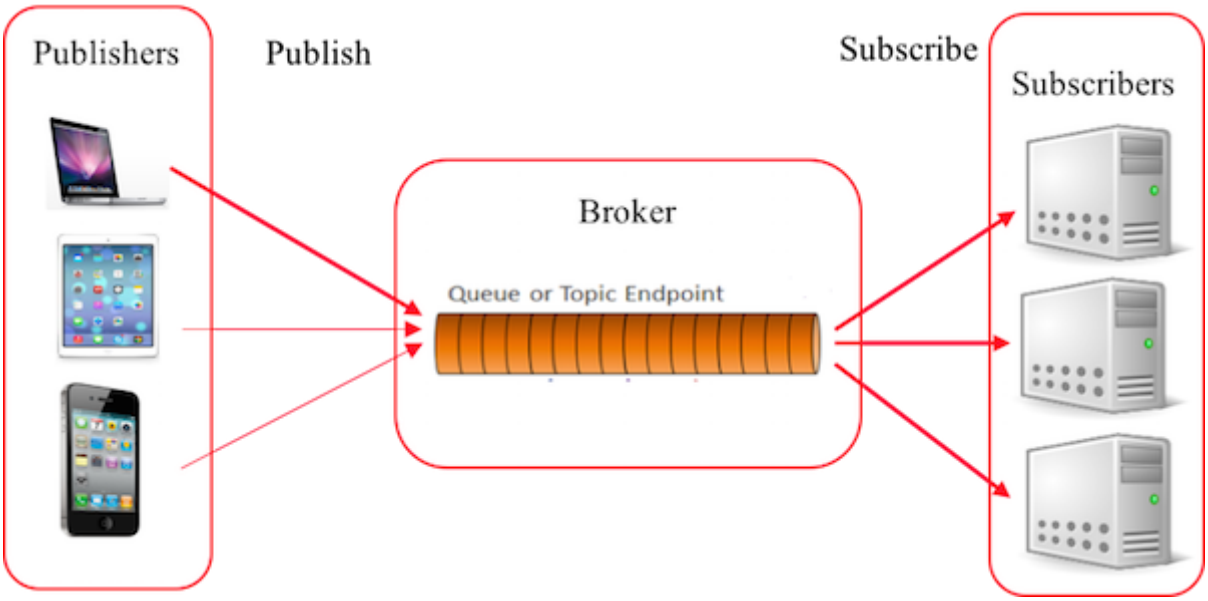
Felhő protokollok

A legtöbb IOT megoldásnak integrálnia kell felhő szolgáltatásokat, általában azoknak is amik majdnem csak az edgen működnek. A felhővel valamilyen felhő protokollon keresztül történik a kommunikáció:

MQTT:

Az MQTT szabványt az IBM mutatta be 1999-ben és az OASIS szabadalmaztatta 2013-ban. Arra lett tervezve, hogy kapcsolatot biztosítson az egyik oldal

alkalmazásai és a másik oldal hálózatai között a kapcsolatot. Egy publish/subscribe architketúrát követ a protokoll. A rendszer három részből áll: publikálókból, feliratkozókból és brókerekből, az architektúra a 4. ábra MQTT architektúra látható [6].



4. ábra MQTT architektúra

SMQTT:

Az SMQTT az MQTT biztonságos változata, ami egy kis erőforrásigényű titkosítást használ. Az SMQTT nagy előnye, hogy képes broadcast titkosításra, ami egy üzenetet titkosít, majd több címre küld ki: ez a szenárió gyakran fordul elő IOT alkalmazások esetén. Az algoritmus négy fázisból áll: setup, titkosítás, küldés, visszafejtés. A setup fázisban minden subscriber és publisher regisztrálja magát a brókerhez, majd kap egy mesterkulcsot a fejlesztő által választott kulcs generáló algoritmus alapján. A kulcsgenerálási algoritmusok nem sztenderdek [6].

AMQP:

Az AMQP egy pénzügyi ágazatra fejlesztett felhő protokoll. TCP felett fut, és az MQTT-hez hasonlóan publish/subscribe architektúra alapján működik. A különbség az, hogy az AMQP-ben a bróker fel van osztva két részre: exchange és queue. Az exchange felelős a publisherek üzeneteiért és azok elosztásáért a queuenak az előre

definiált feladatkörök, feltételek alapján. A queuek írják elő melyik üzenete fogadják el, és küldik tovább annak megfelelően, ki iratkozott fel rájuk [6].

CoAP:

A CoAP felhő protokoll az IETF által arra lett kifejlesztve, hogy alacsony energiafelhasználás mellett biztosítson lehetőséget REST interfacere. A REST a sztandard interface a HTTP kliens és szerver között, de IOT alkalmazások esetén nagy lenne az eredeti protokoll energiafelhasználása. UDP felett fut, de biztosít egy egyszerű megoldást megbízhatóság kezelésére. A CoAP architetkúra két rétegre van felosztva: üzenet és kérés/válasz. Az üzenet réteg felelős a megbízhatóságért és az üzenetek duplikációjáért, a kérés/válasz réteg felelős a kommunikáióért. A CoAP a http-hez hasonlóan GET, PUT, PUSH, DELETE kéréseket használ [6].

IOT alkalmazásai

Rengeteg használati eszközünk már most is okos, de nem kommunikálnak egymással, de ennek a kommunikációnak és információáramnak több praktikus felhasználási területe is van. Ezek a megoldások növelhetik életszínvonalunkat az IOT-nek köszönhetően [5].

Okos otthon, okos épületek

Az okos épületek sok kényelmi funkciót rejtenek magukban, amiket IOT integrálásával lehet implementálni. Egy okos otthonban az olyan háztartási kisgépek mint a mosógép, mosogatógép, ajtók, ablakok, világítás, hűtő, sütő mind csatlakoztatva vannak a hálózatra és manuálisan irányíthatóak. Egy számítógéppel vagy okostelefonnal elérjük a ház energetikai információit és irányítástechnikai rendszerét [5].

Okos farmok

Az IOT segíti a precíziós gazdálkodást, az IOT-t alkalmazó farmok monitorozni képesek a fényt, hőmérsékletet, páratartalmat, talajnedvességet, esőelőrejelzéseket képesek tenni a farmra kitett szenzorhálózat segítségével. Az IOT segíti az automatizált öntözőrendszerek kiépítését és optimalizálást. A szenzorhálózat segít a nagy földek és mennyiségek ellenére részletes adatokat kapni és analizálni a farm kis földdarabjairól, növényeiről [5].

Okos egészségügy

Az egészségügy egyik fontos problémája, hogy a páciensek állandó figyelmet kapjanak, pszichológiai esetek esetén erre különös gondot kell tenni. Okos egészségügyi megoldásokkal a páciensekről folyamatos adatsorokat tud gyűjteni a szenzor, amit a felhő analizálni tud. Az adatsort ezután megkapja a kezelő orvos vagy pszichológus és ez alapján tesz javaslatokat. Ezzel a megoldással az olyan betegségek esetén, ahol gyakran kell vizsgálni a pácienst, csökkenteni tudjuk, a vizsgálatok számát, de azt is látjuk ha szükséges az. Ez csökkenti a kezelés költségét és növeli a hatékonyságát [5].

Okos városok

Az okos városok funkcióinak kiépítése különös tervezést igényel, folyamatos támogatással az önkormányzatokkal, kormánnyal. IOT segítségével növelni lehet a város életének minőségét a közlekedés területén: növelni lehet a tömegközlekedés hatékonyságát, monitorozni lehet a forgalmi torlódásokat és csökkenteni lehet azokat. Előre lehet jelezni és el lehet kerülni baleseteket [5].

Okos hálózat, okos energia

Az okos hálózatok ötvözik az infokommunikációt és a villamos energetikát, így megalkotva egy dinamikus villamos hálózatot, amiben kétirányú kommunikáció történik és az energiaigények folyamatosan monitorozva vannak. A hálozaton lévő szenzorhálózatok segítenek megtalálni hibák esetén a probléma forrását, helyét, így felgyorsítva annak megoldását, más esetben képesek előrejelezni a hibák helyét és elkerülni azokat.

5. Kvantuminformatikai bevezető

A kvantummechanika sok olyan jelenséggel foglalkozik, amik felhasználhatóak az infomratikában. Ebben a fejezetben ennek a területnek az alapját mutatom be.

Kvantummechanika posztulátumai

A körülöttünk lévő fizikai világról nem tudjuk pontosan hogyan működik, de használhatunk modelleket, amelyek jó közelítései a valóságnak és a modelleket mindig finomíthatjuk, ha már nem vagyunk elégedettek a modell adta leírásokkal. [2]

Első posztulátum (állapottér):

*Minden zárt fizikai rendszer leírható egy **V** Hilbert-térben értelmezett komplex együtthatós **v** vektorral, amelyet állapotvektornak hívunk.*

A legegyszerűbb nem triviális zárt fizikai rendszer egy két-dimenziós Hilbert térrel írható le. Ekkor a rendszer állapota: $\mathbf{v} = [a, b]^T = a\mathbf{0} + b\mathbf{1}$, ahol $\mathbf{0} = [1, 0]^{\mathrm{T}}$ és $\mathbf{1} = [0, 1]^{\mathrm{T}}$ a *V* Hilbert tér bázisvektorai és $a, b \in \mathbb{C}$ a komplex együtthatók. Ahhoz, hogy az egységhossz megszorítás teljesüljön az együtthatókra teljesülni kell: $|a|^2 + |b|^2 = 1$.

Második posztulátum (változás):

Minden zárt fizikai rendszer állapotának időbeli változása leírható egy unitér transzformációval, amely csak a kezdő és végállapottól függ.

Harmadik posztulátum(mérés):

*Minden kvantum mérés leírható egy mérési operátorokból álló halmazzal: $\{M_m\}$, ahol m a mérés lehetséges kimenetét jelenti. Az m kimenetel kimérésének valószínűsége egy **v** állapotú rendszeren az alábbi módon számítható:*

$$P(m \mid \boldsymbol{v}) = \boldsymbol{v}^{\dagger} M_m^{\dagger} M_m \boldsymbol{v}$$

és m kimenetel mérése után a rendszer a v' állapotba kerül:

$$\boldsymbol{v}' = \frac{M_m \boldsymbol{v}}{\sqrt{\boldsymbol{v}^\dagger M_m^\dagger M_m \boldsymbol{v}}}$$

Valószínűségszámítást felhasználva felírható az alábbi összefüggés a mérési operátorokra:

$$\sum_m P(m \mid \boldsymbol{v}) = \sum_m \boldsymbol{v}^\dagger M_m^\dagger M_m \boldsymbol{v} = 1$$

eszerint a mérési operátoroknak teljesíteni a kell a <u>teljességi relációt:</u>

$$\sum_m M_m^\dagger M_m = I$$

Mivel a mérések nem visszafordítható folyamatok, ezért kivételt alkotnak az unitér feltételnek. A mérések teremtik meg a klasszikus és a kvantum világ között a kapcsolatot: csak méréseken keresztül tudjuk megfigyelni a kvantum világot.

Negyedik posztulátum(összetett rendszerek):

Egy összetett *W* fizikai rendszer állapottere meghatározható az őt alkotó rendszerek *V* és *Y* állapottereinek tenzorszorzataként:

$$W = V \otimes Y$$

Ha a két rendszer állapota $\boldsymbol{v} \in V$ és $\boldsymbol{y} \in Y$ akkor az összetett rendszer állapota $\boldsymbol{w} = \boldsymbol{v} \otimes \boldsymbol{y}$ lesz.

Qbit és qregiszter

A posztulátumok megteremtik a kvantummechanika leírásához szükséges formalizmusokat, de a gyakorlati alkalmazáshoz további formalizmusokat is bevezetünk, hogy praktikusabb legyen a problémák leírása.

A klasszikus informatika információs alapegysége a bit, amely a 0 vagy az 1 értéket veszi fel. A valós életből egy jó fizikai példa a bitre egy pénzérme, amely vagy fej vagy az írás oldalával felfelé esik le.

Az első posztulátumnak megfelelően a legegyszerűbb kvantum rendszer egy két-dimenziós Hilbert tér komplex együtthatós állapotvektorával írható le. Ezt a rendszert ezután *qbitnek* fogjuk hívni és fizikai megvalósulása lehet például egy foton vagy egy elektron. Az oszlopvektor ***v*** jelölése ezután $|v\rangle$ a Dirac formalizmusnak megfelelően.

Egy qbit állapota két bázisvektor, $|0\rangle$ és $|1\rangle$ állapotvektorok keveréke, a klasszikus 0 és 1 bitértékeknek megfelelően. Egy tetszőleges $|\varphi\rangle$ qbit formálisan:

$$|\varphi\rangle = a|0\rangle + b|1\rangle = a\begin{bmatrix}1\\0\end{bmatrix} + b\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix}$$

$a, b \in \mathbb{C}$ az állapotvektor valószínűségi amplitúdói. A mérés előtt a qbit egyszerre van mindkét állapotban, az amplitúdók nagyságának megfelelően.

A klasszikus informatikával analóg módon egy n qubitből álló egységet *n* méretű qregiszternek hívunk. Egy ilyen egység $N = 2^n$-dimenziós bázisvektorok tetszőleges szuperozícióját tatalmazhatja. Ha ismerjük a qregiszterben lévő qubitek állapotát, akkor a qregiszter állapotvektorát a negyedik posztulátumnak megfelelően a qubitek tenzorszorzataként számíthatjuk formálisan:

Legyen

$|\varphi_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$, $|\varphi_2\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

és nézzük meg az általuk alkotta qregiszter állapotát:

$$|\varphi\rangle = |\varphi_1\rangle|\varphi_2\rangle = |\varphi_1, \varphi_2\rangle = \frac{|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle + |0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle}{2}$$
$$= \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

Vagyis a két qubit állapota előáll egy négydimenziós Hilbert-tér bázisvektorainak kombinációjaként.

6. Shor-algoritmus

Shor híres cikkében két algoritmust ismertetett, az egyik algoritmus a faktorizáció problémáját, a másik algoritmus a diszkrét logaritmus problémát oldotta polinomiális időben. Az első algoritmussal törhetővé válik például az RSA, a klasszikus Diffie-Hellmann kulcscsere, míg a másik algoritmust ki lehet terjeszteni egy elliptikus görbén értelmezett véges csoport diszkrét logaritmus problémájára [10].

Diszkrét logaritmus probléma: Legyen ***G*** egy ***g*** elem által generált *p* prím karakterisztikájú csoport. Adott $y = g^k$. Adjuk meg ***k***-t!

Az algoritmus működése:

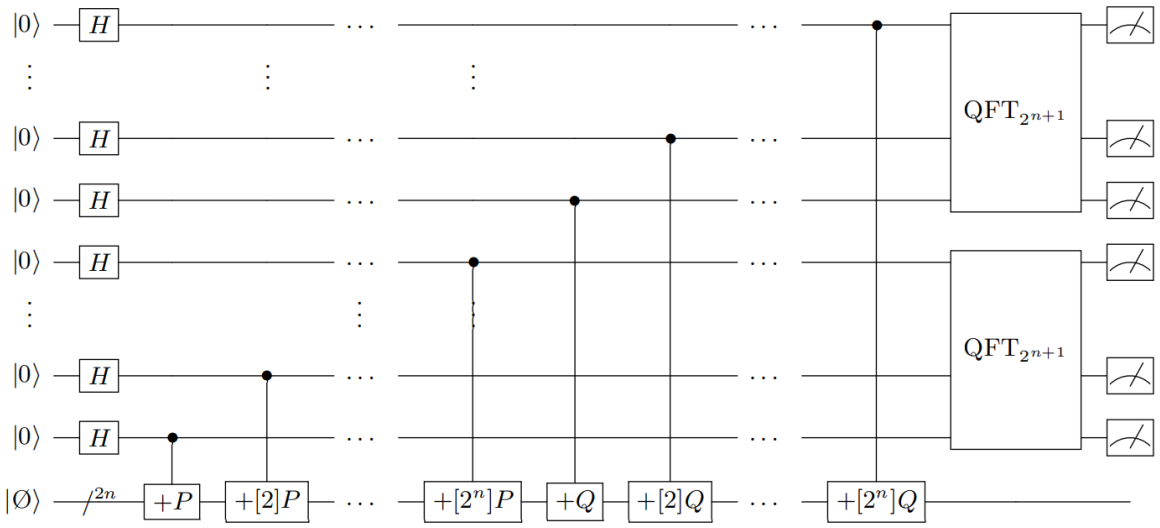
1. Képezzük az $f : (x_1, x_2) \rightarrow g^{x_1} y^{x_2}$ kétváltozós függvényt.
2. Keresünk egy perióduspárt: (ω_1, ω_2), erre teljesül:
$$f(x_1, x_2) = f(x_1 + \omega_1, x_2 + \omega_2)$$
3. Innen: $g^{\omega_1} y^{\omega_2} = 1 \iff g^{\omega_1 + k\omega_2} = 1$, vagyis igaz:
$$k\omega 2 \equiv -\omega 1 (mod q) .$$

Ha $(\omega_1, \omega_2) \neq (0,0)$, akkor $k = -\omega 1/\omega 2 \, (mod q)$. Egyébként újrafuttatjuk az algoritmust. Annak a valószínűsége, hogy megfelelő (ω_1, ω_2) párt találunk [10]:

$$P = \frac{\phi(p-1)}{p-1} \tag{1}$$

Az ECDLP megoldására szóló Shor-algoritmust megvalósító kvantum áramkör az *5. ábra* Shor-algoritmust megvalósító kvantum áramkör látható. Ez annyiban különbözik az eredeti Shor-algoritmustól, hogy az alsó vezetékeken a csoporton értelmezett összeadás, vagyis az elliptikus görbén történő összeadás történik.

ECDLP: Adott $P, Q \in E(F_p)$, keressük azt a k-t melyre $kP = Q$.



5. ábra Shor-algoritmust megvalósító kvantum áramkör

7. Biztonságos kommunikáció megteremtése

Az információ titkosításával foglalkozó tudományterület a kriptológia. Két fontos területből épül fel: A *kriptográfia* foglalkozik az információ biztonságos továbbításának különböző módszereivel és megvalósításával.

A *kriptoanalízis* gyűjti össze azokat az eljárásokat amiknek célja a titkosított üzenetek eredetivé alakítása.

A titkosítás egyik nagyon fontos alkalmazása kommunikáció során történik. Tegyük fel azt a helyzetet, hogy Alice szeretne elküldeni egy üzenetet Bobnak. Üzenet alatt egy karaktersorozatot értünk, pl. ’alma’. Ezt úgy szeretné elküldeni, hogy a külső személyeket

- megakadályozzák az eredeti üzenet információjának megszerzésétől és
- megakadályozzák attól, hogy úgy tehessenek, mintha Bobként üzeneteket küldenének Alicenak.

Ehhez Alice először elkészít egy titkosított üzenetet:

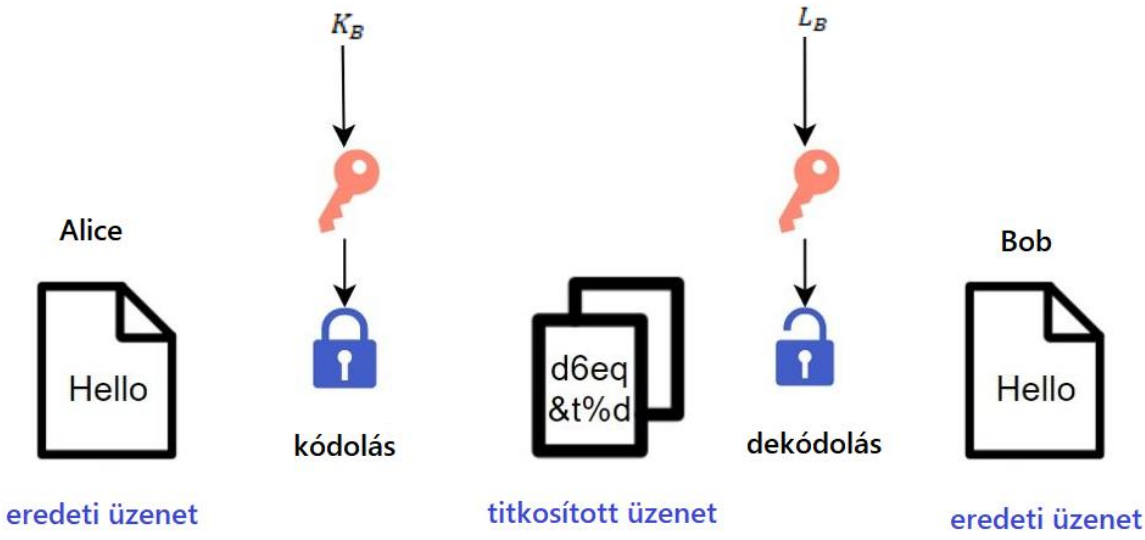
$$E = e_A(P)$$

Majd Bob megkapja a titkosított E üzenetet és dekódolja azt, vagyis kinyeri belőle az eredeti információt:

$$P = d_B(E)$$

Ahol E jelenti a titkosított üzenetet, P jelenti az eredeti üzenetet, e_A jelenti Alice titkosító eljárását leíró függvényt, d_B jelenti Bob dekódoló eljárását leíró függvényt. Tehát E -hez hozzárendel Alice egy másik karaktersorozatot, pl ’körte’, és ezt küldi el Bobnak. Majd Bob ebből a másik karaktersorozatból visszafejti az eredeti ’alma’

karaktersorozatot.



6. ábra Üzenetküldés egy általános titkosítási eljárással

Ahhoz, hogy az információ továbbítása biztonságos legyen, $e_A()$ és $d_B()$ módszereket titokban kell tartani. Ez a módszer a gyakorlatban megjelenő rendszerekben nem praktikus, mert túl sok ilyen eljáráspárt kellene eltárolnunk, minden kommunikáló párhoz különbözőt, és nagyon sokféle eljárást kellene megvalósítani kódolásra, dekódolásra. Képzeljük, hogy Alice leírja az üzenetét egy papírra majd, bezárja azt egy zárral rendelkező dobozba, majd elküldi Bobnak. Minden üzenethez másik zárral rendelkező dobozt kell szereznie. Ezzel szemben sokkal egyszerűbb, ha ugyanolyan dobozokat használ, és a lakatokat külön rakja rá, vagyis a dobozok csak a hozzájuk tartozó zárakban és kulcsokban térnek el, amiket sokkal olcsóbb nagy számban előállítani. (Az eljárás működése az: 6. ábra Üzenetküldés egy általános titkosítási eljárással ábrán látható) Ezzel az analógiával élve egy sokkal jobban skálázható, és így nagy hálózatokban (pl telefonos hálózat) egyszerűbben megvalósítható megoldás, ha a különböző felhasználók mind ugyanazt a kódoló és dekódoló eljárást alkalmazzák, de az eljárás eredmény függ, egy másik változótól is, ezt hívjuk *kulcsnak*.

Formálisan ekkor a kódolás:

$$E = e(P, K_B)$$

és a dekódolás:

$$P = e(E, L_B)$$

alakban írhatóak. A K_B ill. L_A kódoló ill. dekódoló kulcsok. Ekkor az *e()* eljárás más karaktersorozatot rendel hozzá ugyanahhoz a kódszóhoz, ha más a kulcs.pl. ha kulcsnak az 1-et használjuk a titkosított üzenet ’körte’ karaktersorozat lesz, míg 2 kulcs mellett ’banán’ karaktersorozat. formálisan:

$$e('alma', 1) = 'körte'$$
$$e('alma', 2) = 'banán'$$

Szimmetrikus kulcsú titkosítás

A titkosítások legrégebben ismert módja a szimmetrikus kulcsú titkosítás, mert ez az eljárás a legegyszerűbben megvalósítható. Szimmetrikus, mert a küldő ugyanazzal a kulccsal kódol, mint amivel a fogadó dekódol, vagyis $K_B = L_B$.

Ezen az elven működött a legrégebben ismert rejtjelező, az ókori görögök és főleg spártaik által hadjáratokon használt rejtjelező rúd (A rejtjelező rúd a **Hiba! A hivatkozási forrás nem található.** látható). Az eszköz működése a következő: kódoláskor egy szalagot rácsavarunk egy rúdra, majd a feltekert szalagra írjuk a titkosítandó szöveget, a rúd irányával párhuzamosan. Ezután letekerjük a szalagot, és a felírt szimbólumok összekeverednek, ez a titkosított szöveg. Ezután a fogadó fél úgy nyeri vissza az eredeti üzenetet, hogy ő is felcsavarja a szalagot egy rúdra. A dekódoló csak akkor tudja rekonstruálni az eredeti üzenetet, ha ismeri azt rúdvastagságot, amivel készült a titkosított szöveg, rossz vastagságú rúdon nem áll elő az eredeti üzenet.

Aszimmetrikus kulcsú titkosítás

Az aszimmetrikus kulcsú titkosítás azért aszimmetrikus, mert a kódolásra és dekódolásra használt kulcsok különböznek, vagyis $K_B \neq L_B$. Ilyen feltétel mellett megfelelő titkosító eljárás mellett előállhat olyan szituáció, hogy elég, ha csak az

egyik kulcsot tartjuk titokban, ezt hívjuk privát kulcsnak, ezzel zajlik majd a dekódolás. A másik kulcsot publikus kulcsnak hívjuk, ezzel zajlik majd a kódolás, és ezt nyilvánosságra lehet hozni az eljárás során, vagyis a titkos üzenetváltáshoz sem kell titokban tartani. Éppen ezért ezt az eljárást hívják publikus kulcsú titkosításnak is.

Egy centralizált hálózatban egyszerűen megvalósítható a szimmetrikus kulcsú titkosítás kulcsainak szétosztása, de olyan elosztott hálózatoknál, mint például az internet, ez nehezen lenne megvalósítható, mivel nincsen előre meghatározott központi egység. Ilyen esetben sokkal egyszerűbb aszimmetrikus kulcsú titkosítás alkalmazása, éppen ezért az interneten ez a leggyakrabban implementált.

Közös kulcsok megteremtése

Az azonos biztonságú szimmetrikus és aszimmetrikus protokollok közül a szimmetrikusak általában jelentősen gyorsabbak, mert kisebb a hardver igényük. A szimmetrikus titkosítások használatához viszont szükség van a kommunikáló feleknek egy titkos kulcsra, amit csak ők ismernek. A két félnek kommunikálni kell ahhoz is, hogy megteremtsék a közös kulcsot. Ennek a kommunikációnak úgy kell történni, hogy egy támadó a kommunikációból ne tudja előállítani a kulcsot. Ezt a feladatot valamilyen kulcscsere protokollal, aszimmetrikus protokollol valósítják meg. Így a gyakorlatban a bevett módszer az, hogy valamilyen aszimmetrikus protokollal közös titkos kulcshoz jut a két fél, majd ezután elkezdhetnek szimmetrikus protokollal és a közös kulccsal titkosított üzeneteket küldeni és fogadni. Az ilyen rendszereket hibrid kriptorendszernek hívják, amik egyesítik a szimmetrikus és aszimmetrikus titkosítások erősségeit.

Diffie- Hellmann kulcscsere protokoll

Diffie és Hellmann 1976-os cikkében megjelent kulcscsereprotokollja máig is meghatározó a modern kriptográfiában, habár manapság nem az eredeti protokollt használjuk, mert van egy nagy hiányossága [8].

A protokollt tekintsük két résztvevőre, legyenek ők *A* és *B*. *A*-nak és *B*-nek is van egy-egy publikus kulcsa és egy-egy privát kulcsa. A publikus kulcsot elmondhatják másoknak, de a privát kulcsot titokban kell tartaniuk a biztonságos kommunikációhoz. A Diffie-Hellmann kulcscserével *A* és *B* ki tudnak generálni a másiktól kapott publikus kulccsal és a saját privát kulcsukkal egy olyan kulcsot, amit csak ők ismernek.

Az eredeti módszer biztonsága a diszkrét logaritmus probléma és a Diffie-Hellmann probléma nehézségén alapszik. A Diffie-Hellmann-protokoll nem biztosít kulcshitelesítést, ezért aktív támadás ellen nem biztosít védelmet [9].

Elliptikus Görbe Diffie-Hellmann kulcscsere

Az elliptikus görbék (és hiperelliptikus) görbék kriptográfiai alkalmazása viszonylag rövid múltú, de gyorsan fejlődött az aktív kutatómunka miatt. A Diffie-Hellmann kulcscsere protokoll több fajta matematikai problémára alapozva is működtethető. Elliptikus görbékre alapozva használni azért célszerű, mert kisebb méretű kulcsokkal is el tudjuk érni ugyanazt a biztonságot, mintha módszereket, például moduló hatványozást használnánk.

Egy *E* elliptikus görbét *F* test felett az

$$y^2 + a_1xy + a_3y = x^3 + a_2x_2 + a_4x + a_6 \qquad a_i \in F \qquad (2)$$

egyenlet definiál.

Az egyenlet megoldásai Abel-csoportot alkotnak, egy az elliptikus görbére értelmezett „+” művelettel, és az ez alapján definiált „·” művelettel . *A* és *B* is ismer egy *G* generátor elemet.

Protokoll működése:

1. *A* generál egy *x* véletlen számot majd kiszámolja $x \cdot G$ értéket és elküldi *B*-nek.
2. *B* generál egy *y* véletlen számot majd kiszámolja $y \cdot G$ értéket és elküldi *A*-nak.
3. *A* kiszámolja x véletlen számából és a *B*-től kapott $x \cdot G$ értékből $y \cdot x \cdot G$ közös kulcsot.
4. *B* kiszámolja y véletlen számából és a *A*-tól kapott $y \cdot G$ értékből $x \cdot y \cdot G$ közös kulcsot [9].

$$x \cdot y \cdot G = y \cdot x \cdot G$$

vagyis tényleg közös kulcsot kapnak, mert a szorzás kommutatív a csoporton.

8. Támadás analízise

Bár jelenleg nincs akkora kvantumszámítógép, ami komoly veszélyt jelenthetne a mai aszimmetrikus titkosításokra, de egy jövőbeli kvantumszámítógép a mostani adatainkra is veszélyt jelenthet.

A támadó lehallgatva a titkosított üzenetváltást a kulcscserét is beleértve, eltárolhatja a titkosított üzeneteket, majd azokat csak akkor töri fel, ha megfelelő kapacitású kvantumszámítógép áll a rendelkezésére.

Egy kvantumhálózat hardverének nagyságát jellemezhetjük a szükséges qubitek és Toffoli-kapuk számával. Ahhoz, hogy meghatározzuk egy algoritmus töréséhez szükséges hardver nagyságát, a hardver felépítésének pontos ismerete szükséges. Ez elliptikus görbe feletti diszkrét logaritmus probléma megoldásához szükséges mai ismeretek szerint tervezett hardver nagysága $9n + 2\lceil \log_2(n) \rceil + 10$ szerint skálázódik qubitek számában, $448n^3 log2(n) + 4090n^3$ szerint skálázódik Toffoli-kapuk számában, ahol az elliptikus görbe egy legfeljebb *n* bites prím modulus felett van értelmezve [11].

Standard elliptikus görbék

A *p*>3 prím karakterisztikájú elliptikus görbéken definiált csoportokat Weierstrass-görbéknek hívjuk, felírhatók a Weierstrass formula szerint (lásd (2)):

$$y^2 = x^3 + ax + b \tag{3}$$

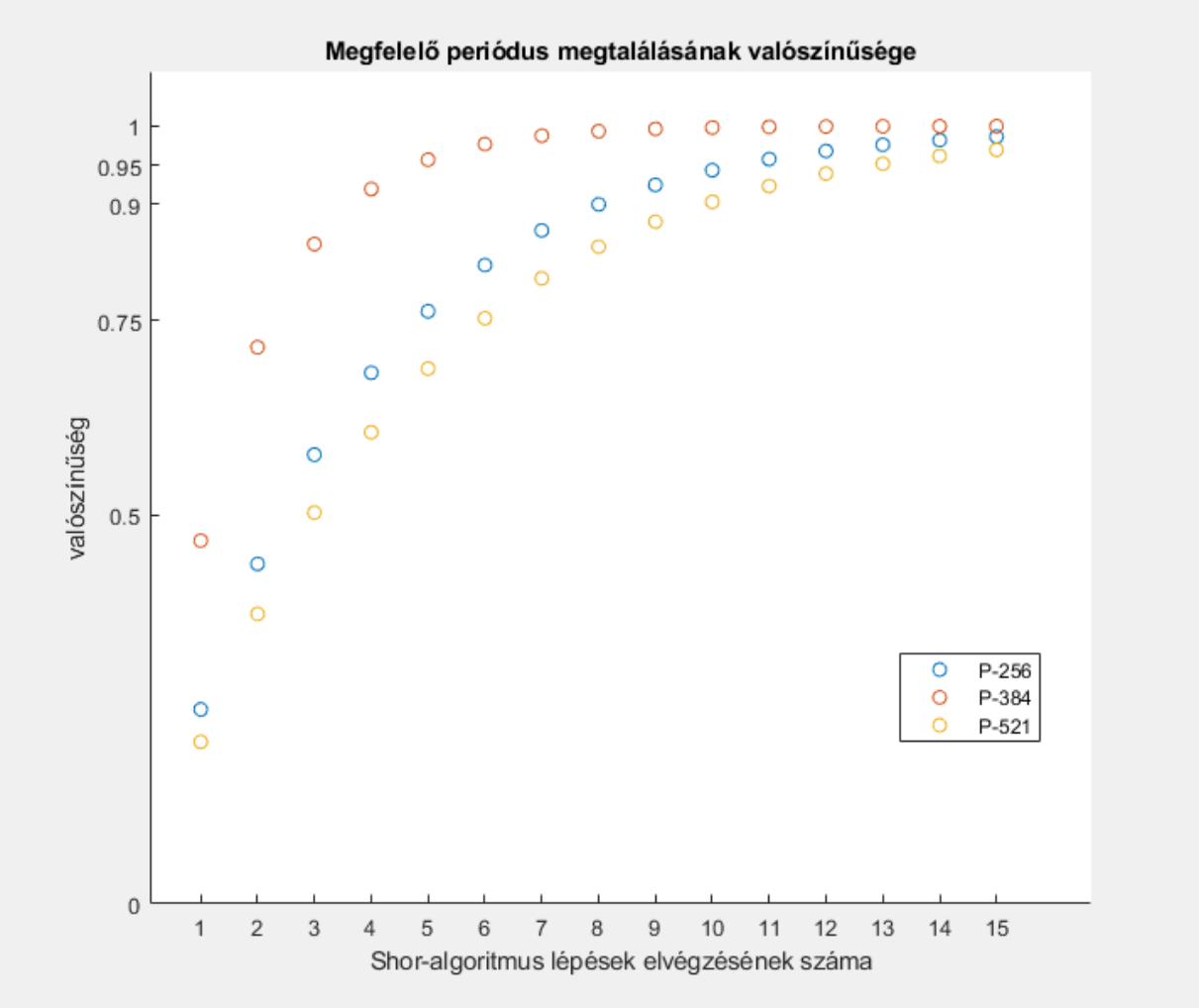
.

Eszerint a formalizmus szerint néhány NIST szabványban is lévő elliptikus görbe hexadecimális paraméterekkel a függelékben látható. (1) alapján számolható annak a valószínűsége, hogy a Shor-algoritmust egyszer futtatva megfelelő periódust találunk. A valószínűségeket matlabban számoltam, de az Euler-függvény számolását nem, mert nagy számoknál nagyon pontatlan a matlab beépített függvénye.

elliptikus görbe	P-256	P-384	P-521
biztonsági szint	128 bit	192 bit	256 bit
$\phi(p-1)$	2.895e76	1.970e115	3.432e156
$P = \frac{\phi(p-1)}{p-1}$	0.24945	0.4666	0.2078

1. táblázat, sorok fentről lefelé: Standard NIST görbék biztonsági szintjei, (p-1) Euler függvénye, annak a valószínűsége hogy a Shor-algoritmus megfelelő perióduspárt talál egy lépés alatt

Ha egy lépés p valószínűséggel vezet sikerre, akkor n lépés után $1-(1-p)^n$ valószínűséggel leszünk sikeresek, a három görbére ez a 7. ábra Megfelelő periódus megtalálásának valószínűsége az elvégzett lépések függvényében látszik.



7. ábra Megfelelő periódus megtalálásának valószínűsége az elvégzett lépések függvényében

Egy lépés a legnagyobb valószínűséggel a P-384 görbe ellen talál olyan periódust, ami megfelelő lesz a diszkrét logaritmus visszaszámolásához. Ezen a görbén 5 lépés után már 95% valószínűséggel sikerrel járunk.

A P-256 görbe ellen 11 lépés után találunk jó periódust 95% valószínűséggel.

Legkisebb valószínűséggel a P-521 görbén találunk jó periódust egy lépés futtatásával, itt a 95% valószínűséghez 13 ciklust kell futtatni.

Kvantumszámítógépek teljesítményének mérése

A kvantumszámítógépek teljesítményének mérésére három mutatót használnak jelenleg: a qubitek számát, kvantum volument és a CLOPSt. A kvantum algoritmusok gyorsaságának futtatását a CLOPS jellemzi vagyis a kvantumszámítógépen a másodperc alatt elvégezhető rétegnyi áramkörök száma [12].

Az IBM jelenleg üzemelő leggyorsabb kvantumszámítógépe körülbelül 1400 CLOPS sebességre képes, így ezt a sebességet fogom feltételezni a továbbiakban.

Kulcsok feltörésének sebessége:

Ebben a részben egy becslést teszek arra, hogy egy jövőbeli kvantumszámítógép mekkora sebességgel lenne képes feltörni a kulcsokat. A becslés során az alábbi feltételezéseket teszem:

1. Rendelkezésünkre állnak a kulcscsere protokoll üzenetei.
2. A kvantumszámítógép 2330 qubites, így a P-256 elliptikus görbét törő Shor-algoritmus futtatható rajta és a teljes áramkör egy réteget alkot [11].
3. A kvantumszámítógép 1400 CLOPS sebességgel képes kvantumáramköröket végrehajtani.
4. A Shor-algoritmus egy lépése $p = 0.2495$ valószínűséggel talál jó periódust.

Ilyen feltételezésekkel élve az alábbi várható törési sebesség érhető el:

$$E(\text{törési sebesség}) = CLOPS \cdot \frac{\text{algoritmus lépés}}{\text{áramkörréteg}} \cdot P(\text{jó periódus}) = 1400 \cdot 0.2495 \ \frac{\text{bit}}{\text{s}}$$

$$= 349.3 \ \frac{\text{kulcs}}{\text{s}}$$

Vagyis egy kulcs feltöréséhez szükséges idő 2.9ms.

9. Összegzés és továbblépési lehetőségek

A jövőben a kriptográfia teljesen új protokollokra fog támaszkodni. Mire elérhetővé válnak olyan kvantumszámítógépek, amik implementálni tudják a Shor-algoritmust egy 256 bites prím modulus ellen, addigra poszt-kvantum szabványokat kell használni.

A jövőbeli kvantumszámítógépek fenyegetést nyújthatnak már ma is az adatainkra, mivel az adatokat le lehet menteni majd később feltörni. A dolgozatban leírt támadás úgy kerülhető el, hogy a kulcscsere alatt történő kommunikáció nem kerül a támadóhoz.

A NIST 2016-ban megkezdte a posztkvantum szabványosítást és 2022-ben kihirdette a kiválasztott szabványokat [14].

Az áttörést várhatóan nem a kvantumáramkör vagy kvantumszámítógép műveleteinek végrehajtási sebessége fogja elhozni, hanem az első olyan hardver, amin implementálni lehet a műveletet.

A kvantumáramkörök alapműveletei mint például az összeadás, kivonás is máshogy vannak implementálva, mint klasszikus áramkörök esetén [15]. Az ilyen alkatrészek optimalizálása jelentősen tudja csökkenteni egy adott algoritmus implementálásához szükséges erőforrást.

Egy kriptográfiai protokoll klasszikus keretek között és kvantum keretek között teljesen más erősségű lehet. Ha egy protokoll klasszikus keretek között erősebb mint más protokollok, abból nem következik hogy ez teljesül kvantum keretek között is. A P-384 görbén alapuló ECDH töréséhez kisebb qubitszámú kvantumszámítógép is elég, de kisebb valószínűséggel törjük egy lépés alatt, mint a P-521 görbén alapuló ECDH-t.

10. Irodalomjegyzék

[1] Samoly, Kevin. „The History of the Abacus”, sz. 65 (2012): 9.

[2] Sarika, Mrs, Vinit Kotak, és Asha Durafe. „A Review Paper on Internet of Things(IoT) and its Applications”, 2019. június 1., 1623.

[3] Imre, Sándor, és Ferenc Balázs. *Quantum Computing and Communications: An Engineering Approach*. Chichester, West Sussex, England: Wiley, 2010.

[4] Packt. „Industrial Internet Application Development”. Elérés 2022. október 16. https://www.packtpub.com/product/industrial-internet-application-development/9781788298599.

[5] „11 Internet of Things (IoT) Protocols You Need to Know About”. Elérés 2022. október 16. https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about.

[6] „Internet of Things Protocols and Standards”. Elérés 2022. október 16. https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/.

[7] „The Architecture of IoT Gateways - DZone IoT”. Elérés 2022. október 17. https://dzone.com/articles/iot-gateways-and-architecture.

[8] Diffie, W., és M. Hellman. „New directions in cryptography”. *IEEE Transactions on Information Theory* 22, sz. 6 (1976): 644–54. https://doi.org/10.1109/TIT.1976.1055638.

[9] Vajda István és Levente Buttyán. *Kriptográfia és alkalmazásai*. Typotex Kft, 2004.

[10] „Algorithms for quantum computation: discrete logarithms and factoring | IEEE Conference Publication | IEEE Xplore”. Elérés 2022. október 31. https://ieeexplore.ieee.org/document/365700.

[11] Roetteler, Martin, Michael Naehrig, Krysta M. Svore, és Kristin Lauter. „Quantum resource estimates for computing elliptic curve discrete logarithms”. arXiv, 2017. október 30. http://arxiv.org/abs/1706.06752.

[12] Wack, Andrew, Hanhee Paik, Ali Javadi-Abhari, Petar Jurcevic, Ismael Faro, Jay M. Gambetta, és Blake R. Johnson. „Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers”. arXiv, 2021. október 28. http://arxiv.org/abs/2110.14108.

[13] „FIPS 197, Advanced Encryption Standard (AES).pdf”. Elérés 2022. november 1. https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf.

[14] Alagic, Gorjan, David A. Cooper, Quynh Dang, Thinh Dang, John M. Kelsey, Jacob Lichtinger, Yi-Kai Liu, és mtsai. „Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process”. *NIST*, 2022. július 5. https://www.nist.gov/publications/status-report-third-round-nist-post-quantum-cryptography-standardization-process.

[15] Cuccaro, Steven A., Thomas G. Draper, Samuel A. Kutin, és David Petrie Moulton. „A new quantum ripple-carry addition circuit”. arXiv, 2004. október 22. http://arxiv.org/abs/quant-ph/0410184.

[16] Feynman, Richard P. „Simulating Physics with Computers”. *International Journal of Theoretical Physics* 21, sz. 6 (1982. június 1.): 467–88. https://doi.org/10.1007/BF02650179.

[17] Vychislimoe Manin YU and I nevychislimoe. “Computable and Non computable (in Russian). Sov.Radio.” In: (Archived from the original on May 10,2013. Retrieved 2013-03-04.), pp. 13–15.

[18] Hitchcock, Yvonne Roslyn. „Elliptic Curve Cryptography for Lightweight Applications”, é. n., 247.

11. Rövidítések jegyzéke

AMQP	Advanced Message Queuing Protocol
BLE	Bluetooth Low Energy
CLOPS	Circuit Layer Operations per Second
CoAP	Constrained Application Protocol
DDS	Data Distribution Service
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDH	Elliptic Curve Diffie Hellmann
IETF	Internet Engineering Task Force
ISM	Industrial, Scientific and Medical
M2M	Machine to Machine
MQTT	Message Queue Telemetry Transport
NIST	National Institute of Standards and Technology
NFC	Near Field Communication
REST	Representational State Transfer
TCP	Transmission Control Protocol
XMPP	Extensible Messaging and Presence Protocol
Wi-Fi	Wireless Fidelity

12. Függelék

P-256 görbe paraméterei

p	0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff
a	0xffffffff00000001000000000000000000000000fffffffffffffffffffffffc
b	0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
G	(0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296, 0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5)
n	0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551
h	0x1

2. táblázat p: prím modulus, a,b: egyenlet paraméterei, G: generátor, n: csoport karakterisztika

P-384 görbe paraméterei

p	0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffeffffffff0000000000000000ffffffff
a	0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffeffffffff0000000000000000fffffffc
b	0xb3312fa7e23ee7e4988e056be3f82d19181d9c6efe8141120314088f5013875ac656398d8a2ed19d2a85c8edd3ec2aef
G	(0xaa87ca22be8b05378eb1c71ef320ad746e1d3b628ba79b9859f741e082542a385502f25dbf55296c3a545e3872760ab7, 0x3617de4a96262c6f5d9e98bf9292dc29f8f41dbd289a147ce9da3113b5f0b8c00a60b1ce1d7e819d7a431d7c90ea0e5f)
n	0xffffffffffffffffffffffffffffffffffffffffffffffffc7634d81f4372ddf581a0db248b0a77aecec196accc52973
h	0x1

3. táblázat p: prím modulus, a,b: egyenlet paraméterei, G: generátor, n: csoport karakterisztika

P-521 görbe paraméterei

p	0x01ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
a	0x01fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffc
b	0x0051953eb9618e1c9a1f929a21a0b68540eea2da725b99b315f3b8b489918ef109e156193951ec7e937b1652c0bd3bb1bf073573df883d2c34f1ef451fd46b503f00
G	(0x00c6858e06b70404e9cd9e3ecb662395b4429c648139053fb521f828af606b4d3dbaa14b5e77efe75928fe1dc127a2ffa8de3348b3c1856a429bf97e7e31c2e5bd66, 0x011839296a789a3bc0045c8a5fb42c7d1bd998f54449579b446817afbd17273e662c97ee72995ef42640c550b9013fad0761353c7086a272c24088be94769fd16650)
n	0x01fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffa51868783bf2f966b7fcc0148f709a5d03bb5c9b8899c47aebb6fb71e91386409
h	0x1

4. táblázat p: prím modulus, a,b: egyenlet paraméterei, G: generátor, n: csoport karakterisztika