ISSN 2047-3338

Acknowledge Enabled Secure Algorithm for Dynamically Updating Programs Installed in Wireless Sensor Nodes

Vishwa Pratap Singh, Kishore Mishra, Jay Shankar Sharma and Ankit shirivastava
Indian Institute of Information Technology, Gwalior, India
Apex Institute of Engineering and Technology, Jaipur, Rajasthan, India
Vishwa.iiit@gmail.com, Kmishra16@gmail.com, jay.maharasi@yahoo.com, shirivastava20109@gmail.com

***Abstract*– There are several programs installed in wireless sensor nodes, time to time these programs need to update in secure manner, such that any unauthorized person could not update the programs. Sensor nodes have several constraints so we cannot use conventional cryptographic algorithms. In this paper we have proposed an algorithm for updating programs installed in wireless sensor nodes in secure manner. We have used distributed key approach and ECC. We have eliminated the shortcoming, "one sided communication between sensor nodes and base station", present in algorithm from paper secure dynamic program update protocol for ZigBee using ECC by Vishwa Pratap Singh and Ashwini Saini and other weakness in a Dynamic Program Update Protocol for Wireless Sensor Networks by Peng Zeng, Zhenfu Cao, Kim-Kwang Raymond Choo, and Shengba Wang in IEEE transactions.**

***Index Terms*– Wireless Sensor Networks, ZigBee, ECC and Distributed Key Approach**

I. INTRODUCTION

WIRELESS sensor networks comprised of several battery powered sensor nodes deployed in hostile environment. Wireless sensor networks [1] are designed in such a way that once sensor nodes are deployed they will work for years without any manual physical maintenance. These nodes are very large in number so the cost of each node should be very low. These nodes run on battery so efficient use of battery is our basic necessity. Several programs are installed in these nodes at the time of deployment and sensor nodes work according to these programs; these programs need to be update time to time. Manually updating programs in these nodes is not a feasible task and updated using wireless communication in secure manner, such that any attacker will not able to update programs installed in these nodes for any evil purpose.

We cannot use conventional cryptography because of two basic reasons. First reason is these nodes are very cheap and temper proof hardware will increase the cost of each node, the attacker can easily retrieve the cryptographic keys stored in nodes. If attacker get successful in capturing the node and retrieved all keys from that node; whole of the security architecture will fail. Second is conventional cryptography requires high computation power and sensor nodes are not rich in computation capability. High computation need large amount of energy and sensor nodes are deficient in it as they runs on lithium ion batteries. Algorithms proposed for updating programs in senor nodes have mainly two shortcomings:

a. They are storing some secrets in sensor nodes, and if temper proof hardware is not used they can be easily compromised.
b. One way Communication- Proposed algorithms are not able to send secure acknowledgment to the base station.
c. Limited Number of updates.

We are proposing scheme based on ECC [7] and distributed key approach for updating programs in sensor nodes dynamically even if attacker can capture the sensor nodes.

II. DISTRIBUTED KEY APPROACH AND ELLIPTIC CURVE CRYPTOGRAPHY

Chien-Wen Chiang, Chih-Chung Lin proposed a scheme which used implicit security to partition the key to many parts. And then the base station assigns these key partitions to every nodes of the network. When a node wants to return the data to the base station, node will recreate these key partitions to an encrypted key and uses this key to encrypt the data. The scheme can reduce the overhead of computing and storage of nodes, hence it can save time and the energy of encrypting data. The scheme is appropriate for WSN due to the scheme can extend survival time of the network.

Approach: This paper proposes a scheme of key distribution which partitions the key into k parts and distributes these partitions over the nodes on WSN [2]. When data want to return to the base station, it will reconstruct key partitions of the path to be a new encrypted key. The node uses this key to encrypt data by symmetric encryption. As our scheme takes the most part of computation on the base station, sensor nodes take lower computation to reducing the encrypting time and the number of keys. It increases the survival time and decreases the response time of the network.

Elliptic curve cryptography (ECC) [6] is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.ECC generates keys through the properties of the elliptic curve equation instead of the

traditional method of generation as the product of very large prime numbers. According to some researchers, ECC can yield a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage. For current cryptographic purposes, an elliptic curve is a plane curve which consists of the points satisfying the equation.

$$y^2 = x^3 + ax + b \quad (1)$$

along with a distinguished point at infinity

Discrete Logarithmic Problem [11] in ECC:

In the multiplicative group Zp* [10], the discrete logarithm problem is: given elements r and q of the group, and a prime p, find a number k such that r =qk mod p. If the elliptic curve groups are described using multiplicative notation, then the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number that Pk = Q; k is called the discrete logarithm of Q to the base P. When the elliptic curve group is described using additive notation, the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number k such that Pk = Q.

III. PREVIOUS EFFORTS

Manik Lal Das and Aakash Joshi presents a novel approach for dynamic program update in wireless sensor networks using orthogonality principle [4] .The protocol is based on the mathematical principle of orthogonality. Mathematically, two vectors are orthogonal if their dot product is zero. Given a vector, we can find infinite vectors which are orthogonal to the given vector over that vector space. Suppose, we have a vector, v, then we can find vectors a,b, c and so on, such that v is orthogonal to a, b, c, i.e., v x a = 0, v x b = 0, v x c = 0. We note that a, b, c are orthogonal to v, but they need not to be mutually orthogonal.

Weaknesses:

- The number of updates is limited.
- Algorithm work on a secret stored in sensor node, but in hostile environment if node get compromised, whole of system fails.
- No mutual authentication, so vulnerable to denial of service attack.

Peng Zeng, Zhenfu Cao identify an inherent aw in their design and demonstrate that the Das-Joshi scheme is vulnerable to an impersonation attack, in violation of their security claim [4]. They then present a modified scheme to eliminate the security vulnerability .The algorithm is again based on orthogonality [7] principles, and able to eliminate some vulnerabilities in Das and Joshi scheme.

Shortcomings:

- The proposed algorithm again have same vulnerabilities, it is not able to update sensor node unlimited number of times in secure manner.
- The proposed algorithm is vulnerable to Denial of service attack, if an attacker intercepts and deletes the message, neither the base station nor the nodes will know about it.
- They don't propose any mechanism for mutual authentication.

IV. THE IDEA AND PROCESS OF PROPOSED ALGORITHM

We have proposed algorithm to transfer updates in constraint and restricted environment, where nodes are vulnerable to retrieval of cryptographic keys stored in sensor nodes. We have used lightweight Elliptic Curve cryptography, which use points on elliptic curve to provide security to data and distributed key approach where cryptographic keys are store in distributed manner on several nodes. We have used this approach in a different ways, which we will discuss in later section. We have worked in two steps first sending updates to sensor node and authentication of update, and second sending acknowledgement to base station.

Assumptions:

We have taken the following assumption related to wireless sensor networks:

- Network is configured in tree topology and all nodes are fully functional device, which can send, store and receive data.
- Sensor nodes have small computation power to do basic calculations and have storage capacity enough for storing key data.
- Attacker is able to retrieve the data stored in nodes but not able to change the stored data.
- Server is fully secured and has very high computation power with unlimited storage
- Hash function is installed in nodes.

Algorithm has two phases:

a. Setup Phase
b. Dynamic node update phase

Setup phase:

This is the pre deployment phase. In this phase before deploying the sensor node in field P and Q is installed in the sensor node. Calculation of P and Q is described in following section.

Step 1: Server chooses an elliptic curve $E_L(a, b)$ over GF(2n) with n should be very large. L is a large prime number.
Step 2: Server chooses a point $e_1(x_1, y_1)$ from chosen elliptic curve.
Step 3: Server choose a random number using pseudo random number generator) d.
Step 4: Calculate hash of d to and D.
Step 5: Server calculate

$$e_2(x_2,\ y_2) = D \times e_1(x_1,\ y_1) \quad (2)$$

Step 6: Let consider point

$$e_1(x_1, y_1)\ as \qquad \alpha_1$$

$$e_2(x_2, y_2)\ as \qquad \beta_1$$

Step 7: Server save α_1 and β_1 in the sensor node before deployed in field and keep D_1 to itself. In same manner Base station calculate points on elliptic curve and store them in sensor node.

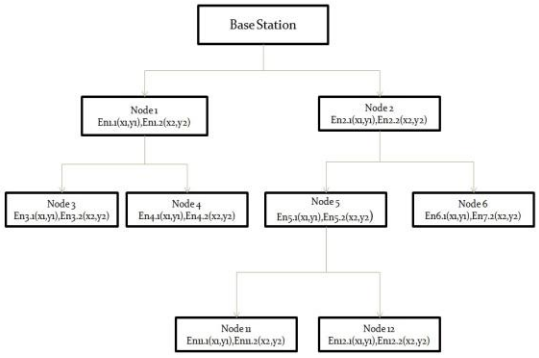

Fig. 1: Stored elliptic curve points on sensor nodes

Dynamic node update phase:

Step1: Server calculates $e_3(x_3, y_3)$, $e_4(x_4, y_4)$ and D_2 in same manner as setup phase for a particular node.

Step2: Let Server wants to send update program M with update number j, id, X_{pid} and program version X_{ver} and T_j time of sending update. Server calculate the hash of

$$(j, t_j, M, X_{pid}, X_{ver}, D_1, \alpha_2, \beta_2) \quad (3)$$

Step 3: Server advertise a message M(j)adv to a particular node.

$$M_{(adj)} = [(j, t_j, M, X_{pid}, X_{ver}, D_1, \alpha_2, \beta_2), H(j, t_j, M, X_{pid}, X_{ver}, D_1, \alpha_2, \beta_2) \quad (4)$$

At node side:

While moving from base station to sensor node, the advertisement message captures intermediate nodes information (path from base station to sensor node id etc.) and append in message.

Sensor nodes receive the M(j) adv and carry out following steps:

Step 1: Node calculate hash of

$$(j, t_j, M, X_{pid}, X_{ver}, D_1, \alpha_2, \beta_2) \quad (5)$$

using Hash function (MD5) stored in sensor node and compare with hash

$$(j, t_j, M, X_{pid}, X_{ver}, D_1, \alpha_2, \beta_2)$$

stored in $M(j)_{adv}$ to check the integrity of the $M(j)_{adv}$, If the calculated hash is equal to the stored hash move to second step, otherwise discard the $M(j)_{adv}$

Step 2: Validate tj with the local current time Clock. If in the equations:

$$|clock - t_j| < \Delta t \quad (6)$$

Holds, then proceed to next step, else reject the message. Here Δt denotes the time of the expected network delay which can be estimated according to different applications.

Step 3: Calculate β_1 using α_1 pre installed in node and D_1 extracted from the $M_{(adj)}$

$$e_2(x_2, y_2) = D \times e_1(x_1, y_1) \quad (7)$$

If calculated β_1 is equal to the installed α_1 in node move to next step otherwise discard advertisement.

Step 4: Install program M in the node and replace the pre installed α_1, β_1 with the new α_2, β_2 n the node.

Step 5: Sensor node send acknowledgement to base station as follows.

Senor node retrieve information of intermediate nodes from the advertisement message and send request to all nodes for hashed value to their points.

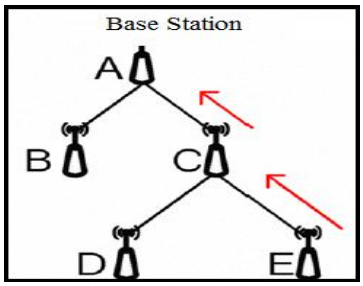

Fig. 2. Sensor node sending request to intermediate nodes for hashed value of points

All the requested nodes send hashed value of their points with request id to their lower requesting node.

Step 7: Sensor node concatenate all the hashes and calculate hash of own points, D and acquired hash from intermediate node and successful or unsuccessful update.

Step 8: Node calculate keyed hash using acquired hash as key and X_{pid}, X_{ver}, successful or unsuccessful bit as body, send it to base station.

Step 9: Base station have all the information about point stored in sensor node and paths. Base station Calculate keyed hash in same manner as node and math with the message got from sensor node.

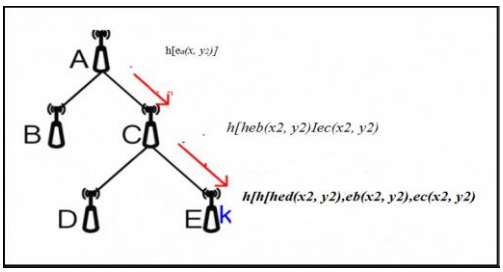

Fig. 3. Retrieving hashed value to points from intermediate nodes

We have implemented the proposed algorithm in java language using jdk1.7, cryptography library BouncyCastle and WiseNet Simulator. We have used merssene twister to generate random numbers because of its very large cycle of 232 that makes the guessing number infeasible. Here Fermat's prime are used to get large prime number as generator in elliptic curve Ep(a, b). We have used the java security package to implement elliptic curve.

Proposed algorithm is simulated using WiseNet simulator and Jcreator. Fig. 4 shows the snapshot we get in WiseNet simulator with 13 nodes and tree topology. We have used WiseNet simulator because of its compatibility with BouncyCastle cryptography library. WisNet simulator support tree topology MAC protocols.

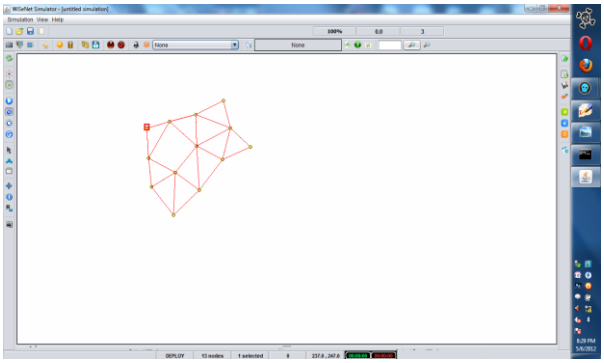

Fig. 4. Snapshot simulating tree topology WiseNet simulator

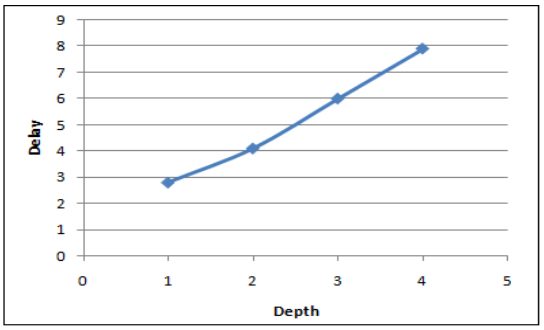

Fig. 5. Delay

Above graph shows the delay we get for acknowledgement. x axis represent the depth of a node for which base station want to update program. Y axis shows the delay we get. We have simulated algorithm using 13 sensor nodes and maximum depth we are getting is 4. The reason for increase in delay is, as we go deeper, packets have to travel more number of nodes and each node hashed their point and then sends to sensor node, which then acknowledges to base station.

V. CONCLUSIONS

Our major findings are listed below:

Unlimited number of updates: We are using elliptic curves to send secure updates to the sensor nodes. We have selected an elliptic curve Ep(a, b) initially and then selecting the points on elliptic curve and multiplying them with random numbers to get another point on the curve and sending the points to the sensor nodes and keeping d as secrete with server.

Security analysis: security of proposed Algorithms lies on the elliptic curves discrete Logarithmic problem, as it is easy to calculate point $e_2(x_2, y_2)$ after multiplication of point $e_1(x_1, y_1)$ with number d. But it is infeasible to get d if we have point $e_1(x_1, y_1)$ and $e_2(x_2, y_2)$ if the generator of chosen elliptic curve EP (a; b) is a very large prime number P. So algorithm immune to forging update as attacker cannot guess d. Update integrity: integrity to update is provided by calculating hash of the message by using 160 MD5 and appends it with the broadcasted message.

- Update Replaying: algorithm is immune to message replaying as a sequence number and clock is added to the broadcast.
- Two way communication: A proposed algorithm is able to reply base station about success of updation in secure manner.
- Immunity to message replaying attack.

Shortcomings:

- Increase delay in reply.
- Only work tree topology.
- In pre deployment phase all nodes have to configure with different elliptic curve points which is a time consuming if done manually and number of nodes are very large.

Table 1: Comparisons with previous efforts and proposed algorithm

Margin	Dynamic program update in wireless sensor networks using orthogonality principal	Security weakness in a dynamic program update protocol for wireless sensor networks	Secure dynamic program update protocol using ECC for ZigBee	Proposed Algorithm
Proposed By	Manik Lal Das and Aakash joshi june 2008	Peng Zeng, Zhenfu cao	Vishwa Pratap Singh And Ashwini	
Based on	Orthogonality	Orthogonality	ECC	ECC and Distributed key Approach
Security if data get retrieved from nodes by attacker	No	Yes	Yes	Yes
Number of Updates	Limited	Limited	Unlimited	Unlimited
Secure two way communication	No	No	No	Yes

ACKNOWLEDGMENTS

I am very thankful to Mr. Subodh Kumar Chaturbedi and Rakesh Kumar for reviewing this paper and helping me with his important suggestions.

REFERENCES

[1]. Gutierrez, J.A. and Naeve, M. and Callaway, E. and Bourgeois, M. and Mitter, V. and Heile, B, "IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks," Network, IEEE, Vol. 15, No.15, pp.12–19, 2001.

[2]. IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006), pp: 1-203, 2007.

[3]. Radmand, P. and Domingo, M. and Singh, J. and Arnedo, J. and Talevski, A. and Petersen, S. and Carlsen, S., "ZigBee/ZigBee PRO security assessment based on compromised cryptographic keys," P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2010 International Conference on,pp.465-470,2010.

[4]. Manik Lal Das and Aakash Joshi, "Dynamic Program Update in Wireless Sensor Networks Using Orthogonality Principle", In IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 6, pp 478-481, 2008.

[5]. Peng Zeng, Zhenfu Cao, Kim-Kwang Raymond Choo, and Shengbao Wang" Security Weakness in a Dynamic Program Update Protocol for Wireless Sensor Networks" IEEE COMMUNICATIONS LETTERS, VOL. 13, NO. 6, JUNE 2009.

[6]. Vanstone, S.A., Zuccherato, R.J., "Elliptic curve cryptosystems using curves of smooth order over the ring Zn" Information Theory , IEEE Transactions on july 1997,pp 1231-1237, 1997.

[7]. Sayed, A., "orthogonality principle "Book Adaptive filters press, Wiley IEEE press, pp: 67-77.

[8]. ZigBee, PRO, "Specification, 2007", San Ramon, California: ZigBee Alliance (October 2007), 2007.

[9]. S. Lee, H. Kim, and K. Chung, "Hash-based secure sensor network programming method without public key cryptography," in Proc. The Workshop on world-Sensor-Web at International Conference on Embedded Networked Sensor Systems, 2006.

[10]. Deschamps, J.-P. and Sutter, G.., "Elliptic-Curve Point-Multiplication over GF (2^{163})", Programmable Logic, 2008 4th Southern Conference on, pp: 25-30, March 2008.

[11]. Smart, N.P., "The discrete logarithm problem on elliptic curves of trace one", Journal of cryptology, Vol. 12, No. 3, pp: 193-196, 1999.

[12]. Vishwa Pratap Singh and Divya Pal Singh, "Dynamic program update protocol for ZigBee using ECC", International journal of engineering research and technology, Vol. 6, 2012.

Vishwa Pratap Singh received the B. Tech. Degree in Information technology and M. Tech degree from Indian Institute of Information technology in Computer science specialized in information Security. He is working as Assistant Professor in Computer Science and Engineering Department. His main research area is information security, sensor networks.

Rakesh Kumar received the degree M. Tech from Kurukshetra University in computer science. His major research areas are Database and Web-Mining.