Eng. &Tech.Journal, Vol. 33,Part (B), No.1, 2015

Digital Signature Based on Quantum Key Exchange Algorithm

Dr. Samer Saeed Essa
Computer Science Department, AL_Rafidain University /Baghdad.
Email:sammersaeed@yahoo.com

Received on: 10/10/2013 & Accepted on: 7/8/2014

ABSTRACT

Digital signatures are growing in importance as a legal standing with traditional handwritten signatures, most secure web transactions today are already dependent on digital signatures, which are included as part of the digital certificates a server presents to a client identify. The digital certificates bind a key exchange algorithm with an identity and enable a receiver to verify the sender's digital signature, or as case when setting up a secure connection with a web server, enable a client to encrypt information.

This paper, describes the quantum computing with the representation of quantum bit, then this paper explain the proposal algorithm which is serve digital signature under unsecure communication channel by applying the quantum key exchange algorithm which is more complex in degree of computational complexity in data security of client than the classical key exchange algorithm.

التوقيع الرقمي المستند على خوارزمية تبادل المفتاحِ الكَمّية

الملخّص

التواقيع الرقمية تزداد أهمية كمقام قانوني مقارنة بالتواقيع التقليدية المكتوبة باليد, أكثر صفقات الويب الأمنة اليوم أصبحت معتمدة على التواقيع الرقمية التي تضمنت كجزء من الشهادات الرقمية التي تميز الزبون. أن تلك الشهادات الرقمية تربط خوارزمية تبادل المفتاح مع أمكانية تحقق المستلم للتوقيع الرقمي للمرسل أو كحالة تضمن أتصال أمن مع خادم الويب من خلال أمكانية تشفير معلومات الزبون.

في هذا البحث, تم وصف الحاسوب الكمي مع تمثيل البت والذي يدعى (البت الكمي) كما أن هذا البحث يوضح الخوارزمية المفترضة التي تخدم التوقيع الرقمي تحت قناة أتصال غير أمنة وذلك بواسطة تطبيق خوارزمية تبادل المفتاح الكمية التي تعتبر أكثر تعقيداً في درجة التعقيد الحسابي لأمن بيانات الزبون من خوارزمية تبادل المفتاح التقليدية.

INTRODUCTION

Public key cryptography has evolved from a mathematical curiosity to an indispensable part of our information technology infrastructure. It has been used to verify the authenticity of software and legal records, to protect financial

https://doi.org/10.30684/etj.33.1B.9
2412-0758/University of Technology-Iraq, Baghdad, Iraq

transactions, and to protect the transactions of millions of Internet users on a daily basis.

Through most of its history, including present day, public key cryptography has been dominated by two major families of cryptographic primitives: primitives whose security is believed to be contingent on the difficulty of the integer factorization problem, such as RSA and Rabin-Williams, and primitives whose security is believed to be contingent on the difficulty of the discrete logarithm problem, such as the Diffie-Hellman key exchange, and the Digital Signature Algorithm (DSA).

While both the integer factorization problem and the general discrete logarithm problem are believed to be hard in classical computation models, it has been shown that neither problem is hard in the quantum computation model. It has been suggested by Feynman and demonstrated by Deutsch and Jozsa that certain computations can be physically realized by quantum mechanical systems with an exponentially lower time complexity than would be required in the classical model of computation. A scalable system capable of reliably performing the extra quantum operations necessary for these computations is known as a quantum computer. [1]

A Quantum Computer is a computer that harnesses the power of atoms and molecules to perform memory and processing tasks. It has the potential to perform certain calculations billions of times faster than any silicon-based computer.

The classical desktop computer works by manipulating bits, digits that are binary, which can either represent a zero or a one. Everything from numbers and letters to the status of your modem or mouse are all represented by a collection of bits in combinations of ones and zeros. These bits correspond very nicely with the way classical physics represents the world. Electrical switches can be on or off, objects are in one place or they're not, etc. Quantum computers aren't limited by the binary nature of the classical physical world, however they depend on observing the state of quantum bits or qubits that might represent a one or a zero, might represent a combination of the two or might represent a number expressing that the state of the qubit is somewhere between 1 and 0. [2]

Signatures are commonly used to authenticate documents. When you sign a physical document, you are authenticating its contents. Similarly, digital signatures are used to authenticate the contents of electronic documents. They can be used with PDF, e-mail messages, and word processing documents. To digitally sign a document, you must have a digital ID. This unique identifier can obtained from various certification authorities on the Web, such as VeriSign and Echo Sign. Once you have a digital ID, you can add register it with programs that support digital.

The digital signature is simply a small block of data that is attached to documents you sign. It is generated from your digital ID, which includes both a private and public key. The private key is used to apply the signature to the document, while the public key is sent with the file. The public key contains encrypted code, also called a "hash," that verifies your identity. Digital signatures can be used to certify or approve documents. Certifying signatures verify the document's creator and show that the document has not been altered since it was signed. Therefore, only the original creator of a document can add a certifying signature. Approval signatures can be added by anyone with a digital ID and are used to approve documents, track changes, and accept terms stated with a document. [3]

Public key cryptography (PKC) signatures have a greater degree of verifiability than signature dynamics signatures. PKC allows for a third party verification of the signature, while signature dynamics signatures require additional steps (including handwriting analysis) to verify the signer of a document. PKC signatures are designed to be immediately verifiable. Signatures using signature dynamics technology are designed to allow future verification of the signature (similar to a non-notarized, paper-based signature).

Public key cryptography (PKC) signatures are affixed to documents using software enhancements to existing applications and web browsers. Signature dynamics signatures require additional hardware to create the signatures. Signature dynamics signatures are easier for the average user to understand, but they do not provide the level of security that is inherent in PKC signatures, which are immediately verifiable with a third-party issued certificate. [4]

Digital Signature [5]

A digital signature is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file, etc.) is authentic. Authentic means: know who created the document and know that it has not been altered in any way since that person created it. Digital signatures rely on certain types of encryption to ensure authentication. Encryption is the process of taking all the data that one computer is sending to another and encoding it into a form that only the other computer will be able to decode. Authentication is the process of verifying that information is coming from a trusted source. These two processes work hand in hand for digital signatures.

There are several ways to authenticate a person or information on a computer:

- **Password -** The use of a user name and password provide the most common form of authentication. Enter the username and password when prompted by the computer. It checks the pair against a secure file to confirm. If either the name or passwords do not match, then you are not allowed further access.

- **Checksum -** Probably one of the oldest methods of ensuring that data is correct, checksums also provide a form of authentication since an invalid checksum suggests that the data has been compromised in some fashion. A checksum is determined in one of two ways. Let's say the checksum of a packet is 1 byte long, which means it can have a maximum value of 255. If the sum of the other bytes in the packet is 255 or less, then the checksum contains that exact value. However, if the sum of the other bytes is more than 255, then the checksum is the remainder of the total value after it has been divided by 256. For example: sum the number of bits for each byte as see in table bellow equal 1151 divided by 256 equals 4.496 (round to 4) and then Multiply 4 X 256 which equals 1024, finally can get checksum (127) by 1151 minus 1024 equals 127.

Table (1) Checksum Example

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Total	Checksum
212	232	54	135	244	15	179	80	1151	127

- **Cyclic Redundancy Check (CRC) -** CRCs are similar in concept to checksums but they use polynomial division to determine the value of the CRC, which is usually 16 or 32 bits in length. The good thing about CRC is that it is very accurate. If a single bit is incorrect, the CRC value will not match up. Both checksum and CRC are good for preventing random errors in transmission, but provide little protection from an intentional attack on data.

- **Private Key Encryption** -Private Key means that each computer has a secret key (code) that it can use to encrypt a packet of information before it is sent over the network to the other computer. Private Key requires that know which computers will talk to each other and install the key on each one. Private key encryption is essentially the same as a secret code that the two computers must each know in order to decode the information. The code would provide the key to decoding the message. Think of it like this.

- **Public Key Encryption** - Public key encryption uses a combination of a private key and a public key. The private key is known only to your computer while the public key is given by your computer to any computer that wants to communicate securely with it. To decode an encrypted message, a computer must use the public key provided by the originating computer and it's own private key. The key is based on a hash value. This is a value that is computed from a base input number using a hashing algorithm. The important thing about a hash value is that it is nearly impossible to derive the original input number without knowing the data used to create the hash value.

Classical Key Exchange Cryptosystem [6]

In asymmetric or two-key cryptosystems the enciphering and deciphering keys differ in such a way that at least one key is computationally infeasible to determine from the other. Thus one of the transformations (Encryption operation by using Key (E_k)) or (Decryption operation by using Key (D_k)) can be revealed without endangering the other.

Secrecy and authenticity are provided by protecting the separate transformations, D_k for secrecy and E_k for authenticity. This principle can be applied to databases, where some users have read-writer authority to the database, while other users have read authority only. Users with read-write authority are given both D_k and E_k, so they can decipher data stored in the data base or encipher new data to update the database. If E_k cannot be determined from D_k users with read-only authority can be given D_k so they can decipher the data but cannot update it. Thus D_k is like a read-key, while E_k is like a write-key (more precisely the deciphering key describing D_k is the read-key and the enciphering key describing E_k the write-key).

In a public-key system each user A has a public enciphering transformation E_A which may be registered with a public directory, and a private deciphering transformation D_A, which is known only to that user. The private transformation D_A is described by a private key, and the public transformation D_A is described by a private key, and the public transformation E_A by a public key derived from the private key by a one-way transformation. It must be computationally infeasible to determine D_A from E_A (or even to find a transformation equivalent to D_A).

In a public-key system secrecy and authenticity are provided by the separate transformations. Suppose user A wishes to send a message M to another user B. If A knows B's public transformation E_B, A can to B in secrecy by sending the ciphertext $C = E_B$ (M). On receipt B deciphers C using B's private transformation D_B.

To send message (M) from user A to user B in secrecy:

$C = E_{kBU}$ (M), here **KBU** is public key of user **B**

To get the message (M) form ciphertext (C):

$M = D_{kBR}$ (C), here **KBR** is private key of user **B**

For authenticity M must be transformed by A's own private transformation D_A. Ignoring secrecy for the moment, A sends C = D_A (M) to B. On receipt, B uses A's public transformation E_A to compute $E_A(C) = E_A (D_A (M)) = M$.

Quantum Computer

A quantum computer is a computation device that makes direct use of quantum mechanical phenomena, as superposition and entanglement, to perform operations on data. Quantum computers are different from digital computers based on transistors. Whereas digital computers require data to be encoded into binary digits (bits), quantum computation uses quantum properties to represent data and perform operations on these data. A theoretical model is the quantum Turing machine, also known as the universal quantum computer. Quantum computers share theoretical similarities with non-deterministic and probabilistic computers, like the ability to be in more than one state simultaneously. The field of quantum computing was first introduced by Richard Feynman in 1982. A quantum computer with spins as quantum bits was also formulated for use as a quantum space-time in 1969.

In the classical model of a computer, the most fundamental building block - the bit, can only exist in one of two distinct states, a '0' or a '1'. In a quantum computer the rules are changed. Not only can a qubit, exist in the classical '0' and '1' states, but it can also be in a superposition of both! In this coherent state, the bit exists as a '0' and a '1' in a particular manner. Let's consider a register of three classical bits: it would be possible to use this register to represent any one of the numbers from 0 to 7 at any one time. If we then consider a register of three qubits, we can see that if each bit is in the superposition or coherent state, the register can represent all the numbers from 0 to 7 simultaneously.

A processor that can use registers of qubits will in effect be able to perform calculations using all the possible values of the input registers simultaneously. This phenomenon is called quantum parallelism, and is the motivating force behind the research being carried out in quantum computing. [7, 8]

Although quantum computing is still in its infancy, experiments have been carried out in which quantum computational operations were executed on a very small number of qubits (quantum bits). Both practical and theoretical research continues, and many national government and military funding agencies support quantum computing research to develop quantum computers for both civilian and national security purposes, such as cryptanalysis. [9]

Quantum Bits (Qubits) [10]

An implementation of qubits for a quantum computer could start with the use of particles with two spin states: "down" and "up" (typically written |1> and |0>). But in fact any system possessing an observable quantity *A* which is conserved under time evolution and such that has at least two discrete and sufficiently spaced consecutive eigenvalues, is a suitable candidate for implementing a qubit. This is true because any such system can be mapped onto an effective spin-1/2 system.

Consider first a classical computer that operates on a three-bit register. The state of the computer at any time is a probability distribution over the $2^3 = 8$ different three-bit strings 000, 001, 010, 011, 100, 101, 110, 111. If it is a deterministic computer, then it is in exactly one of these states with probability 1. However, if it is a probabilistic computer, then there is a possibility of it being in any *one* of a number of different states. We can describe this probabilistic state by eight nonnegative numbers A,B,C,D,E,F,G,H (where *A* = probability computer is in state 000, *B* = probability computer is in state 001, etc.). There is a restriction that these probabilities sum to 1.

The state of a three-qubit quantum computer is similarly described by an eight-dimensional vector (*a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*), called a ket. However, instead of adding to one, the sum of the *squares* of the coefficient magnitudes, $|a|^2 + |b|^2 + \ldots + |h|^2$, must equal one. Moreover, the coefficients can have complex values. Since the absolute square of these complex-valued coefficients denote probability amplitudes of given states, the phase between any two coefficients (states) represents a meaningful parameter, which presents a fundamental difference between quantum computing and probabilistic classical computing.

If you measure the three qubits, can observe a three-bit string. The probability of measuring a given string is the squared magnitude of that string's coefficient (i.e., the probability of measuring 000 $=|a|^2$, the probability of measuring 001 $=|b|^2$, etc..). Thus, measuring a quantum state described by complex coefficients (*a*, *b*,... *h*) gives the classical probability distribution ($|a|^2$,$|b|^2$, …, $|h|^2$) and we say that the quantum state "collapses" to a classical state as a result of making the measurement.

Note that an eight-dimensional vector can be specified in many different ways depending on what basis is chosen for the space. The basis of bit strings (e.g., 000, 001, ..., 111) is known as the computational basis. Other possible bases are unit-length, orthogonal vectors and the eigenvectors of the Pauli-x operator. Ket notation is often used to make the choice of basis explicit. For example, the state (a, b, c, d, e, f, g, h) in the computational basis can be written as:

a|000> + b|001> + c|010> + d|011> + e|100> + f|101> + g|110> + h|111>

Where
e.g., |000> = (1, 0, 0, 0, 0, 0, 0, 0)
|001> = (0, 1, 0, 0, 0, 0, 0, 0)
|010> = (0, 0, 1, 0, 0, 0, 0, 0)
|011> = (0, 0, 0, 1, 0, 0, 0, 0)

The Qubit Measurement Complexity [11]

The computational basis for a single qubit (two dimensions) is |0> = (1, 0) and |1> = (0, 1). Can prepare to look more closely at the measurement complexity principle. Consider the quantum state:

$$\frac{1}{\sqrt{2}}\left(|0\rangle\right) + \frac{1}{\sqrt{2}}\left(|1\rangle\right)$$

If the measure this qubit in the standard basis, the outcome would be 0 with probability 1⁄2 and 1 with probability 1⁄2. This measurement tells us only about the norms of the state amplitudes. To see if can gather any phase information; let us consider a measurement in a basis other than the standard basis, namely:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \text{ and } \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

Quantum Key Exchange Cryptography

The principle of quantum cryptography consists in the use of non-orthogonal quantum states. Its security is guaranteed by the Heisenberg uncertainty principle, which does not allow us to discriminate non-orthogonal states with certainty and without disturbing the measured system. It should be noted that quantum mechanics does not avert eavesdropping; it only enables us to detect the presence of an eavesdropper. Since only the cryptographic key is transmitted, no information leak can take place when someone attempts to listen in. When discrepancies are found, the key is simply discarded and the users repeat the procedure to generate a new key. [12] Public key cryptographic mechanisms are used to provide the authentic channel needed for quantum key exchange. The key exchange sub-system, and hence the overall communications system, will be no more secure than the public key authentication mechanism on which it is based. Moreover, any system using quantum key exchange requires a quantum channel (e.g., an optical fiber) between the communicating parties.

Naturally, such a system would not resist active attacks subsequent to private key compromise. The quantum state of a string of qubits (quantum bits) is used as a key. Storage, distribution and manipulation of these quantum keys, however, require quantum information processing capabilities beyond the reach of current technology.

This system is merely a method for exchanging keys; no messages are involved. The both side in the first publicly choose a finite field $|F_q>$. Then they publicly choose an element $|g> \in |F_q>$ to serve as their “base element” ($|g>$ is preferably, but not necessarily the generator of the group of elements on $|F_q>$). It is a generator of the key. To generate a key, one side chooses a random integer $|a>$ of order of magnitude $|q>$ and keeps it secret, then computes $|g^a>$ *mod* $|q> \in |F_q>$, and makes that public. Another side chooses his own secret random integer *b* and makes public $|g^b>$ *mod* $|q> \in |F_q>$. The secret key is then g^{ab} *mod q* $\in F_q>$. Both them can compute this key. For example, one knows $|g^b>$, (public knowledge) and another own secret $|a>$. and use it for further secure communication. On the other hand, only knows $|g>$,, $|g^a>$ and $|g^b>$, (finding $|a>$ knowing $|g>$ and $|g^b>$), there is no way for him to compute $|g^{ab}>$ only knowing $|g^a>$ and $|g^b>$. [10]

The Proposed Algorithm

This section present the proposed algorithm to describe digital signature based on quantum key exchange that contain six steps of this algorithm.

Initial Public Elements

Step1.1: Sender (**X**) and Receiver (**Y**) publicly choose a prime number |q> from finite field $|F_q>$.

Step1.2: They publicly choose a random element $|a> \in |F_q>$ such that |a> generates a large subgroup of $|F_q>$, where |a> is less than |q>.

Sender (X) Key generation

Step2.1: Sender (**X**) chooses a secret random integer **x** and Convert the **x** chosen number into quantum representation:

$$\mathbf{x} = |\mathbf{x}_1, \mathbf{x}_2 , ... , \mathbf{x}_n\rangle \text{ where } |\mathbf{x}> \text{ is less than } |\mathbf{q}>$$

Step2.2: Sender (**X**) compute public key:

$$|\mathbf{k_x}> = \ |\mathbf{a}>^{|x>} \mathbf{mod} \ |\mathbf{q}> \text{ where } |\mathbf{k_x}> \in \ |\mathbf{F_q}>.$$

Receiver (Y) Key generation

Step3.1: Receiver (**Y**) chooses a secret random integer **y** and Convert the **y** chosen number into quantum representation:

$$\mathbf{y} = |\mathbf{y_1, y_2 , ... , y_n}\rangle \text{ where } |\mathbf{y}> \text{ is less than } |\mathbf{q}>$$

Step3.2: Receiver (**Y**) compute public key:

$$|\mathbf{k_y}> = \ |\mathbf{a}>^{|y>} \mathbf{mod} \ |\mathbf{q}> \text{ where } |\mathbf{k_y}> \in \ |\mathbf{F_q}>.$$

Public and Secret Key

Step4.1: Make **| k$_x$> and | k$_y$>** are public keys.

Step4.2: Make |x> and |y> are secret keys.

Signature Generation

Step5.1: Generate signature of sender(X) by using his secret key |x> with public key **|k$_y$>** of receiver(Y):

$$|\mathbf{S_x}> = |\ \mathbf{k_y}\ >^{|x>} \mathbf{mod}\ |\mathbf{q}>$$

Step5.2: Generate signature of receiver(Y) by using his secret key |y> with public key **|k$_x$>** of sender(X):

$$|\mathbf{S_y}> = |\ \mathbf{k_x}\ >^{|y>} \mathbf{mod}\ |\mathbf{q}>$$

Step5.3: The signature is valid when |**S$_x$**> is equal to |**S$_y$**>.

Verifying of Signature

Step 6.1: The sender(X) can sign the message (encrypt message) by using his secret key |x> while receiver(Y) can check the signature of sender(X) by using the public key **| k$_x$** > of sender(X) with receiver(Y) secret key |y>.

Step 6.1: The receiver(Y) can sign the message (encrypt message) by using his secret key |y> while sender(X) can check the signature of receiver(Y) by using the public key **| k$_y$** > of receiver(Y) with sender(X) secret key |x>.

CONCLUSIONS

1. This paper explains the quantum computing with the representation of bit called quantum bit (Qubit) which is used to increase the complexity degree of data security in a fraction of the time that it would take a conventional computer.

2. Quantum computation has the ability to perform any computation (task) faster than classical computer can do.

3. The quantum key exchange cryptosystem is more complex in degree of computational security than classical key exchange cryptosystem

4. In the proposed algorithm, the key generation has two phases, the first phase is a choice of algorithm parameters which may shared between different user (Sender and Receiver), while the second phase computes public and private keys for a single user (Sender or Receiver), therefore the security of this proposed algorithm is recognized by high secrecy degree and uniqueness of random signature values of keys.

REFERENCES

[1]. Ray A. Perlner and David A. Cooper, "Quantum Resistant Public Key Cryptography", National Institute of Standards and Technology, 2009.

[2]. P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring", In Proceedings of the 35th Symposium on Foundations of Computer Science, 1994.

[3]. www.TechTerms.com, "Digital Signature", software terms, 2009.

[4]. California Secretary of State, "What is Digital Signature", frequently Asked Questions, 4 may 2012.

[5]. www.HowStuffWorks.com , "What is a Digital Signature?", computer security, HowStuffWorks Inc, 1988-2012.

[6]. Dorothy Elizabeth Robling, "Denning, Cryptography & Data Security", CRC Press, 1983

[7]. Neil Gershenfeld and Isaac L. Chuang, "Quantum Computing with Molecules", article in Scientific American, 2001.

[8]. David Deutsch, "Quantum computation", Physics World, 1992.

[9]. Simon, D.R., "On the power of quantum computation", Foundations of Computer Science, 1994.

[10]. Eleanor Rieffel & Wolfgarg Polak, "An Introduction to Quantum Computer for Non-Physicists", www.rieffelpal.xerox.com and www.Polak.pal.xerox.com ,1998.

[11]. Andrew Steane, "Quantum Computing", Department of Physics Vol61. PP117-173, 1998.

[12]. AVINASH G. PILLAI, "Quantum Cryptography", University of Washington, avinashp at cs.washington.edu, 2007.