Smart Verification Algorithm for IoT Applications using QR Tag

Abbas M. Al-Ghaili[1], Hairoladenan Kasim[2], Fiza Abdul Rahim[2], Zul-Azri Ibrahim[2], Marini Othman[1,2], and Zainuddin Hassan[2]

[1] Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), 43000 Kajang, Selangor, Malaysia
[2] College of Computer Science and Information Technology (CSIT), UNITEN, 43000 Kajang, Selangor, Malaysia
{abbas, hairol,fiza,zulazri,marini,zainuddin}@uniten.edu.my

Abstract. A Smart Verification Algorithm (SVA) used with Internet of Things (IoT) applications is proposed performing a verification procedure to enable authorized requests by user to access a smart system with help of Quick Response (QR) tag. It uses encrypted QR-tag values to compare them to original values. Three-layers have been proposed for this verification procedure to attain security objectives. The first layer implements a comparison to preserve the system integrated. In the second layer, original values are stored in offline database storage to disable any access caused by threats; to preserve it available. The third one frequently generates an authenticated QR tag using 1-session private key to prevent both information leakage and an unauthorized access if the key was deduced; to keep it confidential. The SVA aims to increase the system privacy. It is evaluated in terms of security factors. Results confirm that it is faster than other competitive techniques. Additionally, results have discussed SVA's robustness against unauthorized access's attempts and brute force attack.

Keywords: Internet of Things, QR Tag, Smart applications.

1 Introduction

Nowadays, researches focusing on the use of QR tag increase rapidly when comparing the current decade to last three decades. The number of recent studies has shown a rapid increment in research studies dedicated for the QR tag related topics especially beyond the year 2010, as shown in Fig. 1. QR tag features (1, 2), such as storing a huge number of encrypted information in a small shaped-image, is one of the reasons attracting many researchers to propose secure systems (3). Some examples include Internet of Things (IoT) based devices (4) intelligent systems (5), smart access cards (6), and automation processes (7) embedded systems (8) that produce smart services to the user and keep data securely stored.

Smart home applications include several purposes raised from the need of use. For example, monitoring application (9), biometrics-based home access system (10)

…etc. This variety indicates that IoT relies on smart essential tools e.g., QR-tag used as an access tool for IoT applications in which a strong encryption scheme is needed.

QR tag stores a huge number of data in a simple image with small size; many smart systems have exploited such feature(s) so that QR tags are easily scanned. Once, the data stored inside the QR tag has been extracted, the verification process is implemented. The verification process is a very important step to make sure that extracted information is identical to the original one. Thus, the verification process of QR tag is an essential step in smart home applications, because it affects the system privacy.

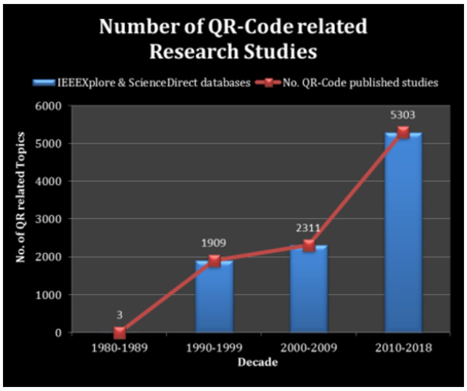

Fig. 1. Number of QR-Code related Research Studies and Topics

However, QR tags are used as different smart tools. The proposed work in (3) has used a QR tag in order to head the robot for measurements. The QR tag acts as a landmark image to ease the recognition process. In (11), the QR tag is generated in such a way to be scanned easily with no much details of black-white-boxes. Other interesting graphical design of QR tag is proposed in (12) to verify documents in terms of authentication and privacy. The original QR tag patterns must be replaced by a specific array of patterns in order for the QR tag to be correctly scanned. In addition, a method to protect the private data stored in the QR tag is designed (13) to overcome the print-and-scan (P&S) operation.

In literature review, there have been many proposed researches using the QR tag with a simple layer of encryption and protection. These methods are suitable for use with smart systems but those which include sensitive data and need a strong protection policy, are vulnerable to threats and attacks. Therefore, to make the QR tag based Smart Verification Algorithm (SVA) achieve a high level of usability and responsiveness, a strong verification policy has been considered in design. In this article, the proposed SVA has considered a number of security issues e.g., integrity and privacy. Additionally, the SVA proposes a simple and reliable QR-tag scanner to verify its contents in terms of authentication. In addition, SVA's verification procedure contains three layers to increase the security. The private key design considers the decryption time caused by an unauthorized action.

This article is organized as follows: Section 2 explains the integrated proposed SVA for IoT applications. Performance analysis and evaluation are discussed in Section 3. Conclusion is provided in Section 4.

2 The Integrated Proposed SVA for IoT Applications

2.1 The Proposed SVA Architecture

The proposed SVA architecture is illustrated in Fig. 2. Once the QR tag is scanned, its information will be verified and then the Reference Value (RV) will be provided to the user to increase the access security. After that, the ID will be checked. Finally, the database is updated based on some security criteria.

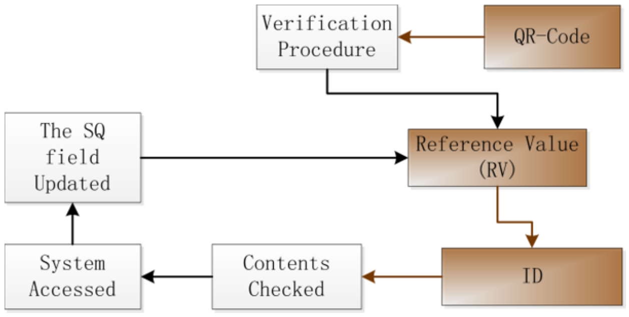

Fig. 2. The Architecture for SVA

2.2 Relation between SVA and Encryption for IoT Applications

After the encryption process has been performed, a hash value is converted to a QR tag logo (as an image) in order to be sent to the user device e.g., mobile app. Then, user is asked to enter an ID value after the system has accepted the QR-tag. Another example of SVA security is that, the RV must be correctly inserted before the access is allowed.

As discussed above, there will be three security levels (QR Scan, RV, ID comparison). Thus, the framework of SVA verification procedure and its relation to the encryption part is illustrated in Fig. 3. The proposed framework has three levels of safety against an unauthorized access and/or modification. In case the system has been modified in an unwanted manner, a wrong RV will be obvious when a hash function is tested. Hence, the system will not allow any access.

This figure shows the SVA relation to the encryption procedure. Once, a user needs to access the system, the encryption procedure immediately is called. The first security step is to show *QR tag by* using a smart device such as a mobile app to allow the user access the system. The pre-generated QR tag which has been stored in the offline database will be used to verify the QR tag with a reference code. If they are identical, the first step is approved; otherwise, the system rejects the process. The second step is to insert an RV to make sure that the right user has displayed the correct QR tag (in the previous step). To increase the security, in this step, the RV is inserted manually by using the mobile app. The RV is frequently changed and the *offline database storage* is updated accordingly. The third security process is *User-id Verification*; it asks the user to enter the *id* number. This number has been previously generated using a complex process. When it is entered, the system will use a special process using a Hash function between the entered and stored ones.

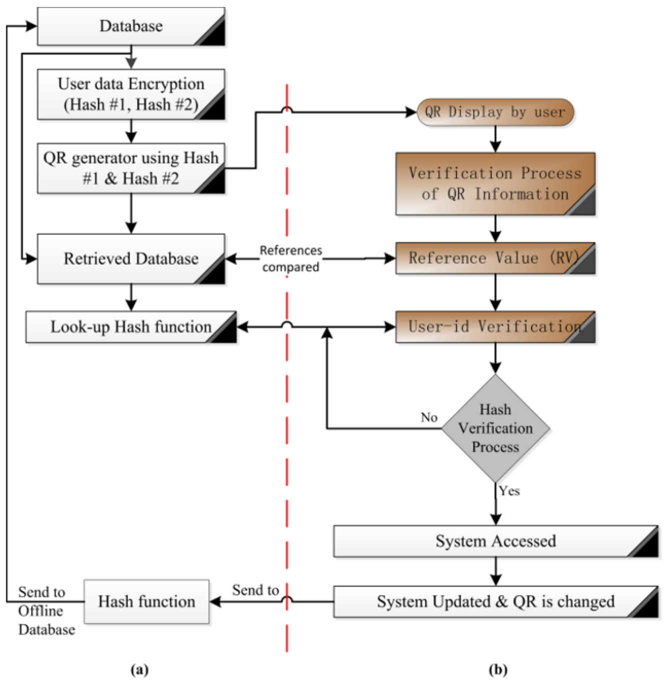

Fig. 3. A SVA Verification Procedure (a) and Encryption (b) Framework

2.3 SVA Flowchart

As discussed earlier, there are four types of verifications which are as follows: a recent face image capturing, fingerprint detection, RV, and user ID. They are discussed in detail as follows:

Face image and Fingerprint Verification Procedure. The first step in SVA is to read information from the offline database and look-up table, once the QR tag has been scanned. The purpose is to do a successful and trust comparison between QR tag and original pre-stored information. The comparison could be illustrated as in Fig. 4. Once information asked by the user and information stored in the database are identical, the system can be accessed. Otherwise, the access is denied.

Hash based Reference Value (HRV) and ID Verifications. The RV is frequently and periodically changed and is expired once it has been used for a short period of time. The SVA needs to update RV in the database. Then, the encryption for the RV will produce a different hash value. The *new* generated QR tag will be updated accordingly. Thus, the new hash value for the user's RV will be periodically verified.

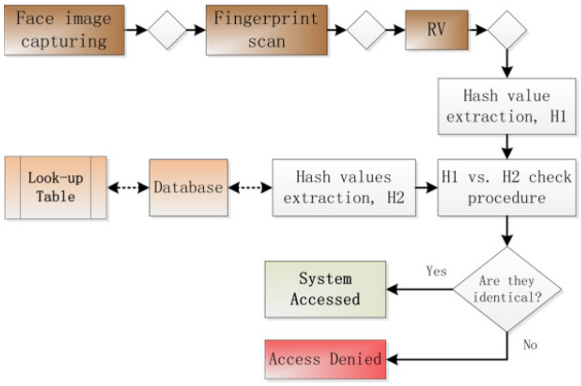

Fig. 4. The Proposed SVA Flowchart

The HRV and ID verifications are applied by using a number of cryptographic operations using algebraic formulas and logic operations. The mathematical procedure is briefly provided in Algorithm 1, Algorithm 2, and a flowchart depicted in Fig. 5.

HRV firstly produces a hash value, h'_{RV}; as expressed in (1):

$$h'_{RV} = hash(RV1 \circledcirc RV2, k_p, M) \quad (1)$$

whereas

- h'_{RV} represents the first calculated hash value
- $RV1 \circledcirc RV2$ is the hash function input and is a produced mathematically value
- $\circledcirc$ represents the mathematical operations applied on $RV1$ and $RV2$
- k_p is the 2nd input of hash function and is a private key used only one time.
- M is the 3rd input of the hash function and is the message obtained from the user's pre-entered data

Once the value h'_{RV} has been calculated, a further encryption process is applied with a rolling function to produce a new hash value, H_{RV}, as expressed in (2):

$$H_{RV} = E(h'_{RV}, k_s) \quad (2)$$

whereas

- H_{RV} is the 2nd hash value and is the encrypted value for $RV1$ and $RV2$
- $E(h'_{RV}, k_s)$ encryption procedure applied on h'_{RV} by using a different secret key, k_s.

The proposed algorithm for this procedure is shown in Algorithm 1:

```
· Start
· Apply pseudorandom number generator (PRNG) on two RVs $RV1$ and $RV2$ values
· Apply various mathematical operations to produce $RV1 \circledcirc RV2$ value
· Produce 1-session private key $k_p$
· Implement a hash function ($h'_{RV}$); Eq. (1)
· Implement an encryption algorithm to produce $H'_{RV}$; Eq. (2)
· Apply a rolling function
· End
```

Algorithm 1: HRV Verification Procedures

The pseudo-code of the ID verification procedure is shown in Algorithm 2.

```
· Start
· Do initialization for the following values:
      -Scan QR tag for user #i
      -Ask the user to enter the correct id
      -Call the stored hash value for the entered id(i) → $Hash(id_{Look\text{-}up}(i))$
· Do the Hash function for id → $Hash(id(i))$
· Extract the $id_{QR}$ value stored inside the QR tag
      -Do Hash function for this id → $Hash(id_{QR}(i))$
· Compare $Hash(id(i))$ AND $Hash(id_{QR}(i))$ to $Hash(id_{Look\text{-}up}(i))$
· Return the comparison value (True OR False)
· End
```

Algorithm 2: ID Verification Procedure

This algorithm checks the user ID collected from different resources. The algorithm compares three collected values. The process compares the encrypted value stored in QR tag, $Hash(id_{QR}(i))$ to the one the user will enter manually which is, $Hash(id(i))$. The result will be compared to the encrypted value stored in the original look-up table, $Hash(id_{Look\text{-}up}(i))$.

3 Performance Analysis and Evaluation

This section performs an analysis and evaluation procedure. The performance of the proposed research work is discussed. To achieve a high level of performance of the proposed algorithm, various parameters and factors have been considered. So that points of views are considered to cover most of the security issue.

3.1 Security Factors Analysis

Confidentiality. Information is transmitted between two authorized parties, using a strong encryption algorithm. The QR tag is periodically generated using a 1-session key to increase information confidentiality and keep it secure.

Integrity. QR tag image patterns are scanned and verified. Additionally, face image and fingerprints are verified. If there is any mismatch, the answer will be wrong. Thus, the SVA has no integrity. Thus, a third party has modified the QR tag contents.

Availability. There will be no access by a third party but only one authorized source is allowed to access thru the offline database given a certain period of time.

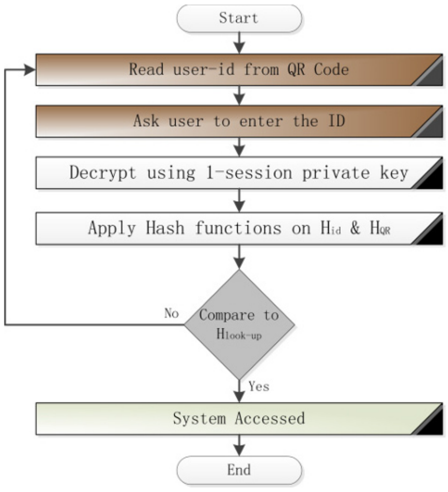
Start
Read user-id from QR Code
Ask user to enter the ID
Decrypt using 1-session private key
Apply Hash functions on H_{id} & H_{QR}
Compare to $H_{look-up}$
No
Yes
System Accessed
End

Fig. 5. Mathematical operations based user-id comparison flowchart

3.2 Other Factors

Authentication. The authentication factor to be verified is denoted: $F_{authentication}$. Here, a small portion of grey-color patterns (pixels' intensities) are merged with a hash value obtained from the user $Hash(user)$ predefined earlier in a secure mode. The hash value is compared to the original one $Hash(db)$; as mathematically expressed in Eq. (3). If it is correct, a certain hash value will be fit with those grey patterns to produce a new QR tag. The SVA is going to scan the new resulted QR tag. This procedure is performed periodically every 24 hours to guarantee authority of QR tag issue. Any mistake of the entered value will appear in this step. Eq. (8) is applied for QR tag authentication.

$$F_{authentication} = \begin{Bmatrix} 1 & Hash(user) == Hash(db) \\ 0 & Hash(user) \neq Hash(db) \end{Bmatrix} \tag{3}$$

If $F_{authentication} = \{1\}$, then, the QR tag is generated by the original source. Otherwise, it is generated by an unauthorized source. That means there is no authenticity for QR tag being used to access an IoT system.

Robustness. In order for the SVA to allow an access to the related system or physical device by using the application, the entries QR tag Scan, RV, and ID are tested. However, the application might ask the user for information collection procedure. These must be correct; otherwise, the system does not allow any attempt to access. If the QR tag has been used successfully and one or more of other entries was wrong, the user can't access. This verification policy increases the security. Hence, it is clear that any error with SVA input will lead to the same output. Meaning, any possibility of error existence during access's attempts, it is rejected. That can give the SVA more robustness against unauthorized attempts; to prevent threats.

Computation Time based Efficiency. To verify the SVA efficiency, the computation time is calculated and compared to other existing algorithms'.

As discussed earlier on how the encryption process is done to apply a hash function (H_p) on user information. Firstly, using SHA-1, the 1-session key is generated. The hash value extracted from the QR tag (H_{QR}) is compared to the original hash value stored in the offline database; i.e., H_p and H_{QR}. Then, the user is asked to enter an RV and ID value as the message being used for this evaluation. Next, the message, session key, and hash values are encrypted to create a public key. This value is being hashed. Finally, the QR tag will be generated using these information and values. The SVA average computation time compared to other schemes is shown in Table 1.

Table 1. SVA Computation Time Compared to Other Schemes

No. Tests	Password; ms	Certificate; ms	(14); ms	Proposed SVA; ms
10	24.9	45.2	31.4	28.6
20	43.7	91.2	63.1	50.2
50	115.6	212.6	149.0	132.1
100	205.1	452.2	313.3	226.7
200	453.2	921.6	645.7	521.4

This comparison shows that the proposed SVA is faster than Certificate and the work proposed in (14). However, the Password system is faster because the time needed for encryption and verification process of SVA takes more time for a more secure procedure. Another reason is that, the SVA has applied the hash function more than one time; that is to increase the safety of the system being accessed. In Fig. 6, the SVA computation time is on the second rank whereas a less computational time is achieved with a good level of safety and security. In Fig. 7, the average computation time is shown whereas its value ranges between 23.8% and 27.2% among others.

Key Length against brute force attack. The SVA is evaluated in terms of brute force attack

Table 2. Key length based decryption-time evaluation

Key length; size in bits	Number of alternative keys	Time required; 10^6 decryption/µs time in years
168	3.7×10^{50}	6×10^{30}
320	2.1×10^{96}	3.4×10^{76}

Table 2 shows that a very long time is needed to decrypt a message with different key sizes. The last column mentions the time needed for a system in which 10^6 keys could be processed in 1 µs. This is computationally secure.

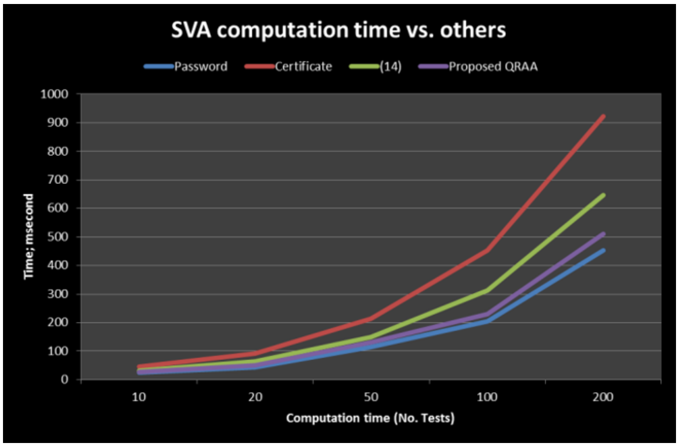

Fig. 6. SVA computation time compared to other methods

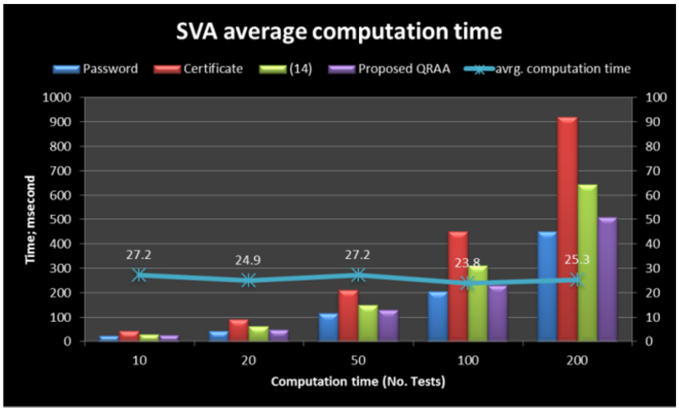

Fig. 7. SVA average computation time

4 Conclusion and Future Works

This paper proposes a simple algorithm used as an access procedure for smart applications and IoT applications such as smart home applications and security gates. The proposed SVA collects user information and encrypts them in order to create a secure QR tag image. The design of SVA has considered the security factors and mechanisms. Results and evaluation have discussed different factors such as integrity, availability…etc. results have confirmed that SVA is strong against brute force attack in terms of time required to crack it. The performance has confirmed it is real-time responsive, reliable, and useable. However, there are some future works suggested. It is recommended, for example, to reduce the computation time during verification process to attain achieving more security and less computation time.

Acknowledgements

Work presented in this paper forms part of the research on Formulation of Evidence Source Extraction Framework for Big Data Digital Forensics Analysis in Advanced Metering Infrastructure, which was partially funded by Universiti Tenaga Nasional Start-Up Grant UNIIG 2016.

References

1. Kirkham T, Armstrong D, Djemame K, Jiang M. Risk driven Smart Home resource management using cloud services. Future Generation Computer Systems. 2014 2014/09/01/;38:13-22.
2. Samuel SSI, editor A review of connectivity challenges in IoT-smart home. 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC); 2016 15-16 March 2016.
3. Nazemzadeh P, Fontanelli D, Macii D, Palopoli L. Indoor Localization of Mobile Robots Through QR Code Detection and Dead Reckoning Data Fusion. IEEE/ASME Transactions on Mechatronics. 2017;22(6):2588-99.
4. Liu Z, Choo KKR, Grossschadl J. Securing Edge Devices in the Post-Quantum Internet of Things Using Lattice-Based Cryptography. IEEE Communications Magazine. 2018;56(2):158-62.
5. Rane S, Dubey A, Parida T, editors. Design of IoT based intelligent parking system using image processing algorithms. 2017 International Conference on Computing Methodologies and Communication (ICCMC); 2017 18-19 July 2017.
6. Huang H-F, Liu S-E, Chen H-F. Designing a new mutual authentication scheme based on nonce and smart cards. Journal of the Chinese Institute of Engineers. 2013 2013/01/01;36(1):98-102.
7. Xiao-Long W, Chun-Fu W, Guo-Dong L, Qing-Xie C, editors. A robot navigation method based on RFID and QR code in the warehouse. 2017 Chinese Automation Congress (CAC); 2017 20-22 Oct. 2017.
8. Ghaffari M, Ghadiri N, Manshaei MH, Lahijani MS. P4QS: A Peer-to-Peer Privacy Preserving Query Service for Location-Based Mobile Applications. IEEE Transactions on Vehicular Technology. 2017;66(10):9458-69.
9. Chen YH, Tsai MJ, Fu LC, Chen CH, Wu CL, Zeng YC, editors. Monitoring Elder's Living Activity Using Ambient and Body Sensor Network in Smart Home. 2015 IEEE International Conference on Systems, Man, and Cybernetics; 2015 9-12 Oct. 2015.
10. Kanaris L, Kokkinis A, Fortino G, Liotta A, Stavrou S. Sample Size Determination Algorithm for fingerprint-based indoor localization systems. Computer Networks. 2016 2016/06/04/;101:169-77.
11. Lin SS, Hu MC, Lee CH, Lee TY. Efficient QR Code Beautification With High Quality Visual Content. IEEE Transactions on Multimedia. 2015;17(9):1515-24.
12. Tkachenko I, Puech W, Destruel C, Strauss O, Gaudin JM, Guichard C. Two-Level QR Code for Private Message Sharing and Document Authentication. IEEE Transactions on Information Forensics and Security. 2016;11(3):571-83.
13. Lin PY. Distributed Secret Sharing Approach With Cheater Prevention Based on QR Code. IEEE Transactions on Industrial Informatics. 2016;12(1):384-92.
14. Kim YG, Jun MS, editors. A design of user authentication system using QR code identifying method. 2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT); 2011 Nov. 29 2011-Dec. 1 2011.