High Speed Data Routing in Vehicular Sensor Networks

Harry Gao, Seth Utecht
Department of Computer Science, College of William and Mary
Gregory Patrick, George Hsieh
Department of Computer Science, Norfolk State University
Fengyuan Xu, Haodong Wang, Qun Li
Department of Computer Science, College of William and Mary

***Abstract*—In this paper, we show through a simple secure symmetric key based protocol design and experiments the feasibility of secure data collection in a vehicular sensor networks. This protocol exhibits high speed data routing for sensor data collection through vehicles. The large communictaion and storage capacities of a vehicle and its mobility facilitates this high speed routing scheme compared with routing through hop-by-hop communication among sensors. We demonstrate that the protocol works in a realistic setting by collecting the real trace data through real implementation.**

I. Introduction

Vehicular network has attracted people's attention in recent years with the vision that it can provide crucial information, such as traffic conditions, to interested parties. The vehicular network architecture is mainly composed of inter-vehicular communications, and communication between vehicles and the roadside sensors. Although the picture is very exciting, we do not expect the technology to be mature in the next couple of years for very practical deployment. This is due to the hurdles along the way: the standardization of the network communications, the effort associated with the deployment of the roadside sensors, and the maturity of the hardware. These processes can be both expensive and time consuming.

We argue in this paper that a simple architecture based on sensors (e.g., Berkeley Motes) can fulfill many important functions envisioned in vehicular networks. Sensors are deployed along the roadside to collect environmental data. For example, the sensors can gather data on highway conditions (e.g. potholes, cracks on the road, ice on the road and blind spots ahead). They can also monitor the environment for scientific purposes, such as monitoring pollution or pollen count. Moreover, they can be used as a temporary storage space for data. For instance, if a vehicle notices a collision ahead, it can send a message to the roadside sensor so that the vehicles behind may know the information when they are within the transmission range of the sensor. This architecture becomes more powerful when the vehicles are harnessed to carry data stored or collected in a sensor to the more sophisticated servers deployed at weigh stations, toll gates, or rest areas. This information can then be processed, analyzed and broadcasted, and can be made available to the general public through services such as Google Map. A small network following this architecture is inexpensive and easy to deploy, because the price of motes continues to drop. A vehicle simply needs to be enhanced with the capability of communicating with sensors.

In this network architecture, it is crucial to provide security support – only authorized vehicles can feed data into sensors and to obtain data from sensors. To block unauthorized and malicious vehicles, data collected by sensors must be encrypted. However, merely encrypting the data cannot prevent a malicious car to obtain the scrambled data. Although the encrypted data is of little use to the malicious vehicle, it is a serious problem when sensors expect vehicles to harvest all the data and carry them to a central station; a malicious vehicle can simply trap the data and leave a hole in the designated data repository. Therefore, authenticating a passing vehicle before transferring any data is indispensable.

A straightforward solution is to use a public-key based scheme, since some (e.g., the ECC) of the schemes can be implemented efficiently on sensor platforms. Taking a closer look at the problem in a real experimental study, we found that authentication takes about one to two seconds in many cases, which is non-negligible for a car traveling at high speed. A car may rush out of a sensor's transmission range after the authentication is conducted. In this paper, we show our security solution to the vehicular sensor networks and give experimental results on a realistic deployment. We show through a simple secure protocol design the feasibility of secure data collection in a vehicular sensor networks.

We deployed sensors along the roadside to test the performance of the communication between the roadside sensors and the sensors in a moving vehicle. We demonstrate the protocol works in a realistic setting by collecting the real trace data through real implementation. We hope this research shows valuable experience in deploying security support for this type of networks.

II. Related Work

There already exist many proposals for some of the challenging aspects, such as the session layer protocol

Manuscript received May 10, 2009; accepted Nov. 17, 2009.

doi:10.4304/jcm.5.3.181-188

Jorg Ott et al. proposed, which addresses disconnection tolerance [11]. This paper strives to improve the security aspect. A survey of the security of vehicular network can be found in [2] [10].

There are many vulnerabilities for an unsecured network, such as jamming, forgery, impersonation, and in-transit traffic tampering [12]. Research shows that a symmetric key scheme is required [10]. A number of researching teams have put forth many solutions to group key and authentication problems [3] [5] [6] [13] [15]. Some of them utilize the Cabernet system where data is delivered opportunistically during travel. While less powerful, it does provide a quick solution that can be implemented without an overhaul [7] [9].

One possible solution is to utilize Wi-Fi connections. They generally involve an open connection where users directly obtain data from various sensors, but many issues, such as not having a standardized handoff scheme, result in “poorly performing and proprietary manufacturer dependent mechanisms and policies” [8]. Other problems include bandwidth limitations, fairness and increased vulnerability as discussed by Bychkovsky et al. [1] [4].

There have also been some research exploring the possibility of using a certificated-based protocol, such as the one proposed by Wang et al. [14]. It uses the short-range radio communication of motes to pass information from various roadside measuring devices to an information-gathering car mote, which then carries the information to a computer to process. However, it does not address some of the issues unveiled by Balfanz et al. [3].

III. Problem Formation

This paper discusses the issues and resolutions of the following problem.

A. Problem Setting

- Many stationary sensors deployed on the side of the road that can detect, measure and record a certain aspect of the traffic pattern, such as the speed of vehicles in its range. Such motes are currently commercially available. It does not possess significant computational power, nor does it have much storage space.
- Another mote similar to the stationary ones placed in a car that can gather information from the stationary motes and then deliver it to a computer in the car. Its responsibility requires it to be able to securely communicate with other motes and with a computer.
- An upload server, which is to be located at the end of the road. It should be a computer with Wi-Fi capacity. It will obtain the relevant information from the computer in a car once in range. This upload server can be located in a toll gate, rest area, or any other similar structures. This server should have the ability to process and analyze the raw data, and notify relevant parties of its findings. As a computer, the upload server has much computational power. It is also assumed that its storage is virtually unlimited. This is justified by the fact it can communicate with external servers across the Internet. However, the upload server cannot communicate directly with the stationary motes.
- An authentication server, which is to be located at the beginning of the road. It should also be a computer with Wi-Fi capacity. It will permit the car mote to communicate with the stationary mote after the server verifies the car mote’s identity. The exact protocol of authenticity between the car mote and authentication server is not discussed in this paper. This server, like the upload server, is assumed to have great computational power with virtually unlimited storage.

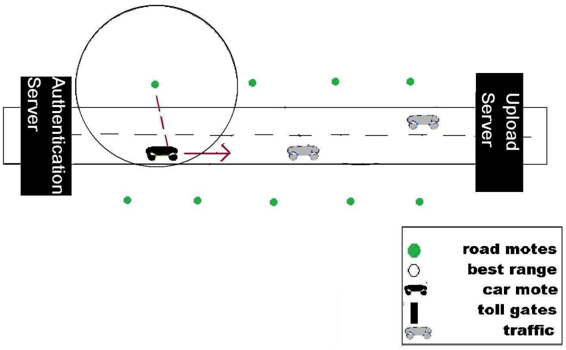

Fig. 1. A scenario of collecting data from roadside sensors in a vehicular networks

B. Goals

The protocol attempts to solve the problem with the equipments outlined above with the following properties:

- **Secure**. To ensure security, instead of granting the car mote the secret key, the authentication server gives the car mote a session key so that it can obtain the necessary information, but cannot decrypt the data. A four-way handshake between the car and the stationary motes allows for both parties to authenticate each other. The upload server, at the end of car mote’s travel, receives, decrypts, analyzes and broadcasts the data obtained.
- **Reliable**. To provide the greatest reliability, we analyzed the frequency and timing of received/dropped packets, as well as the RSSI, at various speeds. This will provide insight to the overall reliability of the implementation, as well as to the most opportune moment for transmission.
- **Efficient**. It is very important for the implementation to be efficient. Since the commercially available motes all use battery power, the efficiency of the implementation directly determines the longevity and sustainability of the infrastructure, or at least the frequency of maintenance required. More crucially, an inefficient implementation may result in too long a handshake process, which may result in the car mote

rushing out of the reception range of the stationary mote before all of the data is transferred. This will greatly reduce the reliability.

- **Deployable**. Lastly, the protocol proposed must be realistic enough to be deployed without an excessive amount of resources. That is, the assumptions and equipments abovementioned must be reasonable and available.

C. Adversary

The adversary is an unauthorized party that wishes to either obtain the secure information gathered by the roadside mote and/or block the car mote from gathering it. The adversary is assumed to have unlimited access to any public information. It may try to impersonate the car mote to hijack the data, or impersonate a stationary roadside mote to provide the car mote with falsified data. Since the car mote is not a part of the infrastructure, it is reasonable to assume that the car mote is malicious. In case the adversary successfully obtains the session key, it still cannot forge inaccurate data to the US, for it does not know the secret key with which the data is encrypted. Ideally the authentication server will accurately identify any malicious parties before granting access.

The adversary is not expected to have the secret key, the protection key, or know the hash functions with which the data is encrypted. It is not able to physically damage or remove any of the structures mentioned above, nor is it able to rewrite any piece of software implemented on either the servers or the motes. The adversary is not expected to be able to crack the security via brute force within any reasonable time frame.

IV. PROTOCOL

The following protocol is designed in order to balance security with efficiency. It provides a 4-key system that safeguards against various malicious parties, as well as a 4-way handshake that allows the roadside mote and car mote to mutually authenticate. The symbols used in the protocol are summarized below:

Variable	**Symbol**
Car mote	*C*
Roadside mote	*M*
The authentication server	*AS*
The upload server	*US*
Secret key shared by *M* and *AS*	*s*
Random number to generate *k*	*R*
The session key generated by *AS*	*k*
Key known only to *M* and *US*	s_p
The data encrypted with s_p	*m*
Randomly generated challenge	*r*
Hash function of *r* and *s*	*hash(r,s)*
The temporary key	TK
Random number used to verify TK	r_t

The first of the two servers provides car mote *C* with the authentication information, while the second uploads and processes the data collected by *C* at the end of *C*'s trip. *M* wants to transfer data reliably to *C* as *C* passes by. On the road, *C* will mutually authenticate with *M* and collect data encrypted with the secret key. At the end of the road, *C* will send all collected data to the upload server, which can use the protection key to decrypted the messages.

A. Pre-distribution

The *AS* shares the secret key *s* with *M*. The *US* shares the protection key s_p with *M*.

B. Communication between AS *and* C:

To provide *C* the ability to transport the information *M* holds, *AS* generates a random number *R*, and use this *R* to form a session key *k*:

$$k = hash(R, s) \tag{1}$$

AS gives *k* and *R* to *C* securely, but withholds *s*. *C* will now possess all the information it needs to collect data from *M*. *AS* also provides *C* with the bitmap it obtained from the *US*, so *C* can notify *M* that the data it transferred to the previous *C* safely arrived at the *US*, and can now be deleted from *M*'s memory.

C. Communication between C *and* M

C broadcasts probe messages containing *R* on the road. When *C* and *M* are within the range of communication, *M* would receive the message and generate the session key *k* = hash(*s*,*R*), which is the same key known to *C*. *M* can then perform a 4-way handshake with *C*:

1) *M* generates a random challenge *r*, and send it back to *C*;
2) *C* generates another random number R_c; compute a temporary key;

$$TK = hash(k, r, R_c) \tag{2}$$

then it generates a MAC for R_c by using this TK and sends both the MAC and R_c back to *M*
3) *M* compute the TK in the same way to verify the MAC. *M* then send back another random number r_t with a MAC generated using this TK. After verification, *C* will confirm that *M* get the TK correctly.
4) After that, both motes are authenticated with each other. *M* can start the encrypted data transfer.
5) (optional) *C* sends the delivery conformation bitmap to *M*; *M* then know that the previous *C* successfully delivered the data to the *US* and the *M* can delete these data from its memory.
6) *C* continues to travel and gather data from other *M*s in the same way.

D. Communication between C *and the* US

Upon arriving in the range of the upload server *US*, *C* uploads all gathered data as well as the session key to the *US* for processing. *US* should have the ability to decrypt the data with the protection key, and it can verify with the *AS* that the session key of the *C* is valid. Once uploaded, *C* has finished its mission and can now reset. *US* can then create a bitmap of all the data successfully decrypted, encrypt it with the protection key and send it to the *US* to pass along to the next car mote.

E. Summery of the protocol

The following figure summarizes the protocol; one car mote and one roadside mote are shown.

AS to *C*	: *R*, *h*ash(R,s), bitmap
C to *M*	: *R*
M computes	: *h*ash(R,s), generate *r*
M to *C*	: *r*
C generates	: R_c
	: TK = hash(*k*,*r*,R_c)
	: MAC=hash(R_c, TK)
C to *M*	: MAC, R_c
M computes	: TK = hash(*k*, *r*,R_c)
	verify MAC, generate r_t
M to *C*	: MAC, r_t
C verifies	: MAC; handshake complete
C to *M*	: bitmap
M to *C*	: *m*
C to *US*	: *m*, *k*
US verifies	: *k*; decrypted *m*
US generates	: bitmap
US to *AS*	: bitmap

F. Security Analysis

The protocol utilizes a four-way handshake, four-key scheme. The four keys are the secret key (known to only roadside motes *M* and *AS*), the session key (generated by *AS* and held by *C*), the protection key (known to only *M* and *US*) and the temporary key (for the one-time use between *C* and a *M*). The handshake protocol allows *C* to authenticate *M* when *M* is required to generate the same session key *C* holds. This is impossible to do without the secret key, so no adversary can forge sensors to provide fake data to the *C*. Meanwhile, the handshake allows *M* to authenticate *C*, since *C* is required to compute a temporary key and to use it to create a MAC for R_c, which is impossible without the correct session key. This would block malicious vehicles from impersonating *C* to obtain crucial information. Moreover, to protect the secret key and protection key, the session key exists so that *C* will not have the most fundamental knowledge of the architecture. Therefore, the protocol provides defense mechanisms against the impoersonation of car mote, malicious car mote, and forged roadside motes.

In the case *C* may drop the data after collecting it from *M*, we used a simple strategy to circumvent this behavior. The upload server notifies the authenticate server what data has been uploaded successfully via a bitmap of successfully decrypted data, and when another vehicle comes to offer assistance in carrying the data, the authentication server passes the already collected data in a bitmap along with the session key keys to the vehicle. When the vehicle moves along the road, it can notify the roadside sensors what data has been collected (the sensor will have to check the validity of the bitmap for the collected data) so that the *M*s can proceed to delete the data from its memory. If the data is not received by the *US*, it will simply need to be re-delivered by later car mote.

V. Experimental Results

To evaluate the proposed protocol, we have implemented a test on two TelosB motes, one of them is used as *M*, while the other *C*. TelosB is powered by the MSP430 microcontroller. MSP430 incorporates an 8MHz, 16-bit RISC CPU, 48K bytes flash memory and 10K RAM. The RF transceiver on TelosB is IEEE 802.15.4/ZigBee compliant, and can have 250kbps data rate. While the hardware directly affects the RSSI and other aspects of the experimental results, TelosB is by no mean the sole platform for the protocol is practical.

A. Metrics and Methodology

In this implementation, we used the following three metrics to better evaluate our results: received/dropped packets, received signal strength indication (RSSI), and the displacement across which the packets are transferred. On an open stretch of road, we drove past the roadside mote *M* at different speeds with the car mote *C* on top of the car and connected to a laptop (via USB), which runs a Java program that reads, records and analyzes the data received. *M* continually sent out radio transmissions in an infinite loop. Upon entering *M*'s range, *C* picked up the encrypted message *hash(r,s)*, decrypted it and responded, and finally obtained the message *m*.

B. Mote to Mote Communication

Three distinct tests were conducted. *C* drove passed *M* at the constant speed of 30, 50, and 70 km/h. Three trials were conducted for each speed. In addition, stationary tests at fixed displacements before and after *M* along the road were conducted at the displacements of 25, 75, 125, 175 and 225 meters. A unique ID number was assigned to every packet received for easier identification. The ID, the RSSI as well as the packet's time of arrive (as an offset from the start time) were recorded for each trial. *C*'s approximate displacement from *M* was also recorded. From this, it was possible to calculate when, where, how many, and at what speed packets were dropped. The location of *C* along the road is expressed in terms of its displacement from *M*, where a negative value X represents that *C* is X meters away from reaching the same point along the road as *M*, a value of 0 represent that it is at the exact same point along the road, and a positive value Y represent that *C* has passed *M* by Y meters.

In order to better analyze the relationship between drop rate of a package and the distance between the two motes, in figure 2, we graphed the total number of packets received/dropped for the three trials at 30km/h. From the graph, it is evident that at the possible range for *C* to receive packets from *M* is around between -150 meters and 50 meters. This represents a window of opportunity of about 200 meters, or about 24 seconds. However, significant number of packets was dropped from -155 meters to about -55 meters, and again resumes to drop significantly at around 25 meters. Therefore, the most reliable window of communication where very few packets (less than 5 percent) is around -55 meter to 25 meters. This 80 meter window represent about 10 seconds.

The data of 50 km/h and 70 km/h show a similar pattern (figures 3 and 4, respectively). However, the optimal window of transmission is halved to just under 6 seconds. This possibility of *C* rush out of *M*'s range is discussed in the limitation/problem section. The possible range of reception, optimal range of reception and approximate time frame to transfer data within the two ranges are summarized in the table below:

Analysis of Possible/Best Packet Transmission Frames

Car Speed	Poss. Range	Optim. Range	Poss. Durat.	Optim. Durat.
km/h	m	m	s	s
30	[-150,50]	[-55,25]	24	10
50	[-150,50]	[-75,5]	14	5.8
70	[-150,50]	[-85,25]	10	5.7

The table is populated with the average of the three trials for each speed. Possible range represents all displacement values where transmission is possible, whereas the optimal range represent the best range for transmission as discussed above. We can use our knowledge of the best window of communication to increase the reliability of our protocol. Possible/optimal duration represents the amount of time in seconds *C* will stay in the respective ranges.

C. RSSI

The signal strength (RSSI) is plotted against displacement in figures 5 to better analyze the relationship between the two. This provides insight to the probability of a successful transmission. For the experiments conducted at 30km/h, three trials show an identical trend with a slight horizontal disparity. The signal strength for the first trial peaked at about -13 meters, second trial at -5 meters, while the third one peaked at -26 meters. It is, however, clear from the graph that RSSI increases as the displacement narrows. The fact all three trials peaked at slightly negative displacement suggests that the radio signals are strongest just before *C* passes *M*. However, there is little difference between about -175 meters and -75 meters in terms of RSSI, suggesting that the strength is not simply inversely proportional to the displacement. This information provides insight to the best timing of the handshake process, for we can use the RSSI as an indicator to show if the communication strength is high enough for the security protocol to start. At 50km/h and 70km/h (figures 6 and 7 respectively), a similar pattern is shown. The results are summarized below:

Approx. Displacement of Best RSSI

Car Speed	Displacement	Time
30 km/h	-10 meters	-1.2 sec
50 km/h	-20 meters	-1.4 sec
70 km/h	-23 meters	-1.2 sec

The trend shows that as speed of the vehicle hosting *C* increases, the ideal displacement from *M* for packet transmission becomes increasingly negative. The ideal point in time for packet transmission, on the other hand, seems to be about just about 1.2 seconds before car mote passes the roadside motes regardless of the speed.

D. Security

The amount of time authentication of this protocol takes is compared with other schemes in order to analyze the efficiency and reliability of the protocol. In particular, experiments show that the amount of time needed to encrypt the message with AES takes less than 1ms with 16 byte-long keys and random numbers. SHA-1 would take 4ms. The proposed protocol can use either one of those two. The speed of encryption/decryption is extremely important in the context. This is because the car mote only has about 6 seconds to communicate with a roadside mote at high speeds, and we must ensure not only that we have enough time for the handshake, but also that the security part takes only a small fraction of the total amount of time we have, and that sufficient amount of time is left for the actual data to pass through.

On the same platform, we found that the ECC-based encryption needs 2 point multiplications, which takes roughly 3.1 seconds on the TelosB. Decryption will take one point multiplication, or about 1.55 seconds. This suggests that an asymmetric key scheme would take much longer to establish the secured connection and is not suitable for a similar set up.

Using the symmetric key protocol as proposed, the total amount of time needed for the motes to encrypt/decrypt is negligible. In addition to the calculation time, however, we also need to consider the amount of time it takes to actually transmit the handshake messages. The protocol requires a 4-way handshake, which requires sending 2 packets from the car mote to the roadside mote, and 2 more packets going the other way. At the experimental rate of 13 packets per second on average, the communication will take roughly 0.31s. Therefore, the total amount of time required for the authentication is about one third of a second, or less than 5 percent of the total amount of time available inside the optimal transmission range. This is very acceptable.

E. Further Observation

At high speed, there would be about 5.6 seconds left for the transferring of interesting data. The TelosB hardware has a maximum transfer rate of 250kbps (as declared on its specifications) and an experimental rate of 200kbps at 70km/h. This means that it will take about 6 cars to unload all 1024K bytes of data the hardware can hold at a time. However, in a realistic setting, we should extract the data long before the onboard flash memory is full. The low percentage of communication time shows the protocol is useful in a realistic setting. In the experiment we did not try to tweak the packet size to maximize the amount of data transferred at a time; ideally, we should be able to reach the rate of 250kbps, and only five car motes need to drive by to collect all data resting on the roadside mote at that rate.

VI. Problems/Limitations

This protocol is simple and easy to understand, but has a number of shortcomings. The most significant problem is the inability to stop repeated attacks aimed at draining the power of roadside motes, which would require a drastic increase of the frequency of maintenance (i.e. battery change).

- While the protocol contains many defenses against the various tactics an adversary may utilize, it has no way of stopping jamming. Therefore, an adversary can block data being transferred to *C* from *M* by simply tie up *M*'s resources via exhausting communication and repeated requests of handshake. Moreover, this strategy can be used over a relative long period of time to drain the battery of *M*, rendering it useless in the future until maintenance addresses the problem.
- How to gracefully drive out of transmission range is not addressed. Therefore, the data transferred need to be small, quick, and atomic.
- A speeding vehicle or a temporary interruption of signal may hinder the reception of the transmission from *M*. Even at reasonable speed such as 50 km/h, the window of opportunity for best transmission is not excessively long, which means any sort of interruption may result in *C* rushing out of the transmission range of *M* prior to a full and secure transmission of data.

VII. Conclusion

In this paper, we show our design of a secure data collection protocol for vehicular sensor networks. We conducted experimental study on this protocol. The experimental data suggest that the proposed protocol can effectively provide a simple solution to vehicular network communication. This architecture based on sensors can fulfill many of the key functionalities envisioned. The protocol can support a large-scaled implementation with many *M*'s and *C*'s. The motes can obtain and transfer various aspects of information interesting to the system (such as traffic patterns) and more sophisticated processing units can process and broadcast the retrieved information.

In the experiment, telosB motes were used and some of the results may be accurate only for the same hardware. Although The exact time and displacement for the best RSSI transmission depend on the hardware, as well as the specific implementation of the protocol and possible external factors beyond a programmer's control (such as the weather), the series of experiments provided in this paper give a rough idea and testimony to the practicality of the protocol.

There are many interesting findings. First of all, at a speed as high as 70 km/h, the motes still have about 5.7 seconds of optimal transmission time, which should suffice under normal circumstances. Another interesting conclusion drawn from the data is that the signal strength is better when *C* is still some distance away from *M* than when the two are at the same point along the road. This is confirmed by the fact the optimal range of transmission is not centered at a displacement of 0 meters, but at around -30 to -15 meters.

We can use this information about the best window of opportunity to improve further implementations. We can, for instance, start the transmission of important data only after the RSSI reaches a certain threshold, (that is, after *C* enters the optimal range of communication with *M*) The data loss rate would then be neglectable, which means the implementation is very reliable. We can lower the threshold if the data to be transferred may require less accuracy or more transmition time, and we can choose to raise the threshold if the data to be transferred is expected to be small but crucial. More experiments under varied and realistic conditions are needed to better estimate the exact RSSI values to use as the thresholds and to reach a perfect balance between reliability and efficiency. It is also very likely that specific applications will change these values to accompany their specific needs.

Acknowledgments

This project was supported in part by US National Science Foundation grants CNS-0721443, CNS-0831904, CAREER Award CNS-0747108, and CSUMS grant DMS 0703532.

References

[1] Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-management in chaotic wireless deployments. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, New York, NY, USA, 2005.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[3] Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddonand, and Hao chi Wong. Secret handshakes from pairing-based key agreements. In *IEEE Symposium on Security and Privacy*, pages 180–196, 2003.

[4] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *MOBICOM*, 2006.

[5] Haowen Chan and A. Perrig. Pike: peer intermediaries for key establishment in sensor networks. In *INFOCOM*, pages 524–535, 2005.

[6] Wenliang Du, Yunghsiang S. Han, Shigang Chen, and Pramod K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, pages 586–597, 2004.

[7] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: vehicular content delivery using Wi-Fi. In *MOBICOM*, 2008.

[8] Anastasios Giannoulis, Marco Fiore, and Edward W. Knightly. Supporting vehicular mobility in urban multi-hop wireless networks. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 54–66, New York, NY, USA, 2008. ACM.

[9] David Hadaller, Srinivasan Keshav, Tim Brecht, and Shubham Agarwal. Vehicular opportunistic communication under the microscope. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 206–219, New York, NY, USA, 2007.

[10] Jun Luo and Jean-Pierre Hubaux. A survey of inter-vehicle communication, epfl. Technical report, 2004.

[11] Jorg Ott and Dirk Kutscher. A disconnection-tolerant transport for drive-thru internet environments. In *Proceedings of IEEE INFOCOM*, pages 1849–1862, 2005.

[12] Maxim Raya, Panos Papadimitratos, and Jean Pierre Hubaux. Securing vehicular communications. In *IEEE Wireless Comm*, 2006.

[13] Harald Vogt. Exploring message authentication in sensor networks. In *Proc. of European Workshop on Security of Ad Hoc and Sensor Networks (ESAS), LNCS*. Springer-Verlag, 2004.

[14] Haodong Wang and Qun Li. Distributed user access control in sensor networks. In *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 305–320, 2006.

[15] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *IEEE Symposium on Security and Privacy*, pages 259–271, 2004.

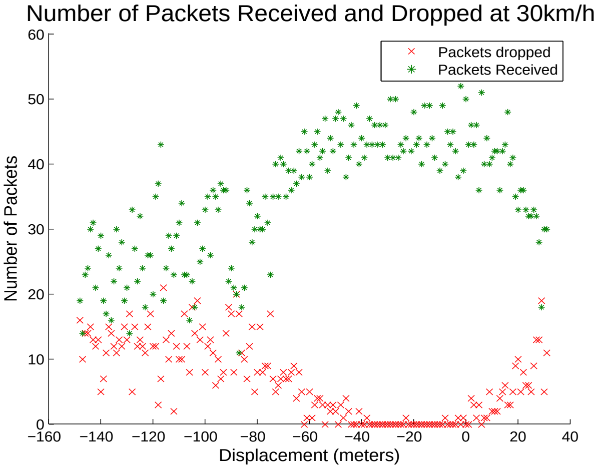

Fig. 2. This graph shows the number of packets received and dropped between the -150m and 50m, and it is the total of three separate trials. It shows that from about -40m to 0m, virtually all packets are successfully received for all three trials. It also shows that close to half of the packets are dropped near the ends of the graph

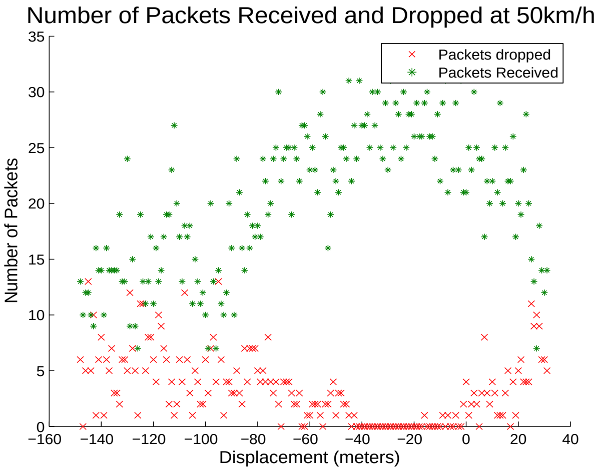

Fig. 3. This graph shows the number of packets received and dropped between the -150m and 50m, and it is the total of three separate trials. It shows that from about -40m to -5m, virtually all packets are successfully received for all three trials. It also shows that close to half of the packets are dropped near the ends of the graph

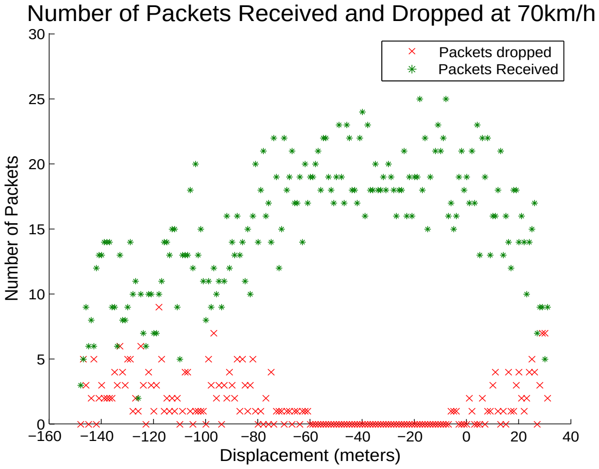

Fig. 4. This graph shows the number of packets received and dropped between the -150m and 50m, and it is the total of three separate trials. It shows that from about -60m to -5m, every single packet is successfully received for all three trials. It also shows that close to half of the packets are dropped near the ends of the graph

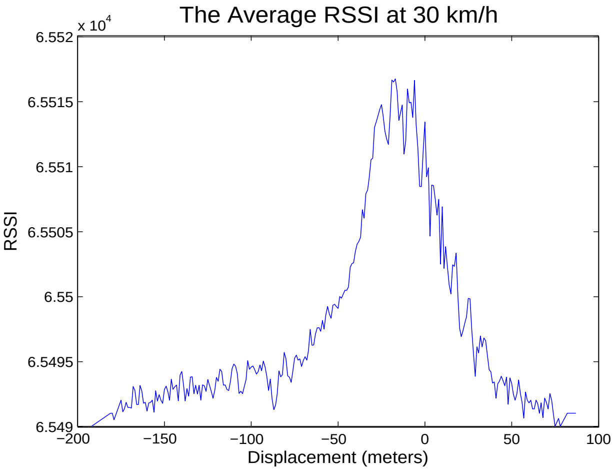

Fig. 5. The RSSI of the packets received within [-200m, 80m]. At about 50m away from the roadside mote the transmission becomes reliable, until it passes the roadside mote by about 25 meters. It peaks shortly before reaching 0m.

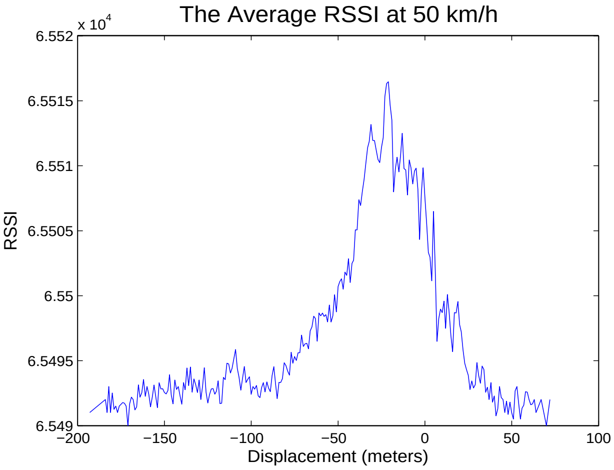

Fig. 6. The RSSI of the packets received within [-200m, 70m]. At about 50m away from the roadside mote the transmission becomes reliable, until it passes the roadside mote by about 25 meters. It peaks shortly before reaching 0m.

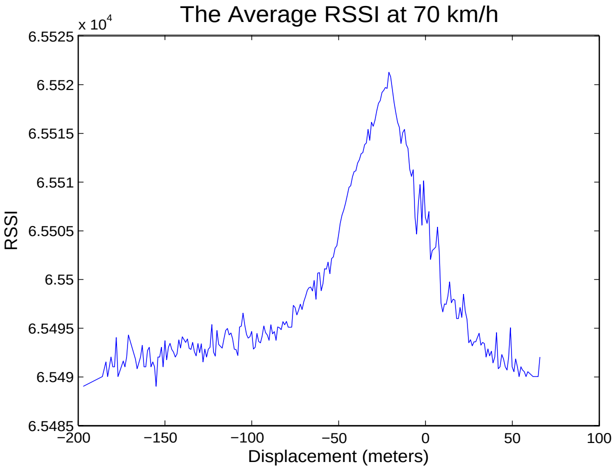

Fig. 7. The RSSI of the packets received within [-200m, 80m]. At about 60m away from the roadside mote the transmission becomes reliable, until it passes the roadside mote by about 15 meters. It peaks shortly before reaching 0m.