

Authenticated Mutual Communication between two Nodes in MANETs

Dega Ravi Kumar Yadav , K. Nikitha Reddy , N. Vamshi Krishna

*Department of CSE,
Vignana Bharathi Institute of Technology
Ghatkesar, Hyderabad, India.*

Abstract— In today's world, we always try to be safe and protected when we communicate with others. But when Mobile Ad-Hoc Networks (MANETs) are considered, they are known to be unprotected due to the nature of message propagation and their openness. For data security we can use Symmetric key cryptography but it has various problems in MANETs like key distribution, key management and scalability problems. Public key cryptography provides solution for the problems raised in Symmetric key cryptography with authenticated key agreement protocols. But implementation of PKC based protocols in MANETs is very difficult because of its constraints like mobility of nodes. In this paper, we propose a new hybrid authentication protocol for MANETs. This protocol uses RSA algorithm concept for authentication between two nodes and protocol security is based on elliptic curve discrete logarithmic assumption. This protocol is energy efficient and uses low memory as we use the concept of Elliptic Curve Cryptography (ECC).

Keywords—Elliptic Curve Cryptography, Two-party Authentication, Key Exchange, Trusted Third Party, RSA Algorithm, Security.

I. INTRODUCTION

One very important property that a Mobile ad-hoc network should exhibit is organizing and maintaining the network by itself. The major activities that an MANET is required to perform for self-organization are neighbour discovery, topology organization, and topology reorganization. MANETs should be able to perform self-organization quickly and efficiently in a way transparent to the user and the application.

The security of communication in MANETs is very important, especially in military applications. The lack of any central coordination and shared wireless medium makes them more vulnerable to attacks than wired networks. In MANETs a node should be capable of having a secure communication with other nodes. For this the node should build a security tunnel of its own. Symmetric key algorithms can be used for building security tunnels but the problem with this is the key pre-distribution among the communicating nodes because key pre-distribution among nodes raises the problem of scalability and key maintenance.

To overcome the problems of Symmetric key algorithms design of Public key cryptography (PKC) based key agreement protocols was proposed. Two-party PKC based

key agreement protocols can establish efficient security tunnels among communicating nodes in MANETS. But these two-party key agreement protocols are susceptible to active attack such as man-in-middle attacks like key compromise impersonation attack.

For two parties to communicate securely over a public network they must be able to authenticate one another and agree on a secret encryption key. Key establishment protocols are used at the start of a communication session in order to verify the parties identifies and establish a common session key. If a private key is compromised then the attacker can impersonate the "corrupted" party to other entities, because entities are identified precisely by their private key. This is known as "key compromise impersonation" attack.

II. RELATED WORK

To propose the described new protocol we need to know about RSA and ECC. RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Here in our protocol we only use the RSA algorithm for key generation and we do not decrypt or encrypt anything here.

Elliptic Curve Cryptography (ECC) is a public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key where as the public key is distributed to all users taking part in the communication.

Every public key cryptosystem requires a set of predefined constants to be known by all the devices taking part in the communication. In the case of elliptic curve cryptography "Domain parameters" are the constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography.

The domain parameters of elliptic curve are a sextuple:

$$T = (P, a, b, G, n, h)$$

An elliptic curve over a field K is a curve defined by an equation of the form

$$y^2 = x^3 + ax + b$$

Where a, b ∈ K and 4a³ + 27b² ≠ 0.

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large.

k is the discrete logarithm of Q to the base P.

III. PROPOSED PROTOCOL

The proposed system uses the concept of RSA and ECC to build a security tunnel among two nodes in MANETs. It is as follows, initially using the concept of RSA a token is generated which is pre- distributed among the two nodes which are in need of communication by the Trusted Third Party (TTP). And once the token is distributed among the source and destination node, the role of TTP ends. TTP doesn't involve in further communication.

On receiving the token from TTP the source node uses the token to calculate session key. And using that session key of source node HMAC is calculated at the source node. And destination node also follows the same procedure and calculates its own HMAC.

Now a authentication request message is sent to destination node to source node which consists of the token given by TTP. On receiving the request from source node destination node verifies received token with its token. If both the token's match destination node replies to source node with a reply message which consists of HMAC of destination node. Source node verifies both HMAC's on receiving the reply message from destination node and if both HMAC's match replies to destination node by an acknowledgement message.

The detailed steps for proposed protocol:

Consider a TTP and communication is between node A and node B. And notations used in protocol.

Q	A large prime number.
AReq	Authentication Request Packet
BRes	Authentication Response Packet
P	Point on elliptic curve
A	Long term secret of node A
B	Long term secret of node B
SKAB	Session key generated between node A and B
SKBA	Session key generated between node B and A

Step 1: TTP calculates the token and distributes it among node A and node B.

Step 2: On receiving token from TTP. Node A selects rA randomly, where 1 ≤ rA ≤ q - 1 and then computes QA = rA ·

P. And node A sends Authenticated request packet AReq (tokenA, QA).

Step 3: After receiving AReq message, node B first verifies node A's token. If the A's token is verified using the B's token given by TTP.

Step 4: Node B selects randomly an integer rB in the range 1 ≤ rB ≤ q-1 and computes QB = rB · P. It then computes SKBA = H ((rB+b) · (QA+PubA)) as a session secret key between A and B.

Step 5: Node B computes HMACB = H(SKBA ||H((QA.x + QB.x)||QA.y + QB.y)). It then constructs a message m consists of HMACB and QB, that is, m = HMACB||QB and generates a signature sigB (m) on m as sigB (m) = (r,s) using the private long-term key b of B with the help of ECDSA signature generation algorithm. Node B finally sends BRes(m, sign (m)) as an authentication reply message to node A.

Step 6: After receiving BRes message, node A first verifies the signature sigB (m) using the public key of node B with the help of ECDSA signature verification algorithm. Node A then computes SKAB = H ((rA+A) · (QB+PubB)) as a session secret key between A and B. And then calculates

HMACA = H (SKAB ||H ((QB.x + QA.x)||QA.y + QB.y))

Step 7: Node A compares both HMACA and HMACB for integrity check and if the check holds then as an initiator node A sends an authentication acknowledgement message to node B. In this way both node A and node B use the secret key future communication.

IV. IMPLEMENTATION

The following pseudo code shows the implementation of the proposed system.

A. Authentication request message

```
Algorithm AReq (tokenA, QA) {
    If tokenA=tokenB then
        Calculate SKBA using tokenB
        Calculate HMACB using HMACB
        Compute message m
        Calculate sigB (m)
    }
```

B. Authentication response message

```
Algorithm BRes (m, sigB (m)) {
    Calculate SKAB using tokenA
    Calculate HMACA using SKAB
    }
```

C. Authentication acknowledgement message

```
Algorithm Ack ( ) {
    Compare SKBA=SKAB
    Compare HMACA=HMACB
    If both conditions satisfy then node A and B are authenticated
    }
```

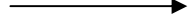


Node A	Node B
1. AReq (tokenA, QA) 	Verifies A's token. If verification is successful, computes $SKBA = H((rB+b) \cdot (QA+PubA))$ $HMACB = H(SKBA H((QA.x + QB.x) (QA.y + QB.y)))$ Constructs a message m Generates sigB (m) = (r, s).
Verify B's signature sigB (m). If these hold, computes $SKAB = H((rA+A) \cdot (QA+PubA))$ $HMACA = H(SKAB H((QB.x + QA.x) (QA.y + QB.y)))$	2. BRes(m, sigB (m)) 
3. ack () 	it stores SKBA for secure communication with A.

Fig 1: Working of the proposed protocol

V. ANALYSIS

The proposed protocol satisfies common security properties of a two party authenticated key agreement protocol such as known key security, perfect forward secrecy, key compromise impersonation resilience, unknown key share, implicit key agreement, key confirmation and explicit key confirmation. In the proposed protocol, key of confirmation is achieved at both communicating parties whereas in other protocols key confirmation is achieved at one end. Overall, we conclude that the proposed protocol is efficient compared with the existing protocols.

VI. CONCLUSION

In this paper we have proposed a new protocol Authenticated mutual communication between two nodes in MANETs and whose security is based on Elliptic Curve Cryptography. By implementing this newly proposed protocol in MANETs the attacker had no chance to do man-in-middle attacks like key compromise impersonation attack and the communication between any two nodes is secured. And the proposed protocol is more energy efficient and uses less memory than the existing protocols. Moreover, the newly proposed protocol is scalable and once the authentication is done and a secured tunnel is built between the two communicating nodes they can use the same secret key for their future communication also irrelevant to the network they are present.

REFERENCES

- [1] Ad Hoc Wireless Networks by C Siva Ram Murthy and B S Manoj.
- [2] M. A. Strangio, On the Resilience of Key agreement protocols to Key Compromise Impersonation, EuroPKI'06, vol.4043, pp. 233-247, LNCS, 2006.
- [3] Kavitha Ammayappan, An ECC-Based Two-Party Authenticated Key Agreement Protocol for Mobile Ad Hoc Networks.
- [4] Maurizio A. Strangio, Password-authenticated Key exchange using efficient MACs, Journal of computers, Vol.1, no.8, PP. 27-35, 2006.
- [5] S. B. Wilson, D. Johnson and A. Menezes, Key agreement protocols and their security analysis, Proceedings of the 6th IMA International conference on cryptography and coding, pp. 30-45, 1997.
- [6] H. Z. Liao and Y. Y. Shen, On the Elliptic Curve Digital Signature Algorithm, Tunghai Science, Vol. 8, pp. 109-126, 2006.
- [7] M. A. Strangio, Revisiting an efficient elliptic curve key agreement protocol, Cryptology eprint Archive, IACR, Report 081, 2007.
- [8] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000.
- [9] S. Wang, Z. Cao, M. A. Strangio and L. Wang, Cryptanalysis and Improvement of an Elliptic Curve Diffie-Hellman Key Agreement Protocol, IEEE Communications Letters, IEEE, Vol. 12, Issue 2, pp. 149-151, 2008.