Int. J. Nonlinear Anal. Appl. 11 (2020) No. 2, 199-208
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2020.4421

Proposing a Novel Method for Increasing Security in the Network Communication

Samiyeh Khosravi*

Computer Engineering Department, Faculty of Engineering, University of Birjand, Birjand, Iran.

(Communicated by Madjid Eshaghi Gordji)

Abstract

Advances in quantum computer technology are threatening cryptographic systems based on mathematical complexity. As an alternative, quantum cryptography has been proposed and developed. Among the reported quantum cryptography systems, quantum key distribution (QKD) constitutes a symmetric key system that can securely distribute a secret key on a quantum channel. While there have been many studies on confidential communication using QKD, only a few of them address its application to the digital signature. In this paper, we propose a new digital signature method with public parameter and signature key through shared symmetric key from QKD. We also analyze the proposed plan from security.

Keywords: Network security, Artificial intelligence, Quantom cryptography.
2010 MSC: 68M10

1. Introduction

Quantum computer technology, which has undergone dramatic improvements in recent years, is expected to solve various problems in the field of science and technology. Meanwhile, it may pose a severe security threat in the field of modern cryptography, where security is based on mathematical complexity. The widely used digital signature is a public key method based on factorization and discrete logarithm [1, 2]. Shor's algorithm, a representative quantum algorithm, can solve this problem efficiently [3, 4]. Therefore, it is necessary to study more secure signature techniques. In this regard, one-time signature (OTS) [5] and arbitrated digital signature (ADS) have been proposed

*Corresponding Author: Samiyeh Khosravi
Email address: skhosravi@birjand.ac.ir (Samiyeh Khosravi*)

Received: October 2019 *Revised:* February 2020

[6]. Both are based on a symmetric key system. An OTS scheme makes use of public parameters. Such parameters are generated through a one-way function that guarantees preimage resistance. And, their signature can be verified. However, there exists a practical problem because the public parameter size must be at least twice the message size [2]. By contrast, ADS can be implemented more efficiently than the signature-based on OTS. However, there have been no proposals about how to share a secret key. Besides, an individual trusted a third party(TTP) is required.

In this paper, we propose a new digital signature that is combined with OTS and ADS. Our digital signature uses quantum key distribution (QKD) to securely share the secret key among signer, verifier, and third party (TP). This symmetric key is in turn used to generate a public parameter for signature and verification. The symmetric key shared with the TP is only employed for making the public parameter. Only the signer and verifier know the symmetric key for signature and verification, so it does not require unconditional TTP.

The concept of the proposed digital signature method using quantum symmetric keys is introduced in Section 2; Section 3 presents an index sift method that constitutes the base technology for extracting public parameters, including analysis of its security. In Section 4, we analyze the security of the signature key, the possibility of forgery of the proposed digital signature, and the security of the non-repudiation.

2. A digital signature protocol using quantum symmetric keys

In the proposed digital signature protocol, Alice generates the signature, Bob receives the signature, and Charlie verifies the signature participate as communication members. The signature protocol consists of a preparation phase, a signature phase, and a verification phase. First, in the preparation phase, three users share a secret key using QKD. Next, in the signing phase, Alice creates a signature pair using an index sift method and sends it to Bob. In the final verification phase, Bob verifies the signature received from Alice with the help of Charlie. Figure 1 shows a schematic diagram of the proposed digital signature.

2.1. Training step

Before proceeding with the signature protocol, Alice, Bob, and Charlie use QKD to share the respective symmetric keys QK_{AB}, QK_{BC}, and QK_{AC} in the preparation phase. The QKD protocol used at this time may be the BB84 protocol [7], which is the most widely applied in QKD. We must remark that QKD is well known to be safe in a quantum computing environment because it uses a quantum channel to which the principle of uncertainty and the no-cloning theorem of quantum states are applied [8,9]. Moreover, the secret key obtained through this protocol enables confidential communication, which ensures the security of the digital signature to be performed afterward.

2.2. Signature step

The signature phase consists of three stages: the signature key generation, signature pair generation, and transmission and verification. In the signature key generation step, with a symmetric key QK_{AC} and a public parameter P_A, Alice extracts the key index I_{AC} using a sift index method, as described in the next section. The parameter P_A is generated through a true random number generator [10]. Next, she creates the signature key K_{sign} with the key index and QK_{AB}. Then, in the signature pair generation and transmission step, she generates a signature pair using the signature key K_{sign} and transmits it to Bob. Finally, in the verification step, Bob verifies the signature pair [19, 20].

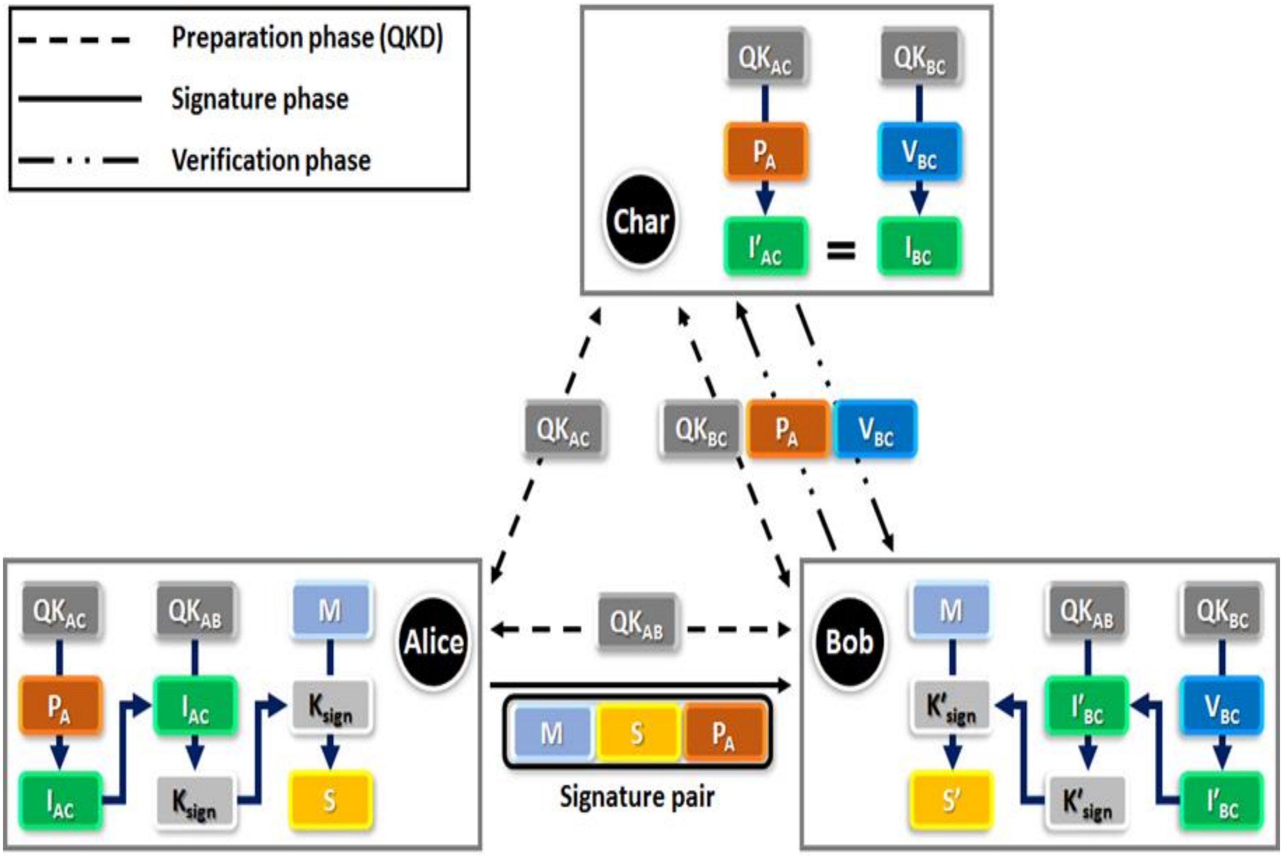


Figure 1: Schematic representation of digital signature with public parameter from quantum symmetric keys.

1. The signature key generation step
   G1. Alice creates a public parameter P_A, which is a random number.
   G2. Alice uses the public parameter P_A as index information to extract the key index $I_{AC} = f_{P_A}(QK_{AC})$ from the quantum symmetric key QK_{AC}. Here $f(\cdot)$ is a function expressing the sift index method – the details of this function are described in Section 3.
   G3. Alice generates a signature key $K_{sign} = f_{I_{AC}}(QK_{AB})$ from the quantum symmetric key QK_{AB} like that described in step **G2** for the key index I_{AC}.
2. Signature pair generation and transmission step
   S1. Alice generates a message M to be sent via a signature, and then she makes a signature $S = f_{K_{sign}}(M)$ using the signature key K_{sign}. In this paper, $M \parallel S = M \parallel f_{K_{sign}}(M)$ is called a signature pair.
   S2. Alice sends the signature pair $M \parallel S$ to Bob along with the public parameter P_A.

2.3. Confirmation step

In the verification step, Bob requests Charlie to verify the signature pair $M \parallel S$ sent to Alice. Charlie generates a verification parameter V_{BC} using the symmetric key, QK_{AC}, QK_{BC} and the public parameters P_A. Next, he sends the verification parameter V_{BC} to Bob, according to Bob's request. Finally, Bob obtains the signature key K_{sign} using the verification parameter V_{BC} transmitted from Charlie and the symmetric keyQK_{BC} QK_{AB}. He employs the signature key to verify the signature pair $M \parallel S$. The procedure of the verification step is as follows [21]:

V1. Bob sends a public parameter$P_A$to Charlie and requests a verification parameterV_{BC}.

V2. Charlie extracts the key index $I'_{AC} = f_{P_A}(QK_{AC})$ from the symmetric key QK_{AC} using the public parameter P_A transmitted from Bob as index information.
V3. Charlie generates a verification parameter V_{BC} that allows the key index I_{BC} to be obtained from the symmetric key QK_{BC} and sends it to Bob, where V_{BC} is a parameter that fulfills$I_{BC} = I'_{AC}$.
V4. Bob obtains the key index $I'_{BC} = f_{V_{BC}}(QK_{BC})$ from the symmetric key QK_{BC} using the verification parameter V_{BC} transmitted from Charlie as index information. The key index obtained as described above is used again as index information to get the signature key $K'_{sign} = f_{I'_{BC}}(QK_{AB})$ from the symmetric key QK_{AB}.
V5. Bob encrypts the message M transmitted from Alice using the signature key K'_{sign}, and applies the hash function to the encrypted message to obtain the signature $S' = f_{K'_{sign}}(M)$. Finally, we verify that the signature $S = f_{K_{sign}}(M)$that Alice sent matches the signature $S' = f_{K'_{sign}}(M)$ [21].

3. Sift index technique

The sift index method proposed in this paper is a technique to obtaining an m result bit sequence $R = (r_1,\ r_2,\ \cdots,\ r_m)$ using an n index bit sequence $I = (id_1,\ id_2,\ \cdots,\ id_n)$ and distance bit sequence $D = (d_1,\ d_2,\ \cdots,\ d_n)$ on an n origin bit sequence $O = (o_1,\ o_2,\ \cdots,\ o_n)$. Here, the index bit sequence $id_j \in \{0, 1\}$ is a random number. Then, the distance bit sequence D is generated by an index bit sequence I where d_j indicates the interval of 1 in the index bit sequence I. If d_j is 0, we do not select o_j. Conversely, if d_j is 1, we select o_j, and the selected bit sequence o_j is stored in the database of the result bit sequence $R_{D(1)}$.
Similarly, if d_j is n, we select o_j, and the selected bit sequence o_j is stored in the database of the result bit sequence $R_{D(k)}$. Finally, $R_{D(1)}$, $R_{D(2)}$, $\cdots$, $R_{D(n)}$ are stored sequentially in the result bit sequence R. As a result, the size m of the result bit sequence R is equal to the number of 1's in the index bit sequence I, and the index order is shuffled. The sift index method can be expressed as a function [22]:

$$R_{D(k)} = \tfrac{1}{2}\sum_{j=1}^{n}\{o_j \cdot d(k)_j \cdot 2^{[\sum_{p=1}^{j} d(k)_p]}\}, d(k)_j = \left\{ \begin{array}{ll} 0, & d_j \neq k \\ 1, & d_j = k \end{array} \right. \cdots \qquad (3.1)$$

When $R_{D(n)}$ is expressed as binary numbers, $o_j \cdot d(k)_j$ it represents a selection value, whereas $2^{[\sum_{p=1}^{j} d(k)_p]}$ represents a selection bit. In this case, if $d(k)_p$ is 0, the corresponding bit is not used, regardless of the o_j value. Therefore, in the proposed sift index method, the origin bit sequence O cannot be restored with the index distance information $d(n)_p$and the result bit sequence R because the knowledge of the bit $dl_p = 0$ is removed [23].

For example, if the origin bit sequence is $O = (0100110100011101)$ and the index bit sequence is $I = (1011100010100011)$, then the distance bit sequence is $D = (102110004020041)$, resulting in $R_{D(1)} = (0011)$, $R_{D(2)} = (00)$, $R_{D(4)} = (00)$, and finally the result bit sequence is $R = (00110000)$. Figure 2 shows an example of the sift index method described above.

However, in the case of $\sum_{p=1}^{j} d(n)_p = j$ (that is when the entire distance bit sequence D consists of bit "1") there is an inverse function. Conversely, when $\sum_{p=1}^{j} d(n)_p < j$, there is no inverse function. Given that the id_p value is generated by a random number generator during the proposed protocol operation, the values of 0 and 1 exist at 5:5. The distance bit sequence D is dependent on the index bit sequence I, and the inverse function of Eq. (1) is $f_I^{-1}(R) \neq O$. In other words, if $\sum_{p=1}^{j} dl_p < j$, then $f(\cdot)$ can be viewed as a one-way function [23, 24].

Generally, the random number sequence of n bits can be found with a probability of $1/2^n$. If the eavesdropper who obtained the information of the index bit sequence I and the result bit sequence

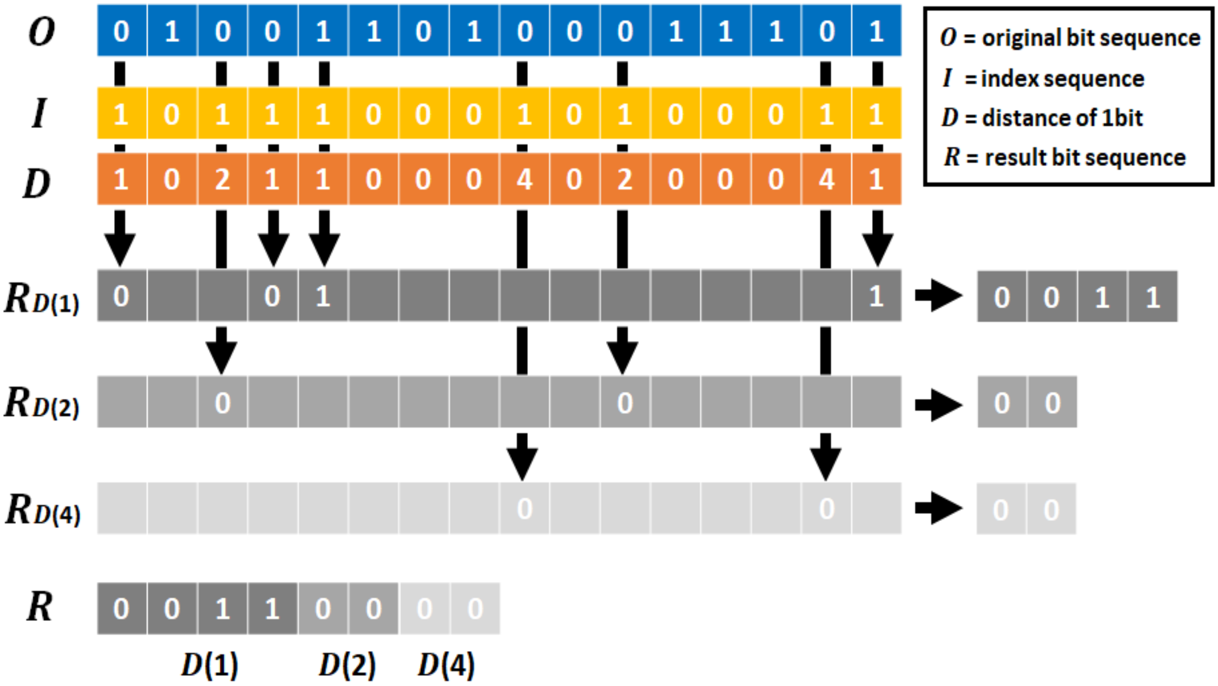


Figure 2: Schematic representation of the proposed sift index method

R tries to find the origin bit sequence O, he can find it out with a probability of $1/2^n$ if the ratio between bit 0 and 1 of the index bit sequence I is 5:5. Thus, if the value of n is enormous, the likelihood of the eavesdropper knowing the information becomes very small. Hence, the safety of the result bit sequence is guaranteed [25].

4. Security investigation

We analyze the security of the signature key, the possibility of forgery of the signature pair, and the non-repudiation to show that the proposed protocol is secure [1, 2, 11-13]. In the proposed protocol, the signature key is generated by applying the aforementioned sift index method and the quantum symmetric key QK_{AC}, which is shared in advance between the signature generator Alice and the verifier Charlie using QKD. Therefore, the security of the signature key is based on the safety of the quantum symmetric key and the sift index method. Forgery of a signature pair is only possible when the signature key is exposed. Non-repudiation is a function that can exclusively be provided by a digital signature, mainly to prevent repudiation of the signature receiver and denial of the signature generator [14-15]. The proposed protocol includes non-repudiation using public parameters and verification parameters. Specific details are described below [26].

4.1. Security of signature key

The quantum keys QK_{AB}, QK_{BC}, QK_{AC} are all made secure through QKD. Given that the signature key uses QKs as the origin bit sequence O of the sift index method, the result bit sequence R generated from the signature key is secured as described in Section 3. Moreover, knowing the signature key K_{sign}requires in turn to know I_{AC} and QK_{AB} orP_A, QK_{AC} , and QK_{AB}. Likewise, understanding K'_{sign} needs to know I_{BC} and QK_{AB} or V_{BC}, QK_{BC}, and QK_{AB}. In other words, Alice only has all the information to render K_{sign}, and Alice's signature key K_{sign} can be secured because K'_{sign} cannot be generated by Charlie alone or by Bob alone [27].

The security of the result bit sequence R in the sift index method is determined by the security of the origin bit sequence O. The id_j of the index bit sequence only determines whether o_j in the origin bit sequence O is used or not. Therefore, if someone exclusively acquires the index bit sequence I, only the position of o_j to be used as the result bit sequence R in the origin bit sequence O can be known.

4.2. Investigation of the counterfeit possibility of signature pair

To create a signature pair, we use the proposed sift index method. It is characteristic of a one-way function (except that the index bit sequence element id_j is all 1s (ex $I = (111111111111))$), so its safety is guaranteed. Moreover, even if Charlie transmits a public parameter P_A, given that there is no QK_{AB}, only the index bit sequence I_{AC} can be generated, whereas K_{sign} it cannot be created. This ensures the security of the signature pair.

4.3. Investigation of non-repudiation

1. Non-repudiation of the signature generator.
   The signature generator Alice may deny Bob the signature pair transmission. To prevent the denial, the protocol establishes that Alice must send Bob a public parameter P_A that can be generated only by Alice with a signature pair. If Bob sends P_A to the verifier Charlie, Charlie can disable Alice's denial based on the public parameter P_A.
2. Non-repudiation of the signature receiver
   On the contrary, the signature receiver Bob may deny this behavior even though Alice's signature pair $M \parallel S = M \parallel f_{K_{sign}}(M)$ had been received and verified. To prevent denial from Bob, he must send a public parameter P_A to Charlie when he requests a verification parameter V_{BC}. Based on the fact that Bob has P_A, Charlie can prevent Bob's non-repudiation.

4.4. Assessment with present systems

One of the traditional digital signatures, i.e., OTS, generates public parameters and provides complete security. However, it presents a critical disadvantage, namely the size of the public parameter must be twice the size of the message. Another digital signature scheme of ADS that adopts the third party for verification of a signature does not have this key size issue. Instead, given that the private key must be shared with the third party as well, a third party must be unconditionally trusted. Remarkably, the proposed signature does not require to have a certain trusted third party and provides secure signature service with the symmetric key generated by QKD, as described in previous section. A comparison of the characteristics of digital signatures is shown in Table 1. For a detailed description of digital signatures of OTS and ADS, see Appendix.

5. Conclusion

In this paper, we generate a public parameter using a quantum symmetric key from QKD and propose a new signature protocol based on it. The proposed protocol consists of a signature generator (Alice), receiver (Bob), and verifier (Charlie). Alice generates a public parameter P_A and creates a signature key K_{sign} using a sift index method. At this time, even if the public parameter P_A is known, Charlie, who does not have QK_{AB} , can safely provide a digital signature. This means that the digital signature operates well without requiring the existence of a particular trusted third party because K_{sign} cannot be detected. Moreover, the proposed protocol can be easily implemented because it mainly uses a sift index method that can be realized with simple logic. Therefore, it is expected to be widely applied.

Table 1: Assessment of Digital Signatures

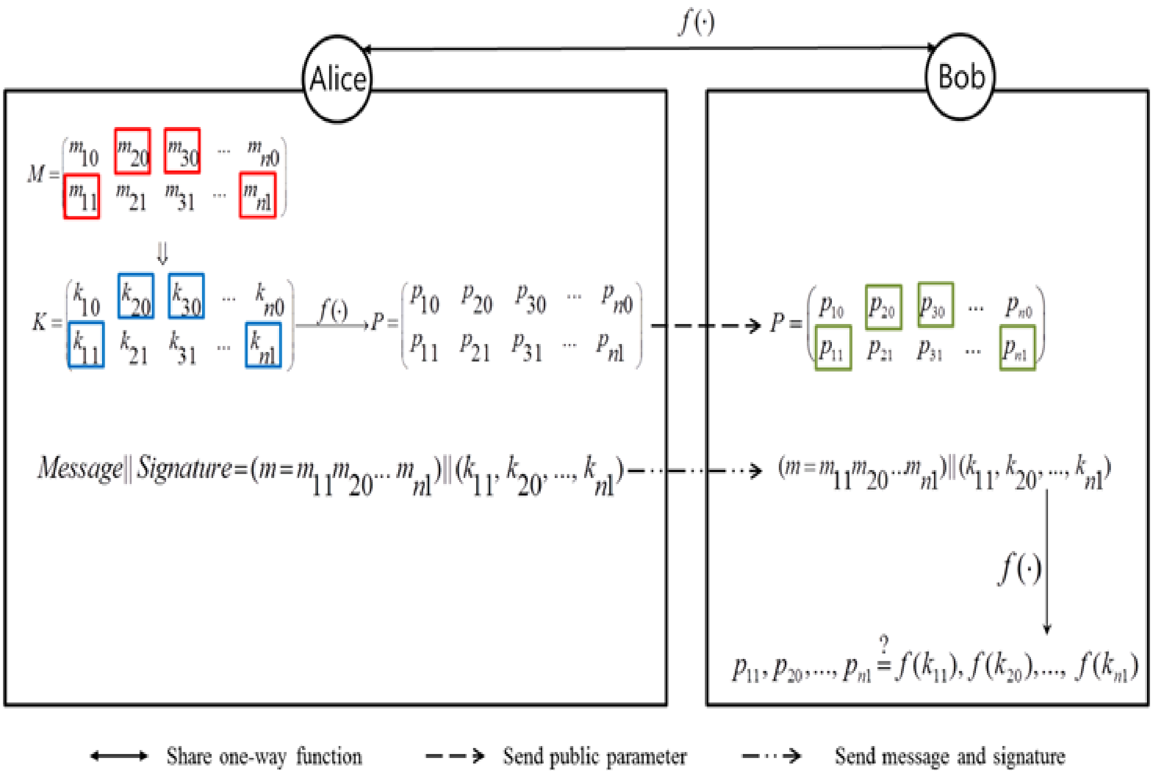
OTS	ADS	Proposed system
-	-	Used Quantum Key Distribution
Used public parameter	-	Used public parameter
-	Unconditionally trusted the third party	Third-party
- Unconditionally secure - The public parameter size is more than twice the message size, so there is a problem in practical terms	- Given that this signature is based on a symmetric key system, it is implemented more efficiently than the signature-based on an asymmetric key system - Problem with sharing the private key - Require unconditional trusted third party	- The symmetric key is shared by QKD - Provide public parameter by the index method - Unconditionally trusted third party is not required

6. Appendix A. One-time signature (OTS)

One-time signature (OTS) was originally proposed by Rabin in 1978, and a similar OTS was proposed by Lamport in 1979. This scheme guarantees safety by creating public parameters through a one-way function rather than a trap-door one-way function [2, 5]. One-time signature is illustrated in Figure A.

It consists of a preparation phase, a signature phase, and a verification phase. The details of each step are described next:

1. Preparation step (P')
   P'1. Alice, who is the signer, shares the one-way function $f(\cdot)$with Bob in advance.
   P'2. Alice generates the message bits m_{ij} $(1 \leq i \leq n,\ j \in \{0,1\})$ where i represent the message length and j represents bit '0' or '1'.
   P'3. Alice generates the private key k_{ij}corresponding to message bits m_{ij}.
   P'4. After creating a public parameter p_{ij} using $f(\cdot)$ and k_{ij}, Alice sends it to Bob.
2. Signature step (S')
   S'1. Alice generates a message $m = m_{11}\, m_{20}...m_{n1}$ by selecting among message bits m_{ij}.
   S'2. Alice selects private keys $(k_{1j},\ k_{2j}, ...,\ k_{nj})$ corresponding to the message $m = m_{11}\, m_{20}...m_{n1}$. Here, selected k_{ij} is called signature.
   S'3. Alice sends the message $m = m_{11}\, m_{20}...m_{n1}$ and the signature $(k_{1j},\ k_{2j}, ...,\ k_{nj})$ to Bob.
3. Verification step (V')
   V'1. Bob computes the signature $(k_{1j},\ k_{2j}, ...,\ k_{nj})$ by applying the one-way function $f(\cdot)$, thus acquiring one-way function values$f(k_{1j}),\ f(k_{2j}),\ ....f(k_{nj})$.
   V'2. Bob selects the public parameters $p_{11}\,,\ p_{20},\ ...,p_{n1}$ corresponding to the message of the public parameter transmitted from Alice.
   V'3. Bob verifies whether. $p_{11}\,,\ p_{20}, ...,p_{n1}$ are equal to $f(k_{1j}),\ f(k_{2j}),\ ....f(k_{nj})$.
   As described in P4 and S3, there is a practical problem arising from the size of the public parameter, which must be more than twice the size of the message.



$f(\cdot)$
Alice
Bob
$M = \begin{pmatrix} m_{10} & m_{20} & m_{30} & \cdots & m_{n0} \\ m_{11} & m_{21} & m_{31} & \cdots & m_{n1} \end{pmatrix}$
⇓
$K = \begin{pmatrix} k_{10} & k_{20} & k_{30} & \cdots & k_{n0} \\ k_{11} & k_{21} & k_{31} & \cdots & k_{n1} \end{pmatrix} \xrightarrow{f(\cdot)} P = \begin{pmatrix} p_{10} & p_{20} & p_{30} & \cdots & p_{n0} \\ p_{11} & p_{21} & p_{31} & \cdots & p_{n1} \end{pmatrix}$
$P = \begin{pmatrix} p_{10} & p_{20} & p_{30} & \cdots & p_{n0} \\ p_{11} & p_{21} & p_{31} & \cdots & p_{n1} \end{pmatrix}$
$Message \| Signature = (m = m_{11} m_{20} \cdots m_{n1}) \| (k_{11}, k_{20}, \ldots, k_{n1})$
$(m = m_{11} m_{20} \cdots m_{n1}) \| (k_{11}, k_{20}, \ldots, k_{n1})$
$f(\cdot)$
$p_{11}, p_{20}, \ldots, p_{n1} \overset{?}{=} f(k_{11}), f(k_{20}), \ldots, f(k_{n1})$
Share one-way function
Send public parameter
Send message and signature

Figure A: Schematic representation of one-time signature:$f(\cdot)$: one-way function, M_{ij}: message bit, K_{ij}: privacy key, p_{ij} : a public key, Red box □ : message from message bit, Blue box □ : signature, Green box □ : selected public parameters.

7. Appendix B. Arbitrated digital signature (ADS)

Arbitrated digital signature (ADS) was proposed by Davies and Price in 1989 [1]. This signature is based on a symmetric key system. It is faster than the public key system. The arbitrated digital signature is illustrated in Figure B.

It consists of a preparation phase, a signature phase, and a verification phase. The details of each step are explained next:

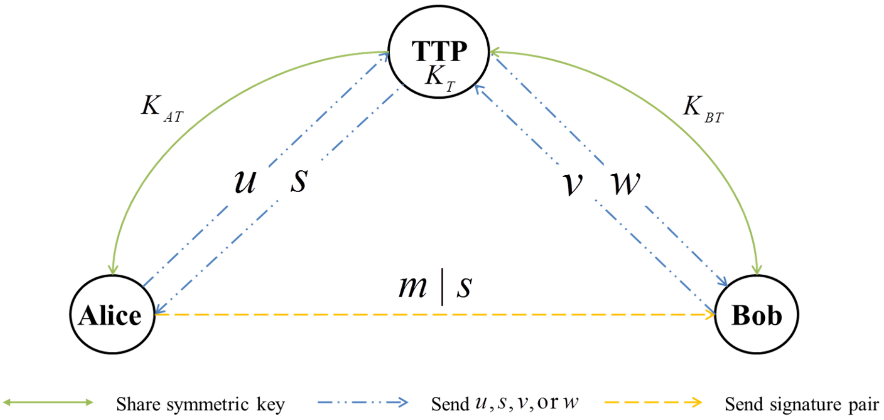


Figure B: Schematic representation of arbitrary digital signature: $h(\cdot)$: hash function, m : message, S : signature, $m||s$: signature pair $E_{K_{AT}}$and $E_{K_{BT}}$: symmetric key, E_{K_T} : private key of TPP, $u = E_{K_{AT}}(H)$, $s = E_{K_T}(H)$, $v = E_{K_{BT}}(s)$, $w = E_{K_{BT}}(w)$, $H = h(m)$

1. Preparation step (P”)
   P”1. The TTP shares symmetric key K_{AT} with Alice and a symmetric key K_{BT} with Bob.
   P”2. The TTP generates his private key K_T.
2. Signature step (S”)
   S”1. Alice computes the message $m$and the hash function $h(\cdot)$ to acquire the hash value $H = h(m)$.
   S”2. Alice encrypts H using K_{AT} to have $u = E_{K_{AT}}(H)$, and then sends it to the TTP.
   S”3. The TTP decrypts u to have $H = E^{-1}_{K_{AT}}(u)$.
   S”4. The TTP encrypts H using K_T to have $s = E_{K_T}(H)$, and then sends it to Alice.
   S”5. Alice sends the signature pair $m||s$ to Bob. Here m is the message and s is the signature.
3. Verification step (V”)
   V”1. Bob encrypts s to have,$v = E_{K_{BT}}(s)$ and then sends it to the TTP.
   V”2. The TTP decrypts v using K_{BT} and K_T to have $H = E^{-1}_{K_T}(E^{-1}_{K_{BT}}(v))$.
   V”3. The TTP encrypts H using K_{BT} to have $w = E_{K_{BT}}(w)$, and then sends it to Bob.
   V”4. Bob decrypts w to have $H = E^{-1}_{K_{BT}}(w)$.
   V”5. Bob computes the received message $m$and the hash function $h(\cdot)$ to obtain the hash value$H' = h(m)$.
   V”6. Bob verifies whether H and H' is equal. This signature has the problem that it must have an unconditionally trusted third party (TTP), which owns all the secret information.

References

1. A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, Handbook of applied cryptography, CRC press, 1996.
2. D. R. Stinson, Cryptography Theory and Practice, ser. Discrete Mathematics and its Applications, KH Rosen, Ed. 2000.
3. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, Foundations of Computer Science, In Proceedings 35th annual symposium on foundations of computer science, Ieee, 1994.
4. M. Darbandi, Proposing New Intelligent System for Suggesting Better Service Providers in Cloud Computing based on Kalman Filtering, Published by HCTL International Journal of Technology Innovations and Research, (ISSN: 2321-1814), 24.1 (2017): 1-9. DOI: 10.5281/Zenodo.1034475.
5. M. Darbandi, Proposing New Intelligence Algorithm for Suggesting Better Services to Cloud Users based on Kalman Filtering, Published by Journal of Computer Sciences and Applications (ISSN: 2328-7268), 5.1 (2017): 11-16. DOI: 10.12691/JCSA-5-1-2; USA.
6. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM review 41.2 (1999): 303-332.
7. L. Lamport, Constructing digital signatures from a one-way function, Vol. 238. Palo Alto: Technical Report CSL-98, SRI International, 1979.
8. D. W. Davies, and W. L. Price, Security for computer networks: an introduction to data security in teleprocessing and electronic funds transfer, Wiley, 1989.
9. C. H. Bennett, and G. Brassard, Quantum cryptography: public key distribution and coin tossing Int, Conf. on Computers, Systems and Signal Processing, Bangalore, India, Dec. 1984.
10. N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Quantum cryptography, Reviews of modern physics 74.1 (2002): 145.

11. W. K. Wootters, and H. Z. Wojciech, A single quantum cannot be cloned, Nature 299.5886 (1982): 802-803.
12. M. Herrero-Collantes, and J. C. Garcia-Escartin, Quantum random number generators, Reviews of Modern Physics 89.1 (2017): 015004.
13. M. S. Kang, C. H. Hong, J. Heo, J. I. Lim, and H. J. Yang, Comment on Quantum signature scheme with weak arbitrator, International Journal of Theoretical Physics 53.6 (2014): 1862-1866.
14. M. S. Kang, C. H. Hong, J. Heo, J. I. Lim, and H. J. Yang, Quantum signature scheme using a single qubit rotation operator, International Journal of Theoretical Physics 54.2 (2015): 614-629.
15. C. S. Yoon, M. S. Kang, J. I. Lim, and H. J. Yang, Quantum signature scheme based on a quantum search algorithm, Physica Scripta 90.1 (2014): 015103.
16. S. Haghgoo, M. Hajiali, A. Khabir, Prediction and Estimation of Next Demands of Cloud Users based on their Comments in CRM and Previous usages, International IEEE Conference on Communication, Computing & Internet of Things, Feb. 2018, Chennai. DOI: 10.1109/IC3IoT.2018.8668
17. H. Lee, C. Hong, H. Kim, J. Lim, and H. J. Yang, Arbitrated quantum signature scheme with message recovery, Physics Letters A 321.5-6 (2004): 295-300.
18. X. Zou, and D. Qiu, Security analysis and improvements of arbitrated quantum signature schemes, Physical Review A 82.4 (2010): 042325.
19. A. E. Bashirov, and S. Norozpour, On an alternative view to complex calculus, Mathematical Methods in the Applied Sciences, 41.17 (2018) 7313-7324.
20. P. Bolourchi, M. Moradi, H. Demirel, and S. Uysal, Feature fusion for classification enhancement of ground vehicle SAR images, In 2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim) 2017, pp. 111-115, IEEE.
21. S. Norozpour, Existence and uniqueness results for Multiplicative Fractional differential equation with three point integral boundary value problem, In THE ABSTRACT BOOK, p. 28. 2018.
22. A. E. Filippov, and S. N. Gorb, Methods of the pattern formation in numerical modeling of biological problems, Facta Universitatis, Series: Mechanical Engineering, 17.2 (2019): 217-242.
23. A. I. Dmitriev, A. Y. Nikonov, W. Österle, and B. C. Jim, VERIFICATION OF RABINOWICZ'CRITERION BY DIRECT MOLECULAR DYNAMICS MODELING, Facta Universitatis, Series: Mechanical Engineering, 17.2 (2019): 207-215.
24. M. Moradi, H. Demirel, and P. Bolourchi, Alzheimer's Disease Detection by Utilizing Key Slice Selection in 3D MRI Images, In 2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim) 2018, pp. 96-101, IEEE.
25. P. Bolourchi, M. Moradi, H. Demirel, and S. Uysal, Random forest feature selection for SAR-ATR, In 2018 UKSim-AMSS 20th International Conference on Computer Modelling and Simulation (UKSim), 2018, pp. 90-95, IEEE.
26. J. Benad, Efficient calculation of the BEM integrals on arbitrary shapes with the FFT, Facta Universitatis, Series: Mechanical Engineering, 16.3 (2018): 405-417.
27. A. E. Bashirov, and S. Norozpour, Riemann surface of complex logarithm and multiplicative calculus, arXiv preprint arXiv:1610.00133 (2016).