Distributed Secure Co-ordinate Control For Multiagent System Under Various Attacks

Shahid shikalgar#1, prof. Prasanna Rasal#2
#1PG Student, Department of Computer Applications, Bharati Vidyapeeth Deemed to be University Pune, India
Yashwantrao Mohite Institute of Management, Karad.
#2Asst.Prof., Department of Computer Applications, Bharati Vidyapeeth Deemed to be University Pune, India
Yashwantrao Mohite Institute of Management, Karad
1shahidshikalgar4@gmail.com
2prrasal123@gmail.com

Abstract—

This paper studies the consensus problem in a multi-agent system. The communication topology is assumed to be directed and fixed. With first order dynamics below the sampled data setting, we first convert the original system into a reduced-order one featuring the error dynamics. Accordingly, the consensus problem is converted into the stabilization of the error dynamic system. Thereafter, based on the theory in stochastic stability for time delay systems, a necessary condition is established in terms of a set of linear matrix inequalities (LMIs). The mean square stability of the error dynamics is shown to guarantee consensus of the multiagent system. By explicitly incorporating the transition possibility of the random delay into consideration, the conservativeness in control design is reduced. A delay-dependent switching control scheme is studied. Based on the solutions of an algebraic Riccati equation and an algebraic Riccati inequality, a procedure to select the control gains is provided and stability analysis is considered by using Lyapunov's method. A distributed, robust, dynamic, control law is studied such that connectivity preserving rendezvous is achieved regardless of the unknown nonlinear dynamics and disturbances

I. INTRODUCTION

Recent years have witnessed an increasing attention on distributed cooperative control of real-world multiagent systems due to its widespread applications in various fields such as distributed control of team robots, design of sensor networks, formation control of vehicles, rendezvous of mobile agents, and synchronization of coupled chaotic oscillators, A fundamental yet interesting issue on this topic is to develop distributed controllers using only relative local information such that as time goes on, all the agents eventually achieve state consensus of the whole group. As an effective consensus seeking approach, consensus tracking
Problem has been widely studied for linear multiagent systems

Distributed secure coordinated control of multiagent systems is an interesting and important problem. Multiagent systems, like all large-scale spatially distributed systems, are vulnerable to cyber-attacks due to the development of network information and communication technologies. Typically, there are two different attack scenarios in a multiagent system: attack on the dynamic behaviors (or closed-loop dynamics) of the agents and attack on the communications among the agents. Both of attacks can dramatically affect the consensus properties of the whole team. Under the assumption that the network is complete, consensus problem was studied in for multiagent systems with adversaries. Shames et al. and Teixeira et al. considered distributed attack detection using unknown input observers for double integrator multiagent systems. In, a distributed attack detection and identification algorithm via a distributed filter was investigated for cyber-physical systems. Note that show that an attack on a specific node is identical to node removal on network graphs. In reality, it is more general to consider the second attack scenario that a number of edges are attacked. In addition, the aforementioned detection techniques and control algorithms are always separated, which implies that there is no feedback to the control parameters when the attacks are detected or identified. Recently, Zhu and Martinez proposed a distributed receding-horizon control method for secure control of multiagent systems by limiting the actions of the adversaries. Moon and Basar and Zhu and Basar modeled attacker-defender interactions as a stochastic game and developed the game-theoretic resilient control schemes for cyber physical systems. So far, how to design effective resilient algorithms is still challenging and of great significance to the distributed secure control problem of multiagent systems

II. EXISTING SYSTEMS

In previous work , two types of attacks:
1) Connectivity maintained
2) Connectivity-broken attacks were studied and a hybrid secure control scheme was provided to achieve distributed secure control of a leader–follower multiagent system. However, on the one hand, the attacks on graphs are modeled by using a deterministic switching signal that determines the switching among various network topologies. That is, it is assumed that the system has complete access to the attacker moves. This similar switching attacks are also considered in from the perspective of sliding mode. On the other hand, sufficient conditions for existence of consensus algorithms are established by solving two linear matrix inequalities (LMIs) to get a common solution for designing Lyapunov functions afterwards. The set of LMIs are dependent on the eigenvalues of the Laplacian matrix of all the information graph topologies. Besides, the time complexity of solving an LMI is O (N2s4), where N and s are the number of

agents and the dimension number of agent dynamics, respectively. Overall, it is conservative by LMI techniques

III. LIMITATIONS

There are several drawbacks over the system

1) The system and the attacker’s machine must be in same network.
2) The server must be on at the time of attack.

IV. PROPOSED SYSTEM ARCHITECTURE

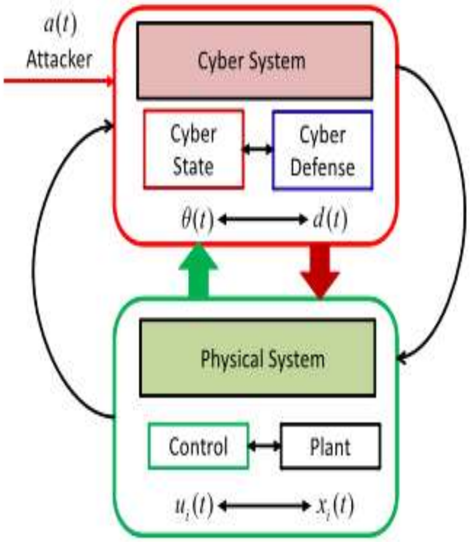


Fig. 1 Architecture of the system

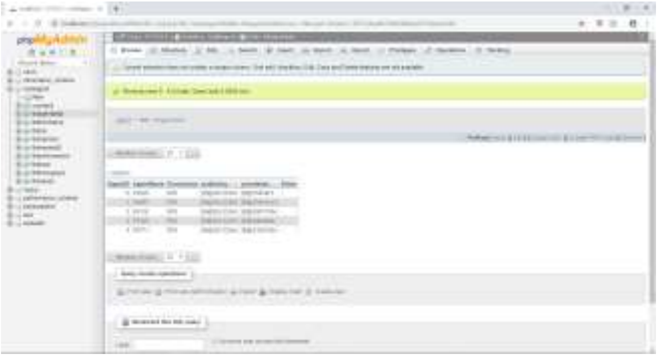
V. IMPLEMENTATION :

We will implement the Agent Security At System as well as Agent level. Gaia methodology and multi-agent systems.

An agent is a software entity that is situated in some environment and is capable of flexible, autonomous action in order to meet its design objectives. It defines an agent as “a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda”. An agent can play one or more roles and interact with other agent to exchange knowledge and coordinate their activities. These interactions occur according to patterns and protocols dictated by the nature of the role itself .

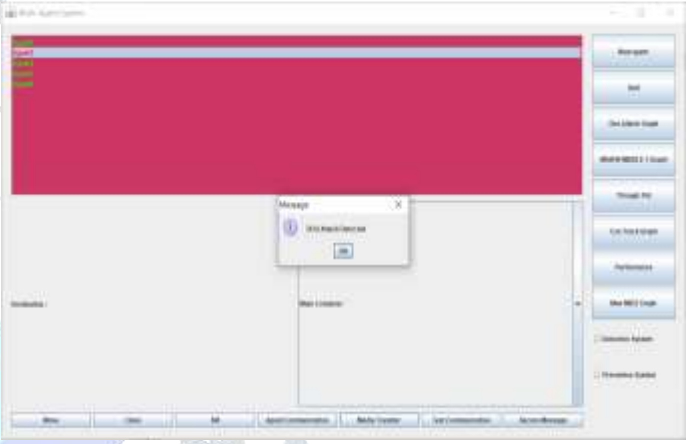
Gaia methodology was developed for agent-oriented analysis and design. The methodology consists of two phases - the analysis phase and the design phase. According to Gaia methodology an organizational model consists of two models, i.e. the roles model and the interaction model. Both models are the result of the analysis phase. The roles model is used to identify all the key roles in the system under development and the interaction model represents the relations between different roles in the system under development. In every system

Front End : ID



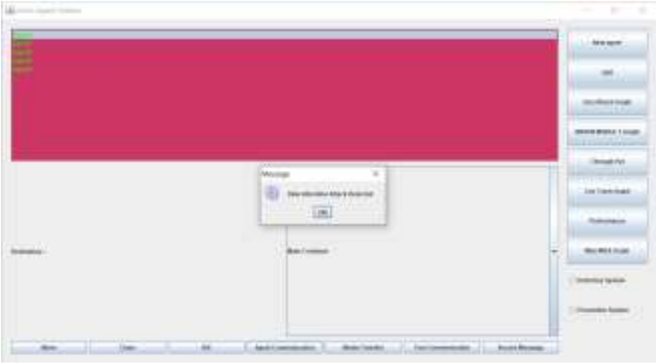
GUI :

1) Detection of DOS Attack :



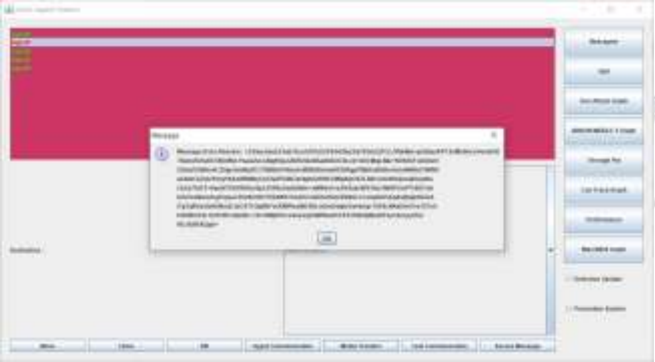
2) DOS Attack Backend :

3) Data Alteration Attack :

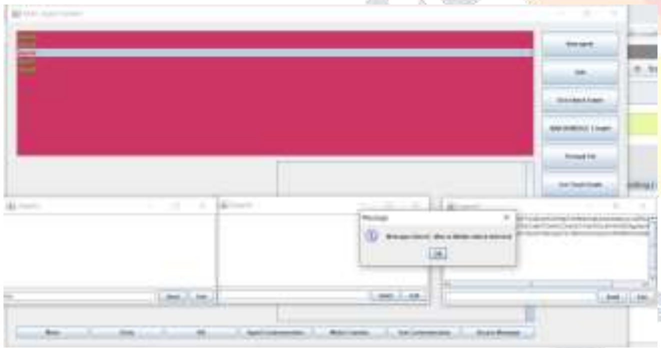


4) Data Alteration attack Backend :

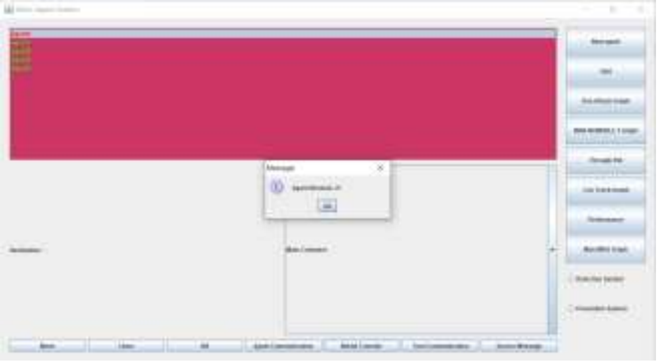
5) Data altered in data Alteration :



6) Man In Middle Attack :



Result : Agent Block



Our Contribution In this Phase

1. Adding ID Parameter for every Agent
2. Generate Public Private Key Pair for every Agent using RSA Algorithm
3. Public Key Cryptography

Unlike symmetric key cryptography, we do not find historical use of public-key cryptography. It is a relatively new concept.
Symmetric cryptography was well suited for organizations such as governments, military, and big financial corporations were involved in the classified communication.
With the spread of more unsecure computer networks in last few decades, a genuine need was felt to use cryptography at larger scale. The symmetric key was found to be non-practical due to challenges it faced for key management. This gave rise to the public key cryptosystems.

Generation of RSA Key Pair

1. Generate the RSA modulus (n)
- Select two large primes, p and q.
- Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

2. Find Derived Number (e)
- Number e must be greater than 1 and less than (p − 1) (q −1).
- There must be no common factor for e and (p − 1)(q − 1) except for 1. In other words two numbers e and (p – 1)(q – 1) are coprime.

3. Form the public key
- The pair of numbers (n, e) form the RSA public key and is made public.
- Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

4. Generate the private key

- Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.
- Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).

We will Store agents data and public private key to the database

We will add the Main Agent which will observe all system and which have full authority about the sub agent.
By Using Gaia Methodology we will add the Roles, Responsibilities, Permissions to the agents.
The Roles Model
The roles model identifies the key roles in the system. Here a role can be viewed as an abstract description of an entity's expected function.

VI. DFD DIAGRAM

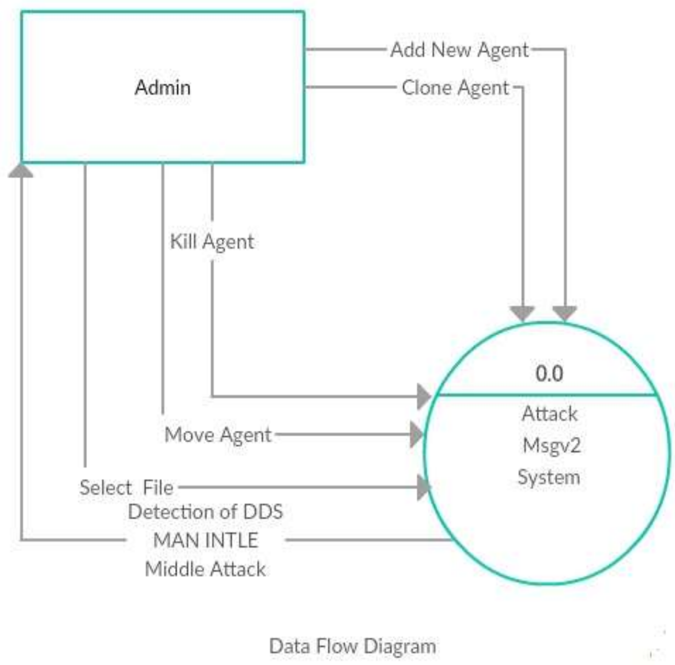


Fig. 3 Data Flow Diagram

VII. SEQUENCE DIAGRAM

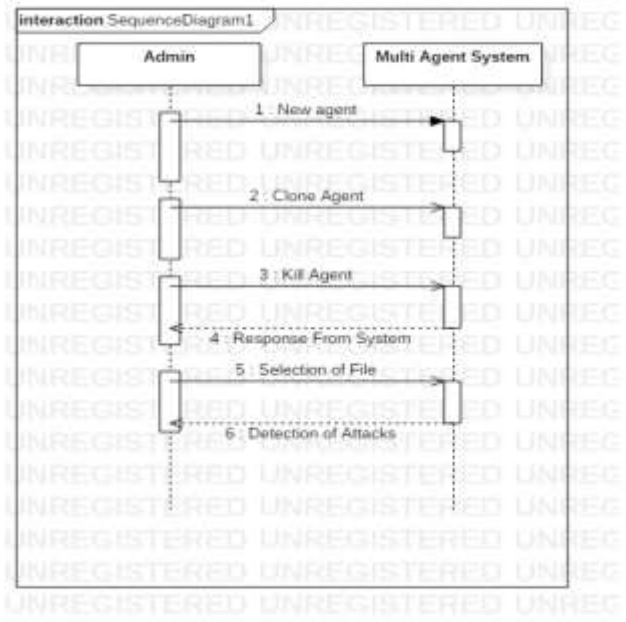


Fig. 2 Sequence Diagram

Responsibilities

Role has certain Functionality i.e Responsibilities of role

Our Contribution
First we detect the Attack
Second we also prevent the Attack

□ Blowfish is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneider and included in a large number of cipher suites and encryption products.

□ Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. (Bruce Schneider)

□ Blowfish was designed in 1993 by Bruce Schneider as a fast, free alternative to existing encryption algorithms.

□ It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use.

VIII. FUNCTINALITIES

1. Node Registration
In this module we register nodes means in real time network machines come in the network we use MySQL database to store the nodes in the database we use RSA algorithm for generation public and private key for nodes
2. To Find Dos Attack and calculate stability
Here if any agent selects the file we detect Denial of Service Attack if Present
3. To Find Data Alteration Attack and Calculate Stability
Here if any agent selects the file we detect Data Alteration Attack if Present
4. to Find Man in the Middle Attack
Here if any agent selects the file we detect Man In the middles Attack if Present
5. To Calculate ARE, ARE and Feedback control gains
6. Block Detected Agents
7. Make Preventive System
8. Analysis of Detection and prevention System
We can calculate throughput, overhead of two system and then make analysis

IX. MATHEMATICAL MODEL

1) RSA Algorithm

The various observations just stated form the basis for the *RSA public-key cryptosystem*, which was invented at MIT in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman.

The public key in this cryptosystem consists of the value *n*, which is called the *modulus*, and the value *e*, which is called the *public exponent*. The private key consists of the modulus *n* and the value *d*, which is called the *private exponent*.

An RSA public-key / private-key pair can be generated by the following steps:

1. Generate a pair of large, random primes *p* and *q*.
2. Compute the modulus *n* as *n* = *pq*.
3. Select an odd public exponent *e* between 3 and *n*-1 that is relatively prime to *p*-1 and *q*-1.
4. Compute the private exponent *d* from *e*, *p* and *q*.

(See below.)

5. Output (n, e) as the public key and (n, d) as the private key.

Key Pair *Public key: n = 55, e = 3* *Private key: n = 55, d = 7*			***Key Pair Generation*** *Primes: p = 5, q = 11* *Modulus: n = pq = 55* *Public exponent: e = 3* *Private exponent: d =* 3^{-1} mod 20 = 7			
Message	***Encryption*** $c = m^3 \bmod n$		***Decryption*** $m = c^7 \bmod n$			
M	m^2 mod n	m^3 mod n	c^2 mod n	c^3 mod n	c^6 mod n	c^7 mod n
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	4	8	9	17	14	2
3	9	27	14	48	49	3
4	16	9	26	14	31	4
5	25	15	5	20	15	5
6	36	51	16	46	26	6
7	49	13	4	52	9	7
8	9	17	14	18	49	8
9	26	14	31	49	36	9

The encryption operation in the RSA cryptosystem is exponentiation to the e^{th} power modulo n:

$$c = \textsc{Encrypt}\,(m) = m^e \bmod n \;\; .$$

The input m is the *message*; the output c is the resulting *ciphertext*. In practice, the message m is typically some kind of appropriately formatted key to be shared. The actual message is encrypted with the shared key using a traditional encryption algorithm. This construction makes it possible to encrypt a message of any length with only one exponentiation.

The decryption operation is exponentiation to the d^{th} power modulo n:

$$m = \textsc{Decrypt}(c)=c^d \;\bmod n \qquad .$$

The relationship between the exponents e and d ensures that encryption and decryption are inverses, so that the decryption operation recovers the original message m. Without the private key (n, d) (or equivalently the prime factors p and q), it's difficult (by CONJECTURE 6) to recover m from c. Consequently, n and e can be made public without compromising security, which is the basic requirement for a public-key cryptosystem.

The fact that the encryption and decryption operations are inverses and operate on the same set of inputs also means that the operations can be employed in reverse order to obtain a digital signature scheme following Diffie and Hellman's model. A message can be digitally signed by applying the decryption operation to it, i.e., by exponentiating it to the d^{th} power:

$$s = \textsc{Sign}\,(m) = m^d \bmod n \;\; .$$

The digital signature can then be verified by applying the encryption operation to it and comparing the result with and/or recovering the message:

$$m = \textsc{Verify}\,(s) = s^e \bmod n \;\; .$$

In practice, the plaintext m is generally some function of the message, for instance a formatted one-way hash of the message. This makes it possible to sign a message of any length with only one exponentiation.

Figure 1 gives a small example showing the encryption of values m from 0 to 9 as well as decryptions of the resulting cipher texts. The exponentiation is optimized as suggested above. To compute m^3 mod n, one first computes m^2 mod n with one modular squaring, then m^3 mod n with a modular multiplication by m. The decryption is done similarly: One first computes c^2 mod n, then c^3 mod n, c^6 mod n, and c^7 mod n by alternating modular squaring and modular multiplication.

Proof :

The proof of this fact is left as an exercise to the reader. *Hint:* Show that the result holds modulo p and q separately, i.e., that for all m,

$$m \equiv (m^e)^d \bmod p$$

Result :

$$m \equiv (m^e)^d \bmod q$$

The result will then follow via the Chinese Remainder Theorem.

2) Blowfish Algorithm

There are two parts to this algorithm;

A part that handles the expansion of the key.

A part that handles the encryption of the data.

The expansion of the key: break the original key into a set of subkeys. Specifically, a key of no more than 448 bits is separated into 4168 bytes. There is a P-array and four 32-bit S-boxes. The P-array contains 18 32-bit subkeys, while each S-box contains 256 entries.

The encryption of the data: 64-bit input is denoted with an x, while the P-array is denoted with a Pi (where i is the iteration).

Key expansion

Blowfish has a 64-bit block size and a key length of anywhere from 32 bits to 448 bits (32-448 bits in steps of 8 bits; default 128 bits).

It is a 16-round Feistel cipher and uses large key-dependent S-boxes. It is similar in structure to CAST-128, which uses fixed S-boxes.

The diagram to shows the action of Blowfish. Each line represents 32 bits. The algorithm keeps two subkey arrays: the 18-entry P-array and four 256-entry S-boxes.
The S-boxes accept 8-bit input and produce 32-bit output. One entry of the P-array is used every round, and after the final round, each half of the data block is XORed with one of the two remaining unused P-entries.

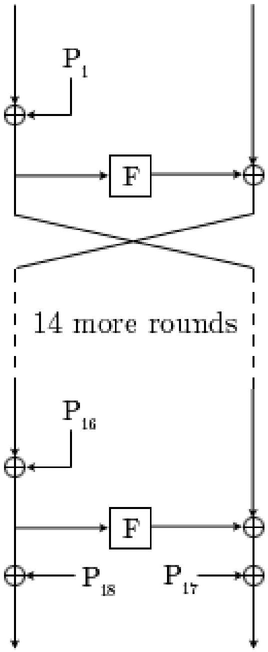


Fig. 4 Key Expansion Diagram

Initialize the P-array and S-boxes
XOR P-array with the key bits. For example, P1 XOR (first 32 bits of key), P2 XOR (second 32bits of key), ...
Use the above method to encrypt the all-zero string
This new output is now P1 and P2
Encrypt the new P1 and P2 with the modified subkeys
This new output is now P3 and P4
Repeat 521 times in order to calculate new subkeys for the P-array and the four S-boxes

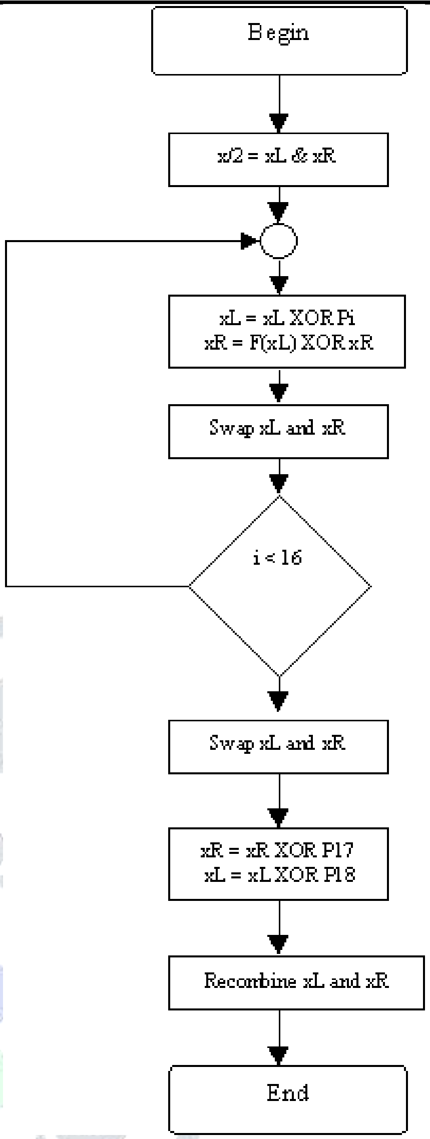


Fig. 5 Blowfish Algorithm

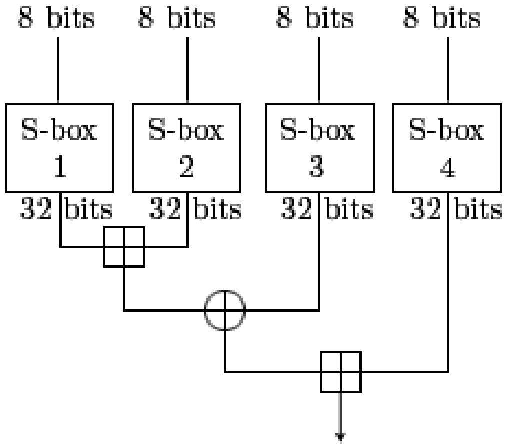


Fig. 6 Blowfish Function

The diagram to the right shows Blowfish's F-function. The function splits the 32-bit input into four eight-bit quarters, and uses the quarters as input to the S-boxes. and XORed to produce the final 32-bit output.

Since Blowfish is a Feistel network, it can be inverted simply by XORing P17 and P18 to the ciphertext block,

Result :

There is no effective cryptanalysis of Blowfish known publicly as of 2005, although the 64-bit block size is now considered too short, because encrypting more than 232 data blocks can begin to leak information about the plaintext due to a birthday attack.
Despite this, Blowfish seems thus far to be secure. While the short block size does not pose any serious concerns for routine consumer applications like e-mail, Blowfish may not be suitable in situations where large plaintexts must be encrypted, as in data archival.

X. CONCLUSION

This research presented a model for securing multi-agent systems, which is developed, based on the concepts and models regarding agent's role and communications. The model provides an efficient way to assure that security requirements and design are integrated with system functionalities during the development process. Moreover, possible attacks on multi-agent systems are presented and threats are categorized. Certain general security requirements at the agent and the system levels are defined and considered in the proposed model. The research has taken into consideration and addressed a system level threats, such as corrupted mobile agents attack the main system host, fake agent, and insecure communication among the platforms as well as the agent level threats, i.e. agent authentication, fake message, modification of agents' interaction by altering the transferring information, message injection, and unauthorized access to agents.

XI. FUTURE SCOPE

- Agents, devices and information sources connected in large scale networks have to share information in effective ways, so as the right information to reach the right agents at the appropriate time, for agents to integrate and interpret data to perform the necessary tasks.
- The distribution, diversity, volatility of data, and, in many emerging applications ubiquity of information sources, make the information sharing task a challenging task. This is important in many real-world settings, where voluminous information from different sources need to reach distant agents.
- The problem becomes even more challenging when agents have different "views" for the meaning of the information they share, when they have to manipulate heterogeneous data from different sources, or when they have to jointly control actuators for which they do not share a common representation. Also, sometimes, information by multiple sources needs to be pre-processed, before being propagated to the right agents: The later may need specific information to be, for instance, extracted, implied, abstracted, or somehow aggregated, in different ways.
-

XII. REFERENCES

[1]. Ruveena Singh, Dr. Balwinder Singh "Design and Development of Smart Waste Sorting System".

[2]. Narayan Sharma, NirmanSingha, TanmoyDutta "Smart Bin Implementation for Smart Cities"

[3]. AmruthaChandramohanet. al. "Automatic waste segregator".

[4]. Kumar, L.M. et. al. "Embedded wireless- enabled low cost plastic sorting system for efficient waste management".

[5]. Twinkle Sinha, K. Mugesh Kumar, P.Saisharan "Smart Dustbin".

[6]. M.K. Pushpa, Aayushi Gupta et. al. "Microcontroller Based Automatic Waste segregator".

[7]. BURT KALISKI ,THE MATHEMATICS OF THE RSA PUBLIC-KEY CRYPTOSYSTEM