An Efficient Geographic Location-based Security Mechanism for Vehicular Adhoc Networks

Gongjun Yan, Stephan Olariu
Computer Science Department
Old Dominion University
Norfolk, VA 23529-0162
Email: gyanx001@odu.edu,olariu@cs.odu.edu

***Abstract*—In vehicular adhoc network (VANET), applications are involved with sensitive and secret information. We address a location-based encryption method that not only ensures messages confidentiality but also authenticates identity and location of communication peers. The authentication of location means that a message can only decrypted by the receiver which is "physically" present inside a decryption region that is specified by location, time and speed. A practical mapping function which converts location, time and speed into a unique lock key is proposed. The determination of the decryption region is addressed in two steps: predicting and updating. The proposed method evaluated by simulations is efficient and secure.**

In a vehicular adhoc network (VANET), vehicles are equipped with wireless transceivers so that they can communicate with other vehicles and roadside infrastructure. Many applications proposed for use in VANETs involve the exchange of sensitive information, such as credit card information in toll and e-commerce services, passwords in Internet access systems, medical information of passengers in VANET medical assistance systems, and secret keys distributed by a certification authority. The threats for these applications include the following:

- Eavesdropping on sensitive information while in transmission.
- Modifying messages to receive goods at another driver's expense.
- "Phishing" to direct traffic to a fake website/shop in order to collect sensitive information.

Authentication of the communication peers and encryption of the secret messages are often used as countermeasures to these attacks. These methods can prevent the eavesdropping and modification threats, because the attackers can neither decrypt the ciphertext of the secret messages nor pretend to be communication peers. However, "phishing" or a similar masquerading attack cannot be solved by authentication and encryption alone. Take, for example, a toll plaza that collects fees through wireless communication. Instead of stopping to pay a toll, each vehicle contains an RFID tag associated with a credit card. As vehicles pass through the toll plaza, their credit cards are automatically charged. In an attack, a vehicle could masquerade as a toll plaza in order to trick another vehicle into paying for the attacking vehicle's toll. In roadside e-commerce, a vehicle could pretend to be a phishing site such as a roadside restaurant in order to collect credit card information. Therefore, to prevent these types of phishing attacks, the locations of the communicating peers need to be authenticated.

In this paper, we address a location-based encryption method that not only ensures message confidentiality, but also authenticates the identity and location of communicating peers. Our method is an extension of *geo-encryption*, proposed by Denning *et al.* [1], [2]. Geo-encryption limits the area inside which the intended recipient can decrypt messages. Our main contributions include: 1) a detailed design of the key composition and recovery mechanism, including techniques to map the location coordinates to a unique value in order to authenticate the communicating peer's location; 2) the prediction of the decryption region in a dynamic vehicular environment. Prediction error is considered by incorporating location prediction deviation, which is dynamically updated based on real locations; 3) the modification of geo-encryption. The population of vehicle is huge and vehicles move from place to place. It is not feasible to use asymmetric cryptographic algorithms like public key infrastructure (PKI). We modify the geo-encryption scheme to adopt symmetric cryptographic algorithms. Moreover, encryption rate is improved by using symmetric cryptographic algorithms.

I. The State of Art

A. Encryption and Authentication

There are two basic types of encryption algorithms, asymmetric and symmetric. In asymmetric algorithms, each node has a public key and a private key. The public and private keys are special in that a message encrypted with a node's public key can only be decrypted using the node's private key, and vice versa. In public key infrastructure (PKI), a well-known mechanism for using and distributing public keys, a certification authority (CA) is responsible for validating public keys and distributing certificates used for authentication. In symmetric algorithms, the communicating peers share a secret key. Both encryption and decryption are performed using the same secret key, thus the secret key must be protected.

PKI and digital signatures are well-explored methods in VANETs [3], [4], [5]. A CA generates public and private keys for nodes. When a node A sends an encrypted message M to node B, A will encrypt M by using the public key of B. Only B has private key, so only B can decrypt the ciphertext. If B

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

wants to sign the message M, B encrypts the message with its private key and sends both M and the signed version of M to A. A then verifies the signature by using B's public key to decrypt the signed version. If the result is M, A will accept M as sent by B, because only B has the private key to generate the unique signature. Laberteaux *et al.* [6] discussed applying a similar method to sign messages in VANETs. The purpose of the digital signature is to validate and authenticate the sender. The purpose of encryption is to disclose the content of message only to the nodes with secret keys. PKI is a method well-suited for security purposes, especially for roadside infrastructure, like roadside e-shops, Internet access points, etc.

But, there are some issues in using PKI in VANETs. The main problem is the need for a trusted CA to distribute public keys and certificates. In order for all vehicles to be able to communicate with each other, all vehicles will have to trust the same CA, a difficult requirement when vehicles are manufactured by different companies in different countries. In addition, bad or mis-used certificates must be revoked. The list of revoked certificates must then be distributed to all vehicles. Another problem is that asymmetric encryption/decryption often takes 1000 times longer to perform than symmetric encryption/decryption [2]. In addition, nodes in VANETs can communicate in groups [7], [8], [9], [5]. In this case, the requirement of public keys for all nodes is not needed, because the vehicles in a group can share the message. In this paper, we improve encryption/decryption speed by using symmetric algorithms. Therefore, we have to design the secret key. The secret key is based on the position of vehicles and no extra cost needed. Moreover, the secret key is a group key which is shared by a group of nodes.

B. Location-Based Encryption

Location-based encryption method is proposed by Denning *et al.* [1], [2] that limits the area inside which the intended recipient can decrypt messages. This *geo-encryption* integrates geographic and mobility information (such as position, time, speed, etc) into the encryption and decryption processes. Denning proposes GeoLock that is computed with the recipient's position, velocity, and time, which is shown in figure 1. Using the same notations in previous section, the GeoLock of A is processed by modulo operation with a secret key Key_S and then the result is encrypted by public key Key_E of B and sent to B which decrypts the ciphertext using private key Key_D of B. The secret key (symmetric key) Key_S is obtained and decrypted the message.

GeoLock is the key function of geo-encryption. Positions are signed and mapped to GeoLock which is like a value of a grid composed by xy-coordinates of positions. However, the mapping function in practice is not specified in Denning's work. If the mapping function is pre-installed tables which is shown as example of the mapping function in [2]. It is extremely hard to ensure the synchronization of the key mapping grid in vehicular networks for two reasons. First, the population of nodes in the network is large, and it would be costly to replace the grid for all vehicles. Second, the mobility of nodes makes the nodes to immigrant from place to place. If the grid is not synchronized, the communication peers will not be able to communicate.

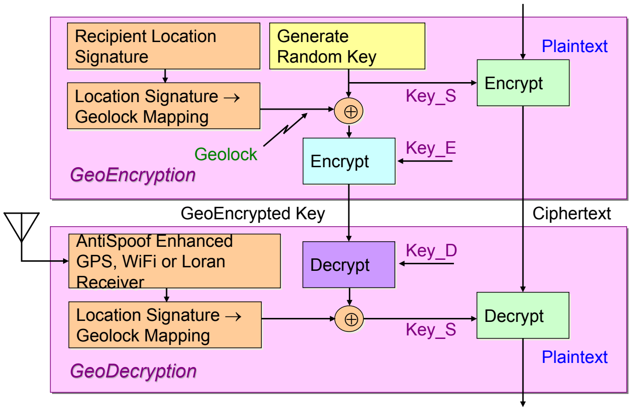

Recipient Location Signature
Generate Random Key
Plaintext
Encrypt
Key_S
Location Signature → Geolock Mapping
⊕
Geolock
Key_E
Encrypt
GeoEncryption
GeoEncrypted Key
Ciphertext
AntiSpoof Enhanced GPS, WiFi or Loran Receiver
Decrypt
Key_D
Location Signature → Geolock Mapping
⊕
Key_S
Decrypt
Plaintext
GeoDecryption

Fig. 1. Denning's geo-encryption [2]

Denning's geo-encryption model did not include details of an implementation of mobility support, so Al-Fuqaha *et al.* [10] proposed a model to provide for mobility when using GPS-based encryption. The decryption zone where the message is allowed to be decrypted contains a mobile node's estimated location. However, the decryption region predicted by Al-Fuqaha is designed for slow or constant mobility nodes. The location predicted by Al-Fuqaha does not include prediction errors. But in VANETs, the nodes have high mobility which will definitely cause a certain prediction errors. Vehicles can move about 33 meters per second (75 miles/hour) and can turn at street intersections, stop, accelerate, decelerate, etc. The dynamics of a vehicle's mobility will make the prediction from a sender difficult and inaccurate. For example, the delay caused by decryption will cause a certain distance of movement of vehicles. This movement will make the decryption prediction inaccurate.

In this paper, we design the key composition/recovery in detail. No mapping tables are needed. Positions can be mapped to a lock on the fly. Since nodes in VANETs are with high dynamics, the decryption region is designed as a series of fixed-size squares in this paper. The area of the square is large enough to cover the error of the decryption region prediction. Moreover, we incorporate the prediction error by using location prediction deviation. We trade freedom of the size and the shape of the decryption region in order to obtain the feasibility and the accuracy of decryption region prediction.

II. An Overlook of Encryption and Decryption

In this paper, we discuss the geographic location-based security in a client-server scenario that the server is a fixed end and its public information such as GPS location and public key are known by all clients. We extend the scheme of encryption/decryption on the basis of geo-encryption algorithm in [2] by removing the public key and private key requirement on vehicles. If vehicles use PKI to communicate, they have to exchange the public key before they can communicate. Constantly broadcasting public key is not efficient for communication, especially in real-time applications. In this paper, only the roadside-servers have to maintain public key and private key pairs. We assume that the symmetric encryption algorithm is used on clients/vehicles because transmitting a secret key to server is more challenging than transmitting a public key. In addition, symmetric algorithm has a faster encryption/decryption rate than asymmetric algorithm.

Our technique involves a security key handshake stage and a message exchange stage, as shown in Figure 2. In the key handshake stage, the client and the server negotiate a shared symmetric key. The client generates two random numbers as keys Key_S and Key_C. Key_S is used to encrypt a message composed of the aggregated location message and Key_C. This encrypted message is $E\{Req\}$. The client generates a GeoLock based on the location of the server. This value is XOR-ed with Key_S and then encrypted using the server's public key Key_E to produce the ciphertext $E\{Key\}$. Both $E\{Req\}$ and $E\{Key\}$ are transmitted to the server through the wireless channel. When the server receives $E\{Key\}$, it is decrypted using the server's private key Key_D to produce the XOR of the GeoLock and Key_S. The GeoLock generated from the GPS location of the server is used to recover the secret key Key_S. Then, Key_S is used to decrypt $E\{Req\}$ to obtain the aggregated location message and the secret key Key_C.

In the message exchange stage, the server and client use the shared Key_C to communicate. When the server wants to reply to a client, it generates a random number, Key_S'. The reply message is directly encrypted using Key_S' to generate a ciphertext, $E\{Rep\}$. Since the aggregated location message contained the client's GPS position, the server can generate a GeoLock of the client vehicle's decryption region. This GeoLock is XOR-ed with Key_S' and then encrypted with Key_C to generate a ciphertext, $E\{Key'\}$. Both $E\{Rep\}$ and $E\{Key'\}$ are transmitted to the client through the wireless channel. $E\{Key'\}$ is then decrypted using Key_C to recover the XOR of the client's GeoLock region and Key_S'. The client generates its GeoLock based on its current location. This is used to recover the secret key Key_S'. $E\{Rep\}$ is decrypted using Key_S', and the reply message is recovered. The client repeats the algorithm in the message exchange stage to communicate with the server.

III. Decryption Region in Vehicular Networks

The geo-encryption protocol allows nodes to securely communicate with nodes at a particular location and time period. We enhance the geo-encryption methods by the special features of vehicular networks. In this paper, we have two improvements of determining decryption region: predicting and updating decryption region. The movement of vehicles is constrained by roads, and the map of the roads can be accessed by all vehicles. Therefore, we can predict vehicles' position based on the map and vehicles' mobility. Based on the prediction of decryption region, the communication messages are checked by geographic location. Because of dynamics of vehicles, there will be a certain prediction error. Therefore the predicted decryption region will be corrected by updating the real positions. The real positions are piggy-backed by communication messages.

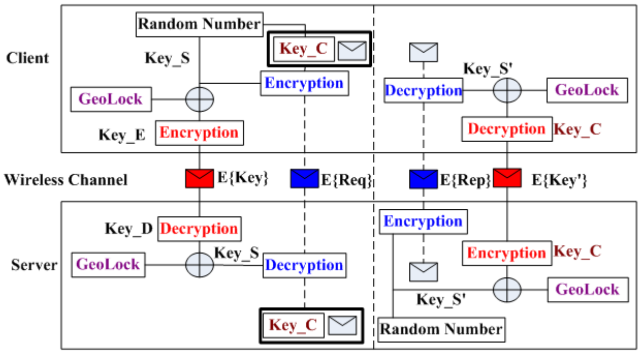


Fig. 2. *Illustrating the proposed encryption and decryption scheme.*

A. Prediction of the Decryption Region

The position coordinates use Universal Transverse Mercator (UTM) [11]. GPS coordinates received by GPS receiver will be converted into UTM coordinates. An example of UTM coordinates is a location called Hilltop 3705 which is located at grid 0577591395 or Zone 13 705775E 3391395N [11]. One of the great features of UTM coordinates is the ability to provide a more precise location by simply adding a pair of digits to abbreviated coordinates [11], for example 8 digits UTM location, accurate to 10 meters, approximately the size of a house. Another example is 10 digits UTM location, accurate to 1 meter. The secret key is used for symmetric cryptography algorithms (for example Data Encryption Standard (DES)) which use trivially related, often identical, cryptographic keys for both decryption and encryption.

Suppose the target decryption region starts from position $P_0(x_0, y_0)$. The decryption region is assumed as a square region. Since a square region must have two components: starting point, length (length equals to width). Since the starting position can be predicted by checking maps and mobility of vehicles, only the length of square needs to be determined. The length of square is listed as a series of scales: L, for example, 10, 20,..., 1000 meters. For 10 digits UTM positions, $1 < L < 10^4$ because the precision is about 1 meter. For 8 digits UTM positions, $10 < L < 10^7$ because 8 digits UTM positions are accurate to 10 meters ([11]). For 6 digits UTM positions, $L < 10^4$ because 6 digits UTM positions are accurate to 100 meters. No smaller than 6 digits can be use in our proposal. Therefore, the length of square is selected from one of the three possible lengths: 10, 100, 1000 meters.

The decryption region can be predicted in several ways in vehicular networks based on the map of roads and mobility of vehicles. The methods that predict the receiver's region are the following.

1) The location of communication peers can be calculated

on the basis of the mobility parameters including current speed, current position, current acceleration, etc. This is a major method. A new position after a certain time interval can be computed. Suppose at time t_0, the target vehicle is at location (x_0, y_0) with speed v_{x0}, v_{y0} and acceleration a_{x0}, a_{y0}, where x_0, v_{x0}, a_{x0} are the x-axis value of initial location, relative speed on x-axis direction, and relative acceleration on x-axis direction; y_0, v_{y0}, a_{y0} are the y-axis value of initial location, relative speed on y-axis direction, and relative acceleration on y-axis direction;. After time interval t, we can roughly predict that the vehicle will at a place near location region: x_1, or

$$x_1 \in [x_0 + v_0 t + \frac{1}{2}a_0 t^2 - \alpha * XDeviation, \\ x_0 + v_0 t + \frac{1}{2}a_0 t^2 + \alpha * XDeviation] \quad (1)$$

; and y_1, or

$$y_1 \in [y_0 + v_0 t + \frac{1}{2}a_0 t^2 - \alpha * YDeviation, \\ x_0 + v_0 t + \frac{1}{2}a_0 t^2 + \alpha * YDeviation] \quad (2)$$

, where $x_1, XDeviation$ are the location prediction on x-axis and the deviation of position value of x-axis; $y_1, YDeviation$ are the location prediction on y-axis and the deviation of position value of y-axis and α is the coefficient which implies the affection of the deviation, $0 \geq \alpha \geq 1$.

2) If the decryption region is a fixed area, we can directly check the map of roads and calculate the GPS coordinates. This is the simplest scenario. Usually the e-business location is known on digital map. A location of a new business can be registered by the digital map generator.
3) If the decryption region is dynamically moving, we can calculate the position of the decryption region by querying the target receiver. This method is addressed in ([10], [12]).

B. Updating The Decryption Region

Although the decryption region is predicted, there are prediction errors of decryption region because of dynamics of vehicles. Therefore, the decryption region needs to be corrected to improve prediction precision for next communication. The predicted position will be updated by the real position which is piggybacked in communication messages. The speed, acceleration and direction of move will be piggybacked as well. Therefore, the updating step includes the following assignment:

$$x_1 = x_{real} \quad (3)$$

$$y_1 = y_{real} \quad (4)$$

$$XDeviation = (1 - \beta) * XDeviation + \beta * |x_{real} - x_0| \quad (5)$$

$$YDeviation = (1 - \beta) * YDeviation + \beta * |y_{real} - y_0| \quad (6)$$

where (x_{real}, y_{real}) is the real position piggybacked, β is the coefficient value which implies the effect of the prediction error $|x_{real} - x_0|$.

The updating frequency is depended on the mobility of receiving vehicles, the precision requirement of decryption region and the bandwidth of control channel. For example, the frequency of updating on highways is much higher than the frequency of updating on urban area because the velocities on highway are much higher than the ones in urban area. Similarly, precision of decryption region and the bandwidth of control channel impact the updating frequency as well.

IV. GeoLock Mapping Function

The GeoLock mapping function converts geographic location, time and mobility parameters into a unique value as a lock. This unique lock value validates that the recipients satisfy certain restriction, for example the decryption region at a certain time interval. The mapping function can be composed by several parameters: position coordinates (x_0, y_0), time interval T, and speed interval V. The concept of GeoLock is proposed by [1], [2]. Our contribution of the mapping function is to provide a feasible and detailed method in VANETs. The mapping function can convert lock value on the fly. There are no preinstalled mapping tables in our proposal.

A. From the sender's view

The process of generating a lock value/key is shown in Figure 3. First of all, all the input parameters are operated respectively. The location (x_0, y_0) will be divided by the length of decryption region (square) L. For example, the length of target decryption region is 100 meters or $L = 100$, each of coordinate number of $P_0(x_0, y_0)$ will be divided by 100. The integral part after division will be obtained. Therefore, bigger L will cause less digital numbers of the output from step one. Less digital numbers will result in weaker lock key. If the value of L is small, there is a risk that a lock key may be computed by brute force attack. Second, the output of the first step is multiplexed or reshuffled. Third, the output of the second step is hashed by a hash function. The hash function in practice can be designed as mod operation or as standard hash function, like Secure Hash Algorithm (SHA) functions which are a set of cryptographic hash functions designed by the National Security Agency (NSA) and published by the NIST as a U.S. Federal Information Processing Standard.

We use Scott and Denning's [2] idea of a GeoLock to map the geographic location of the decryption region of the server into a lock value. This ensures that a vehicle be physically

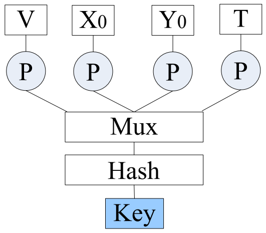


Fig. 3. GeoLock mapping function.

present inside the decryption region to decrypt the message. In Scott and Denning's GeoLock algorithm, the mapping is based on a fixed table which has to be synchronized on all nodes. In our technique, the inputs to the tamper-proof GeoLock function are a GPS position and the length of the square decryption region. The GPS coordinates are divided by the length of the decryption region. The concatenation of the integral remainders of the GPS coordinates are then hashed to produce the GeoLock. To create regions of varying size, we allow the length to be specified as 1 meter, 10 meters, 100 meters, or 1000 meters. An example is shown in Figure 4. First step, two numbers are divided by the length of the region 100, i.e. (042.00, 915.00). The integer part after division, i.e. (042, 915) is kept. Second step, the two numbers: (042,915) are multiplexed as 042915. Third step, the multiplexed number is hashed by SHA to generate the lock value.

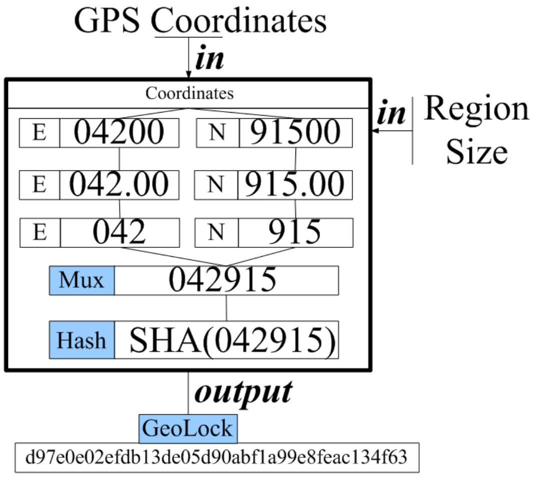
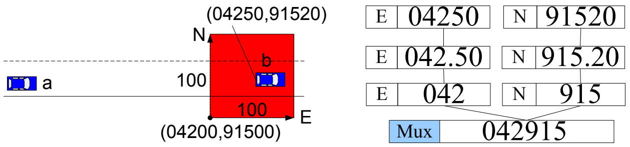


Fig. 4. *An example of GeoLock.*

B. From the recipient's view

After the receiver vehicle b receives the message and decrypts the message by the private key, the secret key will be recovered. The recipient's GPS coordinates are read from the enlisted GPS receiver. The other parameters in figure 3 can be obtained on the recipient vehicle b. The same mapping function discussed in section IV-A is used to compute a lock key. If the vehicle b is restricted by the decryption region in terms of location, time and relative speed, the exact same lock value will be generated. Otherwise, the lock key will be different. The secret key Key_S will not be recovered and the ciphertext $E\{Req\}$ will not be decrypted.

An example of the mapping function on the receiver's view is shown as Figure 5(a) and 5(b). The receiver vehicle b is located at location $(04250, 91520)$ (UTM 10 digital coordinates) shown in Figure 5(a) and the decryption region L is 100 meters. Figure 5(b) shows GeoLock on recipients. First step, the xy-axis coordinates of location $(04250, 91520)$ are divided by the length of the region 100, i.e. (042.50, 915.20). The integer part after division, i.e. (042, 915) is obtained. Second step, the two numbers (042, 915) are multiplexed as 042915. At this point, the multiplexed number is exactly same as the one in key generator on sender side. Hash function (SHA) will generate exactly same key as well. We show that the lock value generated on the receiver side is exactly same as the one computed in GeoLock from the sender's view. It is obvious that the vehicles will pass the geographic validation.



(a) A decryption region specified by a position and a length.

(b) Computing decryption key.

Fig. 5. An example of secret key recovery.

V. SIMULATION RESULTS

A. Simulation Settings

We used SUMO [13] and ns-2 [14] for our simulation. The simulator SUMO is an open source and microscopic road traffic simulation package. The simulator ns-2 is an well-known open source and the second version of network simulator. The SUMO creates a trace file which records the mobility of vehicles. We loaded the trace file into ns-2 to simulate the security. Since our interest is of location-based security, we predicted a decryption region based on the updated vehicle's location. The network routing protocol we were running is the Ad hoc On Demand Distance Vector (AODV). The application is Constant Bit Rate (CBR) with 16 packets every second. The total amount of vehicles is 320. The map is 3.2km x 3.2km. The number of roadside shops is 20. The decryption region of roadside shops is a square of 10m x 10m. Vehicle's decryption region is determined by roadside shops and dynamically changed on the basis of vehicle's mobility. The size of vehicular decryption region is 3m x 3m. We recorded two events:

- Decryption failure, a message is failed to be decrypted.
- Decryption success, a message is successfully decrypted.

First, the decryption ratio over location tolerance/precision is investigated. We measured the decryption ratio as the ratio of successfully decrypted messages over those messages that were received. This ratio is not the delivery ratio but the decryption ratio inside the decryption region. We varied the location tolerance because location detection has precision problem. The square size is set as 10 meters for roadside shops and 3 meters for vehicles. We compared two set of results with different speed (24 meters per second and 14 meters per second) which is shown in Figure 6. As we expected, the increase of location tolerance will cause the decrease of the decryption rate. Besides, the faster speed will cause lower decryption rate. This is because that the increase of location tolerance and the increase of speed will cause the increase of false location of vehicles. Since location-based decryption is based on the location, the false location will cause the failure of decryption. But we consider the deviation of decryption prediction and adjust dynamically. Therefore, the decryption ratio is higher in our algorithm than the one in Al-Fuqaha's algorithm which does not consider the prediction errors.

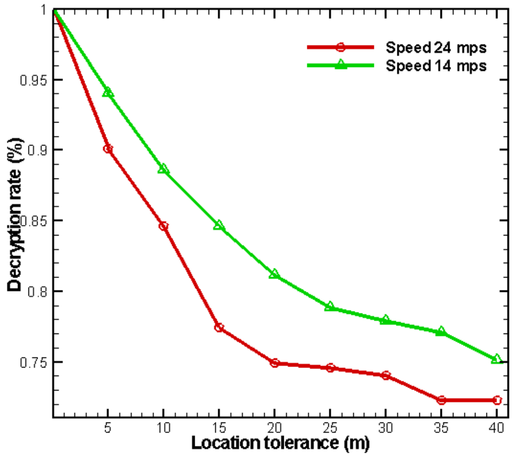


Fig. 6. Decryption ratio

Security is not free but with cost. We measured the overhead (packet size increase) and the decryption time by varying the updating pause time. The velocity of vehicles is set as 20 m/s. The updating pause time is the time interval to update locations of senders and receivers. Figure 7 shows the result that the percentage of overhead increment decreases while the updating pause increases. Al-Fuqaha's algorithm has lower percentage of the decryption ratio than our algorithm. Since we took the updating/control message as the overhead, a short updating pause means a fast frequency of control message which means a large amount of overhead. The large updating pause will cause low location precision of decryption region. In our algorithm, the fixed-size square is less sensitive to the change of the pause than the Al-Fuqaha's algorithm.

VI. Conclusion

We describe a feasible and novel geographic location-based security mechanism for vehicular adhoc network environment on the basis of concepts proposed by [1], [2]. Comparing with the [1], [2], our algorithm is efficient on the basis of simulation. The future work will integrate the model into the existing security methods. The shape of the decryption region will be extended to any shape.

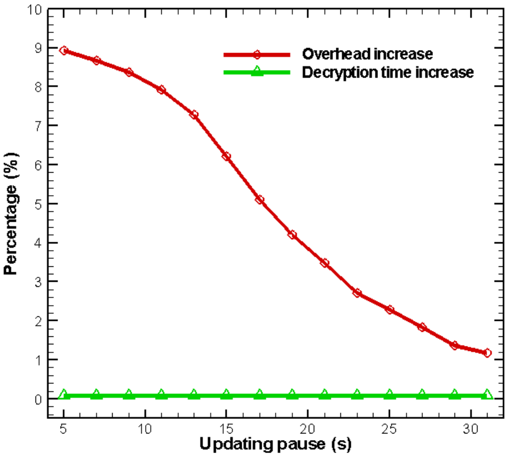


Fig. 7. Overhead and decryption time

Acknowledgment

This work is supported in part from the NSF grant CNS 0721586.

References

[1] D. Denning and P. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud and Security*, vol. 1996, no. 2, pp. 12–16, 1996.

[2] L. Scott and D. E. Denning, "Location based encryption technique and some of its applications," in *Proceedings of Institute of Navigation National Technical Meeting 2003*, Anaheim, CA, January 22-24, 2003 2003, pp. 734–740.

[3] J. Y. Choi, P. Golle, and M. Jakobsson, "Tamper-evident digital signatures: Protecting certification authorities against malware," in *Proceedings of the IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC)*, 2006, pp. 37–44.

[4] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security and Privacy Magazine*, vol. 2, no. 3, pp. 49–55, 2004.

[5] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications Magazine*, pp. 8–15, Oct. 2006.

[6] K. P. Laberteaux, J. J. Haas, and Y.-C. Hu, "Security certificate revocation list distribution for vanet," in *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, 2008, pp. 88–89.

[7] G. Yan, S. Olariu, and M. C. Weigle, "Providing VANET security through active position detection," *Computer Communications: Special Issue on Mobility Protocols for ITS/VANET*, vol. 31, no. 12, p. 2883C2897, 2008.

[8] R. Ramesh and S. Kumar, "Secure position routing using ad hoc network," Dec. 2006, pp. 200–201.

[9] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient Secure Aggregation in VANETs," in *Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET)*, Los Angeles, CA, Sep. 2006, pp. 67–75.

[10] A. Al-Fuqaha and O. Al-Ibrahim, "Geo-encryption protocol for mobile networks," *Comput. Commun.*, vol. 30, no. 11-12, pp. 2510–2517, 2007.

[11] J. N.G. Terry, "How to read the universal transverse mercator (utm) grid," GPS World, April 1996, pp. 32.

[12] L. Scott and D. Denning, "Geo-encryption: Using gps to enhance data security," GPS World, April 1 2003.

[13] Open source, "Simulation of urban mobility," http://sumo.sourceforge.net.

[14] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.