In: Management of E-Commerce and E-Government
Editors: Zhong Yuansheng et al, pp. 51-65

ISBN 1-60021-324-3

Chapter 5

A MOBILE LOCATION-BASED FRAMEWORK FOR SECURE E-COMMERCE AND E-GOVERNMENT

Johnson Thomas[1]*, *Zhe Ming Shen*[2],
Marcin Paprzycki[3] *and Martin Crossland*[4]
[1]Department of Computer Science, Oklahoma State University, USA
[2]Dollar Thrifty Automotive Group, USA
[3]Computer Science Institute, SWPS, Poland
[4]School of Business, Oklahoma State University,USA

Abstract

Mobile e-commerce and e-government using ad hoc networks pose huge challenges due to the constantly changing topology of the network. In this paper we reduce communications link breakage due to mobility by proposing a predictive location triggered approach to obtaining new routes. Simulation results show that fewer e-commerce or e-government sessions are aborted due to the changing topology. Our location triggered approach ensures also that the power expenditure is minimized. To ensure security for e-commerce and e-government applications, we propose a lightweight authentication protocol and a novel non-repudiation protocol.

Key words: Ad Hoc networks, Location Triggered Protocols, authentication, non-repudiation protocols.

1. Introduction

Mobile Ad Hoc Networks (MANETs) consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure [10]. Because of the limited transmission radius of signals, the routes between nodes are created through several hops in such multi-hop wireless networks [10] and host mobility can cause frequent unpredictable topology changes [2]. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. Table-driven and demand-driven protocols have been proposed for MANETs, with the goal of achieving efficient routing [10]. Table-driven

*E-mail address: jpt@cs.okstate.edu

routing protocols such as DSDV, CGSR, WRP [8] [10] attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. The major disadvantage of this approach is the network traffic overhead. Demand-driven routing protocols create routes only when desired by the source node and include: AODV, DSR, TORA, ABR, SSR [5] [9] [10]. The demand-driven routing protocols do not need maintain routing tables, but involve the overhead of (possibly frequent) route discovery.

One of the main constraints of ad hoc networks is the limited power available for computation and communication and restrictions on storage capabilities. While constant progress is being made to increase the capabilities of mobile devices, these performance gains are counterbalanced by a rapidly increasing volume of transmitted data (e.g. the addition of rich media content). The recent availability of small, inexpensive low-power Global Positioning System (GPS) receivers and techniques for finding relative coordinates based on signal strengths, and the need for the design of power efficient and scalable networks have provided justification for applying position-based routing methods in MANETs [13]. Several location aided demand-driven ad hoc network protocols have already been developed to improve the route discovering process by using todays GPS technology [4] [12] [13]. However, most existing on-demand routing protocols do not take full advantage of the information available through the GPS system and continue using a particular route until a link breaks. This results in packet dropping during route reconstruction which results in significant throughput degradation [7].

Although ad hoc networks have enormous portential in both e-commerce (e.g. purchasing) and e-governement (e.g.voting), their usage is almost non-existent due to unacceptable quality of service caused by packet loss and lack of security mechaisms. In electronic commerce or e-government, a trust has to be established along the entire communications route before a business transaction can proceed. If the path is not secure, a node in the path can act maliciously. This problem is particularly severe in mobile ad hoc networks where due to the constantly changing topology, communication links are continuously broken and new links established as nodes move. If a link in the path is broken, a new route has to be set up and security requirements along the path satisfied. Furthermore, due to route changes caused by mobility, receiver and sender nodes can easily repudiate the e-comerce or e-government transaction resulting in disputes. Therefore, from an e-commerce and e-government perspective existing ad hoc network protocols suffer from the following limitations:

1. when a link is broken, the connection is lost and the e-commerce/e-government transaction session has to be terminated; an ongoing transaction is therefore interrupted and the transaction has to be rolled back to a safe state,

2. when a new link is set up to a new neighbor, a new route has to be set up and secured (involving authentication of one or more nodes on the way) and the entire session re-started.

3. due to continous route changes during a single session, it becomes very difficult to ensure non-repudiation of e-business or e-government transactions.

This results in excessive overheads, but also may result in insecure business communications. For these reasons, e-commerce and e-government in ad hoc networks is a largely unexplored area. In this paper, we propose a new protocol called Location Triggered Routing protocol (LTR) that applies location information on a table-driven protocol (in this paper we use DSDV). In LTR, instead of each node periodically sending and receiving messages to maintain its routing table, no routing messages need to be sent unless a node has detected a location change that impacted at least one network route. This results in a significant saving in power as it reduces the number of routing messages. By using location information to predict the moving direction, if a node feels the signal is becoming weak during communication because of its movement, a new route can be discovered automatically to replace the potential broken link even before a link is actually broken. The prediction facilitates not only route discovery, but also provides scope for node authentication before the existing connection breaks thus substantially reducing overhead of sustaining secure e-commerce transactions. In other words, the LTR approach not only results in energy efficiency, but it also permits an existing e-commerce session to be maintained even if a new route is established. However, if a new route and authentication cannot be established before the current route breaks, e.g. due to an incorrect prediction, the session is terminated and a standard approach is applied). We also propose security mechaisms to allow ad hoc networks to operate in a e-commerce or e-government environment. A lightweight authentication protocol is proposed to authenticate new neighbors in a newly established route. We also propose a novel non-repudiation protocol. Non-repudiation is achieved by means of a trusted third party (TTP). The TTP is called when a dispute is filed by the sender or receiver. However, in a mobile environment, the TTP may not be available when a dispute arises. In our protocol, the TTP is not required to be available at the time of dispute. We present the LTR protocol in the next session and the security protocols in section 3.

2. Location Triggered Protocol

We assume that each node has same signal coverage area radius R and that each node has enough cache memory to hold the routing table (as in the DSDV protocol). We also assume that each node has a GPS device to accurately identify its own location. All the nodes within the signal coverage area receive the message sent by any broadcasting node. Although routing is not discussed in detail in this paper, the routing table of each node consists of the Destination (the reachable destination node), the Next Hop (next hop to reach the destination), Metric (the shortest number of hops to reach the destination), Sequence Number (the number that is originally stamped by the source) and the Destination Location (the location value of each node presented by coordinate $[X, Y]$).

2.1. LTR Protocol

In a table-driven protocol like DSDV, each node needs to periodically send and receive routing message in order to maintain its routing table up-to-date [8]. This consumes lots of network resources and power. In LTR, the location information will be used for two purposes: 1) trigger a message exchange process only when a node detects a location change; 2) predict the direction of movement to establish a new route when the current communication has a potential to be broken. In this way we expect to remove the need to periodically send messages and to avoid breaks in communication resulting in lower network overhead and power consumption.

2.2. Routing Table Update Algorithm

By adding location information into the routing table and by using GPS, a node will not send an update message until it discovers a location change. When a node receives an updating message, it compares the message with its routing table to determine if there is any need to update. If there is, the node will update its routing table and broadcast to the neighbors. Otherwise, the node will ignore the message.

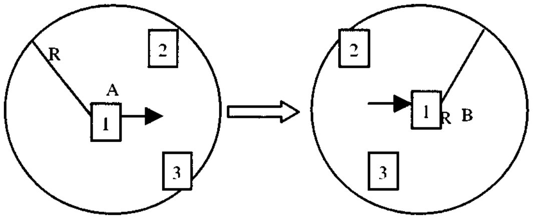

Figure 1. Network snapshot when node 1 moves from A to B

For example, in Fig. 1, R is the signal coverage radius. Node 1 travels from location A to B, while all its direct neighbors 2 and 3 are still within signal range. In this case, no route needs to be updated.

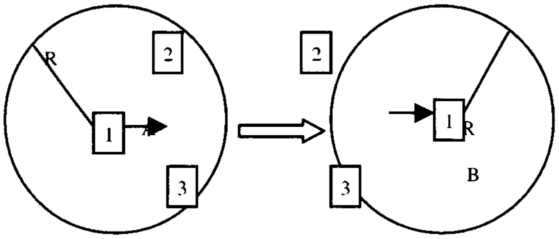

Figure 2. Route update when node 1 moves from A to B

In Fig. 2.2., node 1 travels from A to B. One of its direct neighbors, 2, moves out of its signal range. In this case, the route from 1 to 2 has been disconnected. Therefore, the route needs to be updated.

Protocol: Let us assume that $[X, Y]$ denotes the current location of the node while $[x_i, y_i]$ denotes location of node i stored in the routing table. Each node will receive its GPS location information $[X, Y]$ after a time period t. Small movements (changes of less than 10% of radius R) need be filtered out to eliminate unnecessary location update messages (observe, however that accumulation of those small movements can add up to a significant distance). The moving distance can be calculated by the following formula:

$$\int_{T_0}^{T} V\,dt = D \tag{1}$$

where D is the node moving distance since last location broadcast; T_0 is time of last location broadcast; T is the current time and v is the node moving speed.

Since each node has the knowledge of its location after every period t, the above formula 1 can be represented as:

$$D = \sum_{i=1}^{n} \sqrt{(x_i - x_{i-1}^2) + (y_i - y_{i-1}^2)} \tag{2}$$

where x_i, y_i is the node's current location; x_{i-1}, y_{i-1} is the node's previous location and n is the number of times GPS information received since the last location broadcast.

After calculating the moving distance D:

- If there is no location change ($D \leq 10$ percent of R), nothing needs to be done
- If one node changes its location ($D \geq 10$ percent of R), for any other destinations i such that Metric = 1 and $\sqrt{(X - x_i^2) + (Y - y_i^2)} \leq R$

As defined above, Metric defines the shortest route to the destination in terms of the number of hops. Metric = 1 means the current moving node has direct communication to i. This means no route has been changed, but routing tables are updated with the new l ocation information.

If one node changes its location, for any other destinations i such that:

Metric = 1 and $\sqrt{(X - x_i^2) + (Y - y_i^2)} > R$ or

Metric > 1 and $\sqrt{(X - x_i^2) + (Y - y_i^2)} \leq R$

This means one or more routes have been changed. Tables have to be updated and a new route discovered.

In order to gather the information about the new location, the moving node sends out a query message. After all the direct neighbors receive this query message, they reply with their routing tables to this node. After receiving those data, it will create its new routing table. Since the node has established several new routes, it needs to broadcast its routing table for the rest of network to update.

2.3. Prediction Algorithm

In MANETs, most of the existing routing protocols continue using a route until a link breaks. During the route reconstruction, communication is broken resulting in dropped packets (which causes significant throughput degradation [7}). It therefore causes an e-commerce transaction in progress to abort. In order to solve this problem, we introduce a prediction algorithm.

In Fig. 2.3., node 1 currently communicates with node 3 using path 1 → 2 → 3. Meanwhile, 1 is traveling to another location. When 1 feels the signal received from 2 becoming weak, (1 is currently traveling away from 2), the link with 2 will be broken soon, thus breaking the communication with 3. Existing protocols either wait for the communication

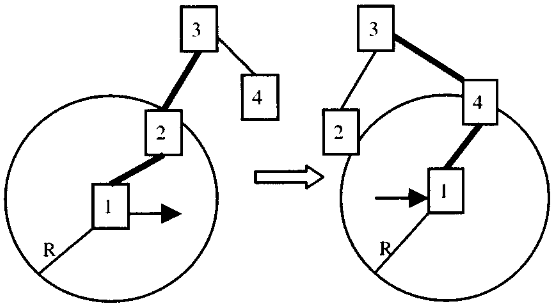

Figure 3. Route update from 1 to 3 when 1 changes location

link to break before starting a route discovery process, which results in a significant loss of packets or start a table updating process, which consumes a lot of bandwidth. Instead, we suggest that 1 discovers a new path to continue communicating with 3 before the link with 2 breaks. In order to do so, 1 needs to predict when (link prediction time) and where (link prediction location) the current communication will be dropped, and locate the potential closest node 4 in its traveling direction, by searching the location information in the routing table.

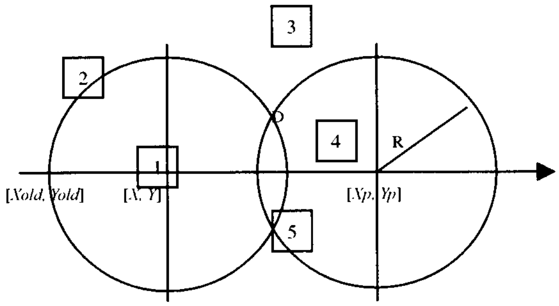

Figure 4. Expected Zone

To explain how node 4 will be chosen, in Fig. 2.4, consider that node 1 travels from $[Xold, Yold]$ to a new location $[X, Y]$. Based on node 1's traveling direction, it can be predicted that $[Xp, Yp]$ will be node 1's next location after it travels time t_p. Any nodes located in an Expected Area D are able to keep communication with 1 after it travels from $[X, Y]$ to $[Xp, Yp]$. Furthermore, if a node is located in D and has the shortest distance to $[Xp, Yp]$, it will still be in the signal range, even if 1 travels beyond $[Xp, Yp]$.

It is beyond the scope of this paper to show the derivation of the time at which the link will be broken. This time is derived in [11]. Since we can calculate the link prediction time, the actual predicted location $[Xp, Yp]$ of node 1 can be derived. Again the derivation is not shown in this paper. The predicted location $[Xp, Yp]$ is derived in [11] and is:

$$X_p = \frac{(x_2^{'} - x_2) * t_p}{t_1} + x_2 \quad (3)$$

$$Y_p = \frac{(y_2^{'} - y_2) * t_p}{t_1} + y_2 \quad (4)$$

Location Triggered Protocol: We now outline the LTR prediction. Assume that node 1 currently has a communication link through 2, and 1 is traveling away from 2. When 1 realizes the signal received from 2 is becoming weak, it will initiate the prediction process:

1. calculate the link prediction time tp; and use equations (2.3) and (2.4) to calculate the predicted location $[Xp, Yp]$,
2. identify all nodes that should have direct communication to predicted location $[Xp, Yp]$,
3. calculate the distance of each satisfied node i from step 2 to the predicted location $[Xp, Yp]$,
4. sort all qualified i according to ascending distances,
5. 1 checks the stability of i; if i is not stable, 1 chooses the next i with smallest distance to the predicted location from the list,
6. if i is stable, 1 send a query message to i to find a new path to destination through i,
7. i replies with its specific routing information,
8. 1 use the received routing information to establish a new path to the communication destination.

2.4. Simulation and Results

To evaluate the performance of the LTR routing protocol, a simulation program has been created. The program is implemented in Java using the JDK 1.2.2. The simulations were done for small and big network sizes, low and high communication ratios, as well as low and high mobility scenarios. The simulations show that the Location Prediction can significantly reduce the number of lost packets compared with No-Prediction routing protocols, especially when communication ratio is high and the mobility in the network environment is also high.

Fig. 2.5 shows the number of packets lost as mobility of nodes in the network increases. Even at low node speeds, the number of lost packets is far lower than for the no prediction case. Moreover, as mobility increases, the rate of increase of packet loss is slower when prediction is applied. A very low speeds, there is no packet loss when prediction is used, that is, there is no break in communications even as nodes move and routes change. Even at very high speeds, a communication route (and hence an e-commerce session) is 6 to 7 times more likely to break if prediction is not employed. This is particularly significant for an e-commerce or e-government environment. Overall, although the LTR protocol does not

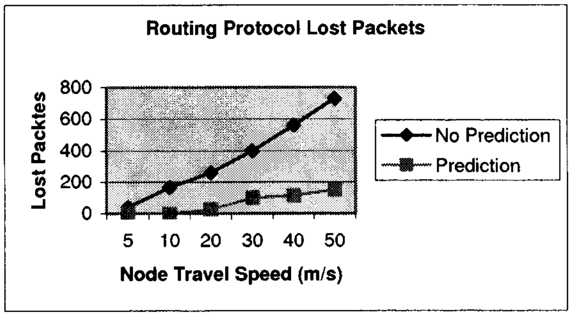

Figure 5. Node Travel Speed vs. Lost Packets

completely avoid the problem of lost packets and broken connections (and thus e-commerce or e-government sessions), it does reduce it significantly (for more details see [11]). The predictive nature of the LTR protocol establishes new routes before the current path is broken.

From an e-commerce or e-governmentperspective, before data can be transmitted on the new route to the new neighbor, it is critical to authenticate the new neighbor. The prediction mechanism provides a window of opportunity for authentication before the current path is broken. Once authentication has been accomplished, the business transaction can continue uninterrupted along the new path. The LTR protocol is therefore particularly well suited to e-commerce applications (compared to other ad hoc network protocols). We next describe the authentication protocol used.

3. Mobile Secure Framework for e-Commerce and e-Government

In electronic commerce, a trust has to be established along the entire communications route before a transaction can proceed. If the path is not secure, a node in the path can act maliciously. This problem is particularly severe in a mobile ad hoc network with a continuously changing topology where communication links are continuously broken and new links established as nodes move. If a link in the path is broken, a new route has to be setup and security requirements along the path satisfied. For example, if a node has a new neighbor who is routing its business transaction to a remote destination such as a bank server, and the new node is not authenticated, the neighbor can eavesdrop, spoof or modify the business data, to name just a few possible security attacks. Existing ad hoc network protocols suffer from interruptions of business transactions, that is, when a link is broken, the connection is terminated and the session is also terminated. An ongoing business transaction is therefore interrupted and the transaction has to be rolled back to a safe state. This introduces the additional overhead of establishing a new route, authenticating nodes along the way, securing the route and re-starting the entire e-commerce session from the beginning. This scenario is very cumbersome from an e-commerce or e-government perspective.

Existing protocols therefore incur enormous penalty in terms of time and computational requirements for a business transaction. The more mobile the ad hoc network nodes are, the higher the penalty. In this section we show that although there is an overhead for pre-fetching, our pre-fetching approach allows a route to be set up, the new neighbor to be authenticated and a session to be maintained without a break. A transaction is therefore not interrupted, rolled back to a safe state and re-started. Only if the prediction is incorrect, the session is terminated and the transaction rolled back to a safe state. The accuracy of the prediction algorithm is therefore important. Simulation results in section two indicate that our prediction based approach results in significantly fewer lost packets and thereby terminated sessions. It is important that the authentication protocol is lightweight, as the time window for authentication is small and power resources are limited in ad hoc network nodes.

3.1. One-way Hash Chains

Since its first use by Lamport [6] one-way key chains have been widely used in cryptography. A one-way key chain is a chain of keys generated through repeatedly applying a one-way hash function on a random number. For instance, if a node wants to generate a key chain of size N, it first randomly chooses a key, say $K(N)$, then computes $K(N-1) = F(K(N)), K(N-2) = F(K(N-1)), ...,$ repeatedly until it gets $K(0) = F(K(1))$. Here F is a pseudo random function [3] that has the property of one-wayness, that is, if $y = F(x)$, it is computationally infeasible to compute x given y and F. So given $K(i)$, anybody can compute $K(i-1), K(i-2), ..., K(0)$ independently, but they cannot compute any keys in $K(i+1); K(i+2); ..., K(N)$.

3.2. Lightweight Authentication Protocol for ad hoc Networks

It is beyond the scope of this paper to describe the secure protocol in detail. When a node first communicates with a neighboring node, the two nodes authenticate each other by means of a certificate signed by a trusted certificate authority (CA). We assume that each node has a public key certificate signed by a trusted CA. The CA has also a public key. We also assume that each node holds the ids of nodes with which it has communicated. However as nodes move and a new connection is setup between the source node and a different node X, a lightweight authentication protocol is followed if there was a previous communication with node X. Authentication is not achieved by means of a certificate. Instead, the next key in the chain is generated and is used for authentication

Notation:

- A, B are the identities or addresses of nodes
- $CertA$ is node A's public-key certificate issued by a trusted CA
- $Sign_A(M)$ is the digital signature of message M, signed with node A's private key
- $K_A(i)$ is node A's ith key in its key chain
- K_{A-} and K_{A+} is private key and public key of node A
- $eK_{A+}(M)$ is encryption of message M with key K_{A+}

- $dK_{A+}(M)$ is decryption of message M with key K_{A+}
- $md(X)$: message digest value of message X
- $X|Y$: concatenation of two messages X and Y

When a node's signal with its current communicating neighbor is getting weak and the new neighbor has been determined, the route is first setup. Route setup was briefly outlined in section 2. The next step is to establish a trust with the predicted new neighbor. If the predicted new neighbor is not in its list of nodes with whom it has communicated previously, it first establishes a trust with the predicted new node B by sending the following message to B:

$$A \rightarrow B : [[CertA, Sign_A(KA(0)), B]eK_{A-}]eK_{B+}$$

After receiving this message, B verifies the authenticity of node A's certificate using the CA's public key in the certificate to verify the signature on the message. The signature contains A's 0th key in its key chain. The message is encrypted with B's public key. An intruder or eavesdropper cannot therefore read the message. B sends a similar acknowledgment

$$B \rightarrow A : [[CertB, Sign_B(KB(0)), A]eK_{B-}]eK_{A+}$$

Both A and B have therefore authenticated each other and can use the 0th key chain for encrypting their messages.

In an ad hoc environment nodes move, contact is lost and new link is established. Assume now node A detects a weakening signal from B and the prediction algorithm determines its new neighbor will be node C. Node A determines that it had an ith communication with node C before. There is no need to go through the authentication process as described above. Node A sends the following short message to C:

$$A \rightarrow C : [K_A(i+1), C]eK_{A-}$$

Node C verifies that A is the sender by authenticating $K_A(i+1)$, that is, by verifying $K_A(i) = F(K_A(i+1))$ if it has $K(i)$ In response node C sends the following short message to A:

$$C \rightarrow A : [K_C(i+1), C]eK_{C-}$$

Node A verifies that C is the sender by authenticating $KC(i+1)$, that is, by verifying $K_C(i) = F(K_C(i+1))$ if it has $K(i)$ Messages transmitted from A to C are encrypted using $K_A(i)$ and messages from C to A are encrypted using $K_C(i)$. Messages are encrypted using the keys $K_A(i)$ and $K_C(i)$ that were transmitted during a previous communication. Some messages are secret, as C is not permitted to read them, but can only route them on to the ultimate destination. Such messages are encrypted with the key $K_A(i+2)$. This key is not known to anyone, but the key will be revealed after a delayed time when all messages have reached the final destination. Thus our approach follows a lightweight authentication protocol after the first initial authentication between any two nodes. The prediction mechanism allows the same session to be maintained even if connection with one node is broken and a new connection established with a different node.

3.3. Novel Non-repudiation Protocol

Given the constantly changing topology of ad hoc networks, there is a need to ensure that the receiver did get the message from the sender and the receiver needs guarantees that the message is indeed from the node which claims to be the sender. In an e-commerce environment, a retailer needs to be guaranteed that the information he received is indeed from the Supplier and a Supplier needs guarantees that the Retailer did actually get the new product detail. In an election scenario, the voter needs to be assured that her vote was registered for the correct person. If a message has been correctly received, the receiver should not be able to claim that the message from the sender was never received or that a message different from what the sender sent was received. There is therefore a need for non-repudiation mechanisms in such environments. In an ad hoc environment, due to the constantly changing topology, messages are transmitted through different intermediate nodes at different times. Although each node in the link is authenticated as outlined above, an authenticated node including the receiver can deny or distort the data received. We propose in this section a new non-repudiation protocol for a path of business or government transactions. A number of non-repudiation protocols have been proposed [14] [15] . All these protocols however deal with two parties who can communicate directly and a trusted third party is involved to resolve disputes. In our mobile scenario, there are many continuously changing intermediate nodes between the sender and receiver. Furthermore, all previous work deal with a trusted third party who is available if a dispute arises due to repudiation. This is an unrealistic assumption in a mobile environment as the third party will have moved and may not be available at the time of dispute. In our approach, a third party is not required to be available at the time of dispute. In Fig. 3.1 sender node A is communicating with receiver node N. Initially the communication is through nodes B, C, D However due to the mobile nature of the network, over time there may be i different paths used for the same session. It is beyond the scope of this paper to detail our novel non-repudiation protocol and therefore it is briefly outlined next.

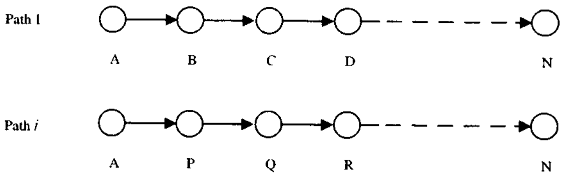

Figure 6. Different paths for A-N

In step 1, the sender generates a secret key randomly and uses it to encrypt the message. It then double-encrypts the secret key (encrypted with the recipients public key and then with the third party authoritys public key). The secret key is encrypted twice for two reasons. Firstly, the sender depends on the third party authority to check the key releasing policy before releasing the key to the receiver and secondly, this will not allow the third party authority to access the key. In step 1 a identification message (id_message)is also generated. This message is encrypted with the public key of the recipient. A signature is also generated on concatenation of the message digest of the ciphered text (the encrypted

content), the message digest of the double encrypted secret key and the message digest of the id of the receiver. All of this information is sent to the recipient in step 1. These steps are outlined below:
K : a symmetric key generated by A (this key is used to encrypt and decrypt the message)
t_id : transaction id
$id\_message$: idenfitication message
msg: message to be sent to N
$em = eK(msg)$
$ek\_from\_S = eK_{N+}(K)$
$dek = eK_{TTP+}(ek\_from\_S)$
$md1 = md(em)$
$md2 = md(dek)$
$md3 = md(N)$

$$id\_N = eK_{N+}(id\_message), id\_TTP = eK_{TTP+}(id\_message)$$

treble signature =

$$t\_id|md1|md2|md3|Sign_{KN-}(t\_id|md1|md2|md3)$$

Message 1: A sends encrypted id_message, encrypted message, double encrypted key and treble signature to B, and B send it to C and so on until it reaches N.

In step 2, A encrypts the id_message with public key of TTP and sends it to TTP.

$$A \rightarrow TTP : eK_{TTP+}(id\_message)$$

When receiving the message of step 1, the recipient N checks the integrity of both the encrypted main message em and the double-encrypted key dek by comparing with the signature. Note that only when the integrity is preserved, the recipient initiates the next step. The progress to the next step implies the recipients confirmation of receiving both the encrypted content and the double encrypted key correctly, so the recipient cannot claim later that he/she had received the wrong encrypted message content. Due to space limitations, we outline the remaining steps. In step 3, the recipient identifies itself to the TTP. For this it decrypts the $id\_message$ (that is, $dK_{N+}(id_N)$) and sends the $id\_message$ to the TTP after encrypting it with K_{TTP+}. The TTP is now able to confirm the identification of N and will send a positive or negative acknowledgment. If it is a positive acknowledgment then (step 4), the recipient forwards the double-encrypted key to the third party authority (TTP), along with its signature to acknowledge the correct receipt of the encrypted message. The recipient is required to send his/her digital signature on the cipher text em, in order to have access to the key. TTP will store the signature temporarily for signature distribution at the end of the protocol. Note that the recipient cannot write a signature on a cipher text em' (where em is actually what the sender had sent and em' is not equal to em) because he/she cannot construct the senders treble signature that contains em' which is needed if there is a lawsuit. In step 5, the TTP decrypts the double-encrypted key and releases the encrypted key to the recipient. Note that TTP is still unable to access the secret key because it is still sealed by the recipients public key. Only the recipient can access the secret key. TTP will log the execution of step 5 in an auditing system. Then, TTP waits for the acknowledgement from the recipient. In case TTP does not receive the acknowledgement within a certain timeout, TTP detects the recipients misbehavior. The sender is protected because TTP has the first signature in step 4 and log of the execution of step 5, which constitutes the undeniable evidence that the recipient receives the message correctly. In step 6, the recipient sends to the third party authority the confirmation of receiving the key. The recipient creates the signature on the digested secret key. The key is digested before being signed with the recipients private key. Message digest uses the one-way hash function, which make it impossible to reconstruct the original content from the digested data. Therefore, TTP cannot access any key information from the signature. Lastly, the protocol ends with TTP forwarding the two signatures in step 4 and 6 from the recipient to the original sender. These two signatures represent the recipients acknowledgments of the receipts of the encrypted ciphertext and the secret key, respectively. TTP collects and forwards these signatures so that the sender does not need the existence of TTP after the transaction is completed. The sender checks if the recipient returns the digital receipts correctly. This can be done because the sender knows what the ciphertext and the secret key he/she had sent. If the sender detects a mismatch with received signatures, it retrieves the execution records of step 4 and step 6 as evidence. The sender can thus prove the recipients misbehavior. Our protocol satisfies the following requirements for a chain of business or e-Government Transactions. Firstly, an intermediate node cannot get key from TTP. Secondly A knows that N got the message. Furthermore, A knows that message N got is correct. A also knows that the key is delivered correctly. In the proposed protocol N can verify that the message is from A. Finally, as this is a mobile environment, the TTP can move on and is not needed at the time of dispute.Space limitations prevent a more detailed presentation of this protocol.

4. Conclusions

Mobile e-commerce or e-government using ad hoc networks poses huge challenges due to the constantly changing topology of the network. Moreover, nodes in an ad hoc network have severe usability limitations and the constantly changing routes pose a security nightmare. Existing ad hoc network protocols are not suitable for e-commerce or e-government applications and very little has been reported in the literature on e-commerce or e-government in ad hoc networks. In this paper we have proposed a predictive location triggered approach to obtaining new routes when communications links are broken due to mobility. Our prediction based approach pre-computes new routes before the existing route is broken. This approach also provides a window of opportunity to authenticate new neighbors in a new route before the old route is broken. Simulation results show that fewer packets are lost and thereby fewer e-commerce sessions are aborted due to the changing topology. A lightweight authentication protocol is also proposed in this paper. Repudiation of origin is a significant and difficult problem in e-commerce and e-government applications. A non-repudiation protocol for mobile environments is outlined in this paper. A number of areas remain for future work. These include performance analysis of the proposed authentication and non-repudiation schemes, reducing the overheads caused by prediction and a more accurate evaluation of the computational and storage overheads for the proposed prediction and authentication mechanism. New lightweight non-repudiation protocols for mobile networks are urgently needed for e-commerce and e-government transactions. The location based approach opens new possibilities for exploring novel security approaches that take into consideration location.

References

[1] M Brown, D Cheung, D Hankerson D, J Hernandez, M Kirkup and A Menezes, PGP in wireless constrained devices, *Proc. 9th USENIX Security Symposium*, (2000) pp. 247–261.

[2] J Broch, D A Maltz, D B Johnson, Y C Hu and J Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, (1998) pp. 85–97.

[3] O Goldreich, S Golswasser and S Mical, How to construct Random functions, *Journal of the ACM*, **33**(4), (1986) pp. 210–217.

[4] Z J Haas and M R Pearlman, *The zone routing protpcol (ZRP) for ad hoc networks (Internet-Draft)*, Aug 1998

[5] D Johnson, D Maltz and Josh Broch, *The dynamic source routing protocol for mobile ad hoc networks (Internet-Draft)*, (1998).

[6] L Lamport, Password authentication with insecure communication, *Communications of the ACM*, **24**(1), (1981) pp. 770–772.

[7] Qin Liang and Thomas Kunz, Increasing Packet Delivery Ratio in DSR by Line Prediction, *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS03), IEEE*, (2003).

[8] C E Perkins and P Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, *Proc. ACM SIGCOMM Symposium on Communication, Architectures and Protocols*, (1996).

[9] C E Perkins and E M Royer, *Ad hoc on demand distance vector (AODV) routing (Internet-Draft)*, (1998).

[10] R Ramanathan and J Redi, A brief overview of ad hoc networks: challenges and directions, *IEEE Communications Magazine*, **40**(5), (2002) pp. 20-22.

[11] M Shen, *Location-Triggered Routing Protocol in Mobile Ad Hoc Networks*, Masters Thesis, Department of Computer Science, Oklahoma State University, (2003).

[12] I Stojmenovic , Position-based routing in ad hoc networks, IEEE *Communications Magazine*, **40**(7), (2002) pp. 128-134.

[13] N H Vaidya and Young-Bae Ko, Location-Aided Routing (LAR), IEEE *Communications Magazine*, (2001).

[14] S. Yang, Stanley Su and H. Lam, A Non-Repudiation Message Transfer Protocol for E-commerce. *Proc. of the IEEE International Conference on E-commerce* (2003).

[15] Zhou and D. Gollmann. A Fair Non-repudiation protocol. *Proc. of 1996 IEEE Symposium on Security and Privacy*, (1996) pp. 55–61.