2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC), and IEEE Intl Conference on Intelligent Data and Security (IDS)

Efficient Implementation of 2D Barcode Verification Algorithm for IoT Applications

Abbas M. Al-Ghaili
Institute of Informatics and Computing in Energy (IICE)
Universiti Tenaga Nasional (UNITEN)
43000 Kajang, Selangor, Malaysia
Corresponding author's e-mail:
abbasghaili@yahoo.com &
abbas@uniten.edu.my

Fiza Abdul Rahim
College of Computing & Informatics (CCI)
Universiti Tenaga Nasional (UNITEN)
43000 Kajang, Selangor, Malaysia
fiza@uniten.edu.my

Feninferina Azman
College of Computing & Informatics (CCI)
Universiti Tenaga Nasional (UNITEN)
43000 Kajang, Selangor, Malaysia
ferina@uniten.edu.my

Hairoladenan Kasim
College of Computing & Informatics (CCI)
Universiti Tenaga Nasional (UNITEN)
43000 Kajang, Selangor, Malaysia
hairol@uniten.edu.my

***Abstract*—This paper proposes a two-Dimensional Barcode-applied Verification Algorithm (2DBVA). The two-Dimensional Barcode (2DB) is used as an access tool for Internet-of-Things (IoT) applications. From a security aspect's point of view, 2DBVA has been evaluated in terms of security factors. To perform this evaluation, a Multiple-Layer Security Architecture (MLSA) has been used in order to attain the three security objectives which are: confidentiality, integrity, and availability. The 2DBVA aims to enable an authorized access to smart IoT applications such as smart city applications. The 2DBVA has adopted the 2DB to encrypt values being used as a smart-key. The performance of the proposed algorithm has considered a number of security factors in order to strengthen the security side of such an IoT application. To evaluate the 2DBVA accuracy, a total number of 166 2DBs has been generated. The proposed 2DBVA was implemented to securely encrypt and verify contents of 166 2DBs. Results have shown a high percentage of accuracy in terms of security for verified 2DBs used as smart keys for IoT applications.**

Keywords—IoT, 2D barcodes, Smart access

I. Introduction

Many researches from a broad variety of fields have exploited the technique of two-dimensional barcode (2DB) in order to design a smart application. 2DB is used as an access card or a key for several applications since a 2DB is easy-to-scan [1-3]. Thus, it has been exploited by many Internet-of-Things (IoT) applications [4-8] performing cryptographic operations. The 2DB technique has been widely used by many IoT applications [9] such as; access systems [10, 11], authentication systems [12], smart home applications [13], smart cities related applications [14], identification systems [15], smart authentication and verification algorithm [16]. However, these applications require a strong 2DB encryption and verification procedures implemented to secure data [17] and also several security were considered.

Even though 2DB based IoT algorithms dealing with smart applications still face challenges to cover security factors. Nevertheless, 2DB features are exploited by many researches to achieve a limited level to which the smart application is expected to reach. For example, the proposed work in [18] has designed a 2DB based authentication method for users achieving good performance in terms of threats prevention. Some other methods [10] used remote user authentication process with smart cards. The 2DB verification procedure is essential in IoT applications because it verifies contents if they are original.

One of the existing solutions applied in this era is that various 2DB related studies have used a simple layer of encryption. A single layer of security was adopted either to use a secure 2DB without adding a further verification algorithm or to validate the time the 2DB has been generated. Using a single layer is adequate with smart applications. But, applications containing sensitive data might be vulnerable to threats and attacks because they consider one or two security objectives. This paper considers five security objectives to verify. Therefore, a Multiple-Layer Security Architecture is proposed and used alongside the two-Dimensional Barcode Verification Algorithm (2DBVA) in order to reach a high level of privacy. Thus, the MLSA versus 2DBVA is aimed to allow a safe access to IoT application.

The proposed MLSA-2DBVA considers several security factors to verify e.g., integrity and authentication in order to verify requests being made to access an IoT application. Additionally, its verification procedure has several layers to increase the security. The MLSA-2DBVA is useful to verify requests that need permissions to access such IoT automatic access systems by using a smart time-based label (TL) to calculate a 2DB age to decide whether the 2DB is authenticated or not and how authenticated it is.

This research is motivated by the need to increase security of IoT applications specifically when there is sensitive and private data. Another important feature of this research is that the main contribution to be achieved is to protect transferred data thru IoT digital platforms and keep data secure and confidential.

This paper is organized as follows: Section II introduces the proposed MLSA-2DBVA. Section III discusses the proposed 2DBVA. The proposed MLSA for making a decision on 2DB authentication is presented in Section IV. Performance analysis and evaluation are discussed in Section V. Section VI summarizes Conclusion.

978-1-7281-0006-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataSecurity-HPSC-IDS.2019.00059

II. The Proposed Multiple-Layer Security Architecture vs. 2D Barcode Verification Algorithm (MLSA-2DBVA)

A. *Graphical Overview*

A simple overview to add a further clarification on the conception is graphically depicted in Fig. 1.

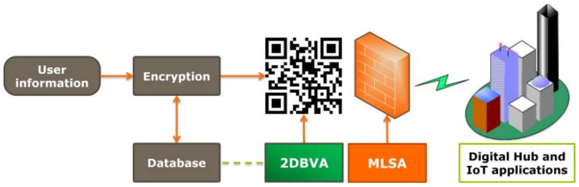

Fig. 1. A Graphical Overview of the Proposed 2DBVA

B. *MLSA vs. 2DBVA Block-Diagram*

The proposed MLSA vs. 2DBVA block-diagram consists of three simple and sequential processes in a one by one order, as depicted in Fig. 2. These processes are briefly introduced as follows: the first process acts as a receiver to contents of 2DB. The second process acts as a sender of verified results. It sends values from 2DBVA to database to store original values. Thirdly, the MLSA receives results from 2DBVA for verification procedures. Hence, a decision is made either to accept user's request or reject.

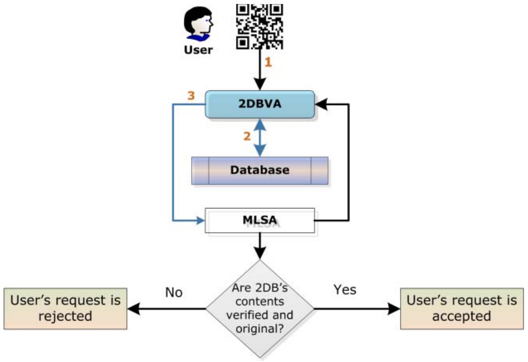
User
1
3
2DBVA
2
Database
MLSA
No
Yes
User's request is rejected
Are 2DB's contents verified and original?
User's request is accepted

Fig. 2. The Proposed MLSA vs. 2DBVA Block-diagram

C. *Introduction to MLSA vs. 2DBVA*

MLSA's and 2DBVA's main units are introduced here. The proposed architecture shown in Fig. 2 has included three units: 2DBVA, database, and MLSA. First and second units are processing units whereas the third one is a decision making unit. The first unit receives inputs from 2DB. Usually, user's request(s) will be first tested; whereas several verification procedures will be applied on contents being verified. In many cases, in order for verified contents being successfully checked, the user request will re-call relatively certain original values from the database unit to be processed. The second unit (i.e., database) basically transfers updated values to the first unit, 2DBVA. Some related values will be sent to the third unit in order to verify requests made by the user. At the third unit (i.e., MLSA), a decision will be made for the user's request using certain values extracted from the user 2DB code's contents. If the verification result is correct, the MLSA accepts such a request and then allows the access to an IoT application.

III. The Proposed two Dimensional Barcode Verification Algorithm for contents (2DBVA)

A. *Definition of the 2DBVA*

The 2DBVA is a processing unit and it is followed by a database processing unit. 2DBVA is an essential unit on which other units, specifically the 3rd unit (i.e., MLSA), mainly rely. The relation between this unit and other units is depicted as a generalized flowchart in Fig. 3.

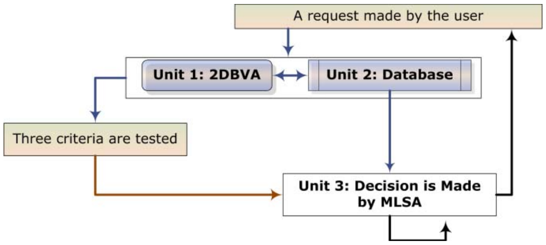
A request made by the user
Unit 1: 2DBVA
Unit 2: Database
Three criteria are tested
Unit 3: Decision is Made by MLSA

Fig. 3. 2DBVA unit's flowchart and its procedural relation to other units

In this figure, the proposed 2DBVA includes three key processes by which 2DB's contents are verified. These verification processes are applied at every time 2DB is tagged by the user. To ease this conception, it is said that these processes are combined together in a single verification procedure. Thus, the 2DBVA's main procedure consists of three secure cascade procedures shown in Fig. 3.

B. *2DBVA Secure Cascade Procedures*

Simplify, the proposed 2DBVA includes three procedures applied on every 2DB in order to do a verification procedure. They are as follows: the first one is the 2DB verification which is dedicated to verify the integrity of contents of 2DB. The second procedure verifies the authentication and confidentiality of 2DB (2DB is the key of the smart IoT application). The third one is to verify availability of 2DB by applying a verification procedure for the database of application (usually 2DB has certain and secret values stored in this database). This verification procedure is applied to ensure that 2DB is preserved available and responsive all times and whenever a 2DB is scanned trying to access the IoT application. These procedures are serially implemented as shown in Fig. 4.

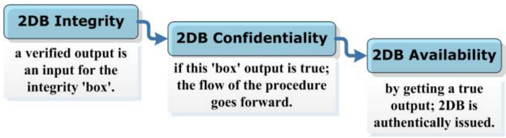
2DB Integrity
a verified output is an input for the integrity 'box'.
2DB Confidentiality
if this 'box' output is true; the flow of the procedure goes forward.
2DB Availability
by getting a true output; 2DB is authentically issued.

Fig. 4. 2DBVA Secure Procedures – a cascade relationship

In this figure, it is clear that each successor is fed by the output of its predecessor. In other words, any of the next procedure will be not able to start unless the previous procedure has completely verified the 2DB. This aims to add a more secure level to contents of 2DB. This guarantees that each step does not process unauthorized 2DB.

C. *Verification of 2DB Integrity*

There are three steps carried out; so that contents of the 2DB is being fully verified. Three steps are proposed in order for this type of verification to be done, which are as follows: verification of biometrics images, verification of serial ID, and verification of Security Question (SQ). The proposed architecture is illustrated in Fig. 5. Once 2DB is scanned, its contents (e.g., biometrics originally stored and pre-encrypted) are extracted and verified. The user will be requested to provide a serial ID. In some cases, a user is asked a secret question and it is necessary to make a responsive answer to IoT application in order to be allowed to access the system. These answers are compared first to database. The result is sent to MLSA to make a decision.

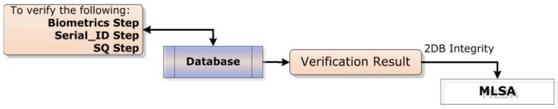

Fig. 5. 2DB Integrity Verification Procedure

1) *Biometrics Images based Verification Step*

In this step, the essential part is to bring together user's information, by scanning the 2DB. Then, a comparison between relatively certain values obtained from the 2DB and original pre-stored contents inside system's database will be done. Once information given by the user and contents stored in database are identical, the system is accessed. Otherwise, the access is immediately rejected.

This is to compare images' values of face to 2DB values. If they are correct, the process goes forward to the next verification step (i.e., next successor 'Serial ID').

```
do {
user_id(i) profile structure is created and assigned a Serial_ID;
} while i is valid and truly generated;
Apply encryption formula to produce a hash value of Serial ID (i)
Scan 2DB for user_id(i);
Extract Serial ID(i) and calculate its hash value;
Compare it to database;
    If (both are equal)
        Send the result to MLSA;
    Else
        Reject a request made by user_id (i);
        Deny 2DB_(i) from an access to IoT application;
    End If
```

Algorithm 1: Serial ID Verification Step

If result of comparison is true; i.e., values are identical. Thus, the system is integrated and contents of the 2DB are authenticated. If there is a mismatch, a possibility of wrong Serial ID is inserted or an unauthorized 2DB might be used.

2) *Serial_ID Verification Step*

Usually, every new user will be provided a distinctive Serial ID. Each Serial ID consists of a very complicated cryptographic and hashed value. This value contains a series of unknown alphanumeric or numerical digits and letters. To implement this verification step, firstly, the user needs to insert the distinctive ID. Then, a number of decryption operations will be applied aiming to extract the hash value. After that, real and original values will be obtained using mathematical calculations. Next, a comparison is done to verify whether the obtained hashed Serial ID is equal to the Serial ID provided by the user or not. As a final step, the comparison's result will be passed to the MLSA unit in order to carry out an appropriate decision. The pseudo-code of this step is further explained in Algorithm 1.

3) *SQ Verification Step*

The aim is to verify two parts. The first part is the one which extracts some specified real values from system's database. The second one extracts the SQ message (M) from the user's 2DB whereas an encryption scheme is applied, together with the real encryption scheme applied earlier; so that both hash values are obtained and verified. This verification is designed in such a way that the SQ is guaranteed to be continuously updated and therefore its related values and mathematical operations are varied. The SQ is generated using a pseudorandom number generator (PRNG) based on certain values obtained from User Activities History (UAH). In this matter, the system's database is used whereas related UAH's values are extracted. This is shown in Fig. 6, marked by green dotted lines. UAH is a changeable variable with the rule: "*each time the user_id(i), has successfully performed an activity, new values could be created and then stored*". Hence, SQ is updated in database.

In Fig. 6, on the right side, in order for the mathematical encryption to be implemented, there will be an SQ related pre-process. The first step of this pre-process is to produce a random number (N_R). Then, the N_R is combined with the SQ using a number of mathematical operations. The result of this pre-process is a combined random number: i.e., $SQ \blacksquare N_R$. The produced number will be one input of the encryption algorithm applied subsequently. This is followed by a rolling function which aims to increase the SQ privacy. Finally, an additional secure procedure is applied, which consists of two secure procedures which are an encryption scheme and then this is followed by a hash function. This is mathematically represented as follows: $H(E(SQ \blacksquare N_R), M, k)$.

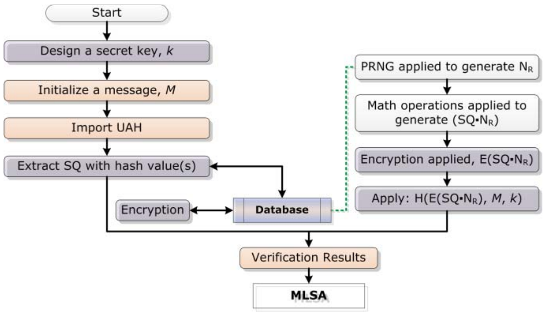
Start
Design a secret key, k
Initialize a message, M
Import UAH
Extract SQ with hash value(s)
Encryption
Database
PRNG applied to generate N_R
Math operations applied to generate (SQ•N_R)
Encryption applied, E(SQ•N_R)
Apply: H(E(SQ•N_R), M, k)
Verification Results
MLSA

Fig. 6. SQ Verification Step Flowchart

In this figure, results taken from both sides of the flowchart (left and right sides), are verified. If they are approved, the result is sent to MLSA; otherwise, the SQ Verification Step will be certainly rejected.

D. *Verification of 2DB Confidentiality*

Usually, each produced 2DB will be assigned a time-based label (TL) to denote 2DB validity starting for the time specified. This is done in encryption scheme. But, in this type of verification, the TL can be verified. The 2DB expiration date is getting controlled by the label, TL. The verification here focuses on the time the 2DB has been generated. This is shown in Algorithm 2. Finally, the TL has considered a periodically generated 2DB policy every an ordered interval of time predefined by the variable *TL*, suppose that TL= 24 hours.

```
Scan 2DB
Match 2DB with its user_id(i);
Extract the time label (TL);
Apply: (Time.Now_function()-TL) on TL;
If (Time.Now_function()-TL)⩽24 hours;  //read current time
    Allow 2DB scan procedure;
Else
    Reject 2DB scan procedure;
```

Algorithm 2: Time-based Label for 2DB

E. Verification of 2DB Availability

The third procedure is designed to protect system's database. It performs two sub-processes. The first one denies an unauthorized access caused by potential threats. The second pre-process arranges the way the authorized resource is trying to do whether a modification or an access to read certain values from database. The 2DB availability verification procedure is stores any updated value inside the database using a periodically and in-offline mode. To reduce access to the database, this procedure has determined the database access upon necessary requests. That means when the process is done, i.e., *process_requests_access==0*, certain values are stored in such a way to reduce the data size and access times as represented in Algorithm 3.

```
While (process_requests_access==0 AND connect_status==0) {
    Close resources;
    Calc_Time(); // function 1
    Assign a resource;
    Disconnect(); // function 2
    Update offline database; }
```

Algorithm 3: 2DB Availability Verification Procedure

In this algorithm, there are two important functions are recalled, which are as follows: the first function is a time, *Calc_Time()*, to assign a certain period of time. The second one is *Disconnect()* function to convert the connection status to 'Disable' and also to update the system's database while there is no connectivity. In regard to Data allocation and management, there is a number of intelligent algorithms proposed to reduce the computation time and increase the simplicity of assigning resources to process data, e.g., [19].

IV. Multiple-Layer Security Architecture (MLSA)

A. Overview

The MLSA is used to verify 2DBVA to increase its security. To achieve the *Integrity* objective, the 2DB is verified thru a database-based verification.

B. The Proposed MLSA for Decision Making

The decision making factor verified is denoted by: *vote*. One of the measures that *vote* needs to verify is the Serial ID predefined earlier. Other required steps to verify their values are however attributes of images of biometrics, SQ, and TL-based expiration date of 2DB. Their hash values are compared to original ones stored in the database. This is shown in Algorithm 4 to guarantee authority of 2DB issue.

```
Scan 2DB; and set: vote=0;
If (verified_value==1) // verify Biometrics_attributes();
        vote1=vote1+20;
If (verified_value==1) // verify Serial_ID();
        vote1=vote1+20;
If (verified_value==1) // verify SQ(); i.e., TL
        vote1=vote1+20;
switch case (TL) { // verify expiry_date(); i.e., TL in hours
      1≤TL≤6 ? vote=vote1+24;
      7≤TL≤12 ? vote=vote1+18;
      13≤TL≤18 ? vote=vote1+12;
      19≤TL≤24 ? vote=vote1+6;
      24<TL ? vote=--vote1; }
```

Algorithm 4: 2DB contents based Decision Making Procedure

Algorithm 4 assigns values to each procedure approved. Every assigned value increases validity scale of 2DB. If it is recently generated within 6 hours, 1≤TL≤6, a value of '24' is assigned and added to '*vote*' and '*vote*=84' if the previous three verifications are valid. This value is the highest value a 2DB gets to ensure a high percentage of validity. Suppose that, 19≤TL≤24, then: '*vote*=66' to point out the lowest value the 2DB can get to pass the authentication test. If '*TL*>24', the 2DB is expired and therefore '*vote*' is reduced by 1, '*vote*=59', to ensure its expiry. This case cannot be accepted by MLSA-2DBVA and access is rejected.

To verify the MLSA-2DBVA robustness, all three verifications must be correct none some. Thus, if the 2DB has been used successfully and one or more of other verifications was wrong or mismatched to original values, the user will be prevented from any access. Meaning, suppose that the two verification processes '*Serial_ID*()' and '*SQ*()' are correct but '*Biometrics_attributes*()' isn't; then the '*vote1*=40' value and if 1≤*TL*≤6, then '*vote*=64<66' will be rejected. Thus, this procedure is used to test the MLSA-2DBVA robustness. Hence, a more robustness level against unauthorized attempts to prevent threats and actions is achieved.

C. MLSA based Security Objectives Verification

There are four security objectives are addressed, i.e.: integrity, confidentiality, authentication, and availability. While the 2DBVA always verifies the user entries and contents of 2DB to ensure that contents are true as well as a hash based verifications for biometrics and ID, the obtained results could verify the integrity of user information thru database to compare between original values and inputs.

In regard to confidentiality, once user's information has been collected, it is encrypted and bounded by a specific time which is TL. To disallow any information leakage, the 2DBVA is designed to make sure that such a decryption attempt needs time more than the TL; i.e., *TL<time of a decryption attempt*. Additionally, UAH, for example, is considered to increase the information system privacy whereas UAH is a personal history and has private values by which it is difficult to track the user.

As for authentication, by updating user information frequently and periodically, the 2DB is generated; every 24 hours in order to authenticate both user and access process. All related values, e.g., SQ are frequently updated in advance in order to authenticate the system's database. That is, with specific interval of time, the system is given a new 2DB to reduce vulnerability in the system design. During the next 24-hours, a new updated 2DB is generated again using different inputs, i.e., SQs to guarantee more security.

Finally, the availability of the 2DB and responsiveness is addressed. The policy of offline database storage is considered and connected to the 2DBVA. Every encrypted value is stored and updated accordingly. The 2DB is generated using updated values re-called from the offline database to guarantee an authorized access to database by other verification procedures anytime and also to prevent an unauthorized access caused by attacks.

V. Performance Analysis and Evaluation

A. Oerview

The 2DBVA is evaluated and compared to some of state-of-art algorithms in terms of computation time. Additionally, security factors are evaluated. MLSA-2DBVA security objectives are evaluated (authentication is an example). Finally, MLSA-2DBVA accuracy is evaluated.

B. Security Factor based Analysis

1) Confidentiality

The 2DB is periodically re-generated using a temporary secret key to increase information confidentiality. The secret key is designed using a very long series of bits to ensure the decryption time being lengthily increased. If the key has successfully decrypted the 2DB, the time needed > TL indicating an expired 2DB.

2) Integrity and Availability

2DB contents are verified in order to ensure the integrity. If there is any mismatch between original values (database) and scanned (2DB), then MLSA-2DBVA has no integrity and therefore the verification process is break. Thus, a third party has modified 2DB contents or it might be the original 2DB is expired. Hence, 2DB and other contents are protected by refusing any attempt to access.

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.

3) Authentication and Robustness

This evaluation performs a vote-based process in order to measure the authority of the 2DB. That is, a several steps of processes are verified to make sure that the 2DB is issued by an authentically original source. Additionally, it measures the percentage of 2DB authentication.

C. MLSA based Security Objectives Evaluation

In this evaluation part, Algorithm 4 is applied on several 2DBs to validate whether they are authenticated or not. If YES, they will be approved to access such an IoT application. Every 2DB is considered for an approval to be used as an access key based on the TL value. Thus, results for which the access is approved are provided in Table I.

TABLE I. APPROVED VERIFIED PROCEDURE STATUSES.

Biometrics	ID	SQ	*vote1*	TL	*vote* (updated)	Status
+20	+20	+20	60	1≤TL≤6 = 24	**84**	Approved
				7≤TL≤12 = 18	**78**	Approved
				13≤TL≤18 = 12	**72**	Approved
				19≤TL≤24 = 6	**66**	Approved
+20	+20	+20	60	TL>24 = -1	**59**	Disapproved

In this table, every 2DB is considered for approval to be used as an access key to an IoT application based on the TL value. There are five cases four of which are considered as ‘approved’ status and one of which is disapproved.

On the other hand, there are however, a number of cases in which the authentication status is considered disapproved. These are provided in Table II, Table III, and Table IV.

TABLE II. DIS-APPROVED VERIFIED PROCEDURE STATUSES; WITH ONE REJECTED STEP.

Biometrics	ID	SQ	*vote1*	TL	*vote* (updated)	Status
+20	+20	0	40	1≤TL≤6 = 24	**64**	Disapproved
				7≤TL≤12 = 18	**58**	Disapproved
				13≤TL≤18 = 12	**52**	Disapproved
				19≤TL≤24 = 6	**46**	Disapproved
+20	+20	0	40	TL>24 = -1	**39**	Disapproved

In this type of evaluation, Eq. (1) is used in order to calculate *vote1*:

$$vote1 = Biometrics + ID + S \qquad (1)$$

In Table II, there is one ‘rejected’ case by the 2DB Verification Procedure which is the SQ Verification Step. The reason it is rejected is that its verified value =0 as shown in Table II, the 3rd column. In Table III, there will be two cases that are rejected. Thus, the verification procedure is considered then rejected and the request made by the user is refused. Thus, IoT application is not accessed.

TABLE III. DIS-APPROVED VERIFIED PROCEDURE STATUSES; WITH TWO REJECTED STEPS.

Biometrics	ID	SQ	*vote1*	TL	*vote* (updated)	Status
+20	0	0	20	1≤TL≤6 = 24	**44**	Disapproved
				7≤TL≤12 = 18	**38**	Disapproved
				13≤TL≤18 = 12	**32**	Disapproved
				19≤TL≤24 = 6	**26**	Disapproved
+20	0	0	20	TL>24 = -1	**19**	Disapproved

In this Table, usually Eq. (2) is applied to calculate *vote*:

$$vote = vote1 + TL \qquad (2)$$

Based on Eq. (2), there is a threshold value being applied in order to make a decision whether the related 2DB is considered verified and validated or not, as mathematically represented in Eq. (3):

$$F_{security}\left(2DBVA\right) = \begin{cases} Approved & vote \geq 66\% \\ Disapproved & vote < 66 \end{cases} \qquad (3)$$

In this design, every 2DB is will be strictly verified using a verification procedure consists of a cascade security scheme. Each 2DB must be fully verified at every step of verification procedures. Then, the 2DB will be assigned a value. This value will be then accumulated in order to achieve a security certain value, denoted by: $F_{security}\left(2DBVA\right)$. Based on this applied equation, some further *‘disapproved’* examples are provided in Table IV.

TABLE IV. DIS-APPROVED VERIFIED PROCEDURE STATUSES; WITH THREE REJECTED STEPS.

Biometrics	ID	SQ	*vote1*	TL	*vote* (updated)	Status
0	0	0	0	1≤TL≤6 = 24	**24**	Disapproved
				7≤TL≤12 = 18	**18**	Disapproved
				13≤TL≤18 = 12	**12**	Disapproved
				19≤TL≤24 = 6	**6**	Disapproved
0	0	0	0	TL>24 = -1	**-1**	Disapproved

It is obviously clear that all ‘*vote*’ values are less than the accepted security value pre-defined by Eq. (3).

Subsequently, the ‘*vote*’ varies based on TL age. That is, ‘*vote’* has an inversely proportional relation with the TL age as shown in Fig. 7.

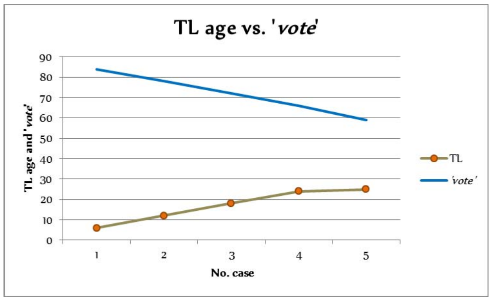

Fig. 7. Relation between TL age and ‘vote’

Fig. 7 illustrates the rule on which the decision is made. That is, when the key age is shorter, its chance to be used as a key for IoT application is higher (TL=6 hours; '*vote*'=84). Thus, '*vote*' value stands for authentication strength.

D. MLSA-2DBVA Accuracy

Here, the MLSA-2DBVA accuracy is measured based on 2DB readability and verified values e.g., Serial_ID and SQ. this is provided in Table V.

TABLE V. MLSA-2DBVA ACCURACY.

Group (#2DB)	***2DB readability***	**Accuracy %**
G1=30	30/30	100 %
G2=40	39/40	97.5 %
G3=45	45/45	100 %
G4=51	51/51	100%
Total		**99.38 %**

The 2DB readability calculates accuracy using Eq. (4).

$$2DB\ readability = \frac{No.\ 2DB\ Generated}{No.\ 2DB\ Scanned} \tag{4}$$

The average accuracy of the proposed 2DBVA to successfully read the 2DB recognize efficiently contents of 2DB equals to 99.38 %.

E. 2DBVA Computation Time

The proposed 2DBVA is compared in terms of computation time to an existing algorithm. The 2DBVA computation time for encryption 2DB is 400 Millisecond for 166 2DBs whereas it is less than the accumulative computation time of [18] is about 500 Millisecond.

VI. CONCLUSION

A 2DBVA has been proposed to authenticate a 2DB. The 2DBVA will assign a value to measure the expiry time of 2DB tag. This aims to authenticate the 2DB that is used as a smart key to access an IoT application. The 2DBVA design has adopted a Multiple-Layer Security Architecture (MLSA) in order to evaluate its overall performance in terms of security factors. The MLSA-2DBVA accuracy for 166 2DBs has been also evaluated in terms of key's validity for IoT applications. Future studies should consider different types of 2DBs with much details of plaintext as well as the computation time needs to be enhanced.

ACKNOWLEDGEMENT

This research work is funded by Universiti Tenaga Nasional Internal Research Grant (UNIIG- J510050738).

REFERENCES

[1] S. S. Lin, M. C. Hu, C. H. Lee, and T. Y. Lee, "Efficient QR Code Beautification With High Quality Visual Content," *IEEE Transactions on Multimedia,* vol. 17, pp. 1515-1524, 2015.

[2] J. Z. Gao, L. Prakash, and R. Jagatesan, "Understanding 2D-BarCode Technology and Applications in M-Commerce - Design and Implementation of A 2D Barcode Processing Solution," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, 2007, pp. 49-56.

[3] A. M. Al-Ghaili, H. Kasim, M. Othman, and Z. Hassan, "Security factors based evaluation of verification algorithm for an IoT access system," in *3rd International Conference of Reliable Information and Communication Technology*, 2019, pp. 384-395.

[4] S. Rane, A. Dubey, and T. Parida, "Design of IoT based intelligent parking system using image processing algorithms," in *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, 2017, pp. 1049-1053.

[5] C. Chen, W. Huang, B. Zhou, C. Liu, and W. H. Mow, "PiCode: A New Picture-Embedding 2D Barcode," *IEEE Transactions on Image Processing,* vol. 25, pp. 3444-3458, 2016.

[6] G. S. Spagnolo, L. Cozzella, and M. D. Santis, "New 2D barcode solution based on computer generated holograms: Holographic barcode," in *2012 5th International Symposium on Communications, Control and Signal Processing*, 2012, pp. 1-5.

[7] M. E. V. Melgar, M. C. Q. Farias, F. d. B. Vidal, and A. Zaghetto, "A High Density Colored 2D-Barcode: CQR Code-9," in *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2016, pp. 329-334.

[8] A. M. Al-Ghaili, H. Kasim, M. Othman, and Z. Hassan, "A new encryption scheme method (ESM) using capsulated-layers conception for verified QR-tag for IoT-based smart access systems," in *Intelligent Systems Reference Library* vol. 154, ed, 2019, pp. 77-103.

[9] J. R. Lee, S. Ruan, and C. H. Lin, "VoiceCode: A 2D barcode system for digital audio encoding," in *2016 IEEE 5th Global Conference on Consumer Electronics*, 2016, pp. 1-2.

[10] H.-F. Huang, S.-E. Liu, and H.-F. Chen, "Designing a new mutual authentication scheme based on nonce and smart cards," *Journal of the Chinese Institute of Engineers,* vol. 36, pp. 98-102, 2013/01/01 2013.

[11] C. H. Hung, Y. Y. Fanjiang, K. C. Chung, and C. Y. Kao, "A door lock system with augmented reality technology," in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, 2017, pp. 1-2.

[12] I. Tkachenko, W. Puech, O. Strauss, C. Destruel, and J. Gaudin, "Printed document authentication using two level or code," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2149-2153.

[13] L. Kanaris, A. Kokkinis, G. Fortino, A. Liotta, and S. Stavrou, "Sample Size Determination Algorithm for fingerprint-based indoor localization systems," *Computer Networks,* vol. 101, pp. 169-177, 2016.

[14] K. Ota, T. Kumrai, M. Dong, J. Kishigami, and M. Guo, "Smart Infrastructure Design for Smart Cities," *IT Professional,* vol. 19, pp. 42-49, 2017.

[15] K. J. Kavitha and B. P. Shan, "Implementation of DWM for medical images using IWT and QR code as a watermark," in *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*, 2017, pp. 252-255.

[16] A. M. Al-Ghaili, H. Kasim, F. A. Rahim, Z. A. Ibrahim, M. Othman, and Z. Hassan, "Smart verification algorithm for IoT applications using QR tag," in *Lecture Notes in Electrical Engineering* vol. 481, ed, 2019, pp. 107-116.

[17] T. Kirkham, D. Armstrong, K. Djemame, and M. Jiang, "Risk driven Smart Home resource management using cloud services," *Future Generation Computer Systems,* vol. 38, pp. 13-22, 2014.

[18] Y. G. Kim and M. S. Jun, "A design of user authentication system using QR code identifying method," in *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, 2011, pp. 31-35.

[19] M. Qiu, Z. Chen, J. Niu, Z. Zong, G. Quan, X. Qin, *et al.*, "Data Allocation for Hybrid Memory With Genetic Algorithm," *IEEE Transactions on Emerging Topics in Computing,* vol. 3, pp. 544-555, 2015.