

# Enhancement of Bluetooth Security Authentication Using Hash-Based Message Authentication Code (HMAC) Algorithm

A. S. Diallo<sup>1</sup>, A. Wajdi<sup>1</sup>, R. F. Olanrewaju<sup>1</sup>, and F. Sado<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, IIUM P.O. Box 10, 50728 Kuala Lumpur, MALAYSIA

<sup>2</sup>Department of Mechatronics Engineering, IIUM P.O. Box 10, 50728 Kuala Lumpur, MALAYSIA

E-mail: [alhassanesaliou@yahoo.fr](mailto:alhassanesaliou@yahoo.fr), {[wajdi](mailto:wajdi@iium.edu.my), [frashidah](mailto:frashidah@iium.edu.my)}@iium.edu.my, [abdfsado1@gmail.com](mailto:abdfsado1@gmail.com)

**Abstract**— Recently, Bluetooth technology is widely used by organizations and individuals to provide wireless personal area network (WPAN). This is because the radio frequency (RF) waves can easily penetrate obstacles and can propagate without direct line-of-sight (LoS). These two characteristics have led to replace wired communication by wireless systems. However, there are serious security challenges associated with wireless communication systems because they are easier to eavesdrop, disrupt and jam than the wired systems. Bluetooth technology started with a form of pairing called legacy pairing prior to any communication. However, due to the serious security issues found in the legacy pairing, a secure and simple pairing called SPP was announced with Bluetooth 2.1 and later since 2007. SPP has solved the main security issue which is the weaknesses of the PIN code in the legacy pairing, however it has been found with some vulnerabilities such as eavesdropping and man-in-the-middle (MITM) attacks. Since the discovery of these vulnerabilities, some enhancements have been proposed to the Bluetooth Specification Interest Group (SIG) which is the regulatory body of Bluetooth technology; nevertheless, some proposed enhancements are ineffective or are not yet implemented by Manufacturers. Therefore, an improvement of the security authentication in Bluetooth connection is highly required to overcome the existing drawbacks. This proposed protocol uses Hash-based Message Authentication Code (HMAC) algorithm with Secure Hash Algorithm (SHA-256). The implementation of this proposal is based on the Arduino Integrated Development Environment (IDE) as software and a Bluetooth (BT) Shield connected to an Arduino Uno R3 boards as hardware. The result was verified on a Graphical User Interface (GUI) built in Microsoft Visual Studio 2010 with C sharp as default environment. It has shown that the proposed scheme works perfectly with the used hardware and software. In addition, the protocol thwarts the passive and active eavesdropping attacks which exist during SSP. These attacks are defeated by avoiding the exchange of passwords and public keys in plain text between the Master and the Slave. Therefore, this protocol is expected to be implemented by the SIG to enhance the security in Bluetooth connection.

**Keywords**—Bluetooth security; authentication; secure and simple pairing; legacy pairing; HMAC algorithm

## I. Introduction

A wireless personal area network (WPAN) “Fig. 1,” is a short-distance wireless network specially designed to support

portable and mobile computing devices such as personal computer (PC), personal digital assistants (PDA), cell phones, printers, pagers, storage devices, and a variety of consumer electronic equipments [1]. Bluetooth technology which was developed to replace the existing wire line connections is used in WPAN with short-range interconnectivity. Moreover, Bluetooth radio operates in the license-free and globally available Industrial, Scientific, and Medical (ISM) band at 2.4 GHz [2] using Frequency-Hopping Spread Spectrum (FHSS) and are capable of transmitting voice and data [3]. Bluetooth provides enough bandwidth that enables data exchange between several mobile devices at a rate up to 1 Mbps [1] for version 2.0 (and earlier) and up to 3 Mbps for version 2.1 (and later) [4][5]. Bluetooth standard is designed for downward compatibility which means that the latest versions can support all features available in old versions. In Bluetooth connection, a piconet is a small network created on an ad hoc basis that includes one master device and up to seven slaves while a scatternet is chain of piconets that allows one or more Bluetooth devices to be a slave in one piconet and act as the master for another piconet, simultaneously [4].

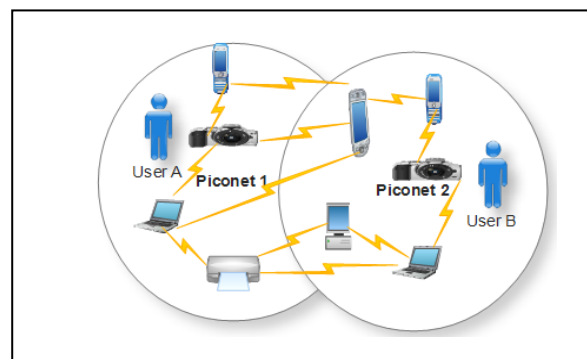


Fig. 1. Introduction to Bluetooth connection

In order for two Bluetooth devices to communicate within a piconet, they need to perform a mutual authentication. During the mutual authentication, pairing is performed in order to establish the connection.

The rest of the paper is organized as follows: section II discusses the existing pairing methods and the corresponding limitations. Section III provides related works on the actual

pairing methods. The proposed authentication scheme is described in section IV followed by an implementation result in section V. Section VI gives the discussion and a conclusion is shown in section VII.

## II. Existing Pairing Methods and their Limitations

### A. Legacy Pairing

The legacy pairing is vulnerable to different security issues such as weak PIN code, passive and active eavesdropping. This method of pairing requires each device to enter a Personal Identity Number (PIN) code in order to perform pairing. Pairing is successful only if both devices enter the same code. In [6], it is stated that many Bluetooth devices today use a 4-digits PIN or a fixed PIN of commonly known values which significantly limit the security of the link key. Therefore, during the pairing procedure there is a very high probability for an attacker to get the used PIN as in [7] [8] [9].

### B. Secure and Simple Pairing (SSP)

In SSP the pairing process is enhanced and became simple and more secure due to the non-use of a fixed PIN code. However, several attacks have been reported recently on it:

**Attacks on the Just Work Model:** A Bluetooth non-input non-out man-in-the middle attack (BT-NINO-MITM) in the just work model was identified in [12] and implemented in [11]. In [13], it is published a novel Bluetooth MITM attack called BT-SPP-Printer-MITM attack against the just work model. Besides that, an attack called Bluetooth-Secure and Simple Pairing- Headset/Hands-Free-Man in the middle attack (BT-SSP-HS/HF-MITM) was proposed in [14]. To perform the attack, the authors exploited the fact that most Headset/Hands-Free Bluetooth devices can be forced to choose the less secure just work model[12][13].

**Attacks on the Passkey Entry Model:** The possibility of successful eavesdropping and MITM attack on the passkey entry model has been mentioned in [10] and the implementation in the GNU radio software framework using the universal software radio peripheral (USRP) as hardware can be found in [5].

**Attacks on the Out of Band Model:** In [14], it is published a Bluetooth-Secure and Simple Pairing- Out of Band-Man in the middle attack (BT-SSP-OOB-MITM) and it is shown that the attack can be performed if the attacker succeeds to have visual contact to the legitimate user's device. The OOB model was suggested to be used as a mandatory model as in [12] [15]; nevertheless, in [11] it is also mentioned that this proposal cannot work.

**Case of the Numerical Comparison Model:** This model is not directly attacked; however attackers can force legitimate users to select a less secure model instead of this secure model. For this reason, in [15] it is mentioned that the numerical comparison model is also found to be not secure.

## III. Related Works

### A. Legacy Pairing

In [16], it was suggested an enhancement of Bluetooth authentication using the concatenation of a master's Clock and a Low Address Part (LAP) to be XORed with the least 42 bits of the Authentication Random Number (AU\_RAND) before being fed into the  $E_1$  algorithm, where the signed response (SRES) is computed. However, this improvement has its drawbacks since it relies on a symmetric key which is not securely shared.

Moreover, in [17] it was designed an improved authentication algorithm using the concatenation of a clock and a part of address values (PAV) to compute the authentication random number:  $AU\_RAND' = f(AU\_RAND, Clock, PAV)$ . However, AU\_RAND which is a public parameter does not need to be changed because this does not prevent to guess the PIN code. Therefore, this enhancement is ineffective.

Reference [18] explored the weakness of the PIN and proposed to add a parameter called authentication ID (au\_id) which is 128 bits in the generation of the initialization key. This au\_id is shared by using Diffie-Hellman key exchange and makes the PIN more robust:  $PIN' = PIN \cup au\_id$ . However, this current approach remains weak due to the use of the unit key and its non-implementation to assess its performance. The Diffie-Hellman key exchange algorithm used is also prone to the MITM attack.

### B. Secure and Simple Pairing

Reference [19] developed an improved authentication algorithm by using SSP which employs Elliptic Curve Cryptography (ECC) that is an analog of Diffie-Hellman Key Exchange. However, one of the weakness of the ECC is that if all ECC users agree on a common set of Elliptic Curve (EC) parameters, to negotiate these parameters, the additional information needed to specify the exact EC might make the effective EC key size to become very large. Another drawback of the ECC is that it increases the size of the encrypted message more than the Rivest-Shamir-Adleman (RSA) encryption. It is also mathematically subtle and more difficult to implement than the RSA.

In [13] it is proposed to add a message saying "The second message has no display and keyboard! Is this true?" in the just work model to solve the BT-NINO-MITM attack. After displaying the message, the user may choose "Proceed" or "Stop". However, it is shown that this proposal does not solve the attack as in [11].

In the view of the above, the authentication procedure in Bluetooth connection needs to be improved.

## IV. Proposed Authentication Method

The proposed model employs a dual authentication which is an authentication concept that requires two verifications prior to establishing any communication.

### A. Description

First of all, a master device is nominated and all Bluetooth devices in a piconet are registered into the database of the master device by assigning a password and a public key to each device such that the password and the public key match the identity (ID) of the device as in "Table I.". This process of registration and updating the database is executed by the administrator of the WPAN. "Table II." describes all involved security entities in the proposed model.

TABLE I. DATABASE OF THE MASTER DEVICE

| No | Identities      | Passwords        | Public Keys     |
|----|-----------------|------------------|-----------------|
| 1  | ID <sub>A</sub> | Pwd <sub>A</sub> | KU <sub>A</sub> |
| 2  | ID <sub>B</sub> | Pwd <sub>B</sub> | KU <sub>B</sub> |
| 3  | ID <sub>C</sub> | Pwd <sub>C</sub> | KU <sub>C</sub> |
| 4  | ID <sub>D</sub> | Pwd <sub>D</sub> | KU <sub>D</sub> |
| 5  | ID <sub>E</sub> | Pwd <sub>E</sub> | KU <sub>E</sub> |
| 6  | ID <sub>F</sub> | Pwd <sub>F</sub> | KU <sub>F</sub> |
| 7  | ID <sub>G</sub> | Pwd <sub>G</sub> | KU <sub>G</sub> |

TABLE II. DEFINITION OF INVOLVED PARAMETERS AND SYMBOLS

| No | Parameters                         | Description                                       | Size     | Status  |
|----|------------------------------------|---|----------|---------|
| 1  | ID = BD_ADDR                       | Identity = Bluetooth device address               | 48 bits  | public  |
| 2  | Pwd <sub>A</sub>                   | Password of slave A                               | 128 bits | private |
| 3  | Kc                                 | Secret key derived from Pwd <sub>A</sub>          | 128 bits | private |
| 4  | IV                                 | Initial Value                                     | 128 bits | private |
| 5  | KU <sub>A</sub> or KU <sub>a</sub> | Slave public key                                  | 128 bits | public  |
| 6  | KR <sub>A</sub> or KR <sub>a</sub> | Slave private key                                 | 128 bits | private |
| 7  | Ks                                 | Session Key for AES                               | 128 bits | private |
| 8  | K <sup>+</sup>                     | HMAC secret key                                   | 256 bits | private |
| 9  | M                                  | HMAC Authentication message                       | 512 bits | private |
| 10 | HMAC                               | Authentication Algorithm HMAC                     |          |         |
| 11 | C <sub>A</sub> & C <sub>m</sub>    | Slave and master commitments values               | 256 bits | public  |
| 12 | , E, D                             | Concatenation, encryption, and decryption Symbols |          |         |

Secondly, we list all initial parameters possessed by both devices:

Slave A: (ID<sub>A</sub>, Pwd<sub>A</sub>, KU<sub>A</sub>, KR<sub>A</sub>).

Master: (ID<sub>A</sub>... ID<sub>G</sub>, Pwd<sub>A</sub>... Pwd<sub>G</sub>, KU<sub>A</sub>...KU<sub>G</sub>).

### B. Different Phases of the Proposed Authentication Scheme

1) *First Authentication Stage (Phase 1)*: This first phase consists of three messages between the master and the slave. It will result to a first verification called first authentication stage or handshaking.

*Message 1*: A slave which would like to establish a secure communication with the master device sends its ID to the master.

*Message 2*: The master receives the ID and checks its database to see whether the received ID exists in the data base or not. If it exists, the master will derive a secret key (Kc) from the corresponding password of the current ID. However, if the ID is not registered previously in the database, it means that none of the seven devices of the piconet has sent its ID. Therefore, the master will ignore the sent ID. Assuming that the ID exists in the database, the master generates randomly a session key (Ks) and derive a secret key (Kc) from the stored password. The derivation of (Kc) is executed as follows:

- If the password length is less than 16 bytes, zero padding is applied to the left most significant bits in order to get a key size of 16 bytes.
- If the password length is exactly 16 bytes, it is used directly as a key without any modification.
- If the password length is greater than 16 bytes, Fanfold operation is applied to get 16 bytes.

A double encryption of the MAC address of the slave (ID<sub>A</sub>) will be executed using the AES encryption with Cipher Block Chaining (CBC) mode which is a recommended mode due to its security. The first encryption is done by using the session key (Ks) and the second encryption is done by using the derived secret key (Kc). The master will send to the slave the double encryption (Cipher 2) with the concatenation of the encryption of the initial value (IV) "(1)".

$$E_{Kc} [E_{Ks} (ID_A)] || E_{KU_a} (IV) = E_{Kc} [Cipher 1] || E_{KU_a} (IV) = Cipher 2 || E_{KU_a} (IV) \quad (1)$$

The purpose of the double encryption is to identify the slave by asking the user to recover cipher 1 as well as to avoid the following attack: assuming that an attacker has eavesdropped on the wireless connection and captured the cipher sent in Message 2, he cannot recover neither cipher 1 nor the IV because only the legitimate user knows the password from which Kc is derived and the private key to decrypt the encrypted IV. Upon receiving cipher 2 with the concatenation of the encrypted (IV), the user recovers the IV by using his private key as in "(2)".

$$D_{KR_a} (IV) = IV \quad (2)$$

After that, the system prompts him to enter his password and derive the same secret key Kc in order to decrypt cipher 2 as in "(3)".

$$D_{Kc} (Cipher 2) = D_{Kc} \{E_{Kc} [E_{Ks} (ID_A)]\} = Cipher 1 \quad (3)$$

If cipher 1 is successfully recovered, it means that the correct password is supplied. Failure to provide the correct password will result to abort the connection.

*Message 3:* The user returns to the master cipher 1 together with his device ID as in “(4)”. Here the master will decrypt cipher 1 by using (Ks) and get the encrypted ID. After that, the master will compare the original ID and the sent ID in Message 3. If they match, it means that Message 2 was not altered and Message 3 is sent by the exact slave. Otherwise, the connection will be aborted. This is the end of the first authentication stage.

$$ID_A \parallel \text{Cipher 1} = ID_A \parallel E_{K_s}(ID_A) \quad (4)$$

2) *Exchange of Secret Key (Phase 2):* This phase consists of one message.

*Message 1:* For the purpose of security, it is recommended to generate a new secret key instead of re-using the same Ks. Thus, the master generates a new session key noted by  $K^+$  to compute the HMAC algorithm. The master will transfer the concatenation of  $K^+$  and a message M by using RSA key exchange which is a secure method to transfer the session key as in “(5)”. With RSA key exchange, secret keys are exchanged securely by encrypting them with the public key of the intended recipient. Only the recipient can decrypt the encrypted key because it requires using his own private key. Indeed, a third party who intercepts the encrypted secret key. Thus, secrecy and privacy of the session key is well obtained. Furthermore, the integrity of the encrypted key is accomplished since there is no way to tamper the transferred key. RSA algorithm is demonstrated to be reliable with high quality, guaranteed security and strong encryption. The selection of RSA key exchange is also motivated by its simplicity on hardware implementation.

$$E_{K_{Ua}}[K^+ \parallel M] = \text{Cipher 3} \quad (5)$$

The slave decrypts cipher 3 using the private key as in “(6)”:

$$D_{K_{Ra}}[K^+ \parallel M] = K^+ \parallel M \quad (6)$$

3) *Second Authentication Stage (Phase 3):* This phase also consists of one message.

*Message 1:* The slave computes the HMAC algorithm to get a commitment value ( $C_A$ ) to be sent back to the master as in “(7)”. Meanwhile the master computes  $C_m$  as in “(7)”. The master device compares  $C_A$  and  $C_m$ . If the two values match, it means that both the session key ( $K^+$ ) and the message (M) are not altered. However, failure to that requires aborting the connection. HMAC is used as the authentication algorithm because it can verify data integrity and authentication simultaneously. HMAC is employed with the hash function SHA256 thus the name HMAC-SHA256.

$$C_A = C_m = \text{HMAC}[K^+ \parallel M] \quad (7)$$

Upon receiving  $C_A$ , the Master compares  $C_A$  to  $C_m$  and decides whether authentication is successful or not.

4) *Exchange of Encrypted Data (Phase 4):* This last phase consists of many messages depending on the data to share. Devices can exchange data securely by using a new generate secret key for encryption. The communication can be ended by either of the devices. However, in this case the slave will end the connection when it will finish exchanging data because it initiated the connection for a specific purpose. It should be noted that the session key is a temporary key; hence for the purpose of security it needs to be changed periodically. “Table III.” summarizes the phases of this proposal and “Fig. 2,” represents the block diagram of the model (page 6).

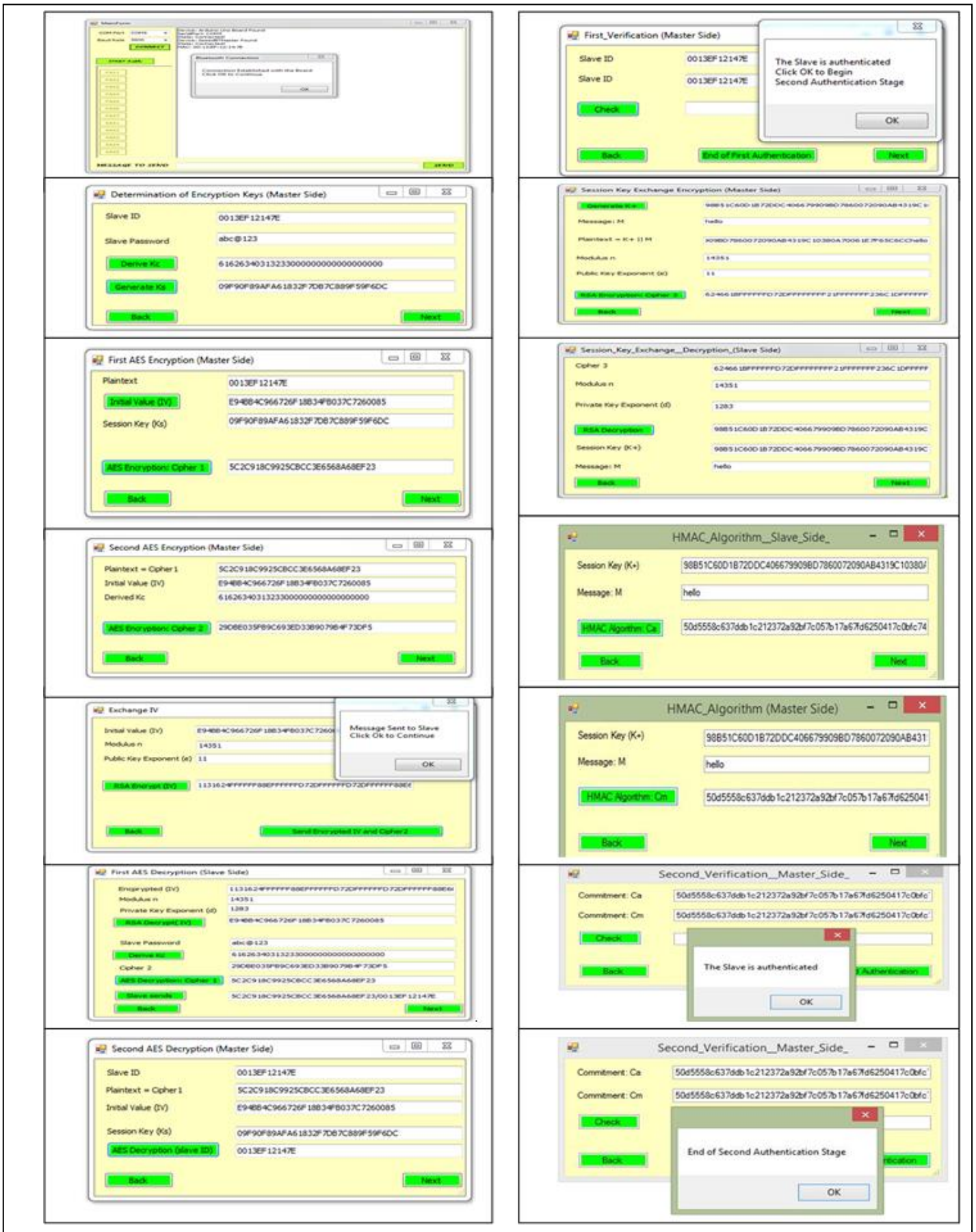
TABLE III. SUMMARY STEPS OF THE PROPOSED MODEL

| PHASES                      | MESSAGES   | BT Devices                          |
|-----------------------------|--|-------------------------------------|
| First Authentication Stage  | $M_1: ID_A$  | Slave to Master                     |
|                             | $M_2: E_{K_c}[E_{K_s}(ID_A)] \parallel E_{K_{Ua}}(IV)$             | Master to Slave                     |
|                             | $M_3: ID_A \parallel E_{K_s}(ID_A)$                                | Slave to Master                     |
| Exchange of Secrete Key     | $M_1: E_{K_{Ua}}[K^+ \parallel M]$                                 | Master to Slave                     |
| Second Authentication Stage | $M_1: C_A = \text{HMAC}[K^+ \parallel M]$                          | Slave to Master                     |
|                             | Meanwhile Master computes:<br>$C_m = \text{HMAC}[K^+ \parallel M]$ |                                     |
|                             | Master compares $C_A$ to $C_m$ and makes decision                  |                                     |
| Exchange of Encrypted Data  | If authentication is successful devices exchange data securely.    | Slave to Master and Master to Slave |

## v. Implementation Results

Security consideration: For a correct and good implementation, the following security concerns need to be taken into consideration: Generation of secret random keys by using a strong random number generator (RNG); Strength of the keys; Changing the keys periodically; Secure protection of keys; Secure key exchange mechanism; Correctness of the used algorithm.

For the implementation on the mentioned hardware and software, the following three libraries have been imported: AES-CBC library, RSA library and HMAC-SHA256 library. The Arduino board and the Bluetooth shield were connected together prior to connecting the board to a laptop running Arduino IDE via a USB A to B cable. The code was developed on the Arduino IDE then linked to a GUI created in Visual studio C sharp environment. “Fig. 3,” shows the implementation result on a GUI.



“Fig. 3” Implementation result on a GUI.



## VI. Discussion

“Fig.3,” shows the workability of the proposal using the mentioned hardware and software. All operations are computed and shown in different windows of the GUI. The result has shown that the security issues in SSP which are passive and active eavesdropping are completely tackled by preventing the exchange of public keys and passwords in clear text. Involved secret passwords and public keys are locally stored and the secret parameters (IV and secret keys) are generated randomly by using a strong random generator number then securely exchanged with the RSA algorithm. Other exchanged ciphers cannot be tampered by an attacker due to the non-availability of the encryption keys to a third user and also the strength of the used encryption systems. Moreover, strong and long passwords are supported to generate  $K_c$  which is used for AES encryption. Passwords are kept secret and refreshed periodically. Therefore, it is highly expected that this proposal will replace for authentication.

## VII. Conclusion

This paper started by given an overview of Bluetooth then discussed the weaknesses in existing authentication methods. Besides that, some improved works have been reviewed prior to outlining the proposed solution for the current security issues in Bluetooth connection. The model was validated by using a BT shield connected to an Arduino board and the obtained result was observed with conformity in a GUI. The entire process of the authentication was clearly shown in snipped windows. The proposal defeats eavesdropping and MITM attacks which exist in SSP and is therefore seen to be the third authentication method in Bluetooth.

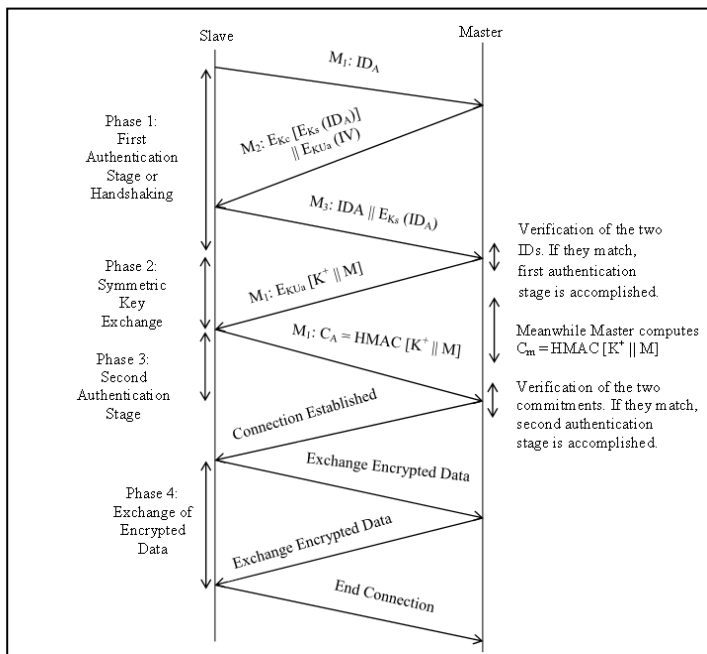


Fig. 2. Block diagram of the model

## References

- [1] V. Garg, *Wireless Communications & Networking: Morgan Kaufmann*, 2010, pp. 653-674.
- [2] C. Gehrmann, J. Persson, and B. Smeets, *Bluetooth security: Artech house*, 2004.
- [3] K. Saravanan and D. Yuvaraj, "An new secure mechanism for bluetooth network," in *Computer and Automation Engineering (ICCAE)*, 2010 The 2nd International Conference on, 2010, pp. 202-205.
- [4] J. Padgett, K. Scarfone, and L. Chen, "Guide to Bluetooth security," *NIST Special Publication*, vol. 800, p. 121, 2012.
- [5] J. Barnickel, J. Wang, and U. Meyer, "Implementing an attack on bluetooth 2.1+ secure simple pairing in passkey entry mode," in *Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012 IEEE 11th International Conference on, 2012, pp. 17-24.
- [6] J. Linksky, *Simple Pairing Whitepaper, revision v10r00. Bluetooth Special Interest Group (SIG). Core Specification Working Group*. 2006, pp.1-23.
- [7] Y. Shaked and A. Wool, "Cracking the bluetooth pin," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 39-50.
- [8] M. T. Alberto, "Sniffing the Bluetooth pairing". 2011. Accessed on 15/03/2013 at 3:30 pm. <<http://www.seguridadmobile.com/bluetooth/bluetooth-security/sniffing-the-Bluetooth-pairing.html>>.
- [9] M. Colin "iOS passwords: Quick tips to maximize your security" (2012). Accessed on 15/03/2013 at 8:00 am. <<http://www.fichnetsecurity.com/6labs/blog/ios-passwords-quick-tips-maximise-your-security>>.
- [10] A. Y. Lindell, "Attacks on the pairing protocol of bluetooth v2. 1," *Black Hat USA, Las Vegas, Nevada*, 2008.
- [11] A. Iman, Al S., Mohammed, Al A Mousa, "Secure public key exchange against man-in-the middle attack during secure simple pairing (SSP) in Bluetooth". *World Applied Sciences Journal* 13 (4): 769-780, ISSN 1818-4952. Computer Science Department Computer Information System Department King Abdullah II School for IT. The University of Jordan. 2011, pp.1-12.
- [12] K. Hypponen and K. M. Haataja, "'Nino" man-in-the-middle attack on bluetooth secure simple pairing," in *Internet, ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on*, 2007, pp. 1-5.
- [13] K. M. Haataja and K. Hypponen, "Man-in-the-middle attacks on bluetooth: a comparative analysis, a novel attack, and countermeasures," in *Communications, Control and Signal Processing*, 2008. ISCCSP 2008. 3rd International Symposium on, 2008, pp. 1096-1102.
- [14] K. Haataja and P. Toivanen, "Practical man-in-the-middle attacks against bluetooth secure simple pairing," in *Wireless Communications, Networking and Mobile Computing*, 2008. WiCOM'08. 4th International Conference on, 2008, pp. 1-5.
- [15] D. Sharmila, R. Neelaveni, and K. Kiruba, "Bluetooth man-in-the-middle attack based on secure simple pairing using out of band association model," in *Control, Automation, Communication and Energy Conservation*, 2009. INCACEC 2009. 2009 International Conference on, 2009, pp. 1-6.
- [16] A. Levi, E. Çetintaş, M. Aydos, Ç. K. Koç, and M. U. Çağlayan, "Relay attacks on bluetooth authentication and solutions," in *Computer and Information Sciences-ISCIS 2004*, ed: Springer, 2004, pp. 278-288.
- [17] P. R. Suri and S. Rani, "Bluetooth security-Need to increase the efficiency in pairing," in *Southeastcon*, 2008. IEEE, 2008, pp. 607-609.
- [18] T. Kumar, "Improving pairing mechanism in Bluetooth security," *Int. J. Recent Trends Eng.*, vol. 2, 2009.
- [19] C.-M. Fan, S. Shieh, and B.-H. Li, "On the security of password-based pairing protocol in bluetooth," in *Network Operations and Management Symposium (APNOMS)*, 2011 13th Asia-Pacific, 2011, pp. 1-4.