APPLYING PUZZLE ENCRYPTION IN THE ON-DEMAND ROUTING PROTOCOLS IN MOBILE AD HOC NETWORKS (MANETS)

Ahmad Alomari[1]

ABSTRACT.

Recently, the use of Mobile Ad Hoc Networks (MANETs) systems in our life has rapidly increased. Nevertheless, we need for this efficiency and privacy routing protocols to exchange the information between the nodes in this kind of networks. We propose a new scheme to apply it on the On-Demand routing because it is one of the most popular and usable in the MANETs. The main goal in our paper is to promote and improve the authentication between the nodes in the MANETs by applying the puzzle encryption before they start exchange the data packet between them. The new scheme is based on combined use of cryptographic puzzles and weakly secret bit commitment (WSBC) function. The scheme has to offer privacy protection of the confidential information stored in the nodes, that is identifier ID and encryption puzzle. The identifier allows the unequivocal identification of the nodes on the mobile networks. The anonymity of messages is crucial to avoid traceability and replay attacks and denial services. The puzzles function use one way encryption functions whereas the keys must be enough to avoid a brute force attack. We use also the bit commitment function with puzzle encryption in our scheme to commit a value without revealing that value. This scheme will offer moderate protection concerning privacy and traceability when a node communicates with each other in the Mobile Ad hoc Networks (MANETs).

Keywords: MANET, AODV, RSA, puzzle function, WSBC.

1. INTRODUCTION

A MANET is a set of mobile nodes that can communicate with each other without the use of predefined infrastructure or centralized administration. Becoming one of the fastest growing areas of research, the mobile ad hoc networks (MANETs) also help the proliferation of cheaper, smaller, and more powerful mobile devices. Due to their capacity to self-organize and to deploy rapidly, MANET can be used with different applications including battlefield communications, emergency relief scenarios, law enforcement, public meeting. The *ad hoc* self-organization also makes them suitable for virtual conferences, where setting up a traditional network infrastructure is a time-consuming high-cost task.

MANETs security solution main goal is to provide security services to the mobile users. These services include authentication, confidentiality, integrity, anonymity, and availability. To achieve this goal, the security solution should provide complete protection spanning the entire protocol stack. We can classify MANET security in fives layers, such as Application layer, Transport layer, Network layer, Link layer, and Physical layer. However, the only layer related to security issues is the network layer, so we only focus on

[1] Ph.D. Student, Faculty of Mathematics and Computer Science, University of Bucharest, Bucharest, Romania, alomari.jordan@gmail.com

it to protect the ad hoc routing and forwarding protocols. From the security design perspective, the MANETs have no clear line of defense. Unlike wired networks that have dedicated routers, each mobile node in an ad hoc network may function as a router and forward packets for other peer nodes. Routing in ad hoc networks has become a popular research topic. Dating back to the early 1980s, there have been a large number of routing protocols designed for multi-hop ad hoc networks. These protocols cover a wide range of design choices and approaches, from simple modifications of Internet protocols, to more complex multilevel hierarchical schemes. Many of these routing protocols have been designed based on similar sets of assumptions. For instance, most routing protocols assume that all nodes have homogeneous resources and capabilities. This includes the transmission ranges of the nodes. Also, bidirectional links are often assumed.

There are mainly two types of ad-hoc routing protocols: first, Proactive routing protocols, where the nodes keep updating their routing tables, by sending periodical messages. We have, for example, OLSR (Optimized Link State Routing protocol) and TBRPF (Topology Broadcast based on Reverse Path Forwarding). Second: Reactive (On Demand) routing protocols, where routes are created only when needed. We have, for example, DSR (Dynamic Source Routing protocol) and AODV (Ad hoc On-Demand Distance Vector Routing protocol) and we choose the on-demand routing protocol to apply our scheme. In some instances, protocols have mechanisms for determining whether links are bidirectional. In these cases, the protocols will then eliminate unidirectional links from consideration for routing. In other instances, protocols can actually utilize these unidirectional links, whereas other protocols simply assume all links are bidirectional. Recently, several researches were introduced to counter these malicious attacks. Most of the previous work has focused mainly on providing preventive schemes to protect the routing protocol in a MANET. Most of these schemes are based on key management or encryption techniques to prevent unauthorized nodes from joining the network. In our paper we proposed a new scheme to apply it on the On-Demand routing protocols because it is the most popular and usable in the MANETs.

The new scheme is based on combined use of crypto-graphic puzzles and weakly secret bit commitment (WSBC) function. The scheme has to offer privacy protection of the confidential information stored in the nodes, that is identifier ID and encryption puzzle. The main objective in our paper is to promote and improve the authentication between the nodes in the MANETs by applying the puzzle encryption before they start exchange the data packet between them. The anonymity of messages is crucial to avoid traceability and replay attacks and denial services. The puzzles functions use one way encryption functions whereas the keys must be enough to avoid and resist many kinds of attacks like eavesdropping, blackhole, wormhole and brute force attacks.

The paper has been organized in sections. Section I: the introduction. Section II contains related and previous works and a short review of on-demand routing protocol. Section III speaks about Weakly Secret Bit Commitment (WSBC) function. Section IV explains some puzzle constructions. In Section V we introduce Puzzle Authentication scheme for on-demand routing protocol.

2. RELATED WORKS

Researchers have attempted to resolve the security concerns related to the use of routing protocols of MANETs and have proposed protocols that claim either to achieve secure authentication or to prevent unauthorized traceability. There are many works that improve the security routing protocols especially in On Demand routing protocols.

In 1996, Rivest, Shamir and Wagner [1] thwarted the related problem of an attacker capable of solving parallel puzzles by means of time-lock puzzles. A time-lock puzzle introduces a computational problem that cannot be solved without continuously running for a precise amount of time, i.e., encrypting some material (e.g., an encryption key) with the result of repeatedly squaring a value with respect to a composite module. Therefore, these puzzles can be used to implement delays, i.e., by setting the amount of computation, the amount of delay controlled. Timelock puzzles use fixed-cost functions, based on super-encryption in RSA, and trapdoors.

Furthermore, Back's hash-cash system [2], first announced in 1997, mainly deals with email spam, and denial of service (DoS) attacks as well, by applying a trapdoor-free non-interactive POW system. A cost-function based on finding partial SHA-1 hash collisions assists in filtering email clients. Implementations (developed in several programming languages) modify the message's header by adding a hash-cash token, which consists of several recipient-related data such as an address, a timestamp and a random nonce.

Jinsong Han, Yunhao Liu [3] proposed Mutual Anonymity for Mobile P2P Systems. This is a scalable secret-sharing-based mutual anonymity protocol, named PUZZLE, which enables anonymous query issuance and file delivery for Mobile Peer-to-Peer Network (MOPNETs) in ad hoc environments by employing Shamir's secret sharing scheme. They presented the design of PUZZLE, analyze its degree of security and anonymity, and evaluate its performance by comprehensive trace-driven simulations. PUZZLE employs Shamir's Secret Sharing (SSS) scheme [4] to anonymously issue queries and utilizes the Information Dispersal Algorithm (IDA) [5] to achieve anonymous downloading. PUZZLE leverages the broadcasting feature of MANETs to achieve mutual anonymity. They introduce reasonable redundancy into MOPNETs to enhance the reliability of this protocol.

SAODV [6]. The Secure Ad hoc On-Demand Distance Vector protocol was proposed to answer the challenge of securing a MANET network. As an extension of the AODV routing protocol, SAODV can be used to protect the route discovery mechanism by providing security features like: non-repudiation, integrity and authentication. SAODV implies that every ad hoc node possess a signature key pair from an appropriate asymmetric cryptosystem. Next, each node can verify securely the coupling between an address of a given ad hoc node and the public key of that node. The accomplishment of this task falls into the responsibility of the key management scheme.

The Secure Efficient Ad hoc Distance vector protocol SEAD (by Hu, Johnson and Perrig [7]). The hash-chains are also used in SEAD, but this time in combination with DSDV-SQ, for the authentication of hop counts and sequence numbers. This protocol doesn't use

asymmetric cryptosystem, but only one-way hash functions. This means that at every given time each node is equipped with its own hash-chain, a secure ad hoc network routing protocol based on the design of the Destination-Sequenced Distance Vector (DSDV) routing protocol.

On Demand Anonymous Routing in ad hoc networks ODAR (is proposed by Denh Sy, Rex Chen and Lichun Bao). ODAR is similar with AODV [8], following the usual route request and reply mechanism in packet formats. The use of Bloom filters (first used by Castelluccia et al. for compressing source route information after the source route is discovered using DSR [9][10]) gives ODAR the storage, processing and communication efficiencies that make it suitable in the ad hoc network environments. The ad hoc networks based on Bloom filters work as follows: the path discovery process is activated when an outgoing packet coming from the application layer cannot find a route to the destination and starts by the node sending out a RREQ message to the network. If the destination receives the RREQ message, it responds with a route reply message RREP to return the complete source route to the source, which provides link, node and path anonymities.

3. PRELIMINARIES

In this section we present a short description of Weakly Secret Bit Commitment (WSBC) function. We also show the emergence of the cryptographic puzzle and some of puzzle constructions.

3.1 Weakly Secret Bit Commitment (WSBC) function

Bit commitment is a way of requiring a principle to commit to a value without revealing that value, so this kind of function is suitable for our scheme and we will use it beside puzzles encryption [14]. We can take this simple example to explain the function:

Alice generates two random bit strings R1 and R2. She commits to a message M by creating h(R1,R2,M) and sending R1,h(R1,R2,M) to Bob.

When she wants to reveal M to Bob, she sends him R2, M. By the properties of hash functions, Bob cannot determine M from the first message Alice sent.

Also by the properties hash function, Alice cannot find R□ □ □2, M□ such that h(R1,R2,M) = h(R1, R□2, M□). First we note that, as the example illustrates, “Bit commitment” is a slight misnomer. This technique could be used to commit to a single bit, but it obviously can be used to commit to much more.

The idea of WSBC is similar to that of bit commitment. The difference is that we want the secrecy of the bit commitment to be breakable within an acceptable bound on time and/ or computation.

The general properties that a WSBC function W should have:

- 2^{nd}-preimage resistance: Given x, it should be computationally infeasible to find X`≠X such that w(x)= w(X`).
- Weak-preimage resistance: For any pre-specified value y of w it should be moderately hard to compute any x such that y = w(x).
- Collision resistance: it should be computationally infeasible to find any X, X` such that w(x)= w(X`).This is stronger than 2^{nd} - pre- image resistance.
- Near-preimage resistance: given y=w(X), it should be hard to find X` such that X and X` differ by a small number of bits. This is not directly similar to any of the hash function properties of although it is probably related to the non-correlation property.

3.2 The emergence of the cryptographic puzzle and RSA Time-Lock Puzzles

Cryptographic puzzles are puzzles used to hide a secret which can only be revealed after some computational effort has been made. Ralph Merkle [15] conceived the first Puzzle system in 1976 to ensure that two parties can communicate securely over an insecure channel. A computational puzzle is a moderate hard problem, the answer of which can be calculated within a reasonable time and verified efficiently. Such a problem is often given to a service requester to solve before the requested services is provided. The two parties will agree on a shared secret by exchanging messages. A Merkle's puzzle consists of a stack of puzzles in the form of an encrypted message with an unknown key. Puzzles use one-way encryption functions whereas the key must be short enough to allow a brute force attack [16] [17]. Cryptographic puzzles have many applications in the field of security. They are most commonly used in eliminating denial of services attacks and a number of researches also proposed the use of cryptographic puzzles in combating connection depletion attacks.

3.3 RSA Time-Lock Puzzles

Jerschow and Mauve introduce a non-interactive RSA time-lock puzzle scheme [19]; inspired by Rivest's time-lock puzzles [1] it enables an author to commit to a document in an offline manner before the deadline and to submit it at some time past the deadline when being online again. The main idea is to let the author solve a modular exponentiation puzzle involving an arbitrary large number of non parallelizable modular squaring operations. They construct the puzzle from the document's cryptographic hash value. The number of puzzle operations is determined by the time period between the deadline and the point in time where the author regains connectivity to the submission server.

They introduce a time-lock RSA puzzle scheme for delayed encryption and signature verification. The basis of our offline submission protocol is a delayed RSA encryption of the document to be submitted using the institution's public key. Having received the delayed submission, the institution verifies the puzzle solution and the assigned level of difficulty by performing an RSA decryption with its private key. Running the offline submission protocol requires the author to hold a computer with a reasonably up-to-date processor and to continuously solve the puzzle from the expiration of the deadline until the

actual online submission. They integrate the time-lock puzzle mechanism with RSA public-key cryptosystem and make the puzzle non-interactive.

Everyone who knows Alice's public puzzle key can solve a puzzle by encrypting an arbitrarily chosen message m. The puzzle complexity is determined by the size of Alice's public key. Alice constructs her RSA puzzle key pair with the artificially enlarged public key by performing the following steps:

- Generates at random two large prime's p and q.
- Compute the modulus n such that
  N = pq
  As the product of two large randomly-chosen secrete prime p and q. and also computes
  φ (n) = (p-1)(q-1)
- Choose a private exponent d randomly, $1 < d < \varphi(n)$ such that gcd (d; φ (n)) = 1 and determine its multiplicative inverse modulo φ (n): $e = d^{-1} \bmod \varphi(n)$.
- Determines the number of squaring operations modulo n per second, denoted by S and a public key operation shall take T seconds,, that can be performed by the solver Bob, and computes t = T S.
- Compute the remainder $r = 2^t \pmod{\varphi(n)}$
- And the public exponential $e\square = 2^t + \varphi(n) - r + e$; and let $z = \varphi(n) - r + e$ denotes the lower bits of e□ which are preceded by a long sequence of 0-bits and finally the leading 1-bit at position t.
- (n; e□) is the public and (n; d) the private key . Since e□ is an extremely large number with lots of 0-bits after the leading *l*-bit, the public key can be efficiently represented by storing the triple (n; t; z). In binary, z is at most twice as long as n.

The inflated public exponent e□ is constructed by adding a large multiple of φ (n) to the regular exponent e. It holds that $m^e = m^{\tilde{e}} \pmod n$ for all $m \in Zn$ since, $e \equiv e\square \bmod \varphi(n)$ and n is a product of distinct primes. e□ has been chosen to be the smallest appropriate exponent which is larger than 2^t.

Solving the Puzzle and the operation of public and private key:

The receiver (Bob) use the public key (n; e□) of the sender (Alice) to encrypt the contest m, $0 < m < n$, in the usually manner, i.e. to compute the cipher-text:

$$c = m^{\tilde{e}} \bmod n$$

Due to the special structure of e□, the fastest way to perform this giant modular exponentiation is to solve the actual puzzle:

$$\sigma = m^{z'} \bmod n$$

In T seconds by repeated squaring and to quickly do the regular-sized modular exponentiation:

$$\Omega = m^{z} \bmod n$$

This yield:

$$C = \sigma . \Omega \bmod n$$

Bob submits the pair (m; c), i.e., the context and the corresponding puzzle solution, to Alice. She verifies the solution by applying her private key (n; d) in the usual manner to decrypt the ciphertext and to compare the result with m:

$$C^{d} \bmod n \stackrel{?}{=} m$$

Since d is of regular size, this operation takes just a few milliseconds. If the verification succeeds, Alice is convinced that Bob has spent about T seconds to solve the puzzle (or even longer, if his computer is not as fast as Alice's high-end reference machine).

4. ON-DEMAND OR REACTIVE ROUTING PROTOCOLS:

On-Demand protocols, routes are created as and when required. On demand protocols create routes only when needed by source nodes. When a transmission occurs from source to destination, it initiates route discovery process within the network. This process is completed once a route is found or all possible route permutations are examined. The route remains valid till destination is achieved or until the route is no longer needed. Once a route is discovered and established, it is maintained by route maintenance procedure until either destination becomes inaccessible along every path from source or route is no longer desired. Some of examples on demand routing protocols are: DSR [10], [11], AODV [9], and TORA [12], [13].

- Dynamic Source Routing (DSR) [10], [11]]

Dynamic Source Routing (DSR) is an on-demand routing protocol, which is based on the theory of source-based routing rather than table-based. This protocol is source-initiated rather than hop-by-hop. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. This is particularly designed for use in multi hop wireless ad hoc networks of mobile nodes. Basically, DSR protocol, like in other On-Demand routing, does not need any existing network infrastructure or administration and this allows the network to be completely self-organizing and self-configuring. The protocol consists of two major phases: route discovery and route maintenance. Every node maintains a cache to store recently discovered paths. When a node wants to send a packet to some destination, it first checks its route cache to determine whether it already has a path to the destination. If it is there, it uses that path to transmit the packet and also attach its source address on the packet. If the node does not have such a route, the sender broadcasts a route request packet to all of its neighbors asking for a path to the destination.

- Ad hoc on-demand distance vector (AODV) [9]

AODV is an improvement of Destination-Sequenced Distance-Vector (DSDV) routing protocol which is collectively based on DSDV and DSR. It aims to minimize the requirement of system-wide broadcasts to its extreme. The nodes are discovered only as and when needed and they are maintained just as long as they are required. The AODV have two main phases; route discovery and rout maintenance:

A. Route Discovery

When a source node S wants to send a data packet to a destination node D, the entries in route table are checked to ensure whether there is a current route to that destination node or not. If it is there, the data packet is forwarded to the appropriate next hop toward the destination. If it does not have a route to D, it initiates route discovery by broadcasting a route request (RREQ) to its neighbors. This routing request contains its IP address, its current sequence number, the destination's IP address, the destination's last sequence number and broadcast ID. This process is repeated until the RREQ reaches the destination node. When receiving the first arrived RREQ, the destination node creates and sends a route reply (RREP) to the source node through the reverse path from where the RREQ came.

B. Route Maintenance

A discovery routing between a source node and destination node is maintained as long as needed by the source node. Since there is movement of nodes in mobile ad hoc network and if the source node moves during an active session, it can reinitiate route discovery mechanism to establish a new route to destination. Whenever there is a broken link between two nodes the route maintenance phase is carried out. The node that discovers the broken link initiates Route Error (RERR) message to the source node by the predecessor intermediate nodes. This process is repeated until the source node is reached. When RERR is received by the source node, it can either reinitiate the route discovery mechanism by sending a new RREQ message or stop sending the data.

- TORA (Temporary Ordered Routing Protocol)

The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive loop-free distributed routing algorithm based on the concept of link reversal [12]. TORA is a distributed highly adaptive routing protocol designed to operate in a dynamic multi-hop network. It uses an arbitrary height parameter to determine the direction of link between any two nodes for a given destination. It is source initiated and provides multiple routes for any desired source/destination pair.

5. PUZZLE AUTHENTICATION SCHEME FOR AODV

5.1 Notations used in our scheme

Our new scheme is combined out of cryptographic puzzle and WSBC function. The scheme has to offer privacy protection of the confidential information stored in the nodes, that is the identifier ID and the encryption puzzle. This identifier allows the unequivocal

identification of the nodes on the network. The anonymization of messages is crucial to avoid traceability and replay attacks. The scheme offers moderate protection concerning privacy and traceability when a single node is considered.

We use AODV routing protocol to communicate between the source and the destination node in Mobile ad ho networks let α_j and s_j symbolize a t-bit random value generated by the destination and the source respectively.

This scheme proposes to use Puzzle encryption between the nodes to protect the privacy of the nodes in MANET, and to protect of the confidential information in the nodes. The anonymization of messages is the most important to avoid traceability and replay attacks.

Some samples use in our scheme:

- S and D denote the two parties of communication in the network, Source and Destination.
- enk_k (x) is asymmetric key algorithm (e.g block cipher AES) that encrypt message X under key. Or symmetric encryption use public / private keys like RSA.
- We still use also hash function in this scheme for example h(a\\b) is hash function concatenation of a and b $P_j = (n_1 \backslash\backslash ID \backslash\backslash \alpha_j \backslash\backslash n_2 \backslash\backslash j)$ represent the cryptographic puzzle sent by D (destination) at the j-th protocol instance. Where n_1 it is a random number and α_j represents the challenge bit, in the low-level distance-bounding exchange.
- w_j^{π} (k) represent WSBC function and we suggest for ID is simply $\{p_j , w_j^{\pi} (k)\}$

In this scheme the destination D randomly selects *l* bits of k and this collection of bits form w_j^{π} (k), and we can use this collection *l* bit k for the encryption and decryption between S and D, and we exchange the K bye Deffie-Helman exchange key.

- $u_j = h (j\backslash\backslash n_1 \backslash\backslash ID \backslash\backslash \alpha_j \backslash\backslash n_3)$ is the pseudonym set by Destination at the j's identification process that mainly has the role of allowing the verification of the Puzzle solution after the Source completes operation.

This j represents the numbers Destination response for the route request on his time life in MANET.

Generally, a pseudonym transmits the static identifier of a Destination with the guarantee of keeping confidential information secret and ensuring the un-traceability of Destination responses.

5.2 Proposal of Puzzle Authentication scheme for On-demand routing protocol

Here we applied our scheme on any On-Demand routing protocol and we choose the On-Demand routing protocol AODV. For initial preparation for this scheme each node in the network has a unique identifier number (ID) and secret key (or private key), which are set in the initialization process.

The steps in our scheme are described in initialization of puzzle encryption function from the destination and also weak secret bit commitment for authentication, and the both source and destination generation of nonce's to challenge response steps between them.

When any node needs any information from another node in the mobile ad hoc network, it starts to send requests for this message and any node starting the communication will be the Source node and the any node that ends the communication will be the Destination node.

- S (source) generates two nonces $\{ n_1, n_2 \}$ and a t-bits S_j random value and commit this value by sending random number n_1 and message γ_j ($\gamma_j = h(n_1 \backslash\backslash n_2 \backslash\backslash S_j)$, after that the Source send the request with n_1, γ_j to the destination.
- When the Destination received the request message it starts to send a response message by generating α_j symbolize a t-bit random value, $0 < \alpha_j < N_i$, and send it to the Source.

Now, we explain what the Destination makes, in details: the destination node generates two large prime number p_i, q_i , (where i = 1, 2 ,3 ,…. represented the number of the node), computes $N_i = p_i q_i$, and also computes $\varphi(N_i) = (p_i - 1)(q_i - 1)$; after that chooses private exponent d_i, $1 < d_i < \varphi(N_i)$, such that: $e_i = d_i^{-1} \bmod \varphi(N_i)$.

Also computes the remainder $r_i = 2^t \bmod \varphi(N_i)$, and public exponential $e\square_i = 2^t + \varphi(N_i) - r_i + e_i$ and we let $z_i = \varphi(N_i) - r_i + e_i$, so now $(N_i, e\square_i)$ is a public key of the destination node and (N_i, d_i) is a private key.

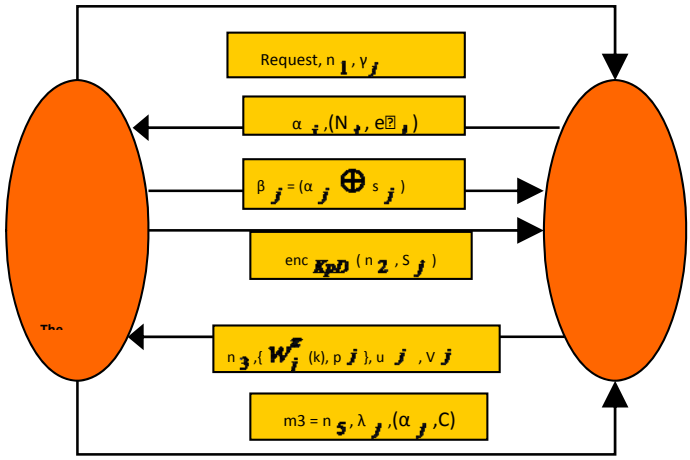

Fig. 1 The security mechanism between the source and the destination

- When the Source received α_j, the source generate random bit s_j and make XoR with α_j to produce β_j ($\beta_j = (\alpha_j \oplus s_j)$ and sent it to the destination
- After the Source and destination complete of the rapid bit exchange, S opens the commitment of the hidden value s_j by sending $\{ n_2 , S_j \}$ and encrypts it by public key of Destination ($enc_{KpD} (n_2 , S_j)$,where $_{KpD}$ is a public key of destination =(N_i, e_i) and sends the value to D.
- When the Destination received the value of that encryption ($enc_{KpD} (n_2 , S_j)$, the destination decrypts this value by private key of destination to produce n_2 , S_j ;($Decrypt_{KprivD}(enc_{KpD} (n_2 , S_j) = n_2 , S_j)$); after that the destination generates two nonce's $\{ n_3 , n_4 \}$ and compute WSBC and Puzzle encryption $\{ w_j^{\pi} (k), p_j \}$, $p_j = (n_1 \backslash\backslash ID \backslash\backslash \alpha_j \backslash\backslash n_2 \backslash\backslash j)$, and $w_j^{\pi} (k)$ which depends on the distance (drt) that separates the source and destination, and finally, message m2 is ended by an authentication message $V_j = enc_k (j \backslash\backslash n_4 \backslash\backslash ID \backslash\backslash n_1)$. After that the destination sends m2 to the source which $m2 = n_3 , \{ w_j^{\pi} (k), p_j \}, u_j , V_j$
- after the Source received the last message from the destination, decrypted it by k and solved the puzzle encryption, it uses the source the public key (N_i, e_i) of the destination node to encrypt also the contest $\alpha_j , 0 < \alpha_j < N_i$, in the usually manner:

  $C = \alpha_j^{\tilde{e}_i} \mod N_i$ and solve the puzzle

$$\sigma = \alpha_j^{2^t} \mod N_i .$$

In T seconds by repeated squaring and to quickly do the regular-sized modular exponentiation

$$\Omega = \alpha_j^{z_i} \bmod N_i ,$$

after that compute $c = \sigma . \Omega \bmod N_i$; after that the source submits the pair (α_j, C) with λ_j nonce n_5 the source generate nonce n_5 and encryption message $\lambda_j = enc_k (j \backslash\backslash n_5 \backslash\backslash ID \backslash\backslash \alpha_j \backslash\backslash \beta_j \backslash\backslash n_4 \backslash\backslash n_1)$ and send $m3 = n_5, \lambda_j, (\alpha_j, C)$ to destination. When the destination receives m3 and decrypt it, the destination can authenticate the source and also verify the solution by applying its private key (N_i, d_i) to compute the ciphertext and compare the result with α_j such as:

$$C^{d_i} \bmod N_i \stackrel{?}{=} \alpha$$

After that the destination is able to check if the messages (challenges and responses) in the rapid bit exchange have not been altered by an adversary. Here we apply this scheme in on-demand routing protocol by sending the request, n^1, γ^j with the routing discovery phase and the response from the destination with the routing replay as challenge and response to achieve the authentication scheme between the source and destination.

When a node receives a RREQ, node checks first before signing create or update a reverse route to the source of RREQ. If RREQ was received with a double signature extension, then the node also store lifetime, which is the value of 'reverse path for lifetime', and signing RREP in the entry path. Intermediate node will reply to RREQ with RREP only if it meets the requirements of AODV node has signed the interview and to put it in the old fields of lifetime of RREP and old signature of the double signature extension. Otherwise, it will be rebroadcast RREQ also has no route temporarily. When the destination receives RREQ, it will respond with RREP signature and one extension. When a node receives RREP, it checks first the signature before create or update a route to that host. If signature validation is successful, it will be stored the route with the signing of the RREP and lifetime. Otherwise be discarded RREP.

The attacks can be classified in two categories [6] [7]:

- Internal attacks – the attacker poses as one of the nodes and gains direct access to the network either by impersonation or by compromising a proper node and using it to do bad (malicious) activities
- External attacks - make the attacker attack from outside the network, he creates congestion in the network traffic by promulgation messages have no meaning, and impairing the complete communication network.

What will happen in the case of a intruder-in-the-middle attacks?

A mobile node and his correspondent node it can derive a shared (symmetric) key to authenticate the MIPv6 Bus sent by the MN (mobile node).

A mobile node (MN) and his correspondent node (CN) will derive the session key using Diffie-Hellman algorithm.

- A random secret value y generated by Correspondent node (CN) and sends g^y mod p to the mobile node (MN);
- The mobile node (MN) chooses a random secret x and sends g^x mod p to its correspondent node (CN)
- The session key share by the mobile node and its correspondent node represent a hash digest of g^{xy} mod p (g and p are known by the mobile node and correspondent node)

As a conclusion for this attack, Diffie-Hellman is known to be vulnerable to the intruder-in-the-middle attack on an unauthenticated Diffie-Hellman key agreement:

$$CN \rightarrow g^{y} \rightarrow \text{int}\, ruder \rightarrow g^{yi} \rightarrow MN$$
$$CN \leftarrow g^{x_i} \leftarrow \text{int}\, ruder \leftarrow g^{x} \leftarrow MN$$

The intruder intercepts g^y which is sent by the correspondent node and after that sends g^{yi} to the mobile node. The intruder also will intercepts g^x sent by the mobile node and send $g^x i$ to the correspondent node. As result, mobile node shares the key g^{xyi} with the intruder. The correspondent node shares the key g^{xi} with the intruder. The intruder can then impersonate the mobile node and the correspondent node.

6. CONCLUSIONS

In this paper, we explore the use of WSBCs and puzzle function as a practical and effective tool to increase the security of on-demand routing protocol on MANETs and the way to resist this attacks in a secure way; also this scheme helps on-demand routing protocol to increase the security between the nodes by enhancement and improving the authentication and confidentiality between the nodes. . The puzzles function use one way encryption functions whereas the keys must be enough to avoid a brute force attack. We use also the bit commitment function with puzzle encryption in our scheme to commit a value without revealing that value. This scheme will offer moderate protection concerning privacy and traceability when a node communicates with each other's in the Mobile Ad hoc Networks (MANETs).

REFERENCES

1. 68. Rivest RL, Shamir A, Wagner DA (1996) Technical report mit/ lcs/tr-684. time-lock puzzles and timed-release crypto. Technical report.
2. Back A (2002) Hashcash. A denial of service counter-measure. Technical report. http://www.hashcash.org/hashcash.pdf

3. Jinsong Han, Yunhao Liu ,Mutual Anonymity for Mobile P2P Systems, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 19, NO. 8, AUGUST 2008.
    a. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, pp. 612-613, 1979.
4. M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance," J. ACM, vol. 36, pp. 335- 348, 1989.
5. Secure Ad hoc On-Demand Distan Mobile Networks Laboratory Nokia Research Center FIN-00045 NOKIA GROUP, Finland ce Vector Routing Manel Guerrero Zapata manel.guerrero-zapata@nokia.com.
6. SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks Yih-Chun Hu; Johnson, D.B.; Perrig, A.; Rice Univ., Houston, TX .IEEE.
7. ODAR: On-Demand Anonymous Routing in Ad Hoc Networks Denh Sy, Rex Chen and Lichun BaoBren School of Information and Computer Sciences and Calit2 University of California, Irvine, CA 92697Emails: {dsy, rex, and lbao}@ics.uci.edu
8. Ad hoc On-Demand Distance Vector Routing Charles E. Perkins Sun Microsystems LaboratoriesAdvanced Development Group Menlo Park ; CA 94025 cperkins@eng.sun.com / Elizabeth M. RoyerDept. of Electrical and Computer Engineering University of California, Santa Barbara Santa Barbara ; CA 93106
    a. Castelluccia and P. Mutaf. Hash-Based Dynamic Source Routing. In IFIP Networking, LNCS 3042, pages 1012–23, 2004.
9. D.B. Johnson and D.A. Maltz. Mobile Computing, chapter Dynamic Source Routing in Ad Hoc Wireless Networks, pages153–181. Kluwer Academic Publishers, 1996.
10. Park and S. Corson, Temporally Ordered Routing Algorithm (TORA) Version 1, Functional specification IETF Internet draft (1998), http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec- 01.txt.
11. Park V. and S. Corson, 2001. Temporary-ordered Routing Algorithm (TORA). Internet Draft, draft-ietf-manettora-spec-04.txt.
12. Syverson P (1998) Weakly secret bit commitment: applications to lotteries and fair exchange. In: Proceedings of the 11th IEEE computer security foundations workshop.
13. Merkle, Secure Communications Over Insecure Channels, CACM, Vol. 21, No. 4, pp. 294-299, April 1978.
14. Groza, B., Using one-way chains to provide message authentication without shared secrets, accepted at IEEE 2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, Lyon, France, 2006.
    a. Shamir, "How to Share a Secret," Comm. ACM, vol. 22, pp. 612-613, 1979.
15. W. Mao, "Timed-Release Cryptography," in SAC 2001: Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography, Aug. 2001, pp. 342–357.
16. Offline Submission with RSA Time-Lock Puzzles,Yves Igor Jerschow, Martin Mauve, Institute of Computer Science, Heinrich Heine University, D¨usseldorf, Germany, IEEE 2010.