RASCP: Providing for a secure group Communication plane using RFID

6.1 Introduction:

Predominantly large distributed networks currently provide support for group oriented protocols and applications. Regardless of the type of distributed network there is a need to provide communication privacy and data integrity to the information exchange amongst the group members. Securing data transactions in distributed networks to support reliable group communications is always desired. This realization of a RFID authentication enabled Secure Group Communication Plane (RASCP) is discussed in this research work. RFID tags are commonly used for the purpose of identification. The RASCP proposed uses the RFID tags for group member identification and protocol initialization.

The communications amongst the group members considered are secured using the commutative RSA cryptographic algorithm. The RASCP proposed adopts the positives from the existent RFID technology and incorporates these features to construct secure group communication planes. The RASCP proposed is compared with the existing RACP secure group communication protocol.

The RASCP overcomes the drawbacks of key exchange, key storage, key distribution and the need of external servers to facilitate key management functions that commonly exist in the existing group communication protocols. The efficiency of the RASCP in terms of the computational complexity exhibited is proved against the RACP protocol through the experimental study presented in this research work.

RFID systems and standards established by IEEE [135, 136] are envisioned. A RFID authentication system primarily consists of a tag and a reader with a database to store the tag details. Tags available are of several types and classes [135] [136] but the research work presented here considers the most commonly available passive RFID tags of class 0. A lot of research is ongoing to provide security to the existing standards and the technology involved in manufacturing and Radio Frequency (RF)
communication systems in place. Currently there exist several threats to the existing RFID deployments like Denial of Service Attacks, RFID Tag Cloning, RFID Tag Tracing, Eavesdropping, Replay Attacks Data Forging, Invading Privacy Information and Hot-listing to name a few [137][138][139][140][141][142][143][144].

More often than not researchers have focused on the eliminating the threats that currently exist in the RFID technology and methods towards improving it. In the research work presented here the use of the existing RFID technology for identification is adopted. The proposed protocol i.e. RASCP assumes that the RFID communication module considered is secure and free of the above mentioned defects/attacks.

Communication provisioning is considered as the basic essentials of any network. The prevalent large scale distributed networks existent provide support for various business, personal, commerce, banking, military, intelligence applications and services. These networks are prone to varied kind of attacks and data compromise issues. To counter the issue of data compromise cryptography is commonly used. Cryptographic algorithms could be broadly classified into two types namely Symmetric and Asymmetric type. The RASCP protocol proposed utilizes the asymmetric commutative RSA Algorithm to provide for security.

6.2 Commutative RSA

A secure plane is realizable provided the data communicated over the plane is protected and cannot be colluded. The use of cryptographic techniques is generally preferred; hence the RASCP proposed in this paper adopts the commutative RSA algorithm. The RASCP considers two prime numbers \( p \) and \( q \) initialized amongst all the group members.
Let \( G_A \) and \( G_B \) represent the group members required to communicate over the secure plane. To compute the encryption keys and decryption key pairs of the commutative RSA algorithm the parameters are computed using the following:

\[
\text{Param} _N^{\text{RSA}} \text{ And } \text{Param} _\Phi^{\text{RSA}}
\]

\[
\text{Param} _N^{\text{RSA}} = [(\text{Param} _P^{\text{RSA}}) \times (\text{Param} _Q^{\text{RSA}})]
\]

\[
\text{Param} _\Phi^{\text{RSA}} = [(\text{Param} _P^{\text{RSA}} - 1) \times (\text{Param} _Q^{\text{RSA}} - 1)]
\]

From the above equations it is clear that \( \text{Param} _N^{\text{RSA}} = \text{Param} _N^{\text{RSA}} \) and \( \text{Param} _\Phi^{\text{RSA}} = \text{Param} _\Phi^{\text{RSA}} \) for \( A \) and \( B \).

The encryption key pair of \( A \) and \( B \) represented as

\( (\text{Param} _N^{\text{RSA}} _A, \text{Param} _E^{\text{RSA}} _A) \) And \( (\text{Param} _N^{\text{RSA}} _B, \text{Param} _E^{\text{RSA}} _B) \) are to be obtained.

The variable \( \text{Param} _E^{\text{RSA}} \) is obtained by randomly selecting numbers such that it is a co prime of \( \text{Param} _\Phi^{\text{RSA}} \) or in other terms.

\[
Fn_{\text{GCD}}(\text{Param} _E^{\text{RSA}}, \text{Param} _\Phi^{\text{RSA}}) = 1
\]

Where \( Fn_{\text{GCD}}(x, y) \) represents the greatest common divisor (GCD) function between two variables \( x \) and \( y \).

The decryption key pair of \( A \) and \( B \) is represented by \( (\text{Param} _N^{\text{RSA}} _A, \text{Param} _D^{\text{RSA}} _A) \) and \( (\text{Param} _N^{\text{RSA}} _B, \text{Param} _D^{\text{RSA}} _B) \) the parameter \( \text{Param} _D^{\text{RSA}} \) is computed based on the following equation

\[
\text{Param} _D^{\text{RSA}} = (\text{Param} _E^{\text{RSA}})^{-1} \text{Mod}(\text{Param} _N^{\text{RSA}}).
\]

Let \( Enc_x \) represent the encrypted data \( x \). The encryption operation is defined as follows:

\[
Enc_x = X^{\text{Param} _E^{\text{RSA}}} \text{Mod}(\text{Param} _N^{\text{RSA}})
\]

The commutative RSA decryption operation on the encrypted data \( y \) is defined as follows:

\[
Dec_y = Y^{\text{Param} _D^{\text{RSA}}} \text{Mod}(\text{Param} _N^{\text{RSA}})
\]
6.3 Commutative Proof of RSA Algorithm by Example

Let $\text{Param}_p^{CRSA} = 293$ and $\text{Param}_q^{CRSA} = 113$

$\text{Param}_N^{CRSA} = [(\text{Param}_p^{CRSA}) \times (\text{Param}_q^{CRSA})] = [293 \times 113] = 33109$

$\text{Param}_N^{CRSA} = [(\text{Param}_p^{CRSA}) \times (\text{Param}_q^{CRSA})] = [293 \times 113] = 33109$

$\text{Param}_\Phi^{CRSA}_A = [(\text{Param}_p^{CRSA} - 1) \times (\text{Param}_q^{CRSA} - 1)] = [(293 - 1) \times (113 - 1)] = 32704$

$\text{Param}_\Phi^{CRSA}_B = [(\text{Param}_p^{CRSA} - 1) \times (\text{Param}_q^{CRSA} - 1)] = [(293 - 1) \times (113 - 1)] = 32704$

The encryption key pair of A obtained is

$$(\text{Param}_N^{CRSA}_A, \text{Param}_E^{CRSA}_A) = (33109, 49423)$$

And the decryption key pair is

$$(\text{Param}_N^{CRSA}_A, \text{Param}_D^{CRSA}_A) = (33109, 32303)$$

The encryption key pair of B computed is

$$(\text{Param}_N^{CRSA}_B, \text{Param}_E^{CRSA}_B) = (33109, 11323)$$

and the decryption key pair is

$$(\text{Param}_N^{CRSA}_B, \text{Param}_D^{CRSA}_B) = (33109, 330067)$$

The commutative nature of the RSA algorithm can be proved if

$$\text{Enc}_B(\text{Enc}_A) = \text{Enc}_B(\text{Enc}_A)$$

Let $X=3569$

$$\text{Enc}_B(3569^{49423} \text{Mod} (33100)) = \text{Enc}_A(3569^{11323} \text{Mod} (33109))$$

$$\text{Enc}_B(30644) = \text{Enc}_A(19419)$$

$$30644^{11323} \text{mod} (33109) = 19419^{49423} \text{Mod} (33109)$$

Thus we find that

$$2166=2166$$
6.4 Prime Number generation:

Prime number generation functions and their application to the arena of cryptography have been extensively studied by researchers. The RACP proposed in this paper utilizes the Sieve of Eratosthenes Algorithm [145] to find a set of prime numbers based on the user RFID tags.

Let $n_{\text{Max}}$ represent a number derived from the user RFID tag. Let us consider a Boolean Set $B_{\text{Tmp}}$ having $n_{\text{Max}}$ Boolean values, each element is represented as $b_{\text{Indx}}$ where $b \in \{T,F\}$ and $\text{Indx}$ represents the index corresponding to the number in $N_{\text{Tmp}} = \{2,3,4,...,n_{\text{Max}}\}$.

Let $Var1 = F_{\text{Least}}(n, N_{\text{Tmp}}, B_{\text{Tmp}})$ represents a function that returns the smallest number $Var1 \in N_{\text{Tmp}}$ in $N_{\text{Tmp}}$ that is greater than $n$ and $b_n = F$ and $b_n \in B_{\text{Tmp}}$. The Sieve of Eratosthenes Prime number generation algorithm is adopted to generate prime number set P. The Sieve of Eratosthenes algorithm adopted is given below:

**Algorithm 1: Sieve of Eratosthenes Prime number generation algorithm**

**Input:** User RFID based Number $n_{\text{max}}$

**Output:** Prime Number Set P

Algorithm Development for Prime number generation:

1) Initialize $N_{\text{Tmp}} = \{2,3,4,...,n_{\text{Max}}\}$

2) Initialize Boolean Set $B_{\text{Tmp}} = \{F_1, F_2, F_3,...,F_{n_{\text{Max}}}\}$

3) Initialize $Var = 2$

4) **Do**

5) Set the Index of all the multiples of $Var1$ to True i.e T occurring between $Var^2$ and $n_{\text{max}}$.

6) $Var = F_{\text{Least}}(Var, N_{\text{Tmp}}, B_{\text{Tmp}})$
7) P= Set of all indexes of 
8) $b_{indx} \in B_{imp} : b_{indx} = F$

From the above algorithm it is evident that the set P obtained contains all the prime numbers between 2 and $n_{max}$.

This algorithm is utilized to obtain the probable $P_{Comm \_RSA}^{Prob}$ and $Q_{Comm \_RSA}^{Prob}$ sets required to initialize the commutative RSA algorithm RACP for each user considered in the communication plane. The computational complexity of this algorithm is $O(n_{Max} \ln \ln n_{Max})$.

6.5 RFID Extended Secure Lock Group Communication Scheme (RSL):

The RSL is a RFID based extended Secure Lock protocol [146]. The RSL protocol considers a central server and a set of group members defined as

$$G^{RSL} = \{ g^{RSL}_1, g^{RSL}_2, g^{RSL}_3, \ldots, g^{RSL}_m \}$$

The RSL protocol incorporates an asymmetric cryptographic algorithm to provide security. Let the private and public of a group member $g^{RSL}_m \in G^{RSL}$ be represented by $(P^{RSL}_m, S^{RSL}_m)$.

The central server also known as the security server establishes a set of $m = |G^{RSL}|$ pair wise relatively prime number $N_1, \ldots, N_m$ from the RFID tags possessed using the Sieve of Eratosthenes Prime number are then assigned to group members $g^{RSL}_m \in G^{RSL}$ and are assumed to be public in nature. To establish a secure plane of communication using the RSL the server computes the following based on the randomly selected key represented as $K^{RSL}$.

$$Lck^{RSL}_m = e^{RSL}_m (K^{RSL}_{m})(mod \ N_m^{RSL})$$
Where \( \epsilon \) represents the encryption operation.

Using the Chinese remainder theorem the server computes \( \text{Lck}^{\text{RSL}} \). The computed value \( \text{Lck}^{\text{RSL}} \) is considered as the lock for the key \( \epsilon_p^{\text{RSL}}(K^{\text{RSL}}) \). The resulting message sent by the server is defined as:

\[
\text{msg}^{\text{RSL}}_m = (\text{Lck}^{\text{RSL}}, \{K^{\text{RSL}}\}K^{\text{RSL}}).
\]

The group member \( g_m^{\text{RSL}} \) on receiving the message \( \text{msg}^{\text{RSL}}_m \) obtains the \( \text{Lck}^{\text{RSL}} \) using the following computations:

\[
\epsilon_p^{\text{RSL}}_m = (K^{\text{RSL}}) = (\text{Lck}^{\text{RSL}})(\mod N_m^{\text{RSL}})
\]

\[
K^{\text{RSL}} = D_p^{\text{RSL}}(\epsilon_p^{\text{RSL}}_m(K^{\text{RSL}}))
\]

Where D represents the decryption operation.

Colluded group members on decryption cannot obtain the lock \( K^{\text{RSL}} \) selected by the server accurately hence providing for security.

The Chinese remainder theorem utilized by the server provides protection by securing the group membership and group size. The use of the Chinese remainder theorem and asymmetric cryptographic scheme render the RSL group communication scheme inefficient and are not scalable.

### 6.6 RFID Extended Access Control Polynomial Secure Group Communication Scheme (RACP):

The RASCP is compared with the Access Control Polynomial Secure Group Communication Scheme based on RFID extensions incorporated similar to the RASCP scheme. The Access Control polynomial protocol considers a central server and a set of group members defined as

\[
G^{\text{RACP}} = \{g_1^{\text{RACP}}, g_2^{\text{RACP}}, g_3^{\text{RACP}}, \ldots, g_m^{\text{RACP}}\}
\]
The members’ $S_m^{RACP} \in G^{RACP}$ possesses a secret key represented as $S_m^{RACP}$.

The secret key $S_m^{RACP}$ is less than a prime number $q^{RACP}$, is positive in nature and is derived from the RFID tag held by each group member. The Sieve of Eratosthenes Prime number generation algorithm is used to derive $q^{RACP}$. The server and the $m^{th}$ group member are aware of the secret key $S_m^{RACP}$. The large prime number $q^{RACP}$ is used to obtain the finite field $F_q^{RACP}$ and

$$f^{RACP} : \{0,1\} \rightarrow \{0,1\}^{l^{RACP}}$$

Where $l^{RACP}$ is a cryptographic hash function and is defined as

$$l^{RACP} = \lfloor \log(q^{RACP}) \rfloor$$

An RACP is a polynomial over $F_q^{RACP}[x]$ and defined as follows:

$$A(x) = \prod_{m \in \mathcal{S}} (x - f(S_m^{RACP}, Z^{RACP}))$$

Where the secret set is represented as $\mathcal{S}$ and the public random integer from $f^{RACP}$ is represented as $Z^{RACP}$.

To maintain security the integer $Z^{RACP}$ changes in accordance to the change in re-computing $A(x)$. If a valid group member with a lock $S_m^{RACP}$ in the set $\mathcal{S}$, $x$ is substituted with $f(S_m^{RACP}, Z^{RACP}(x)$ then $A(x)$ is equated to zero. The value of $A(x)$ is found to be a random number if invalid or random locks are substituted for computation.

The use of “decoys‖ is incorporated in the RACP to establish secure communication.

To establish communication the secret key $K^{RACP}$ to confirm the membership to the set $\mathcal{S}$ the security server of the RACP broadcasts the message defined as follows
(Z^{RACP}, P^{RACP}(x), \{K^{RACP}\}_{K^{RACP}})

Where \(P^{RACP}(X)\) is defined as and \(D^{RACP}w\) represents the decoys.

\[P^{RACP}(X) = A(x)(x - D^{RACP}_1)(x - D^{RACP}_2) \ldots (x - D^{RACP}_w) + K^{RACP}\]

A valid group member \(g^RACP_m \in G^{RACP}\) possessing a secret \(S^{RACP}_m\) can obtain the \(K^{RACP}\) using the following definition

\[K^{RACP} = P(f(S^{RACP}_m, Z^{RACP}))\]

Invalid or colluded group members on computing \(K^{RACP}\) using invalid parameters acquire incorrect random values represented by \(\overline{K^{RACP}}\) and \(K^{RACP} \neq \overline{K^{RACP}}\) providing security to the communication data. The key \(K^{RACP}\) computed is used for encryption and decryption using a symmetric cryptographic algorithm.

The major drawback of the RACP is that there are communication overheads, key storage, key derivation, and key computation overheads when compared to the proposed RASCP protocol.

### 6.7 RFID Authentication Based Secure Communication Plane (RASCP)

Let us consider a set of users who would like to communicate securely represented by a set defined as

\[G = \{g_1, g_2, g_3 \ldots g_m\}\]

Where \(g_m\) represents the \(m^{th}\) user of the group \(G\).

It is assumed that each user \(g_m \in G\) posses a RFID Tag represented as \(T^m\) and an RFID reader. The RFID tags are said to contain data of length \(L^m_T\) where \(m\) represents the \(m^{th}\) user and users associated tag \(T^m\). The secure communication plane is constructed by adopting the commutative RSA algorithm. To initialization of the
The commutative RSA algorithm is based on the RFID tag data $RFID^m_T$, used to obtain the parameters $Param\_P_m^{RSA}$ and $Param\_Q_m^{RSA}$ using the Sieve of Eratosthenes prime number generation algorithm. Each member of the group contributes towards the construction of the commutative RSA sets $PARAM\_P$ and $PARAM\_Q$ defined as:

$$PARAM\_P = \{Param\_P_1^{RSA}, Param\_P_2^{RSA}, ..., Param\_P_m^{RSA}\}$$ and

$$PARAM\_Q = \{Param\_Q_1^{RSA}, Param\_Q_2^{RSA}, ..., Param\_Q_m^{RSA}\}$$

The algorithm used to construct the $PARAM\_P$ and $PARAM\_Q$ sets is as mentioned below:

### 6.7.1 Algorithm Name: $PARAM\_P$ and $PARAM\_Q$ construction

**Input:**

1) Group member set $G = \{g_1, g_2, g_3, ..., g_m\}$

2) Group RFID Tag Associated with each Group Member $g_m$, $T^m$ and its data $RFID^m_T$ and length $L^m_T$

**Output:**

1) $PARAM\_P$
2) $PARAM\_Q$

**Algorithm:**

1) Initialize $PARAM\_P = \emptyset$ and $PARAM\_Q = \emptyset$

2) **For Each** group member $g_m \in G$

3) $Q^m_{Tmp}, P^m_{Tmp} = Split(L^m_T, RFID^m_T)$

4) $P^{Pr\_ob\_Comm\_RSA}_m = Get\_Pr\_imeSet(P^m_{Tmp})$

5) $Q^{Pr\_ob\_Comm\_RSA}_m = Get\_Pr\_imeSet(Q^m_{Tmp})$

6) $Param\_P_m^{RSA} = RandSel(P^{Pr\_ob\_Comm\_RSA}_m, t)$

7) $Param\_Q_m^{RSA} = RandSel(Q^{Pr\_ob\_Comm\_RSA}_m, t)$

8) $PARAM\_P = PARAM\_P \cup Param\_P_m^{RSA}$

9) $PARAM\_Q = PARAM\_Q \cup Param\_Q_m^{RSA}$
10) End For Each

The function Split(x,y) represents a splitting function that obtains the most significant bits (MSBs) of the number Y of length X.

GetPrimeSet(X) represents a function that uses the Sieve of Eratosthenes Prime number generation algorithm to obtain the prime numbers set within X.

RandSel(x,t) represents a random element in the set x selection function based on seed time t.

The communication overheads of this algorithm is $m \times 2D$ transmissions where D represents the size of the message parsed between the m group members.

To construct the commutative RSA secure plane all the m members of the group G require a common $Param_{P_{p^{RSA}}}^{_P}$ and $Param_{Q_{q^{RSA}}}^{_Q}$ to derive their encryption and decryption keys. A time synchronization function $\phi_T$ is adopted to ascertain the

$Param_{P_{p^{RSA}}}^{_P} \in PARAM \ P$ and $Param_{Q_{q^{RSA}}}^{_Q} \in PARAM \ Q$ amongst the group G.

The time synchronization function $\phi_T$ can be considered as RandSel function wherein the seed time is common for all the members $g_m \in G$.

The time synchronization function can be defined as

$\phi_T(x,t_T) = RandSel(x,t_T)$

Where $t_T$ represents the synchronization seed and $\forall g_m \in G : t = t_T$.

The time synchronization function $\phi_T$ is used to obtain $Param_{P_{p^{RSA}}}^{_P}$ and $Param_{Q_{q^{RSA}}}^{_Q}$.

$Param_{P_{p^{RSA}}}^{_P} = \phi_T (PARAM_{P}, t_T)$ and

$Param_{Q_{q^{RSA}}}^{_Q} = \phi_T (PARAM_{Q}, t_T)$.
6.7.2 Derivation of Encryption and decryption keys:
The encryption and decryption keys are derived from \( \text{Param} \_ P_p^{\text{RSA}} \) and \( \text{Param} \_ Q_q^{\text{RSA}} \). Using following algorithm.

6.7.2.1 Algorithm for Encryption and Decryption Key Pair computation

**Input:**
1) Group member Set \( G = \{ g_1, g_2, g_3, \ldots, g_m \} \)
2) \( \text{Param} \_ P_p^{\text{RSA}} \)
3) \( \text{Param} \_ Q_q^{\text{RSA}} \)

**Output:**
1) Encryption Key pair ( \( \text{Param} \_ N_{gm}^{\text{RSA}}, \text{Param} \_ E_{gm}^{\text{RSA}} \) )
2) Decryption Key pair ( \( \text{Param} \_ N_{gm}^{\text{RSA}}, \text{Param} \_ D_{gm}^{\text{RSA}} \) )

**Algorithm:**
1) **For Each** group Member \( g_m \in G 
2) Compute \( \text{Param} \_ N_{gm}^{\text{RSA}} = [(\text{Param} \_ P_p^{\text{RSA}}) \times (\text{Param} \_ Q_q^{\text{RSA}})] \)
3) Compute \( \text{Param} \_ \phi_{gm}^{\text{RSA}} = [(\text{Param} \_ P_p^{\text{RSA}} - 1) \times (\text{Param} \_ Q_q^{\text{RSA}} - 1)] \)
4) Select random number using \( \text{RandSel} (\text{Rnd} \_ \text{Num}, t) \) \( \text{Fn}_{\text{gcd}} \) ( \( \text{Rnd} \_ \text{Num}, \text{Param} \_ \phi^{\text{RSA}} \) ) = 1
5) \( \text{Param} \_ E_{gm}^{\text{RSA}} = \text{Rnd} \_ \text{Num} \)
6) Compute \( \text{Param} \_ D_{gm}^{\text{RSA}} = [(\text{Param} \_ E_{gm}^{\text{RSA}})^{-1} \mod (\text{Param} \_ N_{gm}^{\text{RSA}})] \)
7) Encryption key pair of the \( g_m \)th group member is ( \( \text{Param} \_ N_{gm}^{\text{RSA}}, \text{Param} \_ E_{gm}^{\text{RSA}} \) )
8) Decryption key pair of the \( g_m \)th group member is ( \( \text{Param} \_ N_{gm}^{\text{RSA}}, \text{Param} \_ D_{gm}^{\text{RSA}} \) )
9) **End For Each**

Using the Encryption and decryption Key Pair Computation algorithm all the group members \( g_m \in G \) compute the encryption and decryption key pairs which enable to
construct the envisioned secure communication plane. The RASCP discussed in this research work eliminates the security arising from key exchange neglecting key compromise external server maintenance for key management proving the efficiency in creating a secure communication plane. Let us consider n users of the group G that needs to communicate securely and the secure communication group $\overline{G}$ is defined as

$$\overline{G} = \{g_1, g_2, g_3, \ldots, g_n\}$$

Where $n \leq m$ and $\overline{G} \subseteq G$

The secure communication plane consisting of n group members communicate data by using a series of encryption and decryption operators. The commutative nature of the RSA algorithm adopted in the RASCP ensures that the data communicated is encrypted at least once i.e. the original data is encrypted and then only communicated over the plane thereby securing the data.

The presence of any colluded users within the group represented by $g_c$, on intercepting the data would not be unable to determine the level of encryptions and decryption procedures performed on the data prior to his interception. In the case if the user $g_c \in G$ intercepts the data after the first encryption, $g_c$ would not be able to recover the data as the encryption and the decryption keys are not exchanged and are different for each user $g_c \in \overline{G}$ participating in the secure group communication. Let $g_{Sndr} \in \overline{G}$ represents the sender who needs to communicate data X to $g_{Rcv} \in \overline{G}$ in the presence of group member set $\overline{G}$ securely.

Let us define a set $\overline{G}$ and $G'$ as follows:

$$\overline{G} = G \cap g_{Rcv}$$

$$G' = \overline{G} \cap g_{Sndr}$$

The algorithm to securely communicate amongst $g_{Sndr}$ and $g_{Rcv}$ is mentioned below:
6.7.3 Algorithm for Communication over the Secure Plane

Input:

1) Group Member Set $\overline{G}$
2) Group Member Set $\overline{G}$
3) Group Member Set $G'$
4) Encryption and decryption Key pairs of group member Set $\overline{G}$
5) Data to be transacted $X$ available with $g_{Sndr} \in \overline{G}$

Output:

1) Data $X$ available with $g_{Rcv} \in \overline{G}$

Algorithm:

1) For user $g_m = g_{Sndr} \in \overline{G}$

2) Encrypt the data
   
   i. $Enc_{g_{Sndr}} = \left[ X_{Param \_ E_{g_{Sndr}}^{CRSA}} \mod (Param \_ N_{g_{Sndr}}^{CRSA}) \right]$

   ii. $Enc_{tmp} = Enc_{g_{Sndr}}$

3) End For

4) For Each user $g_m \in G'$

5) Encrypt the data
   
   6) $Enc_{tmp} = \left[ Enc_{Param \_ E_{gm}^{CRSA}} \mod (Param \_ N_{gm}^{CRSA}) \right]$

7) End For Each

8) For Each user $g_m \in \overline{G}$

9) For the first user

10) Decrypt the data

11) $Dec_{tmp} = \left[ Dec_{Param \_ E_{gm}^{CRSA}} \mod (Param \_ N_{gm}^{CRSA}) \right]$

12) End For Each

13) For user $g_m = g_{Rcv} \in \overline{G}$

14) Decrypt to find the final data

15) $X = \left[ Dec_{tmp} \mod (Param \_ N_{gm}^{CRSA}) \right]$

16) End For
Using the communication over Secure plane Algorithm as discussed above the $g_{Rcv}$ is able to receive the data $X$ sent by the user $g_{Sndr}$ using $n$ number of encryption and decryption functions. The algorithm also highlights the fact that the data $X$ to be transmitted is not transmitted in the original form i.e. it is encrypted and transmitted there by securing the data.

The RASCP discussed here utilizes the RFID tags available with each group member $g_m$ to construct the secure communication plane. The RFID tags are often used for identification and tracking. In RASCP the RFID tags are used both for Security provision and identification. As the RASCP adopts multiple encryptions and multiple decryptions to securely communicate data the overheads arising from this could be considered as a drawback of the RASCP.

**PERFORMANCE EVALUATION:**

This secure communication mechanism proposed in this paper is compared with the protocol in terms of the computational costs incurred. The computational cost incurred is proportional to the execution time observed. The and the systems were developed using C#.Net on the Visual Studio 2010 Platform. The tags used were of type 0. The readers were integrated into the platform using VC++.Net. To evaluate the and the secure group communication systems and to observe the computational costs the number of users in the group were varied from 5, 10, 20, 50, 70 and 100 users. The observations were monitored using log files maintained for every operation. The introduction of the tags into the and can be considered as an overhead that exists in reading the tags and the average time observed in reading the RFID tags when the number of group members are varied from 5, 10, 20, 50, 70 and 100 is as shown in Fig 1. It could be observed that the average of the overheads observed reduces as the number of users increase proving that the induction of the based security systems are scalable in nature and do not affect the responsiveness of the systems. The average time taken to read a tag was found to be about 0.76ms.
Figure 6.1: Average Time Observed in Reading RFID Tags with Varying Number of Group Members
The secure group communication system adopts the RSA Algorithm with a key strength of 1024 [19] bits to incorporate secure transmissions amongst the group members. The adopts the commutative RSA algorithm to construct a secure communication plane. The relies on a central server for key initialization, distribution using locks and the verifications is carried out by the group members.

The experimental evaluation conducted considered the protocol initialization phase as the time taken to verify the group membership and derive the cryptographic keys. The computational overheads observed are as shown in Fig.2. It could be observed that the overheads are reduced by about 99.43% in the initialization phase in the protocol. The considers a central server and the verification process of the group members. The overheads resulting from the group membership verification process for the scheme is as shown in Fig 3.

The and the group communication protocols adopt cryptographic techniques to construct a secure communication plane. The overheads arising from the encryption and decryption operations are analyzed for comparisons. The encryption and decryption operations performed using the and the group communication schemes are compared in terms of the computational complexity exhibited.

The results obtained are graphically shown in Fig 4 and Fig 5. Form the figures it is clear that the commutative RSA algorithm adopted in the is computationally less expensive when compared to the RSA cryptographic algorithm adopted in the group communication scheme ye providing security.
Fig 6.2 : Average Protocol Initialization Overheads Vs Group Size
Fig 6.3: Verification Overhead in RSL Scheme
Fig 6.4: Computational Analysis of Encryption Operations
Fig 6.5: Computational Analysis of Decryption Operations
The experimental evaluation discussed in this paper prove that the proposed
group communication protocol introduced in this paper performs better than the existing
scheme by reducing the computational overheads and yet providing security of the data
transacted amongst the group members. The future of the work presented here is to
compare the scheme with other secure group communication schemes
using RFID tags.