Authentication Scheme using Unique Identification method with Homomorphic Encryption in Mobile Cloud Computing

Lim Tsu Chean, Vasaki Ponnusamy
Faculty of Information and Communication Technology,
University Tunku Abdul Rahman (UTAR), Kampar,
Malaysia
Email: vasaki@utar.edu.my
Email: lim.agent@gmail.com

Suliman Mohamed Fati
Faculty of Information Technology-Math & Science, Inti
Inti International University, Persiaran Perdana BBN, Putra
Nilai, 71800 Nilai, Negeri Sembilan
Email: *smfati@yahoo.com*

Abstract—**The purpose of this project is to design and implement a framework that employs unique authentication method that relies on third party identification which is required to accurately identify and authenticate legitimate user in order to reduce the risk of disclosing confidential data to unauthorized party in Mobile Cloud Computing, MCC. Cloud computing is a service provided for user to store and process their data online instead of local device. Mobile Cloud computing which is an extension of the cloud computing service allows user to remotely access their data and services that are being uploaded to the cloud beforehand in any place and any time as long as they have internet connection in their personal mobile device. Throughout the period of the mobile cloud computing era, the security has always been a concern and post as a formidable challenge even for experienced security specialist. Moreover, encryption method such as RSA and Diffie Hellman cryptosystem proven to be useless against the attacks launched by quantum computer. Hence, Homomorphic Signature scheme is introduced along with the Identity Management (IDM) server into the mobile cloud computing in order to address this issue by applying implicit authentication method to differentiate between the genuine and non-genuine user which allows system to authenticate the clients accurately. The details of the framework will be further explained later in this paper, where the user will be authenticated with IDM as medium and no password is used throughout the authentication process, allowing the client to be safely authenticated at the end of the process.**

*Keywords— **Mobile Cloud Computing security, Homomorphic encryption/authentication method, Homomorphic signature, Identity Management, Authentication Framework, Post Quantum Cryptography***

I. Introduction

Mobile Cloud Computing, for short MCC, is a service model that integrates both mobile and cloud computing into one service. Mobile computing provides wireless communication services which can perform transmission of information to other devices through the format of voice, data or in form of visual presentation such as image and video. On the other hand, cloud computing is a method of utilizing the function remote server which is not locally hosted. Through the internet connection, the user can remotely access the server anytime and anywhere [1] to perform store, manage, backup and process the information outside of the device once the specific user is authenticated. Example of the few famous cloud providers are GoogleDrive, Amazon S3 and DropBox. This service provides a lot of convenience for the users as they do not need to consider towards monetary matter such as being charged with extra cost like investing money on setting up infrastructure, hardware and software inspection, maintenance and expansion of their own storage server as most of the work are managed by the remote cloud service provider, where clients only need to concern about how much the cost of the resources which is charged based on their usage. What is MCC? “Mobile cloud computing is an emerging cloud service model following the trend to extend the cloud to the edge of networks.” [2]

MCC is flexible as it is compatible with various mobile operating system platform such as Android, IOS and Windows. By integrating mobile device into cloud computing, MCC enables mobile users to access the cloud as well, where users can monitor their smartphone by utilizing the cloud’s resource to process some the mobile’s computing task. Since the amount of mobile phone owner has been increasing significantly compared to computer user. By implementing the cloud computing technique into mobile device, users are now able to access and receive services from the cloud remotely through their individual portable device. Even though this technology benefits the user by allowing them to conveniently access to the cloud services through their mobile device remotely, it is inevitable that the MCC cannot escape cybersecurity challenges especially in the user authentication aspect.

II. Motivation behind the Proposed Scheme

While the total amount of user for the mobile device increase exponentially, this made security for user authentication to fall into the list of concerns for MCC. Exploitations can be made by the attackers towards the existing mobile security algorithm as long as user’s device is connected to the internet which will allow the attackers to carefully choose their victim and perform data theft. This issue poses as

978-1-5386-3527-8/18/$31.00 ©2018 IEEE

a huge threat and challenges towards the security of user privacy as it allows their log in credentials to be easily exposed and acquired, thus carries a higher risk of getting their access stolen. This will discourage user from using the MCC service due to lack incompetency in their cloud service provider's authentication system in securing the privacy of their data.

By using packet sniffing tools (eg. Wireshark), the attacker is able to track/capture the transmitted packets through wireless connection easily. The content of the sniffed packet might contain user's Sensitive Personal Information (SPI) parameter in it, or commonly known as User ID and password, leading to a scenario where user becomes the victim of unauthorized access and data theft. So when every time user attempts to log in and access the cloud service, there always exist risk of getting their personal log in credentials stolen virtually. By now it should be clear that the security concerns that should be focused for MCC is the authentication process.

The most common authentication process nowadays is user name and password authentication, a process where server will request the client for name and password. Once inputted, the server will query into database to seek out the matching user name and then compare the password. Once all of the required input proved to be matching, it will authenticate the client. This is a simple straightforward authentication process but the SPI inputted by user can be easily stolen in any 3 part of the authentication process. During the first part of the process where user input their SPI into the device, their log in credentials can be easily stolen by logging tools such as key loggers which it records all the key stroke entered. During data transmission between device, which is the second part of the authentication process, the information sent can be easily stolen through utilization of eavesdropping tools for Man in the middle attack such as Ether cap from Kali Linux operating system. Lastly, the server that receives the user's SPI can be malicious as well. The server can be spoofed where attackers can create a fake server with same domain name and then launch Denial of service attack to disable the genuine server, allowing them to impersonate and get their hand on user's SPI. Or perhaps the genuine server itself can be un-trusted as well by providing services to the clients and at the same time collecting their SPI without the knowledge of the client.

Password encryption has been introduced to combat this situation where user's SPI will be encrypted from plaintext format into cipher text format, preventing its original data from falling into unauthorized party. A public key is assigned to the sender in order to allow them to encrypt their message and send it to the receiver. The receiver decrypts the message with a private key instead which is generated along with the public as a pair. A public key can only perform encryption while a private can perform decryption on whatever information that is encrypted by the same pair of public key. This encryption scheme is known as Public key cryptography/ Asymmetrical cryptography, which means different key is used for decryption and encryption process. As for the encryption algorithm, RSA RSA is applied for the public/private key encryption by utilizing the operation on prime number with modulus. Asymmetrical key allows the encrypted message to only be viewed by the receiver that holds the valid private key while allowing the receiver to distribute as many public key as they like towards the sender. This will keep the communication between the receiver and each sender secured from eavesdropping, which is proven to be a far superior compare to Symmetrical key, where the messages can be encrypted and decrypted with the same public key between the sender and receiver.

However, public key schemes like previously mentioned RSA is vulnerable to the attack launched by quantum computer through Shor's algorithm. Shor's algorithm is applicable for quantum computing, where if it is implemented, multiple quantum computer can be used to take down the RSA algorithm, breaking its factorization effectively. Knowing that RSA public key cryptography can be easily defeated is quite devastating. Hence, an effective authentication must be proposed in order to combat the mentioned scenario and allow user to get authenticated without risking their SPI being exposed.

In order to strengthen the legitimacy of authentication process over the cloud, third party Identity Management (IDM) [3] is introduced with its function as to identify a particular person on the bases of claimed values such as name, email, phone number and address. [4] The IDM can be used as a "middle man" that can represent and get identified on behalf of their client towards the cloud server while eliminating the use of password as authentication medium.

III. RELATED WORKS

Apart from the proposed project in this paper that is focused on MCC's authentication that eliminates the use of password, there are several existing solutions that are almost identical in the authentication method that is proposed in this paper. Even though the approaches are different, they still share the fundamental concept of identification which is to deploy a framework that can authenticate and identify user while reducing the need for them to disclose their SPI. The elimination of password during authentication process greatly reduce the chance of allowing user's credential to be stolen as there is insufficient information regarding user's credential to be sniffed out by the attacker which can be used to get authenticated. Below are the related frameworks that can authenticate the client with minimal to zero use of password:

A. PRIME

Known as Privacy and Identity Management for Europe, is a middleware that manages and authenticates user data by preserving the privacy using anonymous credentials. Identity mixer protocol is provided to enable the users to reveal any of the credential's attribute that is obtained from third party identity provider (IdP) without disclosing any information. The problem in PRIME is that to implement it, both client agents and cloud service provider (CSP) are required, where the implementation cost is expensive and would require large sum of budget with precise accounting to ensure the availability of continuous funding for the service provider.

B. *Single Sign-On*

This is a session and user authentication service [5] that allows a single user to access multiple application in their device using only a set of login [6]. This service eliminates the need of further prompting for user to input their SPI in every different application that is inside of their mobile device as long as it is in the same session. On the other hand, there is a risk that allows the user's session to be hijacked by third-party attackers through session injection where the injected malicious codes will overwrite the session, allowing the hijacker to take control over the session, hence, obtaining their victim's authenticated identity.

C. *KodeKey*

Kodekey eliminates the use of password and authenticates the client through a mobile application that identifies user through biometric scanners. It also associates user's biometric information with their Phone number and PIN and it can be integrated with web-based API. If the system detects that there is log in activity that uses phone number and PIN instead of biometric scanner, it will prompt user in their phone in order to verify the legitimacy of the log in activity through requesting the user to perform biometric scan.

D. *LaunchKey*

LaunchKey is a mobile application that helps user to manage all of the authentication process in their devices. It is a multifactor authentication system where user can authenticate themselves with various methods such as biometric, retina, Bluetooth proximity etc… Apart from that, LaunchKey engine does not contain any user's sensitive authentication information and it is only contained within user's own device and the SPI will not be sent out even during authentication process, which ensures the confidentiality of the user's SPI to be secured.

E. *Homomorphic Authentication Encryption (HAE)*

HAE ensures that both privacy and authenticity of the data are secured simultaneously. It is a symmetric-key cryptography that allows its functions to be evaluated by public through corresponding ciphertexts. [10] Basically, HAE is a homomorphic version of Indistinguishability under chosen plaintext attack (IDN-CPA), where it is a security guessing game consist of 2 party which are challenger and the adversary. The adversary first generates 2 different messages with similar length for the challenger to encrypt either one of them. The challenger may randomly choose any one of the message to be encrypted. After that the adversary will attempt to guess which of the message is encrypted by the challenger. The identical message length is used in order to prevent the adversary to compare the message through their length difference and easily win the game.

IV. Proposed work

Based on the related works that are previously mentioned, the framework proposed in this paper assigns a "middleman" to authenticate and represent their client to request for cloud service without the need of using client's own SPI and thus, reducing the risk of identity theft due as the identity of the client is exchanged only between the IDM server and the client's device.

Initially, the client must register with the IDM server. The IDM will synchronize with the client device and generate a type credential known as Identity which consist of a set unique values that can be used to identify that specific client device. The format of Identity is generated through combination device's signature value, user's credential and registration value assigned by the IDM. Once they are synchronized, the IDM itself will store the generated identity in its own database. The IDM will then generate a dummy name which is mapped to the newly created Identity. The IDM will then represent the client to register itself towards the Cloud Service Provider (CSP) using the dummy name.

During authentication process, the client will first send an authentication request to the IDM. The IDM will then verify the request and then generate an instruction for identity splitting process along with a cloud address and then send it back to the client. The IDM itself will split the respective client's identity and send it to cloud, instructing the cloud to homomorphically encrypt the identity that will be sent by the client towards it with matching identity later.

Meanwhile, the client will split its own identity based on the instruction received from the IDM. Once it is done, the client will encrypt the partial identity and send it to the designated cloud for homomorphic encryption process. Cloud matches the partial identity value sent by the client with the one from IDM for verification before initiating the encryption process.

Once the partial identity is encrypted homomorphically, the cloud sends it back to the client and another copy towards the IDM. The client that receives the partial identity will have it combined with another part of the original identity, creating a new identity value before sending it to IDM for final verification.

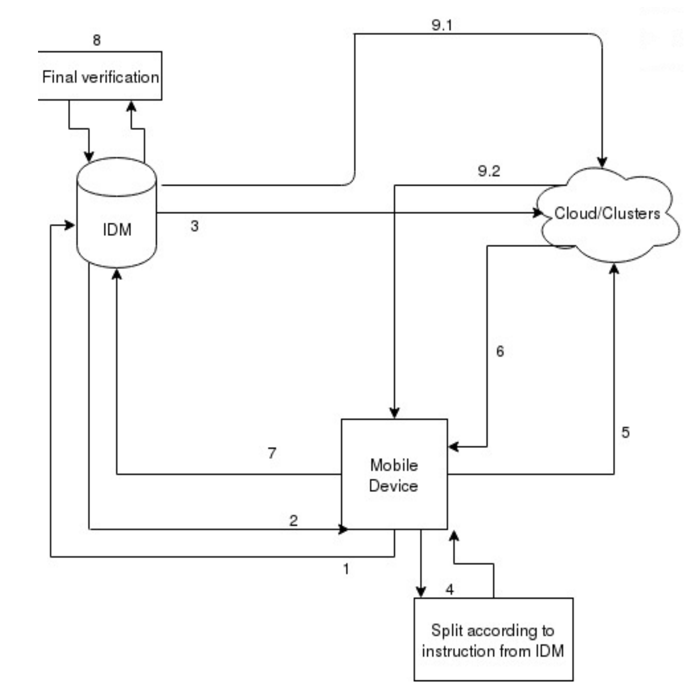

Fig. 1. Homomorphic Identity Management Framework

IDM will also generate the new identity same as what client did for final verification purpose as well. Once the client is verified, the IDM will then represent the client to perform service request towards the cloud using the mapped dummy name along with a designated client's address. Once the cloud verified the request, the service will be provided towards the address as instructed by the IDM. The client that listens to the designated address will then receive the service from cloud, concluding the whole authentication process. The flow diagram of the authentication process:

Authentication process:

1. Mobile device sends authentication request along with its identity (each device have their unique identity) and address to IDM.
2. IDM will then verify the identity of the mobile device and create an instruction that randomly splits out the certain part of the identity which will be sent along with the address of the assigned cloud (for encryption purpose) to the mobile device.
3. The IDM then sends out the same partial value of the identity to the Cloud (cloud won't know what it is), instructing the cloud to only encrypts the matching partial identity that will be sent by the mobile device later.
4. After the mobile device receives the address of the cloud and the split instructions, the mobile device will split out the partial identity according to the instructions received.
5. The partial identity is then being sent to cloud to be encrypted.
6. The cloud will compare the received partial identity value with the ones received from the IDM, once it is matching, it will start the homomorphic encryption process. Once it is completed, the cloud will send the encrypted partial identity back to mobile device and another one to IDM.
7. The encrypted partial identity is then sent back to mobile device to be combined with the other original part of the identity (as header) before being sent to IDM for final verification.
8. IDM will then perform final verification by first comparing the identity header with the one initially received in step 1(IDM knows the specific header for that particular identity as it was the one initially had the identity values split) and then compare the homomorphic side of the identity.
9. Once the final verification is successful, the IDM will send the service request along with mapped dummy name and the mobile device address to the cloud server (can be different cloud server based on user's service request), allowing mobile device to receive service from the cloud.

Simple Experimental Setup

Based on framework shown in Figure 1, A simple setup has been done to simulate the whole process of the framework. This experiment is done in a localhost with a single Window PC and an Android mobile device. The window PC will serve as a back-end server that holds both IDM and Cloud database which can be queried by the android mobile device once connected.

From the Android device's perspective, the device information that is inputted by user is encoded and sent through http POST method from its BackgroundWorker.java class to the IDM's database. From the database perspective, a PHP script will represent the database to receive the data and convert them into queries to be used on database. The script will also manage the result of the queries and perform computations on it before sending the computed results back to the mobile device.

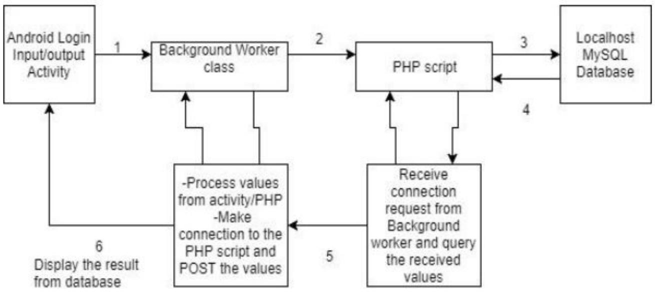
Android Login Input/output Activity
1
Background Worker class
2
PHP script
3
Localhost MySQL Database
4
-Process values from activity/PHP -Make connection to the PHP script and POST the values
Receive connection request from Background worker and query the received values
5
6
Display the result from database

Fig. 2. Data exchange process for the framework's simulation

The IDM database may also interact with the Cloud database by relying on the PHP scripts from both sides. With that, the IDM can register/insert data into cloud's database while instructing the cloud to perform encryption service (non-homomorphic since it is just a simulation on local device) from the partial values sent by android device. This experimental setup will simulate all the data exchanges and also instruction passing process in a local network environment.

A. Quantum Computing

With the current capabilities of the classical computing where a bit from a complete data is made and exist in two states, which is either 1 or 0. However, the bit has limited storage for information and may require excessive amount of resources to process huge chunk of data. In order to fit well into the era of big data, computer with the ability to compute and process complex problem that exceeds the limit of a classical computer is needed. For that, Quantum computer is said to have the power to take on the job and complete it within a short amount of time. Unlike classical computing, quantum computing utilizes quantum bits or known as Qubits [7]. Qubit also has two states but what differentiate it from the usual bit is that they can exist in any superposition of 1 and 0. To put it simply, they can exist as 1 and 0 at the same time. Thanks to the superposition characteristic, the qubits can store more information than just 1 or 0. To further clarify the difference, this can be explained with a sphere that has two poles, each pole represent 1 and 0 bit respectively. With that, a qubit can exist at any point of the sphere freely taking the form of either 1 and 0 bit. This allow the qubit to store tremendous amount of information far more than the classical computer while only consuming minimal amount of energy. This concept enables the creation of computer with processor that is million times faster than the classical computer. However, this is a concern for security as attack from quantum computer is powerful enough to break any existing public key encryption with sheer computing capability. That is why Homomorphic Encryption is proposed in this paper as it is capable of mitigating the attacks launched by quantum computer using complex encryption algorithm [8].

B. Homomorphic Encryption

Perhaps encrypting the plaintext might prove to be too simple and has higher chance of exposing the plaintext's content, risking the confidentiality of user's SPI. Homomorphic encryption allows further encryption computation to be performed on the initially encrypted data, further encrypting the ciphertext it received in order to forge its "hardness". Homomorphic encryption is used on cloud computing platform as heavy amount of resource is needed to perform sophisticated computation on the encrypted data.

A fully Homomorphic encryption that is developed by [9] employing lattice-based cryptography, which is a candidate for post-quantum cryptography that can be used to mitigate attacks launched by quantum computers. Attacks by quantum computer post as a formidable treat towards the modern era security. Public key cryptosystem that are being implemented such as RSA and Diffie Hellman algorithm can be easily taken down through a quantum algorithm that is developed by Peter Shor, which is known as Shor's algorithm. Shor's algorithm can find out the prime factors with the given integer through finding its function's period, allowing it to easily break through RSA as it generates a pair of public and private key for encryption and decryption process respectively. The lattice-based cryptography in fully homomorphic encryption scheme is hard to break due to the nature of the lattice's hardness. Shortest Vector Problem (SVP) is essentially related to the lattice-based computational problem where it will ask for an output of a nonzero lattice vector where its norm is larger than the shortest nonzero lattice to a certain length that is restricted by some approximation factor. The hardness result for the SVP is known to be non-deterministic polynomial-time hardness (NP-hard) or in layman term "at least as hard as hardest problems in nondeterministic polynomial time", which proven to be highly quantum resistant.

The purpose of implementing the Homomorphic encryption in the framework that is proposed in this paper is to utilize the cloud's resource in order to further forge the security during the authentication process. The cloud serves as a platform that helps perform further encryption on the partial identity sent by the client device using homomorphic encryption that even the client's identity is properly secured from eavesdropping. Furthermore, the cloud that received the encrypted value will not know the original value but is able to determine and validate its identity due to the homomorphic nature of the ciphertext.

C. Shor's Algorithm

Instead of launching brute force attack on RSA scheme by trying to all the possible solution at once by comparing every single one of the number to see whether it is smaller than N in order to determine its factor, Shor's algorithm employs number theory through converting the problem of finding the factors of a given number into another problem of finding a different number using the period of a particular periodic function which can be done by[11] Quantum Fourier Transform (QFT). The function is implemented as quantum transform where it will measure the number of repetitiveness of the function as an approximation value of Period.

Example 1 (QFT):

A function f(x)=f(x+5), where it repeats itself every 5 values, thus determine the period=5, where the function would repeat once every 5 values.

Example 2 (explain with RSA's Modulus):

Given a sequence number of 2, 4, 8, 16, 32, 64… (increment of N=N*2) with modulus of 15, where it will give us:

2, 4, 8, 1, 2, 4, 8, 1, 2….

Through the sequence, the period is 4, where it would repeat once it goes through this set of 4 number sequence [2, 4, 8, 1]. With this, it can approximately determine the factor just by finding the period of the RSA modulus through QFT. However, the algorithm would require tremendous amount of time in order for it to factor the prime numbers on a classical computer. This is because the amount of time needed is exponential with the size of input (Let N be the input). For that, quantum computer is a better platform for the Shor's algorithm to run effectively. As compared to the classical computer, the time for quantum computer needed is only polynomial towards in put N, which means less time is required to factorize the prime number. Due to the superposition state of the cubit, it enables the periodic function f(x) to be evaluated simultaneously, thus less time is required for it to break RSA.

V. Conclusion

It is undeniable that the confidentiality of the mobile user's SPI is always at risk when they are used during authentication process as they are connected to the network where they would send their information through internet towards the cloud in order to get verified and authenticated. By doing so, there is a chance that the mobile user will fall victim into man in the middle attack where the packets sent by their device will go through the unauthorized eavesdropper's filter before actually reaching the authenticator. This allows their information to fall into the hands of the attacker, compromising their privacy as well as their identity. Thus granting unauthorized access towards their personal accounts through impersonation by the attackers since the SPI sent out is the only information/prove provided that can be used to identify legitimacy of the user.

So instead of developing a scheme that is solely focused on improving the security of the system through strengthening of the encryption algorithm, a multi-authentication purpose framework should be proposed, developed, tested and implemented on the actual environment [11]. The multi-authentication process that proposed in this paper will remove the use of password or any important credential and instead it uses Identity that is generated together by the IDM and client. This method will allow user to be authenticated in a secure manner without the need of revealing their SPI, providing a better safeguard for the privacy and confidentiality of the user's personal information as well as their identity.

Furthermore, the mobile cloud computing authentication framework also involves a lot of distribution of Identity since the use of SPI has been replaced by it. Although the privacy of the client's SPI is secured, but the risk of getting the identity stolen through information exchange between client and authenticator is not completely eliminated. This is where the failsafe of the framework comes in. Since the identity is generated through the agreement of both IDM and client, each time the Identity that is used for authentication can sequentially or randomly different based on the instruction generated by the IDM. Since the instructions for identity generation is different during every authentication attempt, the attackers cannot re used the values of identity that is previously stolen. As for the method to prevent the values of the identity from being stolen, homomorphic encryption will be performed on the partial identity that is produced and sent by client in the cloud, where further encryption processes will occur upon the encrypted identity before the homomorphically encrypted identity will be sent back for final verification. At the end of the proposed mobile authentication framework, only Identity is used for information exchange throughout the authentication ocess and since the identity's value kept on changing for every authentication attempt and is encrypted along with homomorphic encryption scheme, it will prove to be extremely difficult for the attacker to penetrate this security system. Although the proposed framework looks secured, the attackers will still be able to find ways to exploit the blind spot of the security system [12], so it is inevitable that improvement must be made upon it frequently in order to create and maintain a safe and secured network environment for the user, allowing user to access the online services without worrying of getting their personal information stolen.

References

[1] Q. F. Hassan, "Demystifying Cloud computing". *CrossTalk*(2011), 16-21, 2011.

[2] D. Huang, Z. Zhou, L. Xu, T. Xing, and Y. Zhong,, "Secure data processing framework for mobilecloud computing", in: Proc. IEEE INFOCOM Workshop on Cloud Computing, INFOCOM '11, Shanghai, China, June 2011.

[3] I. Khalil., A. Khreishah, and M. Azeem, "Consolidated Identity Management System for secure mobile". *Computer Networks, I*(65), 99-110, 2014.

[4] S. Smita, M. Deep, "Identity Management issues in Cloud Computing", International Journal of Computer Trends and Technology (IJCTT), volume 9, number 8, Mar 2014

[5] J. Hursti, "*Single Sign-On*." Retrieved December 4, 2016, from http://www.tml.tkk.fi/Opinnot/Tik-110.501/1997/single_sign-on.html, 1997.

[6] H. Mohammed, M. F.Suliman, V. Ponnusamy, B. Y. Ooi, A. Robithoh, and S.Y .Liew,." A Coherent Authentication framework for Mobile Computing Based on Homomorphic Signature and Implicit Authentication." Proceedings of the 6th International Conference on Computing and Informatics, ICOCI 2017, 25-27April, 2017 Kuala Lumpur. University Utara Malaysia.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". SIAM J. Comput.26, 1484–1509 ,1997.

[8] D. Bernstein, J. Buchmann, and J. Ding, "Post-Quantum Cryptography". Springer, Heidelberg, 2008.

[9] G. Craig, and H. Shai, "Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits", Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, p.107-109, October 22-25, 2011 [doi>10.1109/FOCS.2011.94]

[10] C. Joo and A. Yun, "Homomorphic Authenticated Encryption Secure Against Chosen-Ciphertext Attack", in ASIACRYPT: International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C, 2014.

[11] S. Blanda, "Shor's Algorithm – Breaking RSA Encryption", AMS Grad Blog, 2018. [Online]. Available: https://blogs.ams.org/mathgradblog/2014/04/30/shors-algorithm-breaking-rsa-encryption/.

[12] B. Dickson, (2017). "5 authentication methods putting passwords to shame". [online] The Next Web. Availableat:https://thenextweb.com/insider/2016/03/31/5-technologies-will-flip-world-authentication-head/ [Accessed 7 Nov. 2017].

[13] Decompilingandroid.com. (2017). "Top 10 Mobile Security Risks | Decompiling" Android. [online] Available at: http://www.decompilingandroid.com/mobile-app-security/top-10-mobile-security-risks/ [Accessed 7 Nov. 2017].