Intrusion Detection in Bluetooth Enabled Mobile Phones

Kishor Krishnan Nair[*,***] Albert Helberg[**] Johan Van Der Merwe[*]

[*] Council for Scientific and Industrial Research (CSIR), Modelling and Digital Science (MDS), Pretoria, South Africa

[**] Faculty of Engineering, Telkom Centre of Excellence, North-West University, Potchefstroom Campus, South Africa

[***] Corresponding Author: knair@csir.co.za

Abstract— **Bluetooth plays a major role in expanding global spread of wireless technology. This predominantly happens through Bluetooth enabled mobile phones, which cover almost 60% of the Bluetooth market. Although Bluetooth mobile phones are equipped with built-in security modes and policies, intruders compromise mobile phones through existing security vulnerabilities and limitations. Information stored in mobile phones, whether it is personal or corporate, is significant to mobile phone users. Hence, the need to protect information, as well as alert mobile phone users of their incoming connections, is vital. An additional security mechanism was therefore conceptualized, at the mobile phone's user level, which is essential in improving the security. Bluetooth Logging Agent (BLA) is a mechanism that has been developed for this purpose. It alleviates the current security issues by making the users aware of their incoming Bluetooth connections and gives them an option to either accept or reject these connections. Besides this, the intrusion detection and verification module uses databases and rules to authenticate and verify all connections. BLA when compared to the existing security solutions is novel and unique in that it is equipped with a Bluetooth message logging module. This logging module reduces the security risks by monitoring the Bluetooth communication between the mobile phone and the remote device.**

Keywords— BD_ADDRESS, BLA, Bluetooth, Mobile phones, IDS, MPUI, Pairing, PIN, Vulnerability

I. INTRODUCTION

Bluetooth wireless technology connects electronic devices and enables seamless data and voice communication over short range radio connections. The key traits of this technology are its low cost and low power consumption. Bluetooth technology has gained global acceptance mainly through its diverse applications in mobile phones. However, with its popularity in mobile phones, intruders are inclined to find new ways to compromise it [1]. This has created a need to tighten the security in Bluetooth mobile phones. The Bluetooth Special Interest Group (SIG) has categorized different security modes and levels in its security specification [1]. Although this is the case, in reality, mobile phone manufacturers are the ones who decide the level of Bluetooth security required and adheres to their proprietary security implementations [2]. In majority of the cases, Bluetooth security requirements stipulated by the SIG are not met, which ultimately opens the door to serious security vulnerabilities [2].

Even though Bluetooth mobile phones have gained global acceptance and popularity, it also has a significant number of weaknesses. The primary weakness in particular is the crime committed through it which allows criminals to steal valuable information ranging from address books to significant corporate data, eavesdropping in confidential conversations and accessing mobile phone commands surreptitiously [3]. The vulnerabilities also allow intruders to use the attacked phone not only to make calls, but also to get access to Internet. Security vulnerabilities have disastrous consequences to the phone owners. They are often heavily charged and even prosecuted for the criminal's anonymous activities. Intruders today are also increasingly targeting Bluetooth mobile phones as a means for propagating viruses and Trojans into corporate networks. Eighty two percent of businesses worldwide agree that they consider the damage from virus attacks the same or greater on a mobile network than on a fixed network [4].

The inefficient implementation of the Bluetooth protocol stack in mobile phones is considered as the primary cause of the existing security issues [5]. The greater part of Bluetooth mobile phones in the market are subject to one or other implementation issue. The Bluetooth OBject Exchange (OBEX) Protocol and Radio Frequency Communication (RFCOMM) services are prone to severe performance issues [6]. Further the Bluetooth security architecture specification includes a shared Master key for all Bluetooth devices in the Bluetooth networks which is a major security vulnerability [7].

To a certain extent, many intrusions are a result of the lack of Bluetooth knowledge among mobile phone users. A large number of these intrusions are preventable if users turn Bluetooth to a non-discoverable mode when they find themselves in unknown locations. The same should apply when users are not using the Bluetooth feature. The users can also increase the security of their devices and prevent critical information leakage by not connecting with unknown devices. However, it is not practical to develop Bluetooth awareness among millions of mobile users before they start using Bluetooth features [8]. Mobile phones should be equipped with a feature to identify the authenticity of the incoming connections, the possible intrusions and to give alerts to user as and when it occurs. This paper focuses on the above security aspects and develops a Bluetooth security framework known as Bluetooth Logging Agent (BLA) in its aim to address the Bluetooth security issues.

The paper is structured as follows: Section II discusses the state-of-the-art of the current Bluetooth security architecture, its vulnerabilities and limitations. Section III discusses the existing Bluetooth security solutions. Section IV conceptualizes the proposed BLA framework and discusses its various aspects. Section V reviews the proposed framework. Section VI concludes the paper and Section VII looks into the scope for future work.

II. CURRENT BLUETOOTH SECURITY ARCHITECTURE, VULNERABILITIES AND LIMITATIONS

The current Bluetooth security architecture is illustrated in figure 1.

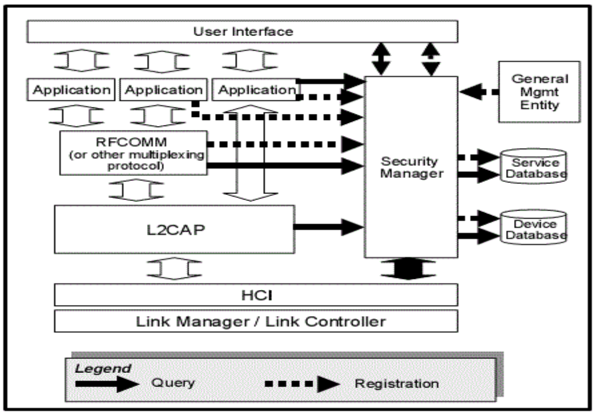


Figure 1. Current Bluetooth security architecture [2]

The key component in the architecture is the security manager and its functionalities are as follows:

- Store security related information in the device database and service database.
- Grant or deny access requests.
- Impose encryption and authentication before a connection attempt.
- Establish a trusted relationship with a device.
- Start pairing process and obtain PIN from a remote device or an application.

Bluetooth is set with its security mechanism both at application level and link level to offer usage protection and information confidentiality. At the heart of Bluetooth's security features is a secret link key. When two devices communicate for the first time, a pairing procedure generates this key and is shared by the devices. All Bluetooth devices have a unique individual address called Bluetooth Device Address (BD_ADDRESS). This allows each device to identify other devices connecting to it. The initialization process in Bluetooth devices uses a PIN which is also known as 'Bluetooth passkey' [9]. According to the Bluetooth specification, Bluetooth devices mainly have three security modes and are as follows [2].

Security Mode 1 - Non secure mode: This is the most insecure mode and is mainly used by applications that do not need any security. The Bluetooth device in this mode is in a promiscuous state or in a discovery mode and it permits other devices to connect with it.

Security Mode 2 - Service level enforced security mode: In this mode, security is imposed in devices after a data link level connection or a Logical Link Control and Adaptation (L2CAP) connection.

Security Mode 3 - Link level enforced security mode: It enforces that devices commence security procedures before setting up an L2CAP connection. This security procedure is the inbuilt security mechanism in the device and it also supports encryption and authentication using a secret link key.

Apart from the above security modes, the security architecture also mandates security levels for services and devices as follows.

Authentication and Authorization: Authentication is the process of identifying a device before establishing a connection and it takes place via the secret link key or by pairing [10]. Authorization is the process of validating whether a particular device has access to a service in another device. Devices that are allowed access are known as trusted devices, and will be indicated as trusted. Unknown devices will have to first acquire authorization from the user before accessing a service [11].

Bonding and Pairing: Bonding refers to the step-by-step procedure to create a relationship based on a common link key [9]. For the period of bonding, the link managers verify that they share a secret key through authentication. After authentication, the link managers create and exchange a link key. The link level procedures of authentication and link key generation are collectively called pairing.

The Bluetooth security vulnerabilities are classified into fundamental vulnerabilities and implementation induced vulnerabilities and they are explained as follows [9].

Fundamental Vulnerabilities: Bluetooth is no different from all the broadcast technologies that disclose some information. Despite Bluetooth's use of the encryption scheme to hide the contents of its message traffic, it is still vulnerable to traffic analysis [12]. A single connection may not have much sensitive information, but the possibility of sensitive information in a large number of connections cannot be ignored. Although Bluetooth devices have a limited range, sensitive Bluetooth receivers can span across a greater distance than the operational distance. Bluetooth is exposed to Denial of Service (DoS) attack [12]. In order to form ad hoc connections with other devices in its proximity, a Bluetooth device continually broadcasts its identity and presence. This broadcast contains information such as the BD_ADDRESS, device name and other sensitive information such as the International Mobile Equipment Identity (IMEI). An intruder can therefore easily make use of this security vulnerability to track and monitor the device. The next section explains implementation induced vulnerabilities.

Implementation Induced Vulnerabilities: Majority of Bluetooth security implementations rarely achieve the security stipulated in Bluetooth security specifications. This is partly due to the difficulty in implementing security specification and partly due to implementation issues [12]. Security specifications mandate that the implementations include (a) random inputs to key generation routines and (b) initialization vectors for performing encryption. These values are either selected randomly or from available sample space. If this is not the case, security implementation may be less secure than what the sample space might suggest. Majority of the devices do not comply with this requirement, and hence they are vulnerable [13]. When Bluetooth applications are implemented, developers are prone to include extra generality and flexibility and intruders exploit this for their own benefits [14].

Bluetooth security PINs are of 1 to 16 octets or bytes (8 bits to 128 bits) in length, depending on the degree of security required in applications. Although security specifications emphasizes the use of long PIN codes, many Bluetooth security implementations use PIN lengths only up to 4 octets. Bluetooth devices that do not have any user interface for entering PIN codes such as headsets, car-kits and modems have fixed pre-defined PIN codes that are values like '0000' or '1111'. These PIN codes can be easily guessed and hence these devices are indeed vulnerable [12, 13]. Based on the security specification, a Bluetooth device that uses a unit key[1] for authentication and encryption can only use a single unit key for all its secure connections at any given time and it opens the door for possible intrusions. In a unit key scheme, all trusted devices participating in group communication know the unit key. Hence, any device can eavesdrop on any communication between the devices in a group. In a group communication, a master device distribute unit key to all the devices. Due to this security vulnerability, any trusted device can compromise any other trusted device, even the master device. Bluetooth security implementations have many crucial limitations and hence are not at all satisfactory [15]. The following scenarios demonstrate the limitations of Bluetooth security architecture.

Scenario 1: Two Bluetooth mobile phones communicate with each other to carry out a task such as file synchronization [2].

Scenario 2: Two or more Bluetooth phones communicate with each other to carry out tasks for which security is not mandatory, such as exchanging business cards [16].

Scenario 3: This would be in scenarios such as a mobile phone requiring access to connect to a banking environment through a Bluetooth link. It will first connect to a Local Area Network (LAN) Access Point. The LAN Access Point will connect to the banking services via a wired or wireless LAN [16].

Based on the three scenarios above, the following security architecture limitations emerge:

➢ Support for 'legacy applications'[2]: In all scenarios, the legacy application will not communicate with security manager since there is no support identified for them in Bluetooth. Therefore, a Bluetooth specific security application must be implemented to set up security procedures with the security manager on behalf of legacy application [2].
➢ It is not the user but only the device that is authenticated. If user needs to be authenticated, other security features will be necessary [15].
➢ Scenario 1 defines no mechanism to cater for separate authorization of each service. A more flexible security policy should be put into practice for this architecture [16].
➢ Bluetooth security architecture does not permit one direction flow on the L2CAP channel. Therefore the security implementations should also cater for options to enforce unidirectional traffic. Such enforcement should occur at application level [15].
➢ The Bluetooth security architecture is based on the Bluetooth baseband security procedure. Hence it only deals with Bluetooth link security and device authentication. To ensure an end-to-end security as in Scenario 3, the security architecture requires an end-to end solution. Since Bluetooth devices access services such as those in Scenario 3, it is important to ascertain that there is appropriate enforcement of security at both ends of the link [2]. If not, several passwords may be required before the link is complete. This would inevitably increase user frustration [16].

Due to the above security vulnerabilities and limitations various companies realized the need for proprietary Bluetooth

[1] A unit key labeled as K_A, is basically a link key which is a 128 bit random number. The unit key is created once at installation of the Bluetooth unit. Thereafter, it is very rarely changed.

[2] When referring to legacy applications and data in information technology, we refer to those that have been inherited from languages, platforms, and techniques earlier than current technology.

security solutions and implemented them. They are laid out in the next section.

III. EXISTING BLUETOOTH SECURITY SOLUTIONS

This section discusses and analyses existing Bluetooth security solutions and they are as follows:

AirDefense BlueWatch™ monitors and identifies all Bluetooth enabled devices and communication between them within a specified range. This product is ideal to be set up in organisations to detect Bluetooth related security threats. In addition, it can take a proactive approach in the prevention of intrusions in a network.

Red-Detect secures wireless network and checks whether an intrusion attempt has occurred in the network. If so, it takes counter measures. It comprises of the following three components: a group of Red-Alert PRO probes, a central server and a Windows management console [17]. The Red-Alert PRO probes capture all the wireless events on a 24X7 basis and stores data in a central server, which then compares the captured data to its internal knowledge base to correlate intrusions.

BlueAuditor is a network auditing tool used to detect and monitor Bluetooth devices in a wireless network. It monitors and displays key information of any device within a distance of 100 meters.

AirMagnet BlueSweep is very analogous to *AirDefence BlueWatch* and *BlueAuditor*. It is a simple, user-friendly, Windows-based utility that can detect and monitor Bluetooth devices in a wireless network environment, within a range of 100 meters.

The above solutions mainly focus in detecting devices within a certain range and demand extensive human intervention to detect intrusions in the network. These Intrusion Detection Systems (IDSs) are available either as a complete hardware or software solution or as a software only solution. IDSs such as *Red-Detect* and *AirDefense* are very expensive and lack critical configuration abilities [18]. Hence, they fail to present important alerts to administrators. Even though they provide greater technical support along with a more user friendly interface for configuration, monitoring, and reporting; one of the biggest disadvantages of these IDSs is their inability to change the antenna [19]. Instead of just changing to a higher gain antenna, there is always the need to buy more sensors to get more coverage [20]. This can result in an increase in the cost of equipment, as well as an increase in the time needed to setup and maintain additional devices.

Further, the current security solutions are not capable of detecting intrusions that are targeted towards a particular device since they are essentially network based IDSs. The biggest shortcoming is that none of the existing Bluetooth IDSs can be deployed in mobile phones as they can only be used in Personal Computers (PCs) or laptops.

A large number of mobile phones from different mobile phone vendors have one or other Bluetooth security limitation [21]. Even though, the vendors are aware of these issues, they are not considering it as critical. As a result, patches or fixes for security issues are not made available regularly [22]. Some of the security issues may be corrected by installing the latest firmware. However, majority of mobile phone owners do not upgrade firmware from time to time. A firmware upgrade is inherently a complicated process in some of the mobile phone models and majority of mobile phone users do not have sufficient knowledge to perform it [23].

Even though numerous file scanners and malware blockers are available in the market for use in mobile phones; they are not particularly suitable for detecting Bluetooth intrusions at the precise moment when they occur. Examples of those in the market include Trend Micro Mobile Security [24], Symantec Mobile Security [25], F-Secure Mobile Security™ [26], McAfee Mobile Security [27], Gold Lock 3G™ [28] and Sophos Mobile Security™ [29].

Many intrusions in mobile phones occur without the user's knowledge. Present mobile phones do not have a solution for users to identify intruders or intrusions that occur while a Bluetooth connection takes place from a remote device. If a mobile user is able to identify an anomalous connection that is taking place, then the user can be safe from many of these attacks. The importance of alerting the user of incoming connections is hereby underlined. The proposed BLA framework will alert the user of any intrusion and make it easy for the user to identify any unidentifiable Bluetooth connection. It is conceptualized in the next section.

IV. CONECPTUALIZATION OF THE BLA FRAMEWORK

The BLA security architecture is illustrated in figure 2.

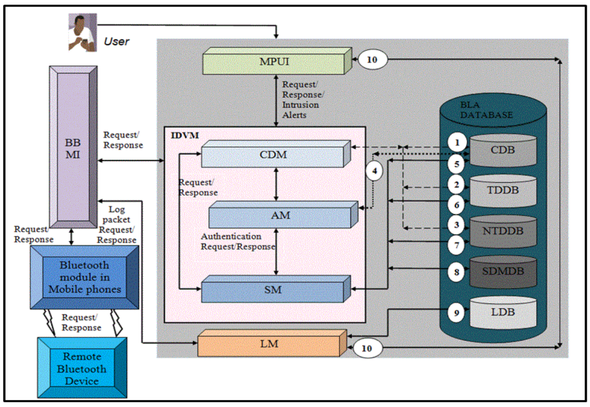


Figure 2. BLA security architecture

Each component of the BLA architecture is analysed in detail in the sections that follow. The numbers in the circles in figure 2 represent the message exchanges between the different modules. The number to message mapping is as follows:

1. Update CDB.
2. CDM/TDDB handshakes. A handshake implies the database update requests and responses.
3. CDM/NTDDB handshakes.
4. AM/CDB handshakes.
5. SM /CDB handshakes.
6. SM/TDDB handshakes.
7. SM/NTDDB handshakes.
8. SM/SDMDB handshakes.
9. LM/LDB handshakes.
10. Communication between the MPUI and LM

The core functionalities that the BLA is intended to perform are as follows:

1. It should work closely and communicate with mobile phone's embedded Bluetooth module stack and profiles such as OBEX, Serial Port, Hands Free and Headset profiles. The Bluetooth Module will indicate to BLA what Bluetooth activities are occurring in the mobile phone.

2. BLA will be presented with a pre-set of trusted devices and their specific parameters. It should only permit a connection from its pool of trusted devices after it has obtained permission from the user. When an incoming request is arriving in the mobile phone, BLA should authenticate the authenticity of the connection with the required parameter checks and via database lookups.

3. BLA should cater for remote Bluetooth device connections, which are not in the trusted device list. These connections should be restrictive, and the mobile phone user should have the freedom to accept or reject such connections.

4. The user should be able to stipulate access rights to mobile phone services in BLA. If the remote device attempts to access a restricted service, BLA will alert the user. The Bluetooth connection with that device will then be terminated. BLA should also alert the user to turn off Bluetooth when it picks up an unauthorised operation.

5. BLA should only accept PIN codes with lengths that are greater than or equal to 8 octets. For the duration of the authentication step, it should not accept any connection that does not meet this criterion.

6. Upon request from the user, BLA should provide the feature of logging all Bluetooth activities.

As illustrated in the architecture of BLA in figure 2, it consists of various modules, where each module fulfills a specific role in intrusion detection. The following sub-sections explain the purpose of each module.

A. Bluetooth Module in Mobile Phones: This is the core module that handles Bluetooth activities and it comprises of Bluetooth hardware, protocol stack, profiles and applications. It sends and receives Bluetooth messages with other Bluetooth devices in its proximity. It communicates with the 'BLA - Bluetooth Module Interface' (BBMI) to deal with the requests from BLA and to alert BBMI with any message from the remote Bluetooth device.

B. BBMI: It is responsible for the communication between BLA and the embedded Bluetooth module in the mobile device.

C. IDVM: The IDVM is the core module of the BLA. It sends and processes messages to and from BBMI, MPUI and BLA database. It consists of three sub-modules, which are described as follows.

Connection and Disconnection Module (CDM): As the name implies, it is responsible for Bluetooth connection and disconnection. When BLA gets a connection request indication, the request is sent to the CDM. At this stage, it becomes the responsibility of the CDM to check if the request is genuine or an intrusion attempt. It performs various steps to ascertain the reliability. As the first step, CDM checks Trusted Devices Database (TDDB) to determine whether the device that is attempting to connect is listed in the database. If the device is indeed listed, CDM will assume that the remote device can be trusted and connection response will be sent. Further, the new connection will be added to the Connection Database (CDB).

If the device that is attempting to connect is not listed in the TDDB, then CDM will check the Non Trusted Devices Database (NTDDB). If the device is listed in the NTDDB, then it will be classified as an intrusion attempt. An alert will then be sent to MPUI. Furthermore, CDM will issue a disconnect response to BBMI. If a device is neither in the TDDB nor in the NTDDB, then the process becomes a bit more complicated. CDM in such situations will have to determine whether the connection is from a valid source or whether it is an intrusion attempt. Under such circumstances, CDM follows certain predefined rules. The list of valid rules includes the following:

1. 'Accept connections from mobile phones only from a list of models of a particular manufacturer'. Models excluded from the list may have some Bluetooth security vulnerability and for this reason, BLA assumes that it is unsafe to accept connections from those devices.

2. If the connect request originates not from a mobile phone but from another class of devices (for example a laptop or PC), it is vital to ensure that the BD_ADDRESS falls within the trusted range of BD_ADDRESS's (marked by the BLA). The BLA can, for example, choose to accept connect requests only if the BD_ADDRESS is within the range of 00:AA:10:EE:21:01 to 00:AA:10:EE:21:0F.

On subsequent handshakes with remote device, CDM collects information and correlates it against the rules. If CDM at any stage finds that the request from a remote device does not match the rules, CDM will then mark it as an intrusion and alert this to the MPUI. CDM will update NTDDB with the newly marked device. In the event that the remote device is from a valid source, CDM accepts the connection and a connection response will be sent. CDM then updates CDB to reflect the new connection and the new remote device is also added to the TDDB.

Authentication Module (AM): AM handles the authenticity of the requests received by the Service Module (SM). AM authenticates the request only if the device is already in the CDB and on condition that the PIN code length in the request is at least 8 octets. If the device is not in CDB, AM classifies the received request as an intrusion attempt which possibly gained access through an anomalous Bluetooth connection. MPUI is then alerted and an authentication failed response is sent to SM. PIN codes of which the length is less than 8 octets are classified as a security threat [30]. The reason for this is that special algorithms can easily compromise PIN codes with a short length [11, 31]. If the device is found in CDB and the PIN code length in the request is less than 8 octets, the AM informs the SM that the authentication has failed. The AM then classifies the received request as an intrusion attempt and MPUI is alerted of this. NTDDB is subsequently updated with the remote device if it is not already in the database. AM also checks the TDDB and if the device is listed, it is removed from the database. The disconnect request is then sent to CDM to close the Bluetooth connection and CDB is updated.

Service Module (SM): SM is in command of offering controlled access to Bluetooth applications or services offered by mobile phone to other devices. It does this by using AM and Services and Device Mapping database (SDMDB). When SM receives a service request; for example if remote *device A* sends a request to access the Internet through our current device, SM will trigger AM to check the authenticity of the request. If authentication fails, AM will take further action and an authentication failure status will be sent to SM.

D. Logging Module (LM): LM is in command of logging all Bluetooth activities occurring in a device, upon request from user. The user, if suspicious of any Bluetooth session, sends a request to LM through MPUI. When LM receives a Log Packet Request, it transfers this request to BBMI and then all transactions in that particular Bluetooth session is logged. LM updates Logging Database (LDB) with the log file attained in the requested Bluetooth session. LDB is very useful for storing and retrieving logs of sessions that are of suspicious nature to the user. Logged files can thus be used as references for tracing all activities that occurred in the respective Bluetooth sessions.

All logged files in the LDB share some common information apart from message exchanges in that session. This includes time in which a session started, time in which a session ended and Bluetooth connections active in that session. If AM returns a successful authentication status, SM will check SDMDB to see if *device A* is allowed to use the Internet service. If it is listed, a positive response will be sent and *device A* will be granted permission to use the service. If device is not listed in the SDMDB, then SM will mark it as an intrusion attempt and MPUI will be alerted. NTDDB will be updated with the remote device and a disconnect request will be sent to CDM to close the Bluetooth connection. In such a case, the remote device will also be removed from CDDB and TDDB. The SM will send a positive or negative response to BBMI, depending on the success or failure of the service access request.

E. BLA Database: It consists of five databases and they have already been explained in the previous sections and are as follows.

1. Connection Database (CDB)
2. Trusted Devices Database (TDDB)
3. Non Trusted Devices Database (NTDDB)
4. Services and Devices Mapping Database (SDMD)
5. Logging Database (LDB)

F. MPUI: The MPUI is the module that directly interacts with the mobile phone user. It exchanges requests and responses with IDVM and LM and is responsible for sending user inputs to these modules. MPUI facilitates the following functions:

1. ‘Start BLA’: As the name suggests, when user selects this option, BLA starts.
2. ‘Accept Request’: This option is to acknowledge requests. The requests may include requests of connection, service or application access.
3. ‘Discard Request’: This option enables the user to decline a request.
4. ‘Log Packet’: This option is to facilitate transaction logging.
5. ‘Stop BLA’: As the name suggests, BLA stops when user selects this option.
6. ‘Power off Bluetooth’: This option enables user to turn Bluetooth off directly from MPUI. This is to safeguard device against intrusions, when IDVM detects an intrusion.

The next section reviews the BLA framework.

V. REVIEW OF THE BLA FRAMEWORK

The BLA framework proof-of-concept implementation consists of two components, namely the BLA and the Bluetooth Intrusion Simulator (BIS). BLA is the component that detects intrusions, alerts mobile phone user and protects the mobile phone from Bluetooth security vulnerabilities. As the name implies, BIS is the component that is developed to generate Bluetooth intrusions, safe requests and random requests in order to test, verify, analyze and emulate the power of the BLA framework. In essence, BIS exactly replicates the behavior of remote Bluetooth devices. BLA and BIS communication is implemented through socket communication, which simulates the Bluetooth link level connection between Bluetooth devices. A connect request in this prototype corresponds to the Bluetooth L2CAP connect request. This is a data link level connection used on top of the Bluetooth link level connection. BLA and BLS is implemented in Microsoft .NET Compact Framework (CF) and the development is carried out in C#. Microsoft Smartphone Simulator is selected as the simulator, which replicates the Bluetooth mobile phone and becomes the platform from where the BLA framework is implemented. The screenshots in figure 3 show the user interfaces of the BLA and the BIS.

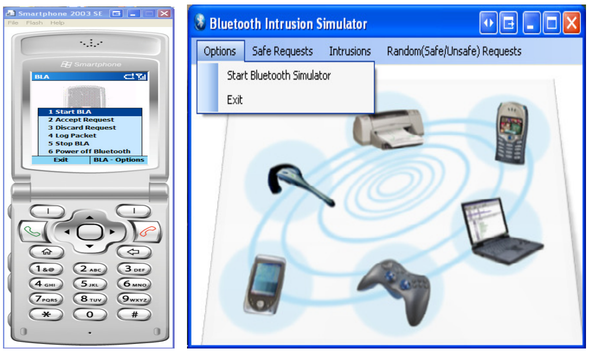
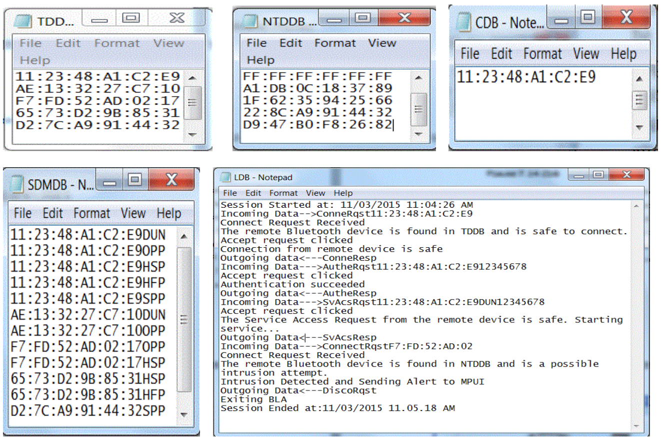


Figure 3. BLA and BIS user interfaces

The BLA Database screenshots are represented as follows in figure 4.


TDD...
File Edit Format View Help
NTDDB ...
File Edit Format View Help
CDB - Note...
File Edit Format View Help
SDMDB - N...
File Edit Format View Help
LDB - Notepad
File Edit Format View Help

```
11:23:48:A1:C2:E9
AE:13:32:27:C7:10
F7:FD:52:AD:02:17
65:73:D2:9B:85:31
D2:7C:A9:91:44:32
```

```
FF:FF:FF:FF:FF:FF
A1:DB:0C:18:37:89
1F:62:35:94:25:66
22:8C:A9:91:44:32
D9:47:B0:F8:26:82
```

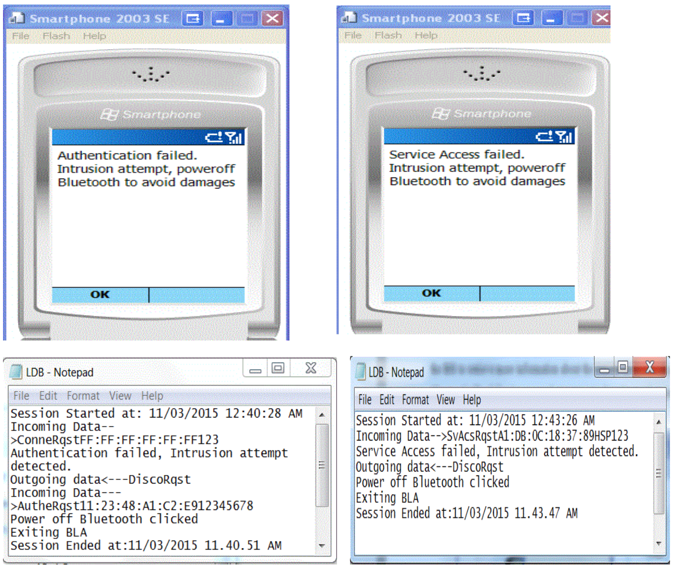
```
11:23:48:A1:C2:E9
```

```
11:23:48:A1:C2:E9DUN
11:23:48:A1:C2:E9OPP
11:23:48:A1:C2:E9HSP
11:23:48:A1:C2:E9HFP
11:23:48:A1:C2:E9SPP
AE:13:32:27:C7:10DUN
AE:13:32:27:C7:10OPP
F7:FD:52:AD:02:17OPP
F7:FD:52:AD:02:17HSP
65:73:D2:9B:85:31HSP
65:73:D2:9B:85:31HFP
D2:7C:A9:91:44:32SPP
```

```
Session Started at: 11/03/2015 11:04:26 AM
Incoming Data-->ConneRqst11:23:48:A1:C2:E9
Connect Request Received
The remote Bluetooth device is found in TDDB and is safe to connect.
Accept request clicked
Connection from remote device is safe
Outgoing data<---ConneResp
Incoming Data--->AutheRqst11:23:48:A1:C2:E912345678
Accept request clicked
Authentication succeeded
Outgoing data<---AutheResp
Incoming Data--->SvAcsRqst11:23:48:A1:C2:E9DUN12345678
Accept request clicked
The Service Access Request from the remote device is safe. Starting
service...
Outgoing Data<---SvAcsResp
Incoming Data--->ConnectRqstF7:FD:52:AD:02
Connect Request Received
The remote Bluetooth device is found in NTDDB and is a possible
intrusion attempt.
Intrusion Detected and Sending Alert to MPUI
Outgoing Data<---DiscoRqst
Exiting BLA
Session Ended at:11/03/2015 11.05.18 AM
```

Figure 4. BLA Databases

Verification and validation of the BLA framework is carried out by providing valid and invalid inputs to get the expected output results as per the scenario. The proof of verification and validation has been provided through log files and screen shots obtained from the BLA framework. The artifacts in figure 5 illustrate the BLA test results obtained on various attack scenarios. BLA detects these attacks and discards the connection attempts accordingly.



Authentication failed.
Intrusion attempt, poweroff
Bluetooth to avoid damages

Service Access failed.
Intrusion attempt, poweroff
Bluetooth to avoid damages

```
Session Started at: 11/03/2015 12:40:28 AM
Incoming Data--
>ConneRqstFF:FF:FF:FF:FF:FF123
Authentication failed, Intrusion attempt
detected.
Outgoing data<---DiscoRqst
Incoming Data---
>AutheRqst11:23:48:A1:C2:E912345678
Power off Bluetooth clicked
Exiting BLA
Session Ended at:11/03/2015 11.40.51 AM
```

```
Session Started at: 11/03/2015 12:43:26 AM
Incoming Data-->SvAcsRqstA1:DB:0C:18:37:89HSP123
Service Access failed, Intrusion attempt detected.
Outgoing data<---DiscoRqst
Power off Bluetooth clicked
Exiting BLA
Session Ended at:11/03/2015 11.43.47 AM
```

Figure 5. BLA test results

V1. CONCLUSIONS

In the BLA framework, an alert mechanism notifies the mobile phone user whenever there is an intrusion attempt. While the alert mechanism adds a slight overhead, it is negligible when compared to mobile phones that run file scanners and antivirus applications in the background that add significant overhead to these devices. Since BLA keeps track all Bluetooth transactions, it allows the user to make flexible decisions such as allowing a basic Bluetooth connection to the user's mobile device from a remote device, thereby safeguarding against intruders. BLA directs Bluetooth transactions only with devices that the mobile phone user permits. This important feature is not available in any of the existing security products. The importance of BLA emerges in that it does not allow any hidden Bluetooth communication. BLA uses its databases extensively to determine the authenticity of a request from a remote device. The databases enable BLA to determine if a request from a remote device is authentic or an intrusion. Further BLA allows the user to participate in Bluetooth activities occurring in the mobile phone and thereby creates Bluetooth security awareness among mobile users.

BLA significantly improves the authentication of the existing Bluetooth security implementation in that it does not authenticate a remote device if the length of the PIN code received is less than 8 octets. BLA prototype, when compared to the existing Bluetooth security solutions, is unique in that it provides a message-by-message logging of each Bluetooth session. This logging feature is advantageous for future use, to retrieve important communication logs and intrusion information. In addition, the logs obtained by the BLA can be used as evidence to claim for major security breaches resulting from an intrusion.

VII. FUTURE WORK

BLA framework may be further enhanced with a rules database. This can be used to store criteria for enabling the Bluetooth transaction with the device under consideration. A Rule Processing Module (RPM) may be incorporated accordingly to IDVM, to process the rules database. The current BLA mobile phone user interface may be slightly changed to present the user with an option to enter new rules. The log messages obtained by BLA may also be automated to derive new rules based on existing log messages. The rules database may also be updated dynamically. Further research may analyse the feasibility of publishing intrusion logs obtained from the logging database to peer Bluetooth devices or to the Bluetooth LAN access points. In doing so, BLA component in those devices will be useful in blacklisting intruders and updating their rules.

REFERENCES

[1] Bluetooth® Core Specification 4.2 (2015), Bluetooth® SPECIAL INTEREST GROUP, [Online]. Available: https://www.bluetooth.org/en-us/specification/adopted-specifications?_ga=1.131133196.246385207.142 6073633.

[2] T. Muller (1999), Bluetooth Security Architecture Specification, Version 1.0, [Online]. Available: http://www.cs.nccu.edu.tw/~lien/Class/Seminar/2000spring/s8547/Bluetooth/1c11600.pdf

[3] T.C. Niem (2002), "Bluetooth And Its Inherent Security Issues", SANS GIAC Security Essentials Certification (GSEC) v1.4b, [Online]. Available: http://www.sans.org/readingroom/whitepapers/wireless/bluetooth-inherent-security-issues-945

[4] M. Banduni (2006), Mobile Enterprise, Secure mobility, [Online]. Available: http://www.networkmagazineindia.com/200607/coverstories04.shtml

[5] G.M.W. Al-Saadoon (2011), "Applying Packet Analysis as New Approach for Discovering Bluetooth Intrusion", in *Proc. of the 2011 ICICS Conference,* [Online]. Available: http://www.icics.info/icics/proceeding/icics.paper/81.pdf

[6] C. Biever (2005), "New hack cracks secure Bluetooth Devices", New Scientist, [Online]. Available: http://www.newscientist.com/ article.ns ?id=dn7461

[7] J. Sun, D. Howie, A. Koivisto, A.J. Sauvola (2001), "Design, Implementation, And Evaluation Of Bluetooth Security", in *Proc. of the 2001 Wireless LANs and Home Networks International Conference Singapore*, [Online]. Available: http://www.mediateam.oulu.fi/publications/pdf/87.pdf

[8] P.Stirparo1, J. Löschner (2013), "Secure Bluetooth for Trusted m-Commerce", Published Online June 2013, [Online]. Available: www.scirp.org/journal/PaperInformation.aspx?paperID=33221

[9] J. Bray, C.F. Sturman (2001), Bluetooth™ Connect Without Cables (2nd Edition), [Online]. Available: http://www.amazon.co.uk/Bluetooth-1-1-Connect-Without-Cables/dp/0130661066

[10] K. Haataja et al. (2013), Overview of Bluetooth Security, Bluetooth Security Attacks, SpringerBriefs in Computer Science, [Online]. Available: https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/seminar giousof_bluetooth.pdf

[11] K. Scarfone, J. Padgette (2008), Guide to Bluetooth Technology, National Institute of Standards and Technology, [Online]. Available: http://www.mcs.csueastbay.edu/~lertaul/BluetoothSECV1.pdf

[12] E. Chai, B. Deardorff, C. Wu (2012), 6.858: Hacking Bluetooth, [Online]. Available: https://css.csail.mit.edu/6.858/2012/projects/echai-bendorff-cathywu.pdf

[13] C. Gehrmann (2002), Bluetooth™ Security White Paper, Bluetooth SIG Security Expert Group, [Online]. Available: http://grouper.ieee.org/groups/1451/5/Comparison%20of%20PHY/Bluetooth_24Security_Paper.pdf

[14] P. Simoneau (2006), The OSI Model: Understanding the Seven Layers of Computer Networks, [Online]. Available: http://courses.cs.tamu.edu/pooch/463_spring2008/BOOKS/WP_Simoneau_OSIModel.pdf

[15] M. Träskbäck (2000), Security of Bluetooth: An overview of Bluetooth Security, [Online]. Available: http://citeseer.ist.psu.edu/400595.html

[16] R. Morrow (2002), Bluetooth: Operation and Use, Publisher: McGraw-Hill Professional, [Online]. Available: http://www.amazon.com/Bluetooth-Operation-Use-Robert-Morrow/dp/007138779X

[17] Red-M (Communications) (2008), Managing Mobility, Enterprise Secure Wireless Control A Red-M Paper, [Online]. Available: http://www.red-m.com/

[18] L. Gade (2008), MobileTechReview: "Bluetooth Networking for your Palm, Pocket PC and Computer", [Online]. Available: http://www.mobiletechreview.com/tips/Red_M_AP.htm

[19] J. Dixon (2009), Wireless Intrusion Detection Systems Including Incident Response & Wireless Policy, [Online]. Available: http://www.infosecwriters.com/text_resources/pdf/Wireless_IDS_JDixon.pdf

[20] A. Laurie, B. Laurie (2003), Serious flaws in Bluetooth security lead to disclosure of personal data, [Online]. Available: http://seclists.org/risks/2003/q4/10

[21] D. Browning, G.C. Kessler (2010), "Bluetooth Hacking: A Case Study, [Online]. Available: http://www.garykessler.net/library/bluetooth_hacking_browning_kessler.pdf

[22] AJ.Solon, MJ.Callaghan, J.Harkin, TM.McGinni (2006), "Case Study on the Bluetooth Vulnerabilities in Mobile Devices", IJCSNS International Journal of Computer Science and Network Security, [Online]. Available: http://paper.ijcsns.org/07_book/200604/200604B07.pdf

[23] A. Walston, J. Barroso, D. Bassett (2009), "Automated double firmware upgrade", US 8386643 B2, [Online]. Available: http://www.google.com/patents/US8386643

[24] Trend Micro™ (2014), MOBILE SECURITY FOR ENTERPRISES Gain visibility and control of mobile devices, applications, and data, , [Online]. Available: http://www.trendmicro.com/cloud-content/us/pdfs/business/datasheets/ds_mobile_security.pdf

[26] F-SECURE SAFE (2015), [Online]. Available: https://www.f-secure.com/en/web/home_global/products

[27] McAfee Mobile Security (2015), THE ULTIMATE OTECTIONYOU'RE YOUR MOBILE LIFE, [Online]. Available: https://www.mcafeemobilesecurity.com/

[28] Gold Lock 3G™ (2013), [Online]. Available: https://www.gold-lock.com/en/home/

[29] Sophos Mobile Security™ (2014), [Online]. Available: http://www.sophos.com/en-us /medialibrary/PDFs/factsheets/sophosmobilesecuritydsna.pdf?la=en

[30] Y. Shaked, A. Wool (2005), Cracking the Bluetooth PIN, [Online]. Available: http://www.eng.tau.ac.il/~yash/shaked-wool-mobisys05/

[31] H.D. Tsague, F. Nelwamondo, N. Msimang(2012), "An Advanced Mututal-Authentication Algorithm Using 3-DES for Smart Card Systems" , in *Proc. of the 2nd International Conference on Cloud and Green Computing,* [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6382887