Applying Puzzle Encryption in the On-Demand Routing Protocols in Mobile Ad Hoc Networks (MANETs)

AHMAD ALOMARI

Faculty of Mathematics and Computer Science, University of Bucharest,
Bucharest, Romania
alomari.jordan@gmail.com

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).

Key-Words: - MANET, AODV, 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.

The security issue in an ad hoc network is especially challenging due to the nature of communication and lack of infrastructure support. A number of security mechanisms has been developed and proposed, but guaranteeing a whole network free from any malicious attacks still proves to be difficult.

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 it to protect the ad hoc routing and forwarding protocols. From the security design perspective, the MANETs have no clear line of defence. 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 [1]: first, Proactive routing protocols, where the nodes keep updating their routing tables, by sending periodical messages. We have, for example, OLSR [2], (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 [3] (Dynamic Source Routing protocol) and AODV [1], [3] (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 cryptographic puzzles and weakly secret bit commitment (WSBC) function [4]. 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 speaks about Weakly Secret Bit Commitment (WSBC) function. Section and explains some puzzle constructions. In Section III we introduce Puzzle Authentication scheme for on-demand routing protocol.

2 Preliminaries

2.1 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 [3]. 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 R̃2, M such that h(R1,R2,M) = h(R1, R̃2, 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:

- 2nd-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 2nd-preimage 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 although it is probably related to the non-correlation property.

### 2.2 RSA Time-Lock Puzzles

Jerschow and Mauve introduce a non-interactive RSA time-lock puzzle scheme [5]; inspired by Rivest’s time-lock puzzles [6] 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 [7]. 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 secret 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 secret prime \( p \) and \( q \), and also computes
  \[
  \varphi(n) = (p-1)(q-1)
  \]

- Choose a private exponent \( d \) randomly, \( 1 < d < \varphi(n) \) such that \( \gcd(d; \varphi(n)) = 1 \) and determine its multiplicative inverse modulo \( \varphi(n) \): \( e = d^{-1} \mod \varphi(n) \).
- 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 \( t = T \cdot S \).
- Compute the remainder \( r = 2^t \mod \varphi(n) \)

And the public exponential \( \hat{e} = 2^t + \varphi(n) - r + e \); and let \( z = \varphi(n) - r + e \) denotes the lower bits of \( \hat{e} \) 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 \( \hat{e} \) is an extremely large number with lots of 0-bits after the leading 1-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 \( \hat{e} \) is constructed by adding a large multiple of \( \varphi(n) \) to the regular exponent \( e \). It holds that \( m^e = m^{\hat{e}} \mod n \) for all \( m \in \mathbb{Z}_n \) since, \( e \equiv \hat{e} \mod \varphi(n) \) and \( n \) is a product of distinct primes. \( \hat{e} \) 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 \( (n; \hat{e}) \) of the sender (Alice) to encrypt the contest \( m \), \( 0 < m < n \), in the usually manner, i.e. to compute the ciphertext:

\[
c = m^\hat{e} \mod n
\]

Due to the special structure of \( \hat{e} \), the fastest way to perform this giant modular exponentiation is to solve the actual puzzle [5]:

\[ \sigma = m^2 \mod n \]

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

\[ \Omega = m^t \mod n \]

This yield:

\[ C = \sigma \cdot \Omega \mod n \]

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 \(m\):

\[ C^d \mod n = m \]

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).

### 3 Puzzle Authentication scheme for AODV

#### 3.1 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 hoc networks let \(\alpha_j\) and \(\gamma_j\) 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.
- \(\text{enk}_x(x)\) 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 \(b\) \(P_j = (n_1 \| ID \| \alpha_j \| n_2 \| j)\) represent the cryptographic puzzle sent by \(D\) (destination) at the \(j\)-th protocol instance. Where \(n_1\) it is a random number and \(\alpha_j\) represents the challenge bit, in the low-level distance-bounding exchange.
- \(w_j^d(k)\) represent WSBC function and we suggest for \(ID\) is simply \(\{p_j, w_j^d(k)\}\)

In this scheme the destination \(D\) randomly selects \(l\) bits of \(k\) and this collection of bits form \(w_j^d(k)\), 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.

\[ u_j = h(j\|n_1 \| ID \| \alpha_j \| n_3) \]

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.

#### 3.2 Proposal of Puzzle Authentication scheme for On-demand routing protocol
Here we applied our scheme on any On-Demand routing protocol and we chose 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.

1. 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 \(\gamma_j\) (\(\gamma_j = h(n_1 \| n_2 \| S_j)\)), after that the Source send the request with \(n_1, \gamma_j\) to the destination.

2. When the Destination received the request message it starts to send a response message by generating \(\alpha_j\) symbolize a t-bit random value, \(0 < \alpha_j < N_j\), and send it to the Source.

3. When the Destination received the request message it starts to send a response message by generating \(\alpha_j\) symbolize a t-bit random value, \(0 < \alpha_j < N_j\), 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 \(i = 1, 2, 3, \ldots\) 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: \(e_i = d_i^{-1} \mod \varphi(N_i)\).

Also computes the remainder \(r_i = 2^t \mod \varphi(N_i)\) and public exponential \(\hat{e}_i = 2^t + \varphi(N_i) - r_i + e_i\) and we let \(z_i = \varphi(N_i) - r_i + e_i\), so now \((N_i, \hat{e}_i)\) 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](image)

- When the Source received \(\alpha_j\), the source generate random bit \(s_j\) and make XoR with \(\alpha_j\) to produce \(\beta_j\) (\(\beta_j = (\alpha_j \oplus s_j)\)) 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 \((\text{enc}_{KpD}(n_2, S_j))\), where \(KpD\) is a public key of destination \(=(N_i, \hat{e}_i)\) and sends the value to D.

- When the Destination received the value of that encryption \((\text{enc}_{KpD}(n_2, S_j))\), the destination decrypts this value by private key of destination to produce \(n_2, S_j\); and compute WSBC and Puzzle encryption \(\{w_j^\pi(k), p_j\}\), \(p_j = (n_1 \| ID \| \alpha_j \| n_2 \| j)\), 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 $V_j = \text{enc}_k(j \parallel n_4 \parallel ID \parallel n_1)$. After that the destination sends $m2$ to the source which $m2 = n_3, \{w_f^j(k), p_j\}, u_j, V_j$. After that 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: $C = \alpha_j^{e_i} \mod N_i$ and solve the puzzle

$$\sigma = \alpha_j^\gamma \mod N_i,$$

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

$$\Omega = \alpha_j^\gamma \mod N_i,$$

after that compute $c = \sigma \cdot \Omega \mod N_i$; after that the source submits the pair ($\alpha_j, C$) with $\lambda_j$, nonce $n_5$ the source generate nonce $n_5$ and encryption message $\lambda_j = \text{enc}_k(j \parallel n_5 \parallel \alpha_j \parallel \beta_j \parallel n_4 \parallel n_1)$ 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 $\alpha_j$ such as

$$C^{d_i} \mod N_i = \alpha_j.$$

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, $n_1, \gamma_j$ 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.

4 Conclusion

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] Secure Ad hoc On-Demand Distans Mobile Networks Laboratory Nokia Research Center FIN-00045 NOKIA GROUP, Finland e Vector Routing Manel Guerrero Zapata manel.guerrero-zapata@nokia.com
[2] 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.