Effective Key Distribution Scheme using Paillier Cryptosystem in Wireless Sensor Networks

Paresh R. Jadhav, Shubhangi D. Sapkal

Abstract: ***Main problem is when setting up Wireless(W)-Sensor(S)-Network(N) intended for node communication arises for security. Discussed here is a three-tier architecture with two polynomial pools containing wireless routers, key generation center and few access points which are also sensor nodes that can be mobile devices that intend to get better security. The crucial feature of three tier architecture authentication, where communication between WN to access(A) point(P) is established from AP to the sensor node, resulted by pair(P)wise(W) key(K) pre-distribution (PD) methodology and the nodes are authenticated utilizing polynomial keys and Paillier cryptosystem algorithm. Presently, the WN attacks the duplication, such as seeing the nodes in the network. In the event that a nasty node is found and if you want to send packets within the network, you need to store many of keys from both pools for validation. But because there are no sufficient keys available and therefore can't communicate with other nodes in the network. This paper describes an effective contrivance for accomplishing security between node communications by formulating three-level security architecture.***

Index Terms: ***Wireless (W)Sensor(S) network(N), Paillier Cryptosystem Algorithm, Pairwise Pre-distribution Key Scheme, Cuckoo Filter, Bloom Filter***

I. INTRODUCTION

As advanced electronic technology, recently paved the way and developed a new (WSN), which consist of many low-power sensor nodes that have the ability to communicate without a wired medium. These SN as it may be utilized in applications, like military monitoring with follow-up, health control and residential control. The information that's examined usually has to be returned to base(B) station(S) for analysis. In case, where field is further from the BS, the process of sending the data over huge distances utilizing multi(M)-hop(H)allows to reduce the advantage of safety. The Schemes use two separate polynomials: the mobile polynomial and the constant polynomial pool. Mobile(M)Polynomial(P) groups are used to create and authenticate between mobile(M) sinks(S) and existing access(A) nodes(N). This method allows mobile(M) sinks(S) nodes to access the SN and collect data. At least one polynomial belonging to mobile pool access network should be compromised by the attacker to access and collect sensor data from network. WSN is one example of world of processing that is widespread idea of small and intelligent sensing devices are cheap will eventually absorb the environment. The (WSN) contains extra small sensor nodes and each sensor node is battery-operated device that has the ability to process data automatically with built-in sensors and limited radio memory capabilities for communication. In the application scenario, SNs are used randomly throughout the region and collect data. WSN deployed for various application like tracing as military operations, over-viewing the environmental features, smart environments monitoring, patient monitoring, etc. [1]. When implementing SN, the vital feature to be considered is security and in friendly environment they should exchange valuable or important information about the environment and the opponents should utilize information to their advantage. The main weakness may result as an attack where spurious data can be injected in the network and the attacker can then disguise itself as network node. For security, communication and encryption are required. Proactive research is performing the security key setting in WSN effectively. Setting a key for safe and secure communication is an important management issue. In a typical network environment, there are three important agreement forms: reliable server schemes, self(s) control schemes with pre(P)-distribution(D)schemes(S). A reliable server layout has a reliable server among two nodes to negotiate keys shared among nodes. The central server is not present in most WSNs so that this kind of format cannot be done on the sensor(S) network (N). Self-control schemes utilize public(P) key(K) algorithms(A), like Diffie-Hellman or RSA. The pre(p)-distribution(d) format uses a secret key to create a pair keys after being deployed.

Revised Manuscript Received on July 05, 2019.

Paresh Rajendra Jadhav, Department of Computer Science and Engineering, Government College of Engineering, Aurangabad 431005, India.

Dr. Shubhangi Deorao Sapkal, Department of Computer Science and Engineering, Government College of Engineering, Aurangabad 431005, India.

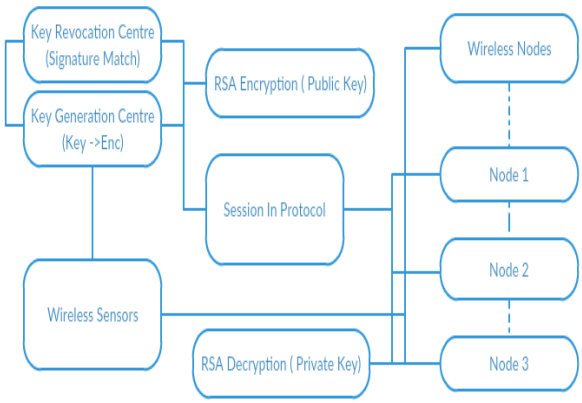

Fig.1 Basic Wireless Node Security Process

Wireless networks send request messages to SN via a static access node. Messages that request the data from wireless network will start a static access(A) node(N) to run the sensor(S)[3][6] node(N) that transmits information to the mobile sink that is requested in use. Schemes use two separate polynomials: constant polynomial and polynomial. By utilizing two distinct key pools and some sensor nodes bring keys from the mobile key group that will help the attacker to initiate the attack, simulating the wireless network on the sensor(S) network(N) more difficult by capturing nodes [2][3]. Although the security methods will make network more flexible in attacking but wireless network replication is compared with a single polynomial primary key distribution model [3]. Creating pairwise coding is important as a part of basic security as it allows sensor(S) nodes(N) to communicate using encryption techniques. Main problem about creation of a general security key is with respect to communication sensor nodes. However, with resource constraints on the sensor node, its' impossible for them to generate traditional dual key with approach like public key encryption (PKE) and zero key distribution (ZKD).

II. RELATED WORK

Seung-Hyun Seo, Salmin Sultana et.al [1] presented important advance distribution patterns that are likely to be established recently for important partners. The idea here is to allow each SN to arbitrarily select a key set against the group of keys afore deploying, therefore two nodes have the possibility of utilizing one key as a commonly shared key i.e., a Trojan key [2][3], and an extended group. Think of this as developing 2 key distribution methods: q-composite key pre-distribution patterns and random key matching schemes. Pre-distribution of key combinations q also uses key groups. For basic probability and pre-distribution patterns of key combinations q when the amount of nodes being compromised raises, the fraction of the affected pair will increase rapidly. As a result, the amount of compromised nodes may reveal extensive fractional keys and still, receive compelling security under small attacks at a more risky price for large-scale attacks. The issue of authentication and the accomplishment of intelligent keys in the sensor(S) network(N) with MS are not resolved when faced with wireless network replication attacks. There are disadvantages among security and vulnerabilities that must be examined the basis on the size of SN and applications.

A. Paillier Cryptosystem

From the viewpoint, the concept of encoding with Paillier [4][15] consists of separate exponential explanations and the creation of sound factors used to hide text. For continuous strengthening, it is known that performance can increase with the base processing power of a certain base. With this adjustment and other methods known, this process can be made much easier. For the creation of audio factors, we use a new method that consists of using pre-calculated sounds to create new sound factors. This will help reduce the bottlenecks of sound calculations as multipliers. When these methods are combined, the efficiency of the encryption will increase dramatically.

B. Key Generation and Secure Choice of Parameters

Since the security of Paillier cryptosystem depends on integer factoring, the same conditions should be considered for n for the modulus size in the RSA [8][10][11] cryptosystem [8][10][11] which is recommended as bits 3072 to 2048 according to NIST instructions.

$$gcd(L(g\lambda mod n2), n) = 1$$

When selecting the parameter g, it is necessary to check whether the selected g command is a multiple of n. In addition, according Paillier [4][15] g, it should be miniscule for performance such as g = 2 in the operations described we choose to work with instead of using other g values.

III. PROPOSED METHOD

The scheme proposed uses two different polynomial pools: a static and dynamic polynomial pool [7]. The polynomial from movable polynomial group is utilized to create authentication with mobile device and the sensor node. Accessing these addresses which will facilitate the mobile (M) sinks(S) network to access the sensor(S) network (N) to collect data. Therefore, the attacker must collect at minimum one polynomial using the mobile pool in order to access the network and collect sensor [7] information. Polynomials [4][5][6][15] from the pool of static polynomial are utilized to concur the authorization and key verification between the static access node and the sensor node. Before deploying wireless networks, each network will randomly select a subset of polynomials from the mobile polynomial group. The pre-selected sensor nodes are known as live access nodes. These are pre-determined authentication point for the network and they make use of sensor nodes to transmit their summed data to the respective mobile sink. Wireless networks transmit request messages to the sensor nodes using a static access node [2][3][4][9]. The sink synch request message will start a static access node to invoke the sensor node to send the collected data to the sink that is requested by the sensor node. From a static polynomial group, the sensor nodes select a random subset of polynomials that can be used with a mobile polynomial with a portable receiver by live access nodes. The benefits of utilizing altogether different group is to monitor the wireless network that is not

completely dependent on the key distribution model used to connect the sensor(S) network(N) [11][14]. This format is divided into two steps: static and mobile distribution and important discoveries between wireless networks [13][15] and sensor nodes.

A. *Mobile node and stationary node Key Discovery phase*

To create a dual smart key between the sensor node u and the wireless network v to find a static access node a. In that area, the node can generate key pairs with both wireless networks v and sensor node u In other words, the access node that depends on the need to create a key pair along with the wireless network and sensor node must search for a mobile polynomial with a wireless sensor node and general polynomial with the sensor node[17]. To find a general mobile / fixed polynomial, the sensor node, I may publish a list of polynomial codes, or in other words, the encoding list α, EKv (α), v = 1. Potential pairwise [11][14][6] keys and other nodes may be available as suggested in [4] and [5],[16],[19]. When creating a secure route between u and v nodes, the wireless network will send the pair Kc key to node a in an encrypted and authenticated message with the shared key Kv, which is a between v and a. If the node a receives the message above and it shares the pairwise key with u, it will send the Pairc [18][19] Kc key to u node in the message that is encrypted and authenticated with Ka. The key pair code between a and u if Failure generate code directly to a wireless network and sensor nodes will need to create a key pair up with the help of other sensor nodes.

B. *Cuckoo Filter*

The cuckoo filter [20] While the filter is a powerful area of query services such as "if the item x is in the set?" They do not support deletion. Variance of activation, often called more space. Cuckoo gives you the flexibility to add and remove dynamic items. The cuckoo filter depends on the cuckoo sound. (And named as a cuckoo filter). It is a hash table that holds the fingerprints of each key. Cuckoo's hash table [12][20] is very small, so cuckoo filters can take up less space than Bloom filters [20] in general for applications that require low false positive rates (<3%).

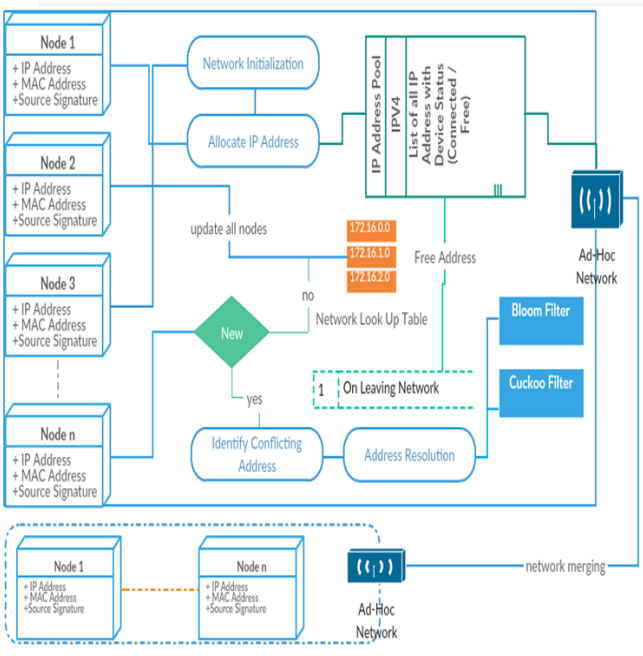

Fig.2 Proposed Architecture

C. *Flow of proposed system*

1) Create Adhoc Network
2) Load Connected Devices
3) Obtain Allocated but Inactive Devices
4) Obtain Free Pool IPV4 Addresses
5) Check for Conflicting IP's and Network Leave Join Scenario
6) Add all existing nodes and non-existing nodes to bloom filter
6) Add all existing nodes and non-existing nodes to cuckoo filter
7) Check for contains, contains all and deletion (only in cuckoo) filter.

D. *Pseudocode for Cuckoo Filter*

```
// create
CuckooFilter<Integer> filter = new CuckooFilter.Builder<>(Funnels.integerFunnel(), 2000000).build();
// insert
   if (filter.put(42)) {
   print("Insert Success!");
      }
      // contains
   if (filter.mightContain(42)) {
   print("Found 42!");
      }
      // count
print("Filter has " +filter.getCount() + " items");
// count
print("42 has been inserted approximately " + filter.approximateCount(42) + " times");
// % loaded
print("Filter is " + String.format("%.0f%%", filter.getLoadFactor() * 100) + " loaded");
// delete
   if (filter.delete(42)) {
   println("Delete Success!");
      }
}
```

The cuckoo filter allows for deletion, such as the filter count. While the count of Bloom filters counts continuously, using more space for the removal of Cuckoo filters [20] is free of charge or time. As with counting various styles of Bloom filters, Cuckoo filters have a limit on the number of times you can insert duplicates. This limit is 8-9 in the current design, depending on the internal status. Access to this limit may cause additional insertions to fail and reduce the efficiency of the filter. Occasionally repetition will not reduce filter efficiency. But will reduce production capacity slightly. Existing items can be deleted without affecting false positive rates or causing false deletions. However, deleting items that have not been added to the previous filter may cause a false rejection. The cuckoo filter [20] supports counting items such as filter counts, blossoming. The maximum number is still limited by a value of up to 7 values, so it should be used to count only small numbers. The measured amount may be higher than the actual number due to false positives. But will not decrease because Cuckoo's filter does not have a negative value When the filter reaches capacity (put () returns false), it's best to create an existing filter or create a

larger filter. Deleting items in the current filter is an alternative. But you should delete at least ~ 2% of the items in the filter before inserting again.

E. Using Hashing Technique

Hash(#) collision attacks[20] can be theoretically done with a cuckoo filter. (Same as the basic structure of the hash table) If this is a problem for your application, use one of the cryptographic hash functions (but slower). The hash function, by default, Murmer3 does. Safe, secure functions include SHA and Sip Hash. All hash including unsafe will be sown and salted. Practical attacks on these things are unlikely. Please note that the maximum filter size supported depends on the hash function. In particular, in the case of Murmur3[20], a 32-bit hash limits the size of the table, even if using a 32-bit hash. The size of table is maximum up to 270 megabytes, with 64-bit hash using 128+ bit hash functions. The library refuses to create a table using incorrect configuration. All operations are safe for threads. Most will work simultaneously to increase efficiency. Important exceptions are copying, serialization, and hash codes, which are required to lock the entire table - run on a single thread until complete.

IV. RESULTS AND DISCUSSIONS

Paillier Cryptosystem [2][4], along with intelligent key pair distribution schemes, such as the primary key set, which consists of many names that are used in this system. For key generation, it means creating a static polynomial pool and a dynamic polynomial pool using the above-stated algorithm. Therefore, this is important for data transactions caused by encryption(enc) and decryption(dec). Of the network key of the network requires information as requested. Therefore, wireless (W) networks(N) collect data from sensors. But cannot communicate directly because it is a three-layer communication added for the safety of the wireless network [6] is a node that has an IP address of the sensor that needs information. In addition, files can be saved according to the specifications. Retrieving the key that occurred at this stage is the key encrypted by the sensor that should decode the wireless network that uses the key. Then, only communications will be done successfully. In addition, the sensor also comprehends information to receive specific requests for specific information, select data from memory and start key retrieval, such as encryption. Of the key, then forward the data and rest for the access node to be at the location or the wireless network to capture the wireless network information. The flag is ready to use and the person with the ability to decrypt the encrypted data can do so when he receives the information.

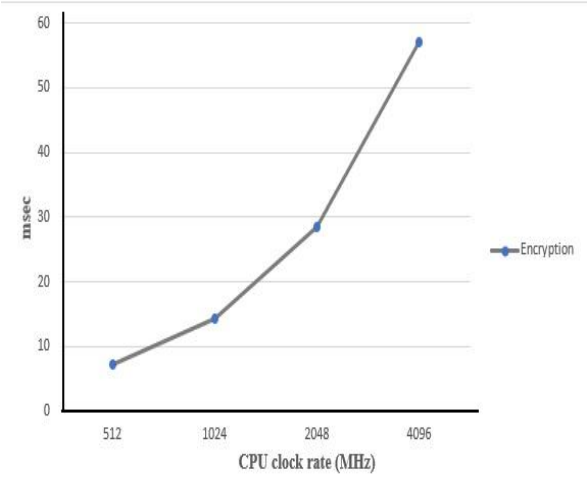

Fig.3 Encryption

key generation step because it is the most time-consuming portion in each of the schemes. Our scheme consists of 512-bit length of key cipher with 64 bits of certainty value. Initial capacity for key cipher is 256 bits that can be increased by (existing capacity * 2) + 2. Average time for key generation is 66.3ms. We have used Intel Core i3-2500M CPU clocked at 2.30 GHz with 2 Giga Bytes of memory installed and 1.85 Giga Bytes of usable memory. We have used Windows 7 with 32 Bit Platform to simulate our model and obtain encryption, decryption results along with key generation values that are mentioned below,

Table I. Encryption Decryption With Key Generation Time

CPU Clock Rate (MHz)	Encryption (msec)	Decryption (msec)	Key Generation (msec)
512	7.13	5.64	14.75
1024	14.26	11.28	29.5
2048	28.52	22.56	59
4096	57.04	45.12	118

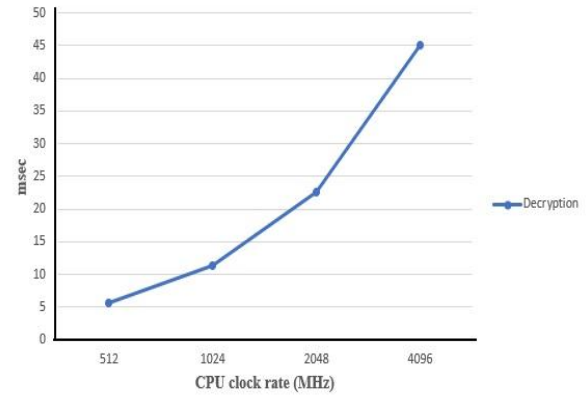

Fig.4 Decryption

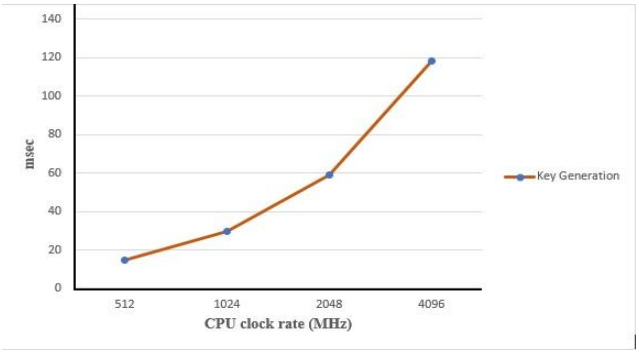

Fig.5 Key Generation

V. CONCLUSION AND FUTURE SCOPE

The larger pool size will reduce the chances of two distinct nodes sharing same key. The amount of data attributes that can be associated with each sensor relies on the size of the Wireless(W) Sensor(S) Network(N) that can improve the network connectivity. Two types of pool based [12] polynomial key allocation with the Paillier Cryptosystem Algorithm [14], significantly improving network flexibility for wireless(W) network(N) replication attacks. Keeping communication costs intact indicates that the creation of encrypted keys and decoding times is less when compared to encryption systems and systems that are more flexible in denial of service attacks which is used for Paillier algorithm which is very difficult because there are not many important numbers. We have observed that there is a lot of scope to improve bloom filter with respect to space complexity and there is lot of room to improve the look up value of bloom filter when it tried to search for a finger print of device and match with new one. Also, there is lot of scope to work on variable key length for networks that support devices from various facades with ipv4 and ipv6 addresses. Implementing ECC is also to be explored along with Paillier cryptosystem.

REFERENCES

1. Seung-Hyun Seo, Salmin Sultana," Effective Key Management in Dynamic Wireless Sensor Networks", IEEE Transactions on Information Forensics and Security, Vol. 10, NO. 2, FEBRUARY 2015
2. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor(S) network(N) s: A Survey," Computer Networks. 38, no. 4, pp. 393-422, 2002.
3. L. Hu and D. Evans, "Using Directional Antenna to Prevent Wormhole Attacks," Proc. Network and Distributed System Security Symp., 2004.
4. J.R. Douceur, "The Sybil Attack," Proc. First Int'l workshop Peer-to-Peer Systems (IPTPS '02), Mar. 2002.
5. B.J. Culpepper and H.C. Tseng, "Sinkhole Intrusion Indicators in DSR MANETs," Proc. First Int'l Conf. Broadband Networks (Broad-Nets'04), pp. 681-688, Oct. 2004.
6. H. Deng, W. Li, and D.P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," Proc. IEEE Comm. Magazine, pp. 70-75, 2002.
7. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor(S) network(N) s," Proc. MobiCom, pp. 56-67, 2000.
8. A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, "Intelligent Fluid Infrastructure for Embedded Networks," Proc. Second ACM Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '04), June 2004.
9. H. Chan, V. Gligor, A. Perrig, G., and Muralidharan, "On the distribution and revocation of cryptographic keys in sensor networks," IEEE Trans. Dependable Secure Computing. 2005, 2, pp. 233–247.
10. H. Chan, A. Perrig, and D. Song, "Random key pre-distribution schemes for sensor networks," In Proceedings of IEEE Symposium on Security and Privacy (SP '03), Washington, DC, USA, 2003, p. 197.
11. S. Chattopadhyay, and A.K. Turuk, "A Scheme for Key Revocation in Wireless Sensor Networks," Int. J. Adv. Comput. Eng. Communication Technol. 2012, 1, pp. 16–20.
12. Y. Jiang, and H.A. Shi, H.A., "Cluster-Based Random Key Revocation Protocol for Wireless Sensor Networks," J. Electron. Sci. Technol. China 2008, 6, pp. 10-15.
13. G. Dini, and I. Savino, "An efficient key revocation protocol for wireless sensor networks," Proc. of International Symposium on a World of Wireless, Mobile and Multimedia Networks 2006, Buffalo, NY, USA.
14. P. Chuang, S. Chang, C. Lin, "A Node Revocation Scheme Using Public-Key Cryptography in Wireless Sensor Networks," J. Inf. Sci. Eng. September 2010, 26, pp. 1859-1873.
15. Y. Wang, B. Ramamurthy, and X. Zou, "KeyRev: An Efficient Key Revocation Scheme for Wireless Sensor Networks," In Proc. of IEEEICC, Glasgow, UK, 24–28 June 2007, pp. 1260–1265.
16. G.N. Purohit and A.S. Rawat, "Revocation and Self-Healing of keys in Hierarchical Wireless Sensor Network," Int. J. Comput. Sci. Inf.Technol. 2011, 2, pp. 2909-2914.
17. C. Wang, T. Hong, G. Horng, and W. Wang, "A key renewal scheme under the power consumption for wireless sensor networks," In Proceedings of the 4th International Conference on Photonics, Networking and Computing, Kaohsiung, Taiwan, 8–11 October 2006.
18. G. Wang, S. Kim, D. Kang, D. Choi, and G. Cho, "Lightweight key renewals for clustered sensor networks," J. Netw., 2010, 5, pp. 300–312.
19. G. Jolly, M. Kusçu, P. Kokate, and M. Younis, M., "A Low-Energy Key Management Protocol for Wireless Sensor Networks," In Proceedings of the Eighth IEEE International Symposium on Computers and Communications, Kiris-Kemer, Turkey, 30 June-3 July 2003.
20. Fan, *Cuckoo Filter: Practically Better Than Bloom*. Retrieved August 22,2016 from https://www.cs.cmu.edu/~dga/papers/cuckoo-conext2014.pdf

AUTHORS PROFILE

Paresh Jadhav Completed B.E. in Computer Science and Engineering, Master of Engineering in Computer Science and Engineering from Government Engineering college, Aurangabad.
Email Address- pareshjadhav550@gmail.com

Dr. Shubhangi Sapkal Completed B.E. Computer Science and Engineering, Master of Engineering in Computer Science and Engineering and Ph.D. in Computer Science and Engineering. Current designation is Assistant Professor of Computer Science and Engineering in Government Engineering College working from year 2000, Aurangabad, Contributing research work on Cryptosystem for multimodal biometric for template security, Information security and neural network.