Wireless Network Security via Per Hop Data Encryption through ECC and Secure Authentication with DSA

Mamta Verma[1] Mr. Akash wanjari[2]
[1]Scholar [2]Assistant Professor
[1,2]Department of Computer Science Engineering
[1,2]Disha Institute of Management & Technology, Raipur, India

Abstract— These networks lack security and are defenseless. Encryption and public key cryptography are essential technologies that are used to conserve data security and integrity, and to reduce information stealing on the public networks. However, the exiting routing protocols are inept of providing secure data transmission on wireless networks. To provides more security transmission data using certificate authentication of digital signature. We say that the cryptography using for data secure transfer in network we use ECC (Elliptic curve cryptography) implemented a prototype of per-hop data encryption protocol on the ns-2 simulator.

Key words: Per Hop Data Encryption, Wireless Network Security

I. INTRODUCTION

It is a familiar fact that the Internet traffic travels through wireless network. These networks need security and are vulnerable. Encryption and public key cryptography are significant technologies that are used to preserve data security and integrity, and to reduce information theft on the public networks. However, the existing routing protocols are unable of providing secure data transmission on wireless networks. To this end, our laboratory introduced the Service oriented Router (SoR) to uphold rich information for the next-generation networks by shifting the current Internet infrastructure to an information-based and an open modernism platform. An SoR can analyze all packet stream transactions on its interfaces and store them in high throughput databases. Using the features of the SoR, we propose a hop-by-hop routing protocol that provides per-hop data encryption. This infrastructure is proposing to preserve both the security and the privacy of data that traverses through public networks. We implemented a prototype of per-hop data encryption protocol using the ECC(Elliptic Curve Cryptographic).that provide the for further security, we will implement the Certification Authority (CA)-based authentication DSA(digital signature algorithms) architecture to authenticate SoRs, clients, and applications.

- A dispatcher wants to send a Hello message to a receiver.
- The original message, also called plain text, is changed to random bits known as cipher text by using a key and an algorithm. The algorithm being used can generate a different output each time it is used, based on the value of the key.
- The cipher text is transmitted over the transmission medium.
- At the receiver end, the cipher text is renewed back to the original text using the same algorithm and key that was used to encrypt the message.

A major worry of communication is the security in data transmission. In many cases, the speed and dependability for transmitting a message with low probability of errors is not enough, but also the transmission has to be permitted in an extremely secure way. In 1949, C.E. Shannon took a decisive step toward showing that if the length of the key is not an inconvenience, a message can be securely sent [6]. Diffie-Hellman Key Exchange (DHKE) protocol is use to swap a cryptography key between two parties that have no previous knowledge of each other and to establish a key exchange between them over apprehensive communication channels. Since the router is a key device for interconnecting networks, a router can access any kind of information integrated in packet streams: packet header information, L2 and L3 information, application information, etc. a new model of Internet infrastructure based on the Service-oriented Router (SoR) [8] To support the necessity of cryptographic key management for secure Internet communication, the Internet Key Exchange (IKE) protocol[9]. The best known cryptographic problem is that of privacy: preventing the illicit extraction of information from communications over an insecure channel[10]. the authors present an authentication protocol for exchanging encrypted messages via an authentication server based on elliptic curve cryptography using El Gamal's algorithm. They chose El Gamal's algorithm for encryption/decryption and, also, for the authentication. The authors motivated this by the fact that using two or more algorithms in the same protocol makes it more vulnerable. They, also, pointed out that using a digital signature algorithm is more secure, but they Opt not to use one because the presented protocol has a big advantage: the receiver does not need to know the senders public key [2]. A short description of the protocol is presented below.

Sending Process (From = Alice; To = Bob) :
A ! S : CS(From; To;N1)
S ! A : CA(CB;N1;N2; Id)
A ! S : CB(M;N2); Id
Receiving Process(Bob receives the message) :
B ! S : CS(To;N3)
S ! B : CB(M;N2);CB(N2;N3)

First, the user A sends a request to the server S consisting in the nickname of the sender (From), the nickname of the receiver (To) and a random message (*N*1). This random message is used to verify the server. These elements are encrypted with the server's public key and sent to the server. After decryption, the server encrypts a mail Id, *N*1, a new *N*2, the receiver's (B) public key with A's public key, and sends the result to the sender (A). The sender *N*1 to see if the message received from the server is legal Next, A encrypts the clear message *M* and *N*2 with B's public key and sends the encryption to S along with the Id. To download his message, B send his nickname and a new Anomalous message *N*3 encrypted with S's key. The server S sends back to B *M* and *N*2 encrypted with B's key along with *N*2 and *N*3, also encrypted with B's key. To verify if the encryption received is from the server S, B checks *N*3. In

[1] are introduced two authentication protocols which use two diverse algorithms: one for encrypting the message and one for generating the key. In both cases, the encryption algorithm is a symmetric one. The authors Counsel the protocol presented in the paper is recommended for the wireless communication, but it does not mean that they cannot be adapted for other communications. They both use the Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication. This algorithm is described below. In [5] it is presented an authentication rank protocol based on elliptic curve infrastructure.. the digital signature algorithms, has three steps:

A. Key Generation

- The ¯rst user selects a random integer *k*1 *2* [2*; n ¡* 2];
- Then the user computes *Q* = *k*1*P*;
- The public and the private key of the user are (*E; P; n;Q*) and *k*1,respectively.

B. Signature Generation

- The user selects an integer k2 2 [2; n ¡ 2];
- Then computes k2P = x1; y1 and r = x1 mod n, if r = 0 then return to step (a).
- Compute k2 -1mod n;
- Compute s = k2 -1(SHA(m)+k1 r) mod n, if s = 0 then return to step (a);
- The signature for the message m is (r; s).

C. Signature Verification

- The second user, after receiving the signature (r; s) computes c = s-1 mod n and SHA(m);
- Then computes u1 = H(m)c mod n and u2 = rc mod n;
- Then computes u1P + u2Q = (x0; y0) and v = x0 mod n;
- If v = r then the signature is valid.

The ECDSA advantages and disadvantages can be studied in [9]. After the verification, the user and the server have to generate a secret key for the encryption. For the protocol presented in [1], generating third key is made through a scalar addition, while in others is used a public-key algorithm. Studying these two protocols we chose a zero knowledge authentication for our protocol. We will prompt this choice in. proposed an elliptic curve authentication protocol in [1]. The protocol uses ECDSA for the authentication and the Diffie-Hellman key exchange scheme to establish session key.

II. METHODOLOGY

A. Per-Hop Data Encryption

There are two methods of encryption to be exact link encryption and end-to-end encryption. In the link encryption or online encryption, all data found along a link is encrypted despite of its content or the protocol. In this method, the payload, headers, and trailer are encrypted as a complete. Therefore, the packets must be decryption at each hop to permit the router or other intermediate device to know where to send the packet next. The router must decrypt the header segment of the packet, read the routing and address information contained in the header, and then re-encrypt the information and send it more. However, in end-to-end encryption, only the payload will be encrypted leaving the headers and trailers legible. This leads to vulnerabilities in learning sensitive information such as the sender details, the destination of the packet, or the type of data it is moving. In case of end-to-end encryption, decryption and encryption of packets at each hop is not compulsory, because the headers and trailers are not encrypted. We use end-to-end encryption in techniques such as ECC

B. Diffie-Hellman (DH) Key Exchange Algorithm

The proposed protocol uses the recognized Diffie-Hellman key exchange algorithm to exchange a symmetric key between neighbors. The shared key can then be used to communicate securely through encryption. To incept the DH key exchange, both the parties should agree on two non-secret numbers. The first number, denoted by m, is the generator, and the second number, denoted by n, is the modulus. These numbers can be transmitted through the public media, and the protocol uses two random prime numbers to denote them. For this process, the protocol uses the inbuilt rand() function to generate a random number and then verifies the primality of the generated number. After generating m and n, both parties share the numbers between themselves through the connected link. In the next step, each party generates their private random secret value ***x***. Then, based on m, n, and ***x*** both the parties generate their public secret value Y using the following formula:

$$Y=m^x \bmod n \qquad (1)$$

Using the above formula, the two parties will generate and exchange their shared secret value. Each party then exponentiates the received public value with the corresponding secret value to compute a common shared-secret value S using the following formula:

$$S=y^x \bmod n \qquad (2)$$

In the above formula, ***x*** refers to their respective private random secret value whereas *Y* and *p* are the shared values between themselves. At the end of this process, both the parties will generate the same shared secret *S*. Any attacker or sniffer listening on the channel will not be able to compute the secret value because even if the m, n, and the public secrets of both parties are known, at least one of the private random secret values is required to generate the common shared-secret value. Because only the communication parties know the private random secret values, it cannot be derived from any of the obtained values; the DH algorithm can create a shared secret value securely using the shared m, n, and Y values. Unless the attacker can compute the discrete algorithm of the above-mentioned equation to recover the x value of either ends, the attacker cannot obtain the shared secret value.

C. Elliptic Curve Cryptography

The authentication protocols based on classic cryptography use public-key crypto systems for establishing the common key. Some of them have been proven to be secure but they require high amount of resources and they need large keys. Applying an elliptic curve authentication protocol the memory and the power consumption are lower. Another advantage is that this kind of protocols is secure Behoove even if a small key is used.

D. DSA digital signature algorithm

National Institute of Standards and Technology brought forward digital signature standard (DSS) in 1991. This

standard adopts digital signature arithmetic (DSA). There are 3 parts in DSS, message abstract, code and decode. The security of DSA lies on the discrete logarithm problem. Message abstract generates by SHA. Digital signature generates by message abstract which has the similar security as Breed by message [4]. p, q, g, y, z are the parameters used in DSA signature. P is prime number between 512bit and 1024bit. q is prime gene of p. $g = t\ p$=1 mod p (1< t < p-1). z is private key, and it's a random number between 0 and 160. $y = g\ z$ mod p . y is the public key generate by z. We can create digital signature by the parameters. The digital signature arithmetic is scheduled arithmetic[12].

DSA arithmetic:

Input: message abstract $H(m)$

Output: signature pairs(r, s)

S1: generate a random number $k(0 < k < p)$,

S2: compute $r = (g^k \bmod p) \bmod q$

S3: compute $s = k^{-1} (H(m) + zr)) \bmod q$;

Return (r, s)

Then, it sends the signature (r, s) to receiver. Receiver needs to validate the signature. The validate arithmetic is listed.

III. RESULT

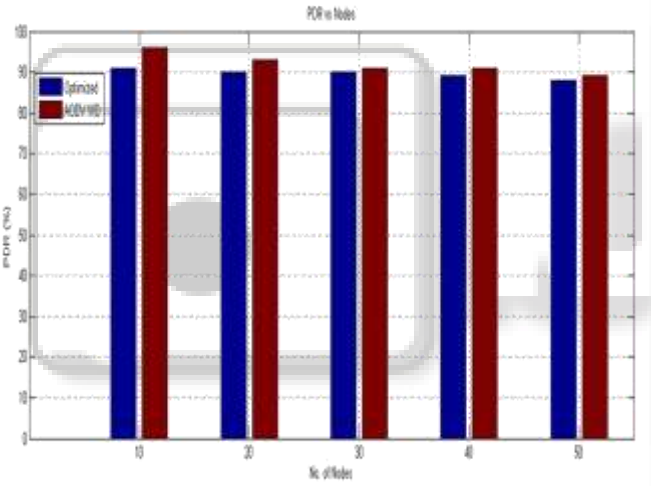

Fig 1: result of optimized AODV

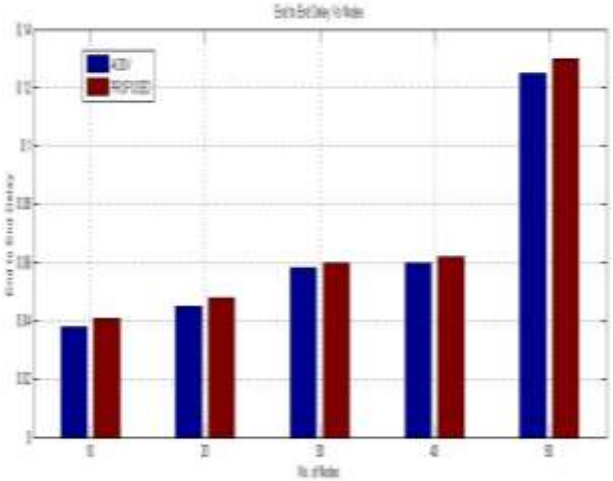

Fig 2: result end to end Delay vs nodes

A. Complexity Computation:

In proposed method performance of each node is calculated and analyzed individually. estimate is done by increasing node number and network size, But proposed method have some limitation .If number of node is increasing or network size is increasing then cost of project increases.

B. Energy Consumption:

From the comparative study it is Comply that energy consumption by using good neighbor node is more than the AODV protocol. For transferring data amidst source to destination node, select only good neighbor node Good nodes have maximum signal strength and flow capacity. So use that node to transfer data fast.

C. End to End Delay:

The end-to-end delay is the average time between data packets sent out from the sources and received at the destination. The delay can be Profess with respect to number of nodes and mobility rate. As the mobility rate increases the end-to-end delay is always increases because the network topology changes more frequently. So selecting good node and create route to decrease end to end delay.

D. Packet Delivery Ratio:

Packet delivery ratio is the ratio of the data packets received at the destination to the data packets sent Alfresco from source. The delivery ratio can be denoted by mobility rate. As the mobility rate increases, the delivery ratio always decreases.

E. Small Key Size:

key size or key length is the size measured in bits of the key used in a cryptographic algorithm (such as a cipher). An algorithm's key length is distinct from its cryptographic security, which is a logarithmic measure of the best known computational attack on the algorithm, also calculated in bits. The security of an algorithm cannot exceed its key length but it can be slighter Most symmetric key algorithms in common use are designed to have security equal to their key length. No asymmetric key algorithms with this property are known elliptic curve cryptography comes the closest with an effective security of roughly half its key length.

IV. CONCLUSION

A per- hop data encryption protocol that can be used to transmit sensitive data over the wireless network.the proposed data encryption protocol can simultaneously provide both secured routing and ordinary routing according to the user requirement and using certification Authority based digital signature algorithms that provide more security to the data .moreover the test result showed. We proposed reliable AODV routing protocol which enhances network performance by selecting stable nodes (i.e, only good neighbor nodes) for network formation This will improve network performance. We presented a relatively simple protocol for a group communication based on elliptic curve cryptography. The protocol has a low complexity mainly because the group's members have the same key pairs, but also because the authentication is made through zero knowledge. Using elliptic curve cryptography provides a

methodology for obtaining high-speed implementations of authentication protocols and encrypted message techniques while using fewer bits for the keys. For establishing the encryption/decryption keys we chose a method were no point on the elliptic curve is made public. Keeping the elliptic curve point private increases the security of the algorithm. This method is also easy to implement, being based on the characteristics elliptic curves and RSA encryption systems.

REFERENCES

[1] M. Aydos, B. Sunar and C. K. Koc, An Elliptic Curve Cryptography based Authentication and Key Agreement Protocol for Wireless Communication, Proceedings of the 2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Dallas (1998),
[2] C. Boyd and A. Mathuria, Protocols for authentication and Key Establishment, Springer-Verlag,2003.
[3] Certicom Research, SEC 2: Recommended Elliptic Curve Domain Parameters, Standards for efficient Cryptography, Version 1.0, Sept. (2000).
[4] K. Chalkias, G. Filiadis and G. Stephanides, Implementing Authentication Protocol for Ex- changing Encrypted Messages via an Authentication Server based on Elliptic Curve Cryptography with the El Gamals Algorithm, IEC (Prague), (2005), 137{142.
[5] N. Constantinescu, Authentication ranks with identities based on elliptic curves, Annals of the University of Craiova, Mathematics and Computer Sciences Series 34 (2007), 94{99.
[6] N. Constantinescu, Elliptic curve cryptosystems and scalar multiplication, Annals of the University of Craiova, Mathematics and Computer Sciences Series 37 (2010), no. 1, 27{34.
[7] N. Constantinescu, The GN-authenticated key agreement, Journal of Applied Mathematics and Computation, Elsevier 170 (2005), no. 1, 531{544.
[8] U.S. Dept of Commerce/NIST, Digital Signature Standard (DSS), FIPS PUB 186-2, Jan. (2000).
[9] IEEE P1363. Standard specifications for public-key cryptography. Draft version 7, Sept. (1998).
[10] Rajitha Tennekoona*, Janaka Wijekoonb, Erwin Harahapc, Hiroaki Nishid a,b,cHiroaki Nishi Laboratory,Department of Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522,Japan
[11] IEEE Standards Association (2011), Media Acces Control (MAC) Security. [Online] Available:
[12] Janaka Wijekoona,1,, Erwin Harahapa, Hiroaki Nishia" Service-oriented Router Simulation Module Implementation in NS2 Simulator" aHiroaki Nishi Laboratory, Graduate School of Science and Technology, Keio University, Japan 2013.
[13] Deng Jian-zhi, Cheng Xiao-hui, Gui Qiong" Design of Hyper Elliptic Curve Digital Signature" Department of Electronics and Computer Science 2009.
[14] K. Inoue, D. Akashi, M. Koibuchi, H. Kawashima and H. Nishi, "Semantic router using data stream to enrich services", 3rd International Conference on Future Internet Technologies (CFI08), Seoul, Korea, June-2008
[15] Janaka Wijekoon, Erwin Harahap, Hiroaki Nishi, Service-oriented Router Simulation Module Implementation in NS2 Simulator, Procedia Computer Science, Volume 19, 2013, Pages 478-485, ISSN 1877-0509.

