VOL. 11, NO. 3, FEBRUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SECURITY ENHANCED CHANNEL PRE-AUTHENTICATION AND DATA ACCESS USING HADAMARD PATTERN IN WIRELESS PERSONAL AREA NETWORKS

B. Nagajayanthi[1], V. Vijayakumari [2] and R. Radhakrishnan[3]
[1]VIT University, Chennai, India
[2]Sri Krishna College of Technology, Coimbatore, India
[3]Vidhya Mandhir Institute of Technology, Ingur (PO), Perundurai (TK), Erode (DT), Tamilnadu, India
E-Mail: nagajayanthi.b@vit.ac.in

ABSTRACT

The inferences of this present work showcase the importance of authentication in Bluetooth. Bluetooth (IEEE 802.15.1) is a multi-communicator link that links network devices over distances typically up to 100 meters and exchange voice, data, photos, video and other information between paired devices. In a Personal Area Network for example the Computer is connected to the mouse, printer and keyboard via cables. The networked nature of devices creates dangers from a forensic standpoint. Bluetooth solves the end-user problems by replacing the cables with radio waves. Bluetooth communication does not require Line-of-Sight (LOS) as the signal is omnidirectional. This technology has importance as the devices which communicate are carried in pockets, bags, etc., which have no line of sight restrictions. In future Bluetooth modems can be envisioned to connect phones to the Internet through public access points. Key problem is to ensure the security of key exchange and Authentication. Authentication includes Pre-Authentication depending on the type of message transfer. Authentication is needed to prove the identities of one piconet to the other. Messages are categorized as confidential and non-confidential messages. Confidential messages are pre-authenticated by using Stronger Encryption techniques which use Hadamard Pattern for authentication. This work presents a secure authentication mechanism for protecting the communication between anonymous Bluetooth peers using *Hadamard' s Pre-Authenticated - Secured Access Algorithm (HPA-SA)*. By increasing the complexity of key generation security is improved.

Keywords: security, ad-hoc, low power, unlicensed band, pre- authentication, encryption, key management, confidential, pairing, global, hadamard pattern, decryption.

1. INTRODUCTION

Bluetooth [1] operates in the globally available unlicensed 2.4 GHz Industrial, Scientific and Medical (ISM) band - the "free band" which is used by cordless phones, Wi-Fi and Microwaves for industrial scientific and medicinal purposes. Almost every mobile and our day to day application like car, mobile have Bluetooth in it. Devices are connected and information is exchanged by pairing. There are also reported instances of making the system inoperable by sending virus along with data. Security is ensured in Bluetooth by performing authentication, authorization, integrity and confidentiality checks. Inspite of the security checks there are recorded instances of reported vulnerabilities in Bluetooth.

Nowadays with the advent of Internet-of-Things it is possible to track an asset using Bluetooth. If Authentication and encryption is not done it is possible for an attacker to impersonate and get hold of the information. In an airport lounge a person wants to read his E-Mail, access his files and print using his Bluetooth enabled Laptop. An attacker can eavesdrop and hack the message during pairing. With a Bluetooth connection, a laptop user can direct a cell phone to establish a connection to the Internet. Bluetooth is by design a peer-to-peer network technology that lacks centralized administration and security enforcement infrastructure. Gap exists between usability and security in Bluetooth which needs to be bridged. Bluetooth is used to link the devices on the road, office or at home.

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) recently issued their guidelines in the Special Publication (SP) 121, Revision 1, and also in the Guide to Bluetooth Security: Recommendations of the National Institute of Standards and Technology, to help organizations protect their Bluetooth devices from security threats and vulnerabilities. Since the publication of the original version, many changes and improvements in Bluetooth technology have been implemented in commercial devices. With new advancements, new threats and vulnerabilities has increased in the information systems.

Bluetooth piconets are often established on a temporary and changing basis. This allows for communication flexibility and scalability between mobile devices, and for easy file sharing and synchronization of information between Bluetooth devices. In addition, designers have implemented Bluetooth using a wide variety of chipsets, devices, and operating systems. Because of these complexities, Bluetooth is particularly susceptible to a diverse set of security vulnerabilities. Publicly documented Bluetooth attacks involve identity detection, location tracking, denial of service, unauthorized control and access of data and unauthorized device control.

The website http://www.bluetooth.com/Bluetooth/Learn/Security contains a general discussion of Bluetooth security and vulnerabilities, and http://www.trifinite.org contains detailed description of Bluetooth attacks.

Basically Bluetooth connections are secured by using long randomly generated passkeys. The strength of Bluetooth security relies primarily on the length and randomness of the passkey used for Bluetooth pairing, during which devices mutually authenticate each other for the first time and set up a link key for future authentication and encryption. Other issues related to Bluetooth security are discoverability and connectivity settings.

The proposed algorithm Hadamard's Pre-Authenticated - Secured Access Algorithm (HPA-SA) is based on Hadamard pattern orthogonality function. In this approach during the pre-authentication phase, connection between two devices is established to generate a common secret pattern as a prerequisite using hadamard's orthogonality function which is exchanged between devices for subsequent authentication. This makes communication secure against passive attacks on the privileged side channel and also from the vulnerable wireless link attacks.

This work is organized as follows:

Section 2 provides a literature review of the existing security measures available in Bluetooth. Section 3 provides the topology. Section 4 provides an overview of the proposed security framework. Section 5 elucidates HPA-SA algorithm implementation .Section 6 illustrates a pictorial view of the implementation. Section 7 interprets the Simulation and Experimental results in different scenarios. Section 8 highlights the significance of HPA-SA algorithm. Section 9 provides the conclusion and ideas for innovative extensions in future work.

2. RELATED WORK

Security experts have analyzed and carried out research in the security architecture of Bluetooth. Major commendable research work is referred in [6] and [8]. Different techniques used for device pairing in wireless network was compared by the authors in [8].They have also provided comparative results of their findings on security protocols.Auxiliary channels ("constrained channels" [18] and "location-limited channels" [12]) were used for authentication between two (or multiple) devices. Specific channels are video by using mobile phone cameras , face matching [20], audio by relying on ambient audio [21], comparing spoken sentences [22] or MIDI tunes [23], motion by common movement [24, 25]. Location based authentication was done in [33] depending on the frequency of visit of a person to a particular location .Improperly secured Bluetooth implementations provide attackers with unauthorized access to sensitive information and usage of Bluetooth devices or networks to which the devices are connected [9]. Different cryptographic protocols for multi-channel authentication have been proposed in [10-17], most of them have been developed for the purpose of secure authentication between devices, users and services, good summary of the most important protocol proposals can be found in [19]. Cryptographic protocols have been developed to exploit their diverse characteristics and detailed security analysis needs to be done for each of them.

3. BLUETOOTH TOPOLOGY

Bluetooth connects heterogeneous devices using Star Topology. The device that initiates the connection becomes the "Master" and the one that gets the information is called the "Slave". Communication occurs between the slave and the Master. One Master and upto seven slaves form a Piconet that is why the required address space is limited to 3. Bluetooth consists of both Hardware and Software. Hardware is used to link the devices using RF and software is used by the Bluetooth devices to discover each other and initiate connection via inquiry and paging. A Master can become a slave in another Piconet using Time Division Multiplexing whereas a Master cannot become another Master in a different Piconet. If more devices are added to the Piconet the throughput decreases. Connected piconets are referred to as "Scatternet". All the devices in the Piconet are synchronized to the Master's clock and the hopping sequence. The hopping sequence is determined using the 48 bit Bluetooth Device Address .To avoid interference the Piconets have a unique hopping sequence. Since Bluetooth operates in the unlicensed band in which other devices like microwave ovens, cordless phones etc., also operate so interference is more likely to occur. To prevent interference Bluetooth uses Frequency Hopping Spread Spectrum (FHSS).

The Bluetooth protocol stack comprises of the Controller part and a Host part. The Controller part comprises of the Bluetooth Radio, Baseband Layer and the Link Manager Protocol which forms the hardware. Host deals with high level data, and is usually built in software. Between the Host and the Controller there is a Host/Controller Interface.

The Radio layer specifies the air interface details like transmit power, frequency and Modulation. The range of communication can be increased from 10 to 100m by increasing the transmit power. The Baseband layer enables the physical links between the devices and defines synchronous data packets for voice transfer and asynchronous data packets for data transfer. The Link layer sets up and controls the link characteristics such as transmit power, QoS , level of security etc., between the Bluetooth devices. This layer measures the received signal strength of the received packets and is also responsible for generating and exchanging the keys needed for Authentication and Encryption.

4. PROPOSED SECURITY FRAMEWORK

Key management is left to the software layers. Authentication and Privacy is maintained by the Bluetooth Physical layer and the software protocol layer.

Authentication involves Initialization, Pre-authentication, Key Generation and Key Verification. Keys are generated for Authentication and Encryption.

Hadamard transform for secure access

Properties of Hadamard matrices such as orthogonal, symmetric and self-inverting nature make them useful for data encryption [28], signal processing [29], data compression algorithms [30], randomness measures [31] and so on. Efficient security algorithms can be designed using the Binary nature of elements in basis vectors [27]. Hadamard transform H_a is a $2^a \times 2^a$ matrix that transforms 2^a real numbers x_n into 2^a real numbers X_k. The Hadamard transform can be defined in two ways: recursively, or by using the binary (base-2) representation of the indices n and k. Recursively, the Hadamard transform H_a for $a > 0$ is defined by:

$$H_a = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{a-1} H_{a-1} \\ H_{a-1} -H_{a-1} \end{pmatrix} = H_1 \otimes H_{a-1} \quad (1)$$

where the $1/\sqrt{2}$ is a normalization that is sometimes omitted.

5. PROPOSED ALGORITHM: HADAMARD's PRE-AUTHENTICATED CHANNEL - SECURED ACCESS ALGORITHM (HPA-SA)

Based on the application scenario, the proposed algorithm supports different modes of operation depending on the type of data that needs to be transmitted via the channel. The data communicated is categorized as secured and common depending on whether the message is confidential or not.

Secured Data Transmission (SDT): This is applicable for unicast and multicast messages where confidentiality is required. In the Proposed algorithm pre-authentication is done to ensure secured data transmission.

Common Data Transmission (CDT): This is applicable for broadcast messages where pre-authentication is not required.

Proposed algorithm for secured communication involves the following phases:

- **Initialization**
- **Pre-Authentication**
- **Authentication and authorization**
  - Key Generation
  - Key Verification
- **Data confidentiality**
  - Encryption
  - Decryption

Parameters

P_k	Public Key
N	Piconet (collection of Bluetooth nodes), i ∈ N
N_i^h	i^{th} Node header data
N_i^{ip}	i^{th} Node IP address
N_i^m	i^{th} Node message
N_i^{sh}	Slave node - header data
N_i^{sip}	Slave node - IP address
N_i^{sr}	Slave node - raw data
N_m^n	Master node - name
N_m^{ip}	Master node - IP address
N_m^r	Master node - raw data
N_m^k	Master node - key
N_i^{sn}	Slave node - name
M_k^p	Master node - Private key
S_k^f	Slave node - Final key
A_k^p	Attacker node - Private key
D_k	Decryption key
T_p	Temporary variable
	Master PN Sequence
	Generated Master Final key
M_{id}	Master Id
N_i^n	i^{th} Node name
N_i^{id}	i^{th} Node ID
N_i^r	i^{th} Node raw data
N_i^k	i^{th} Node key
N_i^{sk}	Slave node - key
N_i^{sid}	Slave node - ID
N_i^{sm}	Slave node - message
N_m^h	Master node - header data
N_m^{id}	Master node - ID
N_m^m	Master node - message
N_a^{id}	Attacker node - ID
S_{ps}	Slave PN Sequence
S_k^p	Slave node - Private key
M_k^f	Master node - Final key
A_k^f	Attacker node - Final key
M_{pk}	Malware protection key
T_{p1}^m,T_{p2}^m	Temporary variable1 & 2 at master M_{ps}
R_{fk}^s	Received Slave Final key G_{fk}^m
M_k^p	Malware Protection key

Functional notations

f_{db} → converts a nonnegative integer decimal $\vec{D}$ to a binary matrix B. Each row of the binary matrix B corresponds to one element of $\vec{D}$.

f_{ri} → returns an array containing integer values drawn from the discrete uniform distribution on I_{max} and I_{min}.

f_{ir} → reads a grayscale or color image from the image file specified by the string $\mathcal{F}$.

f_l → loads data from file A into workspace or data from an ASCII file A into a double-precision array.

f_{xor}→ returns the bitwise XOR of arguments P_k ,S_k^p *or* P_k ,M_k^p

Initialization

The master node in the piconet creates station ids for legal slave nodes (N_i^{sid}). This station id (N_i^{sid}) is configured in all legal slave nodes. All legal slave nodes receive control messages from the master node (assuming a possibility of having multiple master nodes in a scatternet). The control messages contain master node information like public key(P_k), Network id/BSS etc. The slave initiates a pre-authentication process with the following parameters initialized (N,P_k, N_i^{sid}, N_i^{sn}, N_i^{sm}, N_i^{sr}). The first process is to $f_{XOR}(P_k, N_i^{sid})$ to generate a 32-bit temporary key S_k^t.

$$S_k^t = (P_k \otimes N_i^{sid}) \quad (2)$$

[32-bit, 32-bit]

Algorithm: Nodes and parameter initialization

Input : Number of nodes N, nodes locations in 2-D plane N(X, Y) = $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

Output : Master and Slave Nodes are initialized in a 2-D plane dimension with initial set of parameters

N; I_{min}; I_{max}; δ = 128; $\delta_0 = 64$; δ_1=32; δ_2=37; δ_3=5; à = 1; Ĵ = 2; $\delta_4 = 4$; M_{id} = 10;

Ĵ = 16;

x_{dm};y_{dm};

ξ; λ_m ; λ_s ; $\delta_1 z_u$ =′ unicast′; $\delta_2 z_u$ =′ broadcast′; A =′ datafile. txt′

$P_k = f_{db}(f_{ri}(I_{min}, I_{max}), \delta)$;

for i = 1 : N

N_i^n= null; N_i^{id}= 0; N_i^h = 0; N_i^{ip} = 0; N_i^r = null; N_i^m = null; N_i^k = 0;

end

% Node deployment in a 2-D plane location

% N(X,Y) = $(x_1, y_1), (x_2, y_2). \ldots (x_n, y_n)$

N(i).x = $v_j * x_{dm}$; N(i).y = $v_j * y_{dm}$;

$0.25 \leq v_j \leq 0.75$

% Initialize Master node

$N_m^n \leftarrow \xi$;

$N_m^{id} \leftarrow \lambda_m$;

% Initialize Slave nodes

for i=1: N-1

$S_k^t \leftarrow 0$;

end

% initializing unique ids to Slave nodes

m ← 1;

for i = 1 : N

if N_i^{sid} ~= λ_m

$N_i^{sn} \leftarrow \lambda_s$;

$N_i^{sip} \leftarrow f_{db}(N_m^{id}(m), \delta_1)$;

m ← m+1;

end

end

% Assume 2 slave nodes – Assign inputs to the slave nodes

$N_1^{sh} \leftarrow \delta_1 z_u$;

$N_i^{sm} \leftarrow f_{ir}$(Ғ);

$N_1^{sh} \leftarrow \delta_2 z_u$;

$N_i^{sr} \leftarrow$ A;

$N_i^{sm} \leftarrow f_l$(A);

% Generate 32 bit of Slave Station id and Public key

$N_i^{sid} \leftarrow N_i^{sid}(1{:}32)$;

$P_k \leftarrow P(1{:}32)$;

% Generate 32-bit Slave temporary key

$S_k^t \leftarrow f_{xor}(P_k, N_i^{sid})$;

Pre-Authentication

After initializing with the required parameters, pre-authentication is triggered. The 32-bit temporary key (S_k^t) generated by legal slave is given as the input to Hadamard Code pattern (f_{hp}) process (32 X 32 Hadamard matrix pattern), that dynamically picks a Slave Hadamard pattern (S_{hp}) from a random rowed (r) and the generated pattern (S_{hp}, r) is sent through the channel. Master receives the slave Hadamard pattern and the selected row ID from the slave (S_{hp}, r) and initiates the pre-authentication validation process for all its legal slaves, generating a master temporary key (M_k^t) by XOR ing its public key (P_k) with its i^{th} slave node id(S_{id}^i).

$$M_k^t = (P_k \otimes N_i^{sid}) \quad (3)$$

[32-bit, 32-bit]

Using the master temporary key (M_k^t) generated for i^{th} slave node id(S_{id}^i), Hadamard pattern for the master(M_{hp}) is derived for the row (r). Thus Hadamard code pattern process for the master(M_{hp}) is generated for the selected row (M_k^t, r). In the pattern verification process the master Hadamard pattern (M_{hp}) is compared with the slave Hadamard pattern (S_{hp}), for the legal i^{th} slave resulting in successful pre-authentication. Each legal slave is pre-authenticated by the master and the devices are paired successfully. Our improved robust pairing protocol proves to be efficient, user-friendly and it solves critical security problems of Bluetooth by sharing the pattern information rather than exchanging the key. Pre-Authentication using Hadamard Code Pattern generation process is represented in the Figure-1.

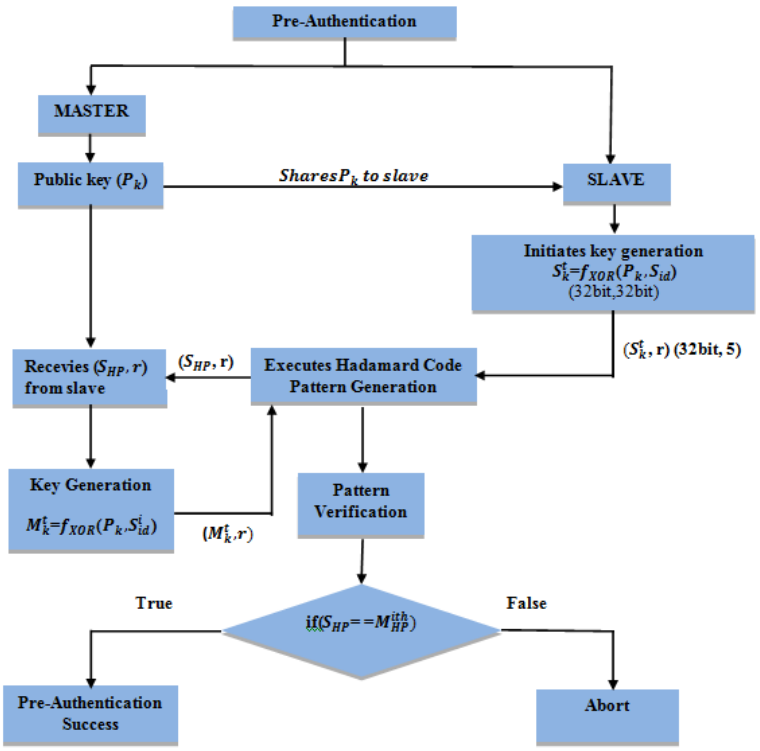


Figure-1. Pre-Authentication using Hadamard Code Pattern generation process.

Algorithm: Channel Pre-Authentication process

Functional notations

f_{hc} → horizontal concatenation of matrices T_{p1}^s *and* T_{p2}^s

f_{rm}→ creates a large matrix tiling of copies of S_k^p

f_{xor}→ returns the bitwise XOR of arguments P_k , S_k^p *or* P_k , M_k^p

f_{bd}→ converts binary vectors to decimal value

f_{hp}→ Hadamard matrix.

f_l → returns the length of the vector

for i = 1 : N

if $N_i^{id} \neq \lambda_m$

C ← N_i^h;

% identifying message as to be confidential or non-confidential

switch(C)

case $\{\delta_1 z_u, \delta_2 z_u\}$

% Hadamard process at slave side

r ← f_{ri}(1 32);

$S_{id} \leftarrow N_i^{sid}$

$S_{id} \rightarrow$ *slave station id*

$S_{id}^k \leftarrow S_{id}$ (1:32);

$S_{id}^k \rightarrow$ 32 *bit slave station id*

$P_k^t \leftarrow P_k$ (1:32);

$S_p^t \leftarrow f_{xor}(P_k^t, S_{id}^k)$;

$S_{hpattern} \leftarrow f_{hp}$(32,r);

for l =1: f_l ($S_{hpattern}$)

for k = 1: f_l (S_p^t)

if $S_{hpattern}$(1,l) == k

S_{hp} (1,l) ← S_p^t(1,k);

break;

end

end

end

% Hadamard process at master side

$N_m^{sid} \leftarrow N_i^{sid}$;

$N_m^r \leftarrow$ r;

$M_{hpattern} \leftarrow f_{hp}(32, N_m^r)$;

$N_m^{sid} \leftarrow N_i^{sid}$;

$M_p^t \leftarrow f_{xor}(P_k^t, N_m^{sid})$;

for l =1: f_l ($M_{hpattern}$)

for k = 1: f_l (M_p^t)

if $M_{hpattern}$(1,l) == k

M_{hp}(1,l) ← M_p^t (1,k);

break;

end

end

end

if (M_{hp} == S_{hp})

f_{dp} ('pre-authentication successfull');

end

Authentication and authorization

After pre-authentication, the devices have to be further authenticated and authorized for service initiation. This process comprises of Key-Generation and Key-Verification stages. Each stage has a unique algorithmic structure that makes services flow between devices in a sequential phased manner. It helps in quicker authentication and authorization with robust security structure in WPAN environment.

Key generation

In the key generation process, the key is generated both at the slave side and the master side.

The following entities referred in Table-1 are used for key generation:

Table-1. Keys used for authentication and encryption.

Entity	Key size	Description
BD_ADDR	48 bits	Address of the Bluetooth Device
Node-id	32 bits	Master generates a node-id for each slave within the piconet.
Private user key, authentication	128 bits	Authentication keys are derived from the master and the slave using the node-id of the slave and the pre-commitments exchanged between the master and the slave.
Private user key, encryption	8-128 bits	Using a part of the key used for authentication and a malware protection key the data is encrypted .

Key Generation - Slave: The Slave id (N_i^{sid}) [16-bit] and slave pattern (S_{hp}) [16-bit] is concatenated using the function horizontal concatenation (f_{hc}) and is stored in a slave pattern sequence (S_{ps}).

$$S_{ps} = (N_i^{sid} \parallel S_{hp}) \quad (4)$$

[16-bit, 16-bit]

The i^{th}unique node id (N_i^{sid}) [32-bit] and the slave pattern sequence (S_{ps}) [32-bit] is concatenated using the function horizontal concatenation (f_{hc}) and is stored in slave private key (S_k^p).

$$S_k^p = (N_i^{sid} \parallel S_{ps}) \quad (5)$$

[32-bit, 32-bit]

The slave private key (S_k^p) [32-bit] is repeated twice using function repeat(f_{rp}) to generate a $S_k^p$of 64-bit.

$$S_k^p = f_{rp}\,(S_k^p, S_k^p) \quad (6)$$

[64-bit] [32-bit, 32-bit]

The public key (P_k) [64-bit] and slave private key (S_k^p) [64-bit] is XORed and stored in slave final key(S_k^f) [128-bit].

$$S_k^f = (P_k \otimes S_k^p) \quad (7)$$

[128-bit] [64-bit, 64-bit]

Key Generation - Master: The slave id (N_i^{sid}) and master pattern (M_{hp}) [16-bit] is concatenated using the function horizontal concatenation (f_{hc}) and is stored in a master pattern sequence (S_{ps}).

$$M_{ps} = (N_i^{sid} \parallel M_{hp}) \quad (8)$$

[16-bit, 16-bit]

The i^{th}unique node id (N_i^{sid}) [32-bit] and the master pattern sequence (M_{ps}) [32-bit] is concatenated using the function horizontal concatenation (f_{hc}) and is stored in master private key (M_k^p).

$$M_k^p = (N_i^{sid} \parallel M_{ps}) \quad (9)$$

[32-bit, 32-bit]

The master private key (M_k^p) [32-bit] is repeated twice using function repeat(f_{rp}) to generate a master private key which is 64-bit.

$$M_k^p = f_{rp}\,(M_k^p, M_k^p) \quad (10)$$

[64-bit] [32-bit, 32-bit]

The public key (P_k) [64-bit] and master private key (M_k^p)[64-bit] is XOR ed and stored in master final key(M_k^f) [128-bit].

$$M_k^f = (P_k \otimes M_k^p) \quad (11)$$

[128-bit] [64-bit, 64-bit]

Algorithm: Key generation

% Key generation at slave side

$S_{ps} \leftarrow f_{hc}\left(T_{p1}^s(\grave{a}:\hat{J}), T_{p2}^s(\grave{a}:\hat{J})\right)$;

$S_k^p \leftarrow f_{hc}\,(N_i^{id}, S_{ps})$;

$S_k^p \leftarrow f_{rm}\,(S_k^p, \grave{a}\,, \hat{J}\,)$;

$S_k^f \leftarrow f_{xor}(P_k\,, S_k^p)$;

% Key generation at master side
$M_{ps} \leftarrow f_{hc}\left(T_{p2}^{m}(\grave{a}:\hat{J}), T_{p1}^{m}(\grave{a}:\hat{J})\right)$;
$\psi \leftarrow f_{db}(N_m^{id}(N_i^{id}), \delta_1)$;
$M_k^p \leftarrow f_{hc}(\psi, M_{ps})$;
$M_k^p \leftarrow f_{rm}(M_k^p, \grave{a}, \xi)$;
$M_k^f \leftarrow f_{xor}(P_k, M_k^p)$;

Key verification

The slave final key (S_k^f) is converted from binary to decimal using function (f_{bd}) and stored in the slave final key variable (R_{fk}^s).

$$R_{fk}^s = f_{bd}(S_k^f) \quad (12)$$

The master final key (M_k^f) is converted from binary to decimal using the function (f_{bd}) and stored in the master final key variable (G_{fk}^m).

$$G_{fk}^m = f_{bd}(M_k^f) \quad (13)$$

Algorithm: Key verification

% Key Verification at master side
$R_{fk}^s \leftarrow f_{bd}(S_k^f)$;
$G_{fk}^m \leftarrow f_{bd}(M_k^f)$;
if $R_{fk}^s == G_{fk}^m$

f_d("Authorized access by legal Slave - Authentication Successful");
else
f_d("Unaauthorized access by illegal device - Authentication failed");
end

Data confidentiality

Protection of transmitted data against passive eavesdropping and Man-In-The-Middle (MITM) attacks.

Malware Protection Key Generation for Ciphering using Hadamard Code Pattern

Cipher is an algorithm to perform encryption and decryption process. In our approach a [64x64] Hadamard matrix is generated to draw a 32-bit pattern, r $\leftarrow f_r[1, 64bit]$. A 32-bit pattern and row id is given as input to Hadamard process and the pattern (P_t) is generated $P_t \leftarrow f_{hp}$(32bit, r). The length of the pattern (l_p) is calculated using the function (f_l) where pattern is represented as ($\boldsymbol{P_t}$) [32-bit] ,$l_p \leftarrow f_l(P_t)$. The length of the slave final key (S_k^f) [32-bit] is calculated and stored in $l_{sfk} \leftarrow f_l(S_k^f)$.

The temporary value of the slave final key (S_k^f) [32-bit] is stored in the variable temporary slave key (T_{sk}).

$$T_{sk} = S_k^f \quad (14)$$

[32-bit]

The temporary value of the master final key (M_k^f) [32-bit] is stored in the variable temporary master key (T_{mk}).

$$T_{mk} = M_k^f \quad (15)$$

[32-bit]

Algorithm: Hadamard Code Pattern Generation

% Hadamard pattern generation
R $\leftarrow f_{ri}(1 : \delta_0)$;
$P_t \leftarrow f_{hp}(\delta_1, R)$;
$l_p \leftarrow f_l(P_t)$;
$l_{sfk} \leftarrow f_l(S_k^f)$;
for ɲ =1:l_p
for ƙ = 1: l_{sfk}
if $P_t(1, ɲ) ==$ ƙ
$T_{sk}(1, ɲ) \leftarrow S_k^f(1, ƙ)$;
$T_{mk}(1, ɲ) \leftarrow M_k^f(1, ƙ)$;
break;
end
end
end
$N_i^k \leftarrow f_{bd}(T_{sk})$;
$N_m^k(N_i^k) \leftarrow f_{bd}(T_{mk})$;

The Malware protection key (M_k^p) and length and breadth of the image is given as the input to key generation function and a key is generated.

$$\forall = f_{kg}(M_k^p, \beta) \quad (16)$$

where, $\forall \rightarrow$ key $\beta \rightarrow$ length and breadth of image file.

Similarly, Malware duplicate key (M_k^d) and length and breadth of the image is given as input to key generation function and a decryption key is generated.

$$[\overrightarrow{K_d}] = f_{kg}(M_k^d, \beta) \quad (17)$$

Where, $\overrightarrow{K_d} \rightarrow$ key

Algorithm: Malware Protection Key Generation

Functional notations

$f_{sz} \rightarrow$ returns the Size of array
$f_{kg} \rightarrow$ encrypts and decrypts the image file
$M_k \leftarrow M_k^p$;

for i=1:20
if $M_k > 1$
$M_k \leftarrow M_k/10$;
end
if $D_k > 1$
$D_k \leftarrow D_k/10$;
end
end
$I_f \leftarrow \acute{Y}_l$;
f_d("Original Image");
$[\vec{n}\ \vec{m}\ \vec{k}] \leftarrow f_{sz}(I_f)$;
$\beta \leftarrow \vec{n}*\vec{m}$;
%Malware key generation for ciphering
$\forall \leftarrow f_{kg}(M_k, \beta)$;
$M_k^d \leftarrow D_k$;
$[\overrightarrow{K_d}] \leftarrow f_{kg}(M_k^d, \beta)$;

Encryption

SDC Information being exchanged between Bluetooth devices are encoded in a way that eavesdroppers cannot decode its contents.

Data encryption - Slave

The sensor data ($\acute{Y}_{df}$) is stored in a data file (D_f) and a key $[\overrightarrow{K_d}]$ is generated based on the size of the sensor data using key generation function,

$$[\overrightarrow{K_d}] = f_{kg}(M_{pk}, S_i) \quad (18)$$

The sensor data (D_f) and the key $[\overrightarrow{K_d}]$ is XORed and an encrypted data file is created.

$$E_{data} = f_{xor}(D_f, [\overrightarrow{K_d}]) \quad (19)$$

Now the slave will send the encrypted data file to the master.

Algorithm: Data encryption

Functional notation

$f_p \rightarrow$ returns the filename parts such as path, file name, and file name extension for the specified file
$f_{op} \rightarrow$ opens a file for binary read access
$f_{ts} \rightarrow$ reads formatted data from text file or string
$f_{scp} \rightarrow$ compares strings
$f_{imed} \rightarrow$ perform encryption and decryption to the image file
$f_{dataed} \rightarrow$ perform encryption and decryption to the data file
$D_f \leftarrow \acute{Y}_{df}$;
$S_i = f_{sz}(D_f)$;
$M_{pk} \leftarrow M_k^p$;
%Encryption key is generated
$[\overrightarrow{K_d}] = f_{kg}(M_{pk}, S_i)$;
$[\overrightarrow{K_d}] = f_{id}([\overrightarrow{K_d}])$;
for i = 1: S_i
$[\overrightarrow{K_d}](i) = f_{floor}([\overrightarrow{K_d}](i)*100)$;
end
% Data file is encoded using the encryption key
$E_{data} = f_{xor}(D_f, [\overrightarrow{K_d}])$;

Data verification - Master

Master validates the file types and permits the legal file and rejects the illegal file from the slave. Master validates each file types using file parts (f_p) function.

$$[\sim, \sim, \Delta\omega] = f_p\ (N_1^r) \quad (20)$$

particular file extensions are stored in fileID

$$f_{id} = f_{open}(\beta\Delta.\text{txt}) \text{ where, } \beta\Delta \rightarrow \text{filetypes} \quad (21)$$

Read the fileID and store it in file type using text scan function and close the fileID

$\beta\Delta = f_{ts}(f_{id}, s)$

Master analyses the data content transmitted from Slave, if $\acute{Y} == 0$, where $\acute{Y} \rightarrow$ file, Master blocks the file indicating as threats detected. Whereas, if the file is validated as legal file type, then ($\acute{Y}$ (1) ~= 0) would be true, Master accepts the legal file content and performs decryption over SDT. Master analyses the data content from the file and perform encryption and decryption on the data file.

$$f_{imed}(M_k^p, D_k, \acute{Y}) \quad (22)$$

Algorithm: Data verification

%Slave - Internal Attacker starts transmitting corrupt file to the master

%Data Verification Process - Master analyzes the data content transmitted from Slave

$[\sim, \sim, \nabla\omega] \leftarrow f_p(N_1^r)$;
$\ddot{e}^1 \leftarrow \nabla\omega$;
$\acute{Y} \leftarrow 0$;
$\boldsymbol{\vartheta} \leftarrow f_{op}(\acute{Y})$;
$\beta\Delta \leftarrow f_{ts}(\boldsymbol{\vartheta}, s)$; % s → string
$f_c(\beta\Delta)$;
$\omega \leftarrow f_l\ (f_{tp}\{1,1\})$
for d = 1: $l\ (\omega\{1,1\})$
if $f_{scp}(\ddot{e}^1, \omega\{1,1\}\{d,1\})$
$\acute{Y} \leftarrow N_1^m$;

end
end
if′Y == 0
f_d("Threats detected - File blocked");
end
if′Y (1) ~= 0
f_d("Decrypt the data");
end
Decryption

Data decryption - Master

Upon successful validation against slave content transmitted, master performs decryption process to generate the original data. Decryption mechanism is triggered when the master generates a decryption key ($\overrightarrow{K_{d1}}$) using a key generation function (f_{kg}).

$$[\overrightarrow{K_{d1}}] = f_{kg}(M_{pk1}, ssi) \quad (23)$$

The encrypted data (E_{data}) and the decrypted key $[\overrightarrow{K_{d1}}]$ is XORed to generate the decoded/original data.

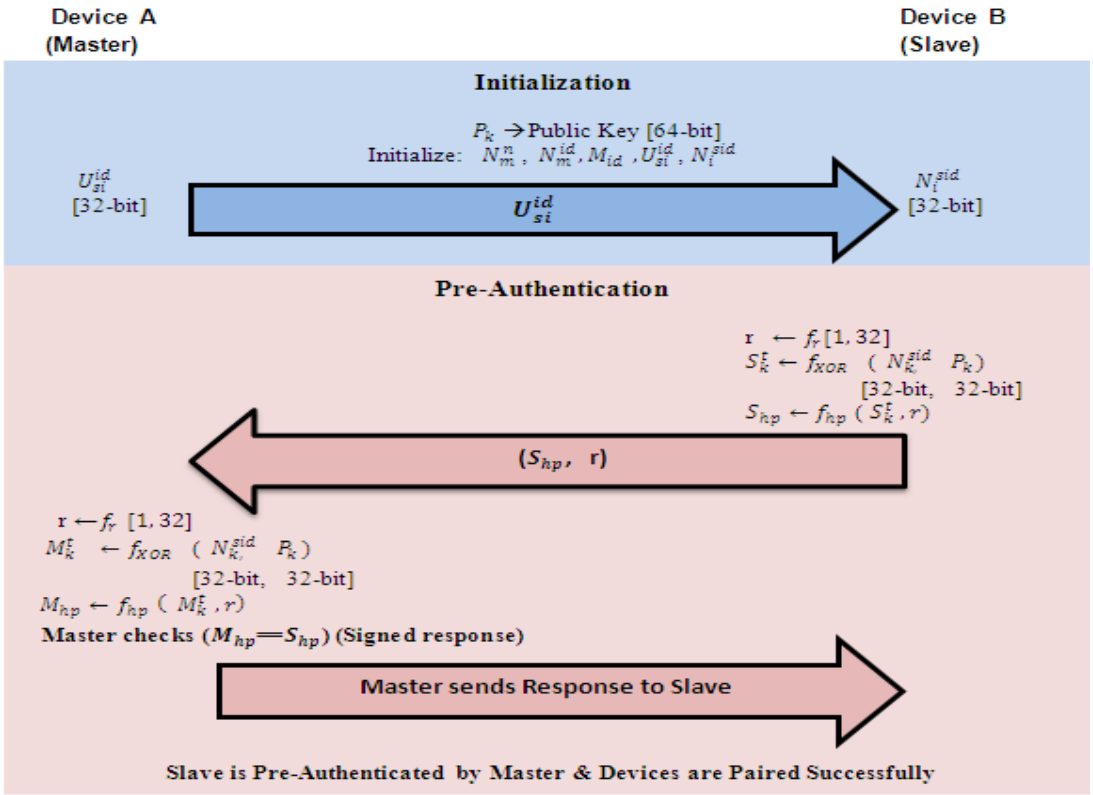
$$D_{data} = f_{xor}(E_{data}, [\overrightarrow{K_{d1}}]) \quad D_{data} \rightarrow \text{Decoded data [Original data]} \quad (24)$$

Algorithm: Data decryption

$M_{pk1} = D_k$;
$[\overrightarrow{K_{d1}}] = f_{kg}(M_{pk1}, ssi)$;
$[\overrightarrow{K_{d1}}] = f_{id}([\overrightarrow{K_{d1}}])$;
for i = 1:ssi
$[\overrightarrow{K_{d1}}](i) = f_{floor}([\overrightarrow{K_{d1}}](i)*100)$;
end
$D_{data} = f_{xor}(E_{data}, [\overrightarrow{K_{d1}}])$;

6. ILLUSTRATION OF BLUETOOTH PICONET SECURITY OPERATION USING HPA-SA

The Complete HPA-SA algorithmic phases are diagrammatically illustrated below in Figure-1:



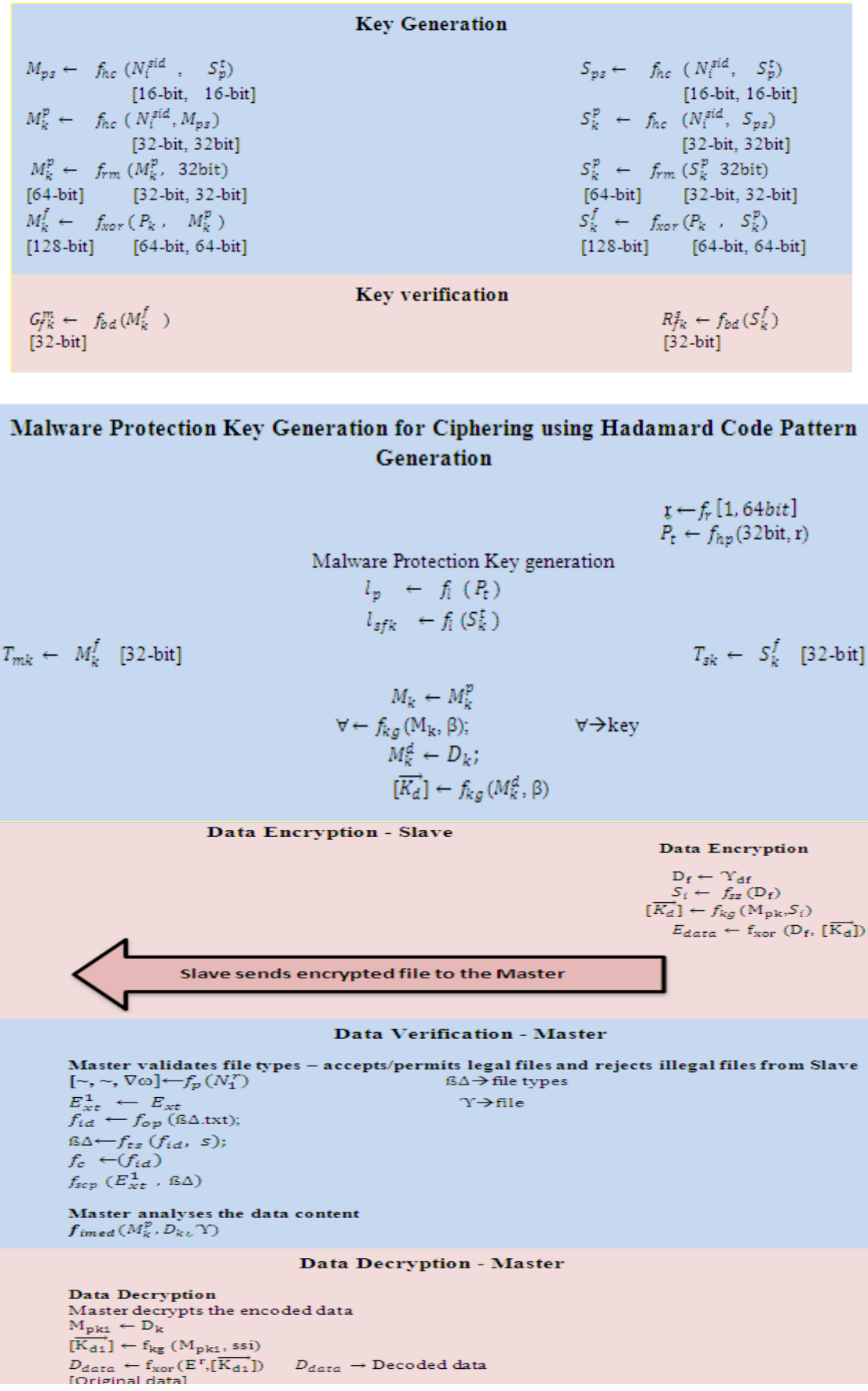


Figure-2. Sequences involved in Pattern Generation using HPA-SA.

7. SIMULATION AND EXPERIMENTAL ANALYSIS

Simulation was developed using MATLAB. A piconet with one master and 4 slaves is formed. An attacker was introduced in the piconet and the corresponding security issues were analyzed and tested using HPA-SA algorithm. Simulation results enunciate authentication between the master and the slave with an without attacker. For the test instances, an image and a sensor data have been chosen to understand the probabilistic reception of the data by the slave in the specified transmit duration. The transmit configurations as named subsequently; can be categorized into broad terms as master and slave communication using HPA-SA without Attacker, HPA-SA with Attacker. In addition, the effect of internal and external attacker on the transmission is also illustrated by applying the HPA-SA with Attacker.

Scenario 1: HPA-SA without Attacker: Master pre-authenticates the slave device in Figure 3 depending on the type of message and authenticates the slave for further data transmission. As designed and theoretically observed, it can be inferred that the original image file shown in figure 3(a), is encrypted as shown in figure 3(b) and the encoded data is transmitted from the master to the slave, which is decrypted by the slave as shown in figure 3(c) , to retrieve the original image. In addition, the sensor data, was chosen to get tested over the same algorithm and the original ,encrypted and decrypted data is shown in figure 3(d),3(e),3(f), respectively.

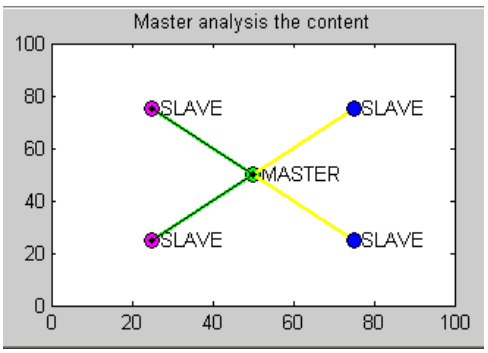


Figure-3. Master pre-authenticates the slave.

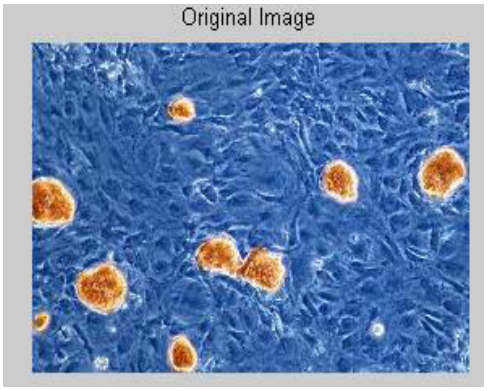


Figure-3(a). Original image.

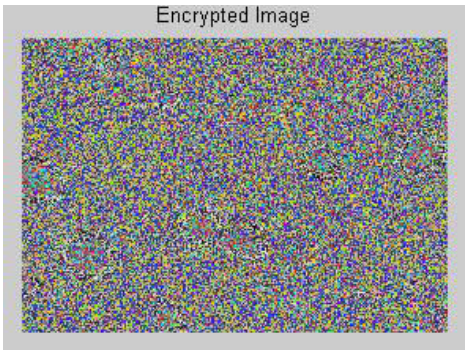


Figure-3(b). Encrypted image.

Figure-3(c). Decrypted image.

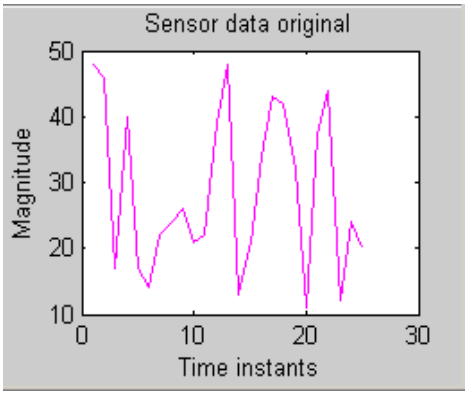


Figure-3(d). Sensor data sent.

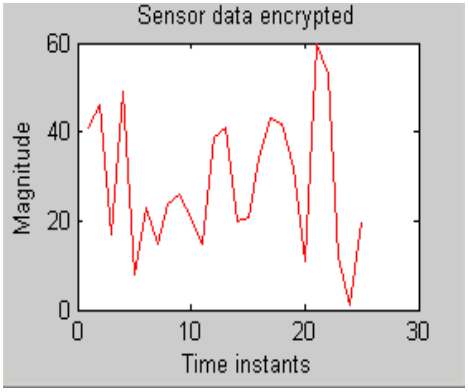


Figure-3(e). Sensor data encrypted.

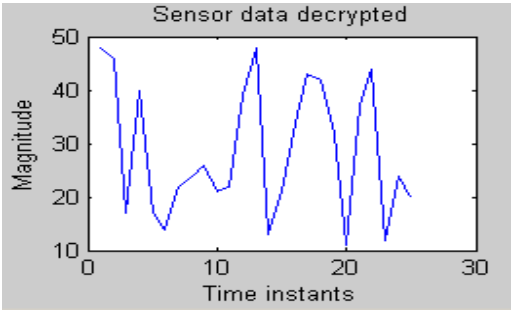


Figure-3(f). Sensor data decrypted.

Scenario 2: HPA-SA with external attacker: The following scenario takes place between master, slave and external attacker device. Master pre-authenticates the slave device and authenticates for further data transmission. An external attacker copies the Hadamard pattern, sent through the channel, but fails to pre-authenticate as a legal slave device with Master in Figure-4. During the same process, the attacker node using the pattern, tries to generate the decryption key to decrypt the encoded image in Figure-4b and fails to retrieve the original content shown in Figure-4c. In addition, the sensor data was chosen to get tested over the same algorithm and the original, encrypted and decrypted data is shown in Figure-4(d). 4(e), 4(f) respectively. The External Attacker fails to retrieve the sensor data.

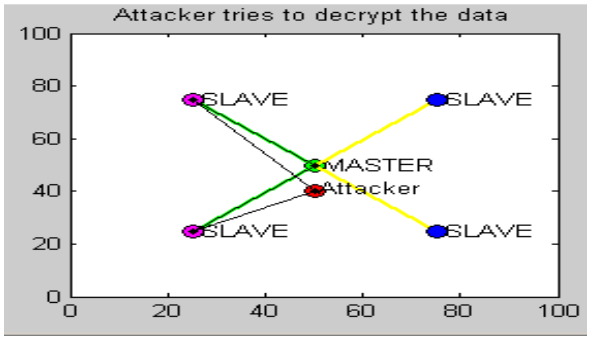


Figure-4. Attacker copies the identity of the slave and fails to get pre-authenticated by the master.

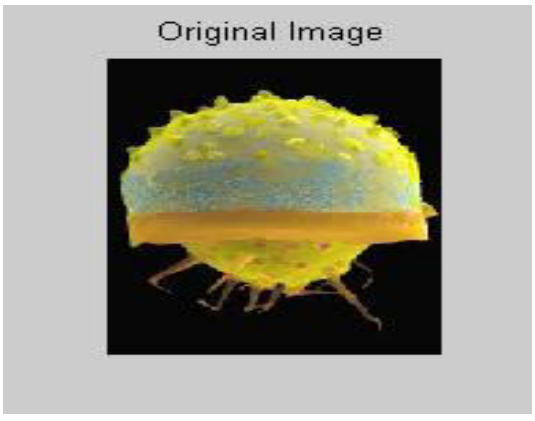


Figure-4(a). Original image sent from the master.

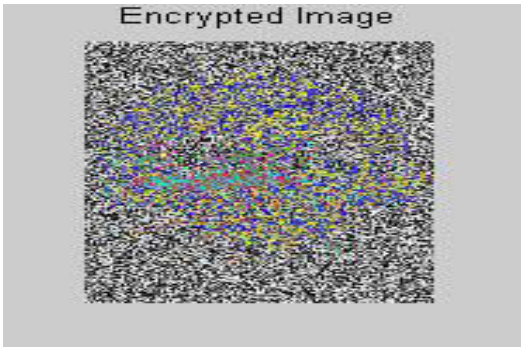


Figure-4(b). Encrypted image.

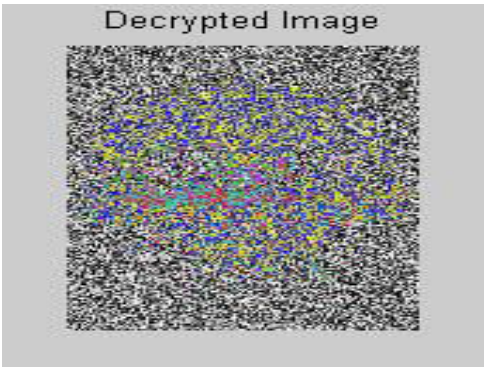


Figure-4(c). Attacker is not able to decrypt the image.

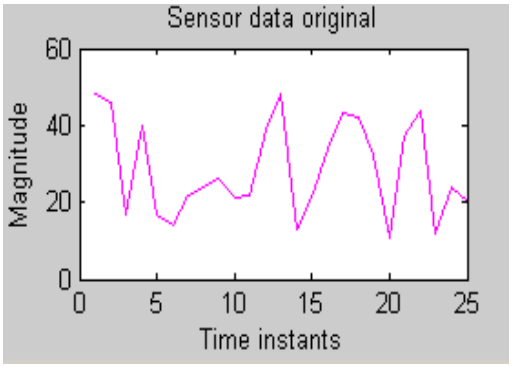


Figure-4(d). Sensor data sent.

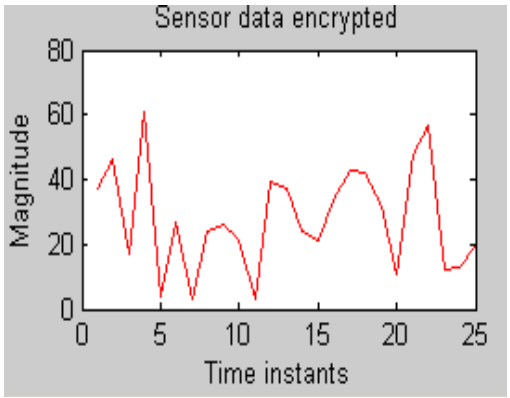


Figure-4(e). Encrypted data.

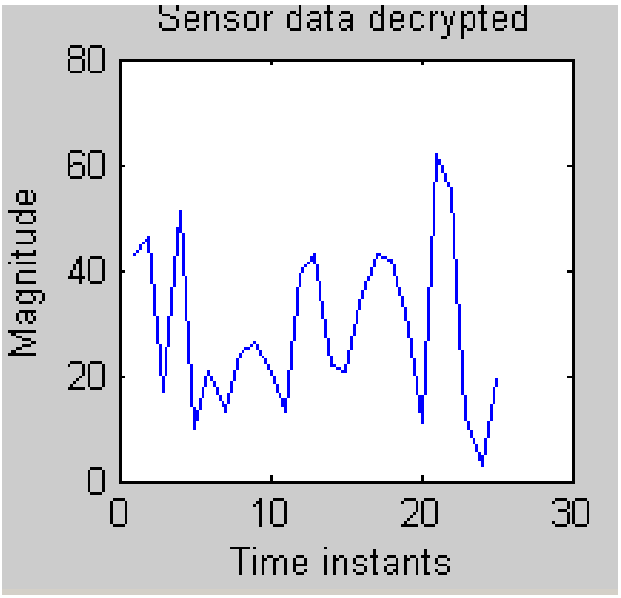


Figure-4(f). Attacker is not able to decrypt the data.

Scenario 3: HPA-SA with Legal Slave Device as an Internal Attacker: Master pre-authenticates the legal slave device and authenticates for further data transmission. Legal slave transmits a malware file to the master, which after content verification by the master is identified and blocked. So only the valid file from legal slaves is accepted by master thus ensuring data verification authenticity. Figure-5 referred below illustrates the following scenario that takes place between master and legal slave device where the slave acts as an internal attacker and transmits a malware file, but fails to pre-authenticate as a legal slave device with Master. During the same process, the attacker node using the pattern, tries to retrieve the data from the encrypted data shown in Figure 5c and fails to retrieve the original content.

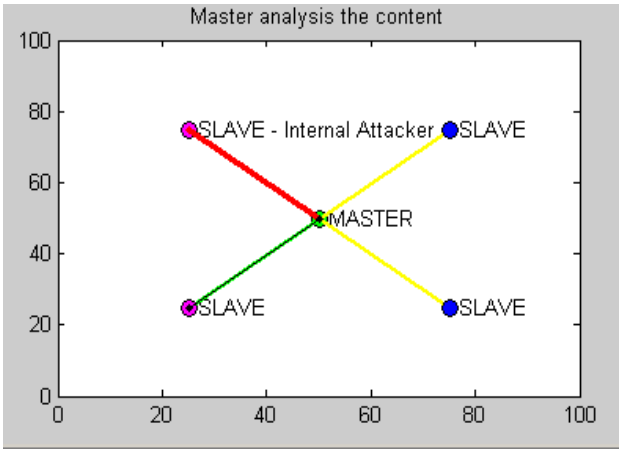


Figure-5. Slave transmits a malware file to the master acting as an internal attacker.

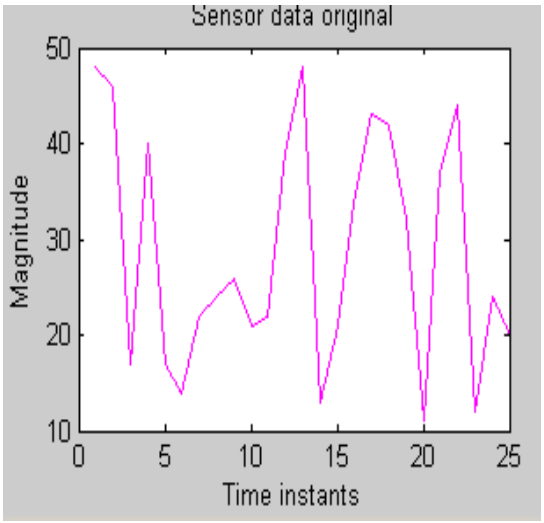


Figure-5(a). Sensor data sent.

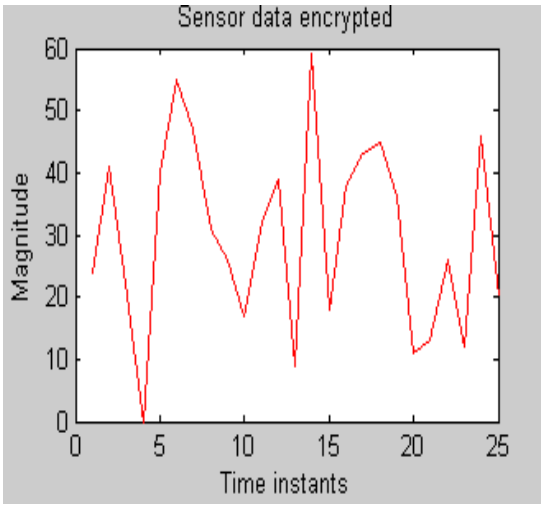


Figure-5(b). Encrypted data.

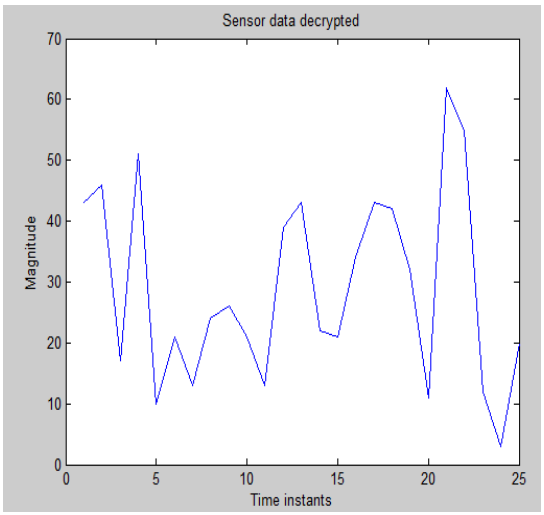


Figure-5(c). Slave is not authenticated.

8. RESULTS AND DISCUSSIONS

Our proposed method was architected considering the dynamic nature of network and their topologies with respect to application based architecture. Methods like Triple DES, AES, BLOWFISH, ECDH, IDEA, SAFER+ etc have shown considerable improvement in the security level but their performance in terms of Encryption frequency, Data throughput and Encryption time were not commendable against new challenges. Based on the analysis, the HPA-SA algorithm required minimum encryption time and maximum encryption frequency when compared to all the existing algorithms due to the enhanced pre-authentication mechanism implemented using Hadamard process and the pattern generated was used for generating the malware protection key that was used for data encryption and decryption that proved to be efficient compared to other existing algorithms.

The number of hits required to attack various algorithms were compared and shown in Figure-6. The security level is much enhanced for the algorithm proposed since the number of hits is found to be maximum due to the introduction of the dynamically generated Hadamard Pattern.

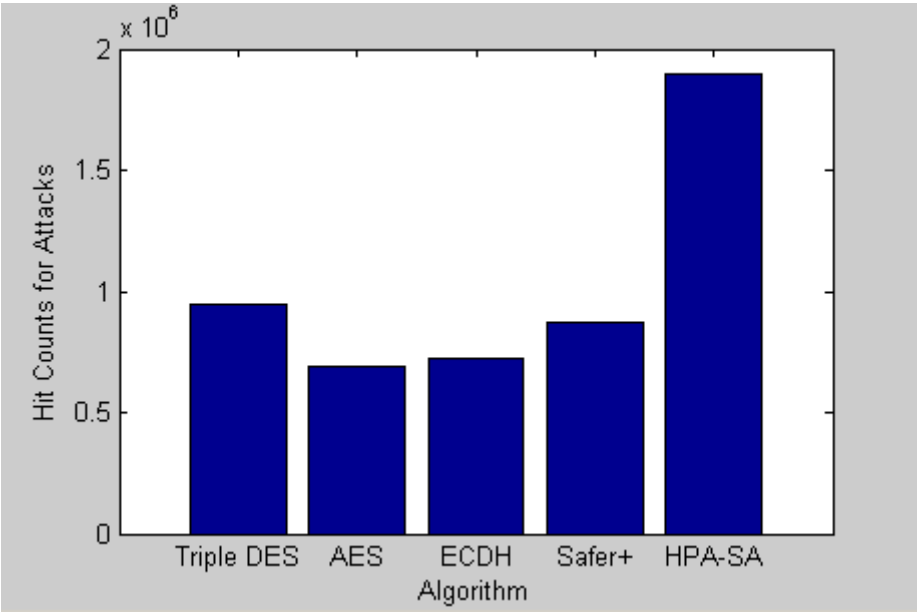


Figure-6. No. of Hit counts for HPA-SA Vs Other algorithms.

Authentication Delay is a factor for differentiating the performance of the security protocols. Authentication Delay is calculated considering Propagation Delay and Process Delay. The Propagation delay is calculated on one-direction involving the Master and the slave Node. In a Piconet scenario the propagation delay is not a deviating factor as consider in other networks due to its limitation in geographical area of RF coverage (Usually within few 10 meters). Hence the factor of Process delay constitutes the main content in overall authentication delay. From the assumptions and simulation analysis HPA-SA method takes less computation time to construct the pattern and communicate with full authentication compared with conventional algorithm.

Higher security level has always been an inverse factor to bandwidth and data throughput in wireless networks. When the security level increases, it consumes certain bandwidth by decreasing overall throughput due to frequent updates of control messages. HPA-SA method was architected in keeping view of this data throughput variations and has proven from the experimental analysis that it outperforms having low overhead than existing algorithms.

Experimental analysis conducted for various applications like audio (*.WAV,.*.MP3), image (*.JPG,*.JPEG, *. .TIFF, *.DOC, *.TXT , *.PDF etc) and data have shown that the encryption time taken for HPA-SA is lesser by 25% - 35% compared to existing algorithms. This shows our algorithm is better suitable for application centric future wireless network. Though our Encryption time is shorter, the security level is higher due to the usage of Hadamard pattern exchange instead of symmetric/asymmetric keys (bits/bytes). Our algorithm uses few bytes (1/2 bytes) for verifying the Hadamard pattern than compared to existing algorithms (4/16/32 bytes). Hence our algorithm is a light-weight security protocol for fast authentication and encryption-decryption process. Encryption frequency was also proven to be better than existing algorithm, showcasing our faster encryption process.

Various existing algorithms were analyzed and compared with the proposed algorithm based on the parameters such as Encryption frequency, Data throughput

and Encryption time - Security level and the results are shown in Figure-7 Based on the analysis, the HPA-SA algorithm required minimum encryption time and maximum encryption frequency when compared to all the existing algorithms due to the enhanced pre-authentication mechanism implemented using Hadamard process and the pattern generated was used for generating the malware protection key that was used for data encryption and decryption, that proved to be efficient compared to other existing algorithms.

The Figure-7 illustrates the comparison between various algorithms against HPA-SA algorithm.

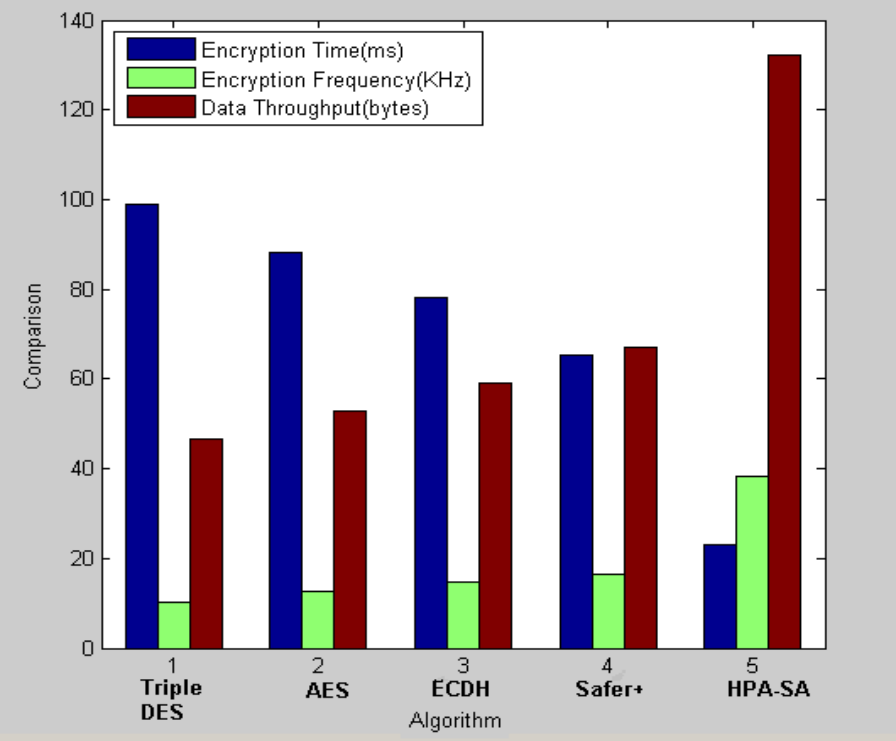


Figure-7. Encryption time, encryption frequency and data throughput vs various algorithms.

9. CONCLUSION AND FUTURE WORK

Currently wireless network challenges the existing security mechanism due to its dynamic nature with respect to topology and application centric architecture. In this paper, thorough analysis and research was done to bring out an efficient, unified and robust security mechanism to meet the current and future requirements. HPA-SA was formulated in keeping view of all these challenges and it has outperformed in terms of security level with existing algorithms like Triple DES, AES, ECDH and SAFER+. Significantly its performance is Noticeable in terms of execution time for authentication and encryption. The hit count on an average is improved by 58% when compared to the other existing algorithms. Data throughput was enhanced and the result proeved 60% higher data rate when compared to the other existing algorithms. This is due to the faster execution time and low control overhead provided by the low byte one way Hadamard pattern used in the algorithm. Upon analyzing with application scenarios, the encryption time taken for HPA-SA is lesser by 25% - 35% compared to existing algorithms. In this paper, HPA-SA an unified security protocol for device authentication proved to be robust and can be still enhanced in areas of IoT and in areas where security concerns are more.

REFERENCES

[1] Bluetooth SIG. Bluetooth specifications 1.0, 1.1, 1.2, 2.0+EDR and 2.1+EDR. Technical specifications, https://www.bluetooth.org. 1999-2007.

[2] Bluetooth Security-Systems and Network Analysis Center Information Assurance Directorate, https://www.nsa.gov/ia/_files/factsheets/i732-016r-07.pdf.

[3] Keijo Haataja. 2009. Security Threats and Countermeasures in Bluetooth Enabled Systems. Kuopio University Library. pp. 55-62.

[4] G. Lamm, G. Falauto, J. Estrada, and J. Gadiyaram. 2001. Security Attacks against Bluetooth Wireless Networks. In: Proceedings of the 2001 IEEE

Workshop on Information Assurance and Security, pages 265-272. U.S. Military Academy, WestPoint, NY

[5] D.Singelee and B. Preneel. 2004. Security Overview of Bluetooth. COSIC Internal Report.

[6] Anindya Bakshi. 2007. Bluetooth Secure Simple Pairing, wirelessdesignmag.com, cover story.

[7] Ersin Uzun, Kristiina Karvonen, N. Asokan. 2007. Usability analysis of Secure Pairing Methods. Nokia Research Center Technical Report NRC-TR-2007-002,2007.http://research.nokia.com/tr/NRC-TR-2007 002.pdf.

[8] Fast Reliable and Secure Digital Communication using Hadamard Matrices – Pal S.K, DRDO Scientist Analysis Group, Published on Computing: Theory and Applications. 2007 ICCTA IEEE Publications.

[9] Paraskevas kitos, Nicolas sklavos, Kyriakos Papadomanolakis and Odysseas Koufopavlou university of patras, Greece. Hardware Implementation of Bluetooth Security. IEEE CS and IEEE Communications Society - January to March 2003. pp. 21-29.

[10] C. Gehrmann, C. J. Mitchell, and K. Nyberg. 2004. Manual authentication for wireless devices. RSA Cryptobytes, vol. 7, no. 1, pp. 29-37.

[11] S. Laur and K. Nyberg. 2006. Efficient mutual data authentication using manually authenticated strings. in Proc. CANS 2006. Springer-Verlag. pp. 90-107.

[12] D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong. 2002. Talking to strangers: Authentication in ad-hoc wireless networks. in Proc. NDSS'02. The Internet Society.

[13] J.-H. Hoepman. 2004. The ephemeral pairing problem. In Proc. 8th Int. Conf. Financial Cryptography. Springer-Verlag. pp. 212-226.

[14] S. Vaudenay. 2005. Secure communications over insecure channels based on short authenticated strings. in Proc. CRYPTO 2005. Springer-Verlag.

[15] S. Creese, M. Goldsmith, R. Harrison, B. Roscoe, P. Whittaker, and I. Zakiuddin. 2005. Exploiting empirical engagement in authenticated protocol design. In: Proc. SPC 2005. Springer-Verlag. pp. 119-133.

[16] M. C˘ agalj, S. C˘ apkun, and J.-P. Hubaux. 2006. Key agreement in peer-to-peer wireless networks. IEEE (Special Issue on Cryptography and Security). 94: 467-478.

[17] F.-L. Wong and F. Stajano. 2006. Multi-channel protocols. In: Proc. Security Protocols Workshop 2005. Springer-Verlag.

[18] T. Kindberg and K. Zhang. 2001. Context authentication using constrained channels. HP Laboratories, Tech. Rep. HPL-2001-84, [Online]. Available: http://www.hpl.hp.com/techreports/2001/HPL-2001-84.pdf.

[19] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. 2007. Secure ad-hoc pairing with biometric: SAfE. in Proc. IWSSI 2007, September 2007, pp. 450–456.

[20] S. Sigg and D. Schuermann. Secure communication based on ambient audio. IEEE Transactions on Mobile Computing (TMC), 2012, accepted for publication.

[21] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. 2006. Loud and clear: Human verifiable authentication based on audio. in Proc. ICDCS 2006. IEEE CS Press. p. 10.

[22] C. Soriente, G. Tsudik, and E. Uzun. 2007. HAPADEP: Human asisted pure audio device pairing. Cryptology ePrint Archive, Report 2007/093.

[23] R. Mayrhofer and H. Gellersen. 2007. Shake well before use: Authentication based on accelerometer data. in Proc. Pervasive 2007: 5th International Conference on Pervasive Computing, ser. LNCS, vol. 4480. Springer-Verlag. pp. 144–161.

[24] Manoop Talasila, Reza Curtmola, Cristian Borcea. 2014. In Collaborative Bluetooth-based location authentication on smart phones. Pervasive and Mobile Computing.

[25] J. H. Hoepman. 2005. Ephemeral Pairing on Anonymous Networks. In Proceedings of the Second International Conference on Security in Pervasive Computing (SPC05), Lecture Notes in Computer Science, LNCS 3450, pages 101-116. Springer-Verlag.

[26] Fast, Reliable and Secure Digital Communication using Hadamard Matrices - Pal S.K, DRDO Scientist Analysis Group, Published on "Computing: Theory and Applications. 2007 ICCTA IEEE Publications.

[27] L. J. Yan and J. S. Pan. 2007. Generalized discrete fractional Hadamard transformation and its application on the image encryption. International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE Computer Press. pp. 457-460.

[28] C. C. Gumas. 2006. A century old fast Hadamard transform proves useful in digital communications. Chip Center quest link.

[29] W. Ouyang, W.K. Cham. 2010. Fast algorithm for Walsh-Hadamard transform on sliding windows. IEEE Trans. on Pattern Analysis and Machine Intelligence. 32, 165-171.

[30] S. Kak. 1971. Classification of random binary sequences using Walsh-Fourier analysis. IEEE Trans. On Electromagnetic Compatibility EMC-13, pp. 74-77.

[31] Rene Mayrhofer, Jurgen FuB, Iulia Ion. 2012. UACAP: A Unified Auxiliary Channel Authentication Protocol. Transactions on Mobile Computing. 1(1).