Key Generation and Identity Verification using Quantum Teleportation

Xiangyu Gao
New York University
xg673@nyu.edu

Eddy Z. Zhang
Rutgers University
eddy.zhengzhang@gmail.com

ABSTRACT

The advent of quantum computers introduces new challenges to the fields of encryption and authentication. Traditional cryptographic systems, such as RSA and ECC, rely on mathematical problems that are computationally hard for classical computers to solve. However, quantum computers have the potential to render these algorithms ineffective by leveraging Shor's algorithm, which can efficiently factor large numbers and solve the discrete logarithm problem. This capability threatens the security of many widely deployed encryption and authentication schemes. There have been a number of post-quantum cryptography methods proposed in the past. However, they are either relatively difficult to implement in current small-scale quantum computers or they do not take noise and error mitigation into account. In this paper, we propose a new framework of post-quantum key generation and identity verification mechanisms which are relatively easy to implement and adapt. Our approaches take the noisy nature of current quantum devices into consideration. As a result, our proposed approach can resist attacks from malicious quantum computers and the successful attack rate could quickly decay to zero as the number of qubits increases.

CCS CONCEPTS

• **Security and privacy** → **Social network security and privacy**;

KEYWORDS

Quantum computing; quantum teleportation; key generation; identity verification; network security;

ACM Reference Format:
Xiangyu Gao and Eddy Z. Zhang. 2023. Key Generation and Identity Verification using Quantum Teleportation. In *1st Workshop on Quantum Networks and Distributed Quantum Computing (QuNet) (QuNet '23), September 10–14, 2023, New York, NY, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610251.3610559

1 INTRODUCTION

Identity verification involves the authentication and validation of the identity of communicating entities in a network. In a secure networking environment, it is crucial to ensure that the parties involved in communication are who they claim to be. Identity verification mechanisms employ various techniques, such as digital certificates, public key infrastructure (PKI), and digital signatures. These mechanisms use cryptographic algorithms to verify the authenticity of an entity's identity and validate their digital credentials. By verifying identities, network systems can establish trust, prevent unauthorized access, and protect against impersonation or data manipulation attempts.

Rivest, Shamir, Adleman (RSA) [11] and Elliptic Curve Cryptography (ECC) [5] are the algorithms widely used asymmetric encryption algorithms in PKI. They use a pair of mathematically related keys: a public key for encryption and a private key for decryption. However, in the age of quantum computing, these authentication algorithms are poised to face significant challenges. Quantum computers possess immense computational power and the ability to perform calculations exponentially faster than classical computers. For instance, by leveraging Shor's algorithm [12], quantum computers could potentially break RSA by quickly factoring large numbers, rendering one of the most widely used encryption algorithms vulnerable. Shor's algorithm can break a 500-digit RSA code in 2 seconds with a 2.2 GHz quantum processor, while it would take 10^{12} years to do the same with a 2.2 GHz classical processor[1].

As a result, we need to develop post-quantum key generation and identify verification techniques in order to ensure robust security in the face of quantum computing advancements. In particular, we need to take into account the noisy nature of the near-term intermediate scale quantum (NISQ) devices, which prior post-quantum cryptography approaches may not have systematically taken into consideration. In this paper, we propose a simple and effective key generation and identify verification method through teleportation. Our method has the benefits of easy implementation and relatively good performance in terms of the verification accuracy. Our main contributions are as follows:

(1) We leverage the reversibility property in quantum gate operation to do key generation and verification.

(2) We develop a multi-qubit key generation and verification procedure to improve the security level compared to the single-qubit scenario.

(3) We provide simulation results to show that our proposal can achieve good security performance with the potential to use fewer number of bits than other verification methods.

We have open sourced our code along with instructions to replicate this paper's results at https://github.com/XiangyuG/ID-verification. This work does not raise any ethical issues.

[1]Source: Institute for Quantum Computing,"John Preskill - Introduction to Quantum Information (part 1) - CSSQI." 2012

QuNet '23, September 10–14, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0306-5/23/09…$15.00
https://doi.org/10.1145/3610251.3610559

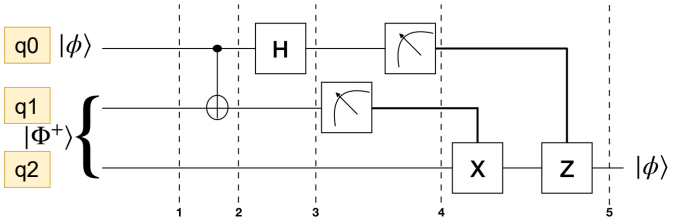

Figure 1: Quantum teleportation between two nodes. Node A owns 2 qubits (q0 and q1) while Node B has 1 qubit (q2). Node A will deliver its qubit q0 to node B through teleportation.

2 BACKGROUND

In this section, we would introduce part of the important features within the quantum computing. More detailed concepts (e.g., qubit, gates, measurement) can be referred from [6].

2.1 EPR Pair

EPR pair or bell's state [6] is used to entangle 2 qubits with the format to be $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. According to such format, we can find that the state of each qubit would either be $|0\rangle$ or $|1\rangle$ with equal probability before measurement. After selecting any qubit and measuring its state $|s\rangle$, the other qubit will be certain to collapse to the same state $|s\rangle$. Therefore, we could say that there is a hidden correlation between these 2 qubits. EPR pair can be useful in quantum communication between different nodes. Specifically, we could distribute one EPR pair between two nodes beforehand and then it is possible for us to transmit one qubit's quantum state to the other qubit through a series of quantum gates through quantum teleportation.

2.2 No Cloning Theorem

In quantum computing field, it is impossible to clone one random qubit unless we know the generation mechanism of such a qubit. Even if there is a communication channel between 2 nodes, it is usually hard for one node to pass the parameters (e.g., α and β) of its qubit $\alpha|0\rangle + \beta|1\rangle$ to the other one because these parameters could be complex or irrational numbers and we need infinite number of bits to precisely represent their values. More detailed proof is shown in [2].

2.3 Quantum Teleportation

When sending qubit information between two geographically separate nodes, we need quantum teleportation to finish such transmission. Figure 1 illustrate the general process.

In the beginning, one EPR pair $|\Phi^+\rangle$ is generated and shared by node A and node B. After generating one qubit $|\phi\rangle$, which should be passed to node B, node A will first of all use the CNOT gate to change the state of one qubit within the shared EPR pair. Afterwards, node A will measure that qubit and the result will determine the operation implemented on node B's side (CX gate). Concurrently, node A will update its own qubit through one Hadamard gate and measure its output, which will decide whether or not node B should implement a Z gate over its qubit.

The whole process can achieve the goal that node A successfully passes its generated qubit to node B. Note that based on the no-cloning theorem, if node B receives the qubit, node A will no longer maintain it any more. If node A wants to transfer multiple qubits to node B, multiple shared EPR pairs are required.

2.4 Reversability of Quantum Gates

Quantum computation consists of a series of operations mathematically represented as complex square unitary matrix (U). By definition, all of the complex square unitary matrices are invertible with the inverse to be their conjugate transposes. Therefore, if a particular qubit has experienced a series of gates operations (U_1, U_2,...,U_n), we could reverse those operations by developing another series of conjugate transpose gates (U_n^H, ..., U_2^H, U_1^H) and implement them in the reverse order.

In fact, if we only focus on the quantum gates over the real number space, inversing them would be even simpler. For instance, typical quantum operation matrices include single-qubit gates (e.g., H, X, Z) and two-qubit gates (e.g., Control-NOT gates), all of which are not only inversible but also Hermitian. Therefore, if we implement one single-qubit gate twice, the qubit will return to its previous state since the extra single-qubit gate will offset the previous one.

Reversability is an important feature used by some researches. For instance, Qraft [8] leverages this feature to transform one generated qubit back to its original state for error mitigation. In our paper, we want to use this feature to check whether the given qubit is the same as the previously generated qubit or not.

2.5 Noisy Nature of Quantum Computers

One obstacle of quantum computation originates from the fact that qubits are susceptible to errors, including coherence errors, gate errors and preparation and measurement errors. The noise comes from various sources, like disturbances in Earth's magnetic field, local radiation from Wi-Fi or mobile phones, cosmic rays, and even the influence of neighboring qubits. Quantum systems are highly sensitive to factors like temperature, electromagnetic radiation, and vibrations, which disturb the fragile quantum states of qubits. Imperfections in control operations and manufacturing processes, as well as interference from stray electromagnetic fields and crosstalk, further contribute to the noise.

The existence of these errors can corrupt qubits' states, producing incorrect outcomes during program execution. Taking the appearance of unreliable results from quantum computation as a given fact and developing good ways to deal with this issue is a critical area of research in quantum computing.

2.6 Identity Verification

Identity verification is used to prove a participant is who they claim to be so that we can make sure that we are communicating with a trustworthy user. One traditional method is password-based verification where anyone is required to provide their password before entering into the system. Then only people with valid account name and password combination can have access to the system. However, this is vulnerable when attackers decode one user's password and then become in the disguise of authorized users.

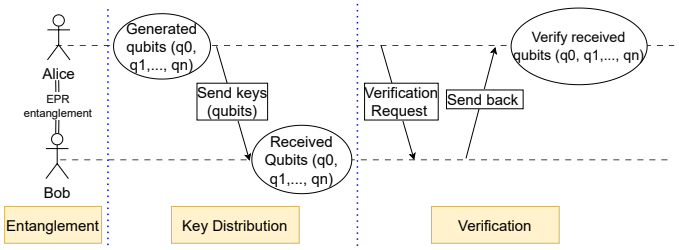

Figure 2: Workflow to distribute key and verify identity

Multi-factor authentication(MFA) [7] is another emerging way for authentication, working by requesting multiple forms of ID from the user at the time of account login. This increases the security level to prevent malicious users from entering into the system because they have to pass multiple authentication processes before logging into the system. Public key infrastructure (PKI) [3] is even more popularly used in settings such as cryptocurrency trading and remote access to a computer using Secure Shell Protocol (SSH). PKI uses asymmetric encryption methods to ensure that messages remain private and only visible to authenticated receivers. Specifically, a user generates a pair of keys (private key and public key) locally and send the public key to others through public channel. When other users deliver a message to this user, they would encode such message using public key during transmission and only the user with corresponding private key can decode this message. The security guarantee of PKI is based on the trap door functions - e.g. functions that are easy to compute in one direction, but very hard to reverse. In other words, it might take forever for classical computers to guess the private key from the public key through testing all possible candidates. Therefore, as long as the private key is not released, PKI can guarantee that the message is visible to authenticated users.

3 MOTIVATION AND OUR APPROACH

This section provides an overview of the problem, which is followed by the general workflow of our proposal presenting in Figure 2. We also give more detailed explanation of each steps within the workflow in this section.

3.1 Motivation

How to efficiently generate and safely distribute one key to the trusted target? There are several dimensions we want to consider for the key generation and distribution process. First of all, we want such generation process to be simple so that not many resources are required for key generation and storage; in addition, we want to securely transmit the key to our trusty target.
How to verify one target's identity? Before sending the message to the other node, it is important to verify its identity for security reasons. Only after making sure that the counterparty is a trustworthy identity, we could continue sending subsequent data to it. Therefore, we want a fast and precise way to finish the verification process.

3.2 Approach 1: Key Distribution through Teleportation

In our setting, we could regard the generated qubits as the keys for later verification. Key distribution phase is visualized in Figure 2. In the initial stage, a sender will generate a qubit $\alpha|0\rangle + \beta|1\rangle$ through some defined procedure and then distribute this qubit to a receiver through quantum teleportation. The receiver should store this qubit and later send it back for identity verification purpose.

There are some benefits for this type of distribution. First of all, the sender and receiver have already shared one EPR that is necessary for quantum teleportation beforehand in the entanglement stage, thus the sender could only distribute its key to this particular receiver without worrying about giving the qubit to incorrect nodes. Additionally, nobody except the sender could understand the details within such a qubit; the no-cloning theorem in quantum computing makes sure that other potentially malicious nodes cannot copy this qubit. The receiver also does not have motivation to share this qubit with others because as long as it distributes this qubit, the receiver will lose it, making it fail to pass the identity verification in the future.

3.3 Approach 2: Verification through Reverse Operation

This is more related to the verification phase in Figure 2. During the identity verification phase, the sender would request the previous receiver to send back the qubit that was distributed before. Suppose the qubit we receive is in the format $\alpha|0\rangle + \beta|1\rangle$. Even if it is infeasible to check directly the value of α and β, the sender can still verify it by leveraging the reversibility feature within quantum computing. Here is the detailed verification procedure.

Since it is the sender that produces this qubit, the generation process is known to the sender. According to the reversibility feature within the quantum computation scope, the sender could reverse its computation and implement it on the received qubits. In theory, after going through all the reversed computation gates, it should reach one state (either $|0\rangle$ or $|1\rangle$) for sure. At that point, the sender is able to do the final verification through measuring the state of qubit.

3.4 Approach 3: Error Mitigation

Current Noisy Intermediate-Scale Quantum (NISC) [9] machines suffer from a limitation that could output erroneous output. The error may originate from quantum gate operation error, qubit coherence errors and measurement errors. Efforts [4] [13] have been made to mitigate the error occurrence to obtain the correct output. Such error appearance would make a negative impact on the final measurement result. On the one hand, it is possible for the verifier to falsely reject one qubit; on the other hand, it is also possible for the verifier to falsely accept one randomly generated qubit.

In our setting, we also want to assume that the output might be unreliable and thus develop better mechanism to avoid such negative outcome. To be specific, in the beginning, the sender would generate n different qubits, where n is great than 1, and teleport to the receiver. During the verification phase, the receiver will teleport back to the sender and then the sender could verify all these n qubits independently. When the total number of qubits that

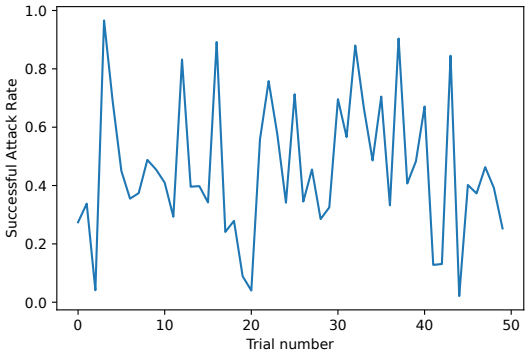

Figure 3: Successful attack rate for randomly generate quantum circuits

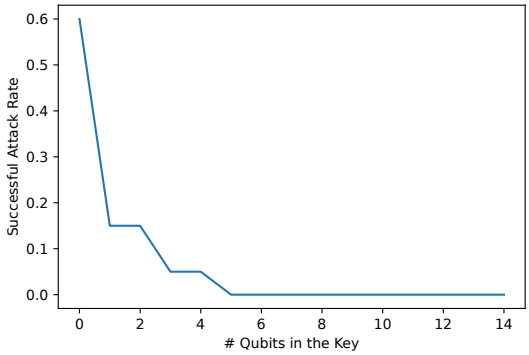

Figure 4: Successful attack rate for multiple qubits

pass the verification exceeds a predefined threshold, it could decide that the counterparty is trustworthy. Developing multiple qubits and verifying all of them could reduce the probability of wrong decision caused by the noise within quantum machine.

4 EVALUATION

In this section, we show the simulation results of our proposed approach. We first test the security level of our proposal against the random attack; then, we increase the security level to send multiple independently generated qubits and show the improvement compared with the situation where we only send 1 qubit. Finally, we extend to the simulated noisy environment and measure such proposal under a more realistic scenario.

4.1 Experiment Setup

Our experiment consists of two phases: key generation and identity verification. During the key generation phase, we would randomly select a series of single-bit gates to generate 1 qubit; then the verification process would be composed of another series of gate operations by reversing the gates' operation order in the key generation phase. Our metrics is the rate of false verification, including reject true key and accept false key.

In our existing setup, we only choose from 4 single-qubit gates (X, Y, Z, H) for simplicity, but it is easy to include more complex gates into such set. All our code is based on the open-source Qiskit [10] package.

4.2 Security Level Analysis

4.2.1 Measurement results against randomly generated qubits. We want to show how effective it would be to defend potential attackers. We use successful attack rate as the metric and equation (1) shows the formula to get the successful attack rate.

$$\text{Successful Attack Rate} = \frac{\#\text{ Measurement result is } |0\rangle}{\#\text{ trials}} \quad (1)$$

To be specific, suppose the initial state of the qubit is $|0\rangle$, therefore the reversibility within quantum computation gates means that when we verify the received key, it is supposed to go back to $|0\rangle$ and the measurement result should be $|0\rangle$ with 100%. As for verifiers, if the measurement result is $|0\rangle$, it means the received qubit comes from the trustworthy counterparty.

Figure 3 shows the simulation results of successful attack rate. We randomly generate one encoding and decoding situation which consists of a series of quantum gates. As for each trial, our simulator randomly outputs 50 qubits per trial and the verifier simulates the measurement process for 1000 times for each qubit. In our setting, we generate 50 random qubits to test the probability to pass the verification using randomly generated qubits. We also consider measuring 1000 times because the measurement results for one qubit might be different. For instance, the qubit $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ has equal probability to collapse to state $|0\rangle$ or state $|1\rangle$ and if we measure it twice, with the probability 50% we may get different results. Measuring 1000 times and taking the average over all measurement results can take into consideration the parameters of each state.

If the measurement result is $|0\rangle$, it means we accept this randomly generated qubit (because the initial state to generate the key is $|0\rangle$). Each time we make a trial, we record the successful attack rate of this particular trial to find that people have approximately 50% possibility to successfully attack the system. It means this is not a good way to transmit only **ONE** qubit as the key.

4.2.2 Extension to multiple qubits. Given the fact that the generation and verification process are independent across qubits, a natural fix to the issue mentioned above is to increase the total number of qubits inside one key. The attack will become a failure unless all qubits within the key pass the verification. Figure 4 plots the relationship between the successful attack rate and the number of qubits within the key. It shows that the successful attack rate decreases exponentially as the number of qubits within the generated key increases. For example, if there are 6 qubits inside one key, the successful attack rate would decrease downto 0%. Note that the trend might not be monotonically decreasing when the number of qubits is small because of nondeterminism from measurement and random qubit generation. However, in general the trend is decreasing exponentially and stay at 0 when the number of qubits is large enough.

In comparison, the minimum size for clear RSA keys and secure RSA keys on the public key data set (PKDS) is 512 bits [1]. This

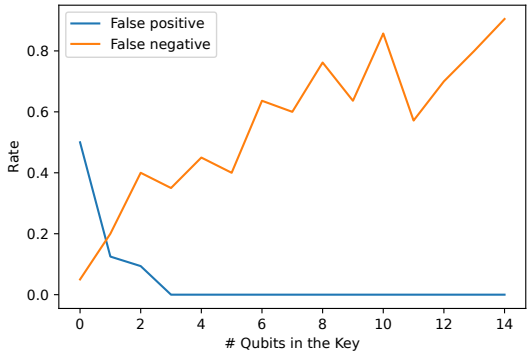

Figure 5: Verification results under noisy setting

gives our proposal the potential to achieve the same security level with fewer number of bits to generate the key.

Therefore, we can easily fix the problem that the probability to erroneously regard a randomly qubit as the correct key is too high. However, this fix is at the cost of more generated EPR pairs since we require one EPR pair to teleport each qubit. Generating and distributing a large number of EPR pairs in the beginning might be required to avoid the shortage of EPR pairs later.

4.3 Noisy Environment

In addition to the ideal situation, we simulate the noisy situation where different types of errors might happen. This is due to the fact that quantum machines suffer from noise, making its result inconsistent with the correct output. In our simulator, we use parameters to set different types of error such as incorrect measurement results and incorrect gate operation. The criteria for the verification to pass keeps unchanged: the user will be considered as a trustworthy counterparty only if all of its qubits pass the verification.

The Figure 5 presents the results, where *False positive* means we accept a malicious user and *False negative* means we reject a trusty user. It shows that even under the noisy setting, the false positive rate can quickly decay to 0. However, the false negative rate also increases mainly due to the fact that our requirement is too strict. The existence of noise would lead to the possibility that some trusty qubits might fail to pass the verification. As the number of qubits increases, the false negative rate should increase as well. A compromised solution could be as long as the number of the received qubits passing the verification exceeds **one threshold**, we can consider the user as the trustworthy one.

Determining the optimal threshold of accepting or rejecting the received qubits is orthogonal to our proposal and will be left for future work.

5 LIMITATIONS AND FUTURE WORK

We now briefly discuss main limitations of our proposal along with avenues for future work. First, our proposal makes an assumption that EPR pairs can be shared safely between the host and the users; however, this is not straightforward since we cannot guarantee that the sender shares the EPR pair itself with the correct receiver in the first place. In other words, if an adversary tricks the sender to share the EPR pair, it could as well get the generated key later. Second, even if we can guarantee the safe sharing of EPR pair, we do not consider the cost of sharing and using EPR pair. During the quantum teleportation process, one EPR pair is used to transmit one qubit. The generated key could be used at most once, making the sender and the receiver require a large number of EPR pairs in the beginning. Third, the current experiment results are based on simulated situations and we should take into account more realistic scenarios. All these will be left for future work to keep improving our proposed key generation and identity verification process.

6 CONCLUSION

We propose a new method to do key generation and identity verification by leveraging several features within quantum computing. The simulated experiment result shows that the security level increase exponentially as the number of qubits within the verification key increases. This mechanism is easy to implement and has a potential to achieve good security level using fewer bits than other verification ways such as RSA. We hope this proposal can prompt further research on using quantum computing to improve the security.

ACKNOWLEDGEMENTS

We are grateful to the anonymous reviewers for their valuable comments on previous drafts of this paper. We also thank Yuwei Jin and Tiancheng Hou for their insightful discussions over the pros and cons of our proposal, together with the experiment design. This work was supported in part by grants from Rutgers Research Council, NSF-CCF1628401 and NSF-FET-2129872.

REFERENCES

[1] 2021. Size considerations for public and private keys. https://www.ibm.com/docs/en/zos/2.3.0?topic=certificates-size-considerations-public-private-keys. (2021).
[2] 2023. No-cloning theorem. https://en.wikipedia.org/wiki/No-cloning_theorem. (2023).
[3] Johannes Buchmann, Evangelos Karatsiolis, Alexander Wiesmaier, and Evangelos Karatsiolis. 2013. *Introduction to public key infrastructures*. Vol. 36. Springer.
[4] Poulami Das, Swamit Tannu, and Moinuddin Qureshi. 2021. JigSaw: Boosting Fidelity of NISQ Programs via Measurement Subsetting. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*. Association for Computing Machinery, New York, NY, USA, 937–949.
[5] Neal Koblitz. 1987. Elliptic Curve Cryptosystems. *Math. Comp.* 48, 177 (Jan. 1987), 203–209.
[6] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. USA.
[7] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen, and Yevgeni Koucheryavy. 2018. Multi-factor authentication: A survey. *Cryptography* 2, 1 (2018), 1.
[8] Tirthak Patel and Devesh Tiwari. 2021. Qraft: Reverse Your Quantum Circuit and Know the Correct Program Output. In *ASPLOS*. Association for Computing Machinery, New York, NY, USA, 443–455.
[9] John Preskill. 2018. Quantum computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.
[10] Qiskit contributors. 2023. Qiskit: An Open-source Framework for Quantum Computing. (2023). https://doi.org/10.5281/zenodo.2573505
[11] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* 21, 2 (feb 1978), 120–126.
[12] P.W. Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 124–134.
[13] Swamit S. Tannu and Moinuddin Qureshi. 2019. Ensemble of Diverse Mappings: Improving Reliability of Quantum Computers by Orchestrating Dissimilar Mistakes. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. Association for Computing Machinery, New York, NY, USA, 253–265.