Enhanced Management of Certificate Caching and Revocation Lists in VANET

Sadiq H. Abdulhussain
Computer Engineering Department
College of Engineering
University of Baghdad

ABSTRACT
Vehicular network security is an important field and it is agreed that digital signature certificates are becoming the main authentication technique in this environment. The high number of vehicles and their continuous location change bring some difficulties in the exchange of these certificates and in their revocation. This paper covers these two cases and proposes an enhancement to the certificate revocation list (CRL) that is efficient when entire set of certificates belonging to same party are revoked. Then it proposes a solution to the exchange of certificates between vehicles by using the road side units as caching servers. It is shown that cooperation between caching servers enhances the distribution operation.

Keywords
Vehicular Network, Security, Certificates, Certificate Revocation List.

1. INTRODUCTION
Vehicular networks are getting a wide interest in the research community aiming to enhance safety and efficiency of transportation systems. Vehicular ad-hoc network (VANET) is a special case of the mobile ad-hoc network (MANET), with difference in its higher speed of motion. The communication is achieved either Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I). The infrastructure is represented by Road Side Units (RSU) or Base Stations (BS) as shown in figure (1). The research interest and focus is spread among several aspects: the physical communication infrastructure, the Medium Access Control protocols, the network protocols and routing algorithms, the wide range of possible applications, and the privacy and security of information and users. This research depends on the standard IEEE 1609, for wireless access in vehicular environments (WAVE), it includes 1609.1 (resource manager) [1] 1609.2 (security services) [2] 1609.3 (networking services) [3] 1609.4 (lower layers) [4]. It should be mentioned that the lower layers should not affect the other layers.

Certificates, digital signatures, and consequently public key infrastructure (PKI) are becoming a must in this environment since the availability of an online central server is not feasible as stated in many works [5][6].

Certificates are issued by some trusted authorities (may be the car manufacturers or by regional authorities), and these certificates are used by vehicles to authenticate to each other. Thus message authenticity and integrity is achieved by digitally signing the message. This mechanism also provides the non-repudiation property. To accomplish that: "the different CAs will have to be cross-certified so that vehicles from different regions or different manufacturers can authenticate each other. This will require each vehicle to store the public keys of all the CAs whose certificates it may need to verify" [6]. By this manner certificates will be the base for authenticating different parties, and consequently construct the trust relationship between different vehicles and trust between vehicles and the infrastructure.

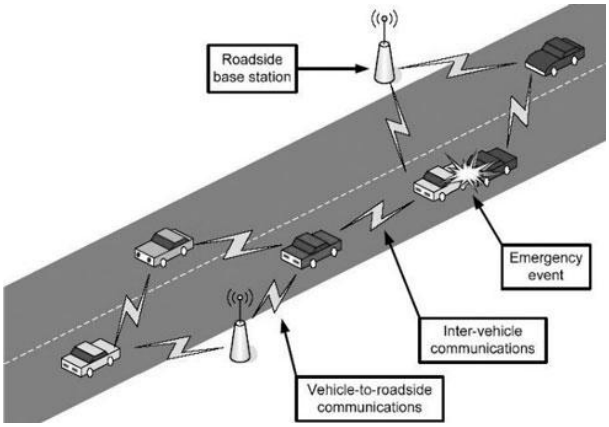


Fig 1: Vehicular Network Scenario

Signed messages can be trivially linked to the certificate of the signing node, and by this way a vehicle can be tracked according to the location of signed messages, and as a result the driver is tracked too. This is because the certificate will have a Global ID (GID) that is related to the vehicle. "Drivers value their privacy and are unlikely to adopt systems that require them to abandon their anonymity" [7]. But it should be mentioned that the license plate of each vehicle is an identifier of the vehicle and the owner, so driver's privacy is not achieved. Balance between the security of the system and driver privacy should be considered [7].

To preserve privacy (provide anonymity) each vehicle is given a set of different pseudonyms from the CA; each pseudonym is attached with a private/public key pair in a certificate. So each vehicle will be supplied with a set of certificates that cannot be related to each other and cannot be related to the GID. The pseudonym and certificate used are changed continuously to prevent tracking. And under special cases the pseudonym can be connected to the GID by the CA only [8].

The requirement of different pseudonyms for each vehicle requires that the vehicle stores these credentials in a secure storage in its On Board Unit (OBU). Another problem that will be faced is the certificate revocation problem. In some situations the certificate and the corresponding pseudonym are revoked before they expire, so a notification must be given to other vehicles to reject any message from this pseudonym. The revocation process has several solutions, but Certificate Revocation List (CRL) is the most accepted one.

This work will address two aspects; the first part is the enhancement of revocation of a complete set of pseudonyms, while the second part proposes an enhancement to the distribution of certificates among vehicles by adopting the cache principle.

2. CERTIFICATE REVOCATION

As mentioned previously each vehicle is given a set of different pseudonyms and a certificate for each pseudonym. This will lead to the fact that each vehicle will possess a large number of certificates in each moment, even that only one of these will be used at any point. These certificates may be revoked entirely, and to realize the revocation of a given vehicle there are two solutions. First solution is to give the Tamper Proof Device (TPD) of the revoked vehicle a command to stop using its credentials [9]. This solution requires identifying the location of the revoked vehicle and ensuring that the TPD of the target receives the message and stops using the cryptographic credentials. The second solution is the widely used one; adding all revoked certificates to CRL. This will present an overhead especially when the number of entirely revoked vehicles increases.

Delta CRL and sliding window delta CRL [10] is an enhancement over the traditional CRL method, but for a node that has not received the base CRL still has to get the entire large CRL.

[9] also proposes the use of Bloom filters to compress the CRL. Bloom filters significantly reduce the size of CRLs, but have the problem of false positives, so there is a very small probability that a certain certificate is found in the list where in fact it is not revoked. But it does not have false negatives, which means that when a certificate is not found in the list means that it is definitely not revoked.

[11] proposes the use of regional CA, and thus using CRLs containing only regional revocation information. Partitioning the CRL creates many lists but with reduced size and any vehicle only has to get the list of the region currently positioned into.

2.1 Proposed Revocation Solution

In order to reduce the size of the CRL in vehicular networks we can make use of the fact that all the certificates of a vehicle are generated by the CA at the same time, but there is no information that relates them so the privacy of the vehicle is preserved. If we can make a shared ID that correlates all the pseudonyms' certificates of a certain vehicle we can use this shared ID in the CRL when the vehicle is entirely revoked. To achieve that, it is needed to state the fact that creating this sharing violates the privacy of vehicle, making the tracking possible, but only movement history not future movement, and the knowledge of the shared ID does not lead to the knowledge of the GID and the owner. But it should be related to the GID only under the approval of the right authorities; by this the pseudonyms will be related to each other but not related to the real identity of the vehicle and driver. This shared ID must only be revealed when all the certificates of the vehicle are revoked so there are no more shared pseudonyms to be used in the future, and the vehicle will not be traceable if the id is revealed because a new set of certificates and pseudonyms will be used that is not related to the revoked one.

The point that must be taken in consideration is that old pseudonyms will be correlated after revocation so the history of vehicle movement will be available but only under some considerations. To be able to retrieve the movement history of a vehicle, first the adversary must have been recording information about a wide set of pseudonyms in a wide period of time and in a wide geographic region, such that when a vehicle is revoked entirely and its pseudonyms are exposed to public, the adversary can find these pseudonyms in the information he was recording. In order to find the complete history of the vehicle movement it is required that the adversary records all the information about all pseudonyms at all the locations all the time. This may be a very difficult assumption if not impossible. Also the probability to find two or more pseudonyms that belong to the same revoked vehicle in the recorded list increases as the recorded list is enlarged. This means more resources (more area to be covered by sensors, more storage area, and more processing power and time).

It is important to distinguish between two cases, a case where the pseudonyms of a vehicle are related and the movement history is revealed, and the case where the pseudonyms are related to the owner which means the owner movement is revealed which is a more important case. When using a shared ID and it is revealed as stated in the proposed solution only the pseudonyms are related but not related to the owner, so the adversary will get the history of a vehicle but does not know which vehicle or who the owner is.

2.2 Storage Cost Estimation

To have an idea of the storage required to monitor the vehicles in a region for some period of time we can assume there is one point to monitor having a coverage radius of R meters as in figure (2).

$$Area = \pi \times r^2 \qquad m^2$$

Assume a car density D (vehicles/m^2)

So number of vehicles in region:

$$N = D \times \pi \times r^2 \qquad vehicles \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

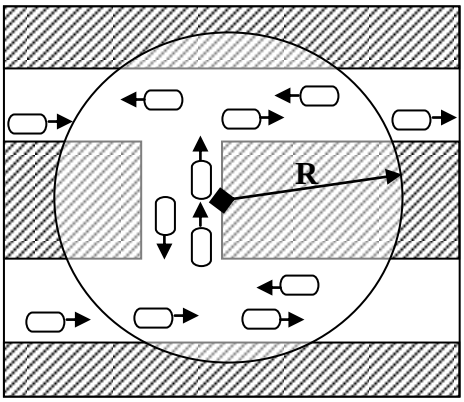


Fig 2: Vehicle Monitoring Scenario

Assuming an average car speed of S (m/sec), and that vehicles on average will stay for a distance of 2×R meters inside the covered region.

Then at average each car stays in the region for a time = 2×R/S sec

So the adversary must log information at (2R/S) intervals to be able to log every vehicle passing through the region. In fact the interval must consider the worst case, i.e. the minimum time any vehicle stays in the region which means it must take the minimum distance and the maximum speed expected.

Assume I is the number of bytes logged for each vehicle, then for N vehicles the size of data logged is N×I bytes.

The number of bytes recorded per second (logging rate LR) = N×I / (2R/S) = N×I×S/2R

And from equation 1:

$$LR = D \times Pi \times R \times I \times S/2 \quad bytes\ per\ second \quad \ldots\ldots\ldots\ldots(2)$$

It is clear that the size of data recorded is linearly proportional to:

- Vehicle density, more vehicles means more information to be recorded.
- Radius of region, as the region is increased the number of vehicle is increased.
- Size of information to be recorded for each vehicle, more details means larger log.
- Speed of vehicles, because if the speed is increased it means the number of vehicles passing per second will be increased too.

Taking for example:

R = 100 m, D = 0.001 vehicle / m^2, I = 100 bytes, S = 50 km/hr = 13.8 m/sec

LR = 216.66 bytes/sec

LR = 17.8 MB per day

This number is for one point of monitoring that covers an area = 3.14×100^2 = 31,400 m^2 = 0.0314 km^2. So to cover an area of 1 km^2 it requires ≈ 32 regions. So the logging rate LR = 32×17.8 = 569.6 MB per day for 1 km^2 of area. This area is a small area to track vehicles in and the period is very small knowing that the pseudonym of vehicles may change once a day only.

2.3 Solution Details

To achieve the goals, a secret (S_i) for each node *i* is generated by the CA which represents the shared ID. Then a hash function is applied to produce the field *Y* to be added to the certificates.

For *N* pseudonyms for vehicle *i*, *N* certificates are required, each having a different *Y* field.

$$Y_i(j) = h(S_i, j) \qquad j = 1\ to\ N$$

Where $h(m)$ is the hash function over the message *m*. The hash function is a one way function such that knowing $Y_i(j)$, which is a public information it is hard to find S_i. In the same manner we can use a symmetric encryption algorithm such as AES, DES, 3DES, and others, so the *Y* field is produced by the following:

$$Y_i(j) = E_{Si}(j) \qquad j = 1\ to\ N$$

Where $E_k(m)$ is the encryption of *m* using the key *K*. To revoke a vehicle *i*, instead of including the certificates' serial numbers in the CRL we include the secret S_i of vehicle *i*, so when the nodes receive the CRL containing S_i they must produce the Y_i fields of all the certificates by using the shared secret S_i.

$$Y_i(j) = h(S_i, j) \qquad j = 1\ to\ N$$

Now each node will check the certificates of vehicles to check for revocation, they compare the serial number of the certificate with the CRL serial numbers, and then they compare the Y field in the certificate with Y fields generated from Si. If there is a match in the serial numbers then that certificate is revoked but the same vehicle may have other certificates that are valid. When there is a match in the Y fields it means that the whole set of certificates of this vehicle is revoked. When there is no match at all then the certificate is valid.

Assuming the function used to produce *Y* fields is the hash, one of the algorithms is the SHA-2 hash algorithm [12]. One of the possible outputs of the SHA-2 hash function is 256 bits, so $Y_i(j)$ is a 256 bit field. The secret S_i can be of any size in the concern of the hash function, but since it represents a key it must have enough length to be hard to break. So we can choose 256 bit as its size. In this way $h(S_i, j)$ will be the hash operation on (S_i *xor* *j*). The maximum number of pseudonyms and certificates for each vehicle identifies the size of *j*. if we take 256 pseudonyms per vehicle then *j* is an 8 bit number.

According to the above assumptions the size of certificates will be increased by 256 bits (the size of $Y_i(j)$). On the other hand the change in size of CRL depends on the number of vehicles that are entirely revoked (all certificates belonging to that vehicle are revoked). The proposed solution reduces the size of CRL for the entirely revoked vehicles part only.

Figures (3) and (4) show simplified structures of the original CRL and new CRL respectively. It also shows the size of each. The figures with sizes are based on the following assumptions:

- N vehicles are completely revoked.
- Each vehicle has 256 certificates.
- M certificates of different vehicles are revoked that do not represent a complete set of any vehicle.
- The serial number (S.N.) of certificates has a size of 10 bytes (80 bits), this size differs according to the standard but it is rarely more than 20 bytes.
- Si size is 256 bits.

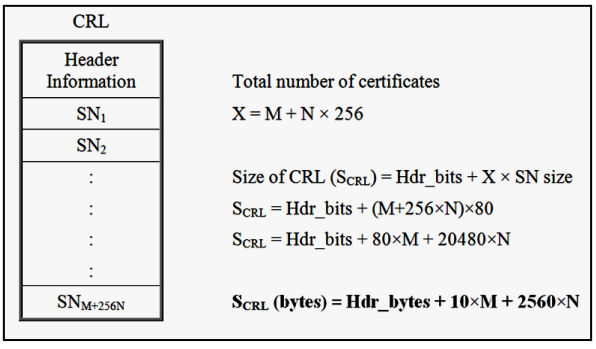


Fig 3: Original CRL and Its Size

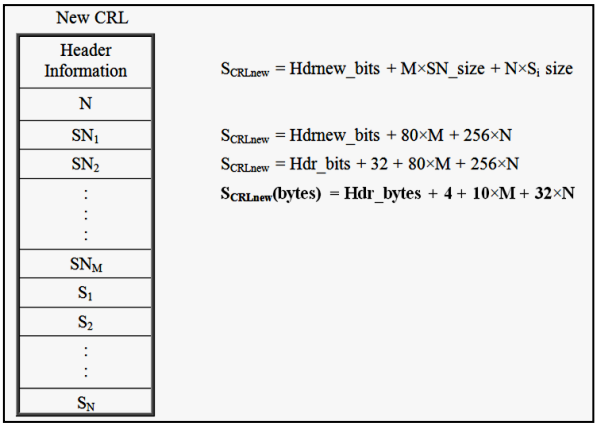


Fig 4: New CRL and Its Size

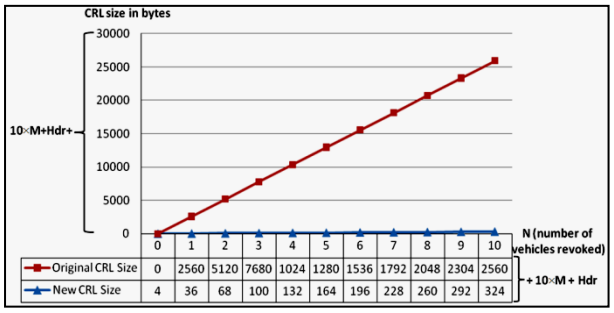


Fig 5: Size of New and Original CRL with respect to N

In general the header will be increased by adding a new field representing the number of revoked vehicles N. So if we assume this field is 32 bit then the CRL can contain up to 2^{32} = 4G revoked vehicles.

By comparing S_{CRL}(bytes) with S_{CRLnew}(bytes) we notice that (Hdr_bytes+10×M) is common in both. So assuming M to be constant both represent a linear equation with respect to N, but S_{CRL} has a very larger slope (2560) than S_{CRLnew} (32), this is shown in figure (5). So in all cases the new CRL will have a smaller size except in one case where there are no vehicles revoked (i.e. N=0). In this case the size of the new CRL will be 32 bits (4 bytes) larger, this is the new field added to the header to represent the number of revoked vehicles.

It is important to state that the proposed solution does not prevent the use of other enhancements to CRL, so it is possible to use the Bloom Filters to compress the CRL. But it should contain two parts, one filter to compress the *M* different certificates, and another filter to compress the *N* revoked vehicles represented by their corresponding S_i. Also it is useful to use the regional CA, so each region will have its own CRL and this list is implemented using our proposed solution. In the same way Delta CRL are possible too.

3. CERTIFICATES DISTRIBUTION

At any given time each vehicle has a certificate issued by a trusted CA, and when a message is to be sent from vehicle B to A the operations shown in figure (6) are carried out to send the message and achieve message authenticity.

If A stores the certificate of B ($Cert_B$) after checking its authenticity then for subsequent messages from B it does not have to do step 1 again. This significantly reduces the computation time to around the half. So if each vehicle caches the used certificates that are verified to be authentic, the performance is improved. This is a general case in every system that uses certificates, and it is suggested in the standard IEEE 1609.2 [2]. Here we want to extend the caching to include the base stations such that they cache the certificates of vehicles in their region and in neighbor regions by cooperation. All this is to reduce the number of verifications carried out by the vehicles since these operations are time and resource consuming.

Signature and verification times vary according to different parameters such as the algorithm used, the key size, available resources and platform, and implementation but it is possible to compare different options at least relatively to each other. We are interested in two candidates: RSA and ECDSA. Table 1 taken from [13] shows performance as tested on Pentium III 500 MHz, table 2 taken from [14] shows the results on StrongARM CPU 206 MHz. Table 3 shows our results tested on Pentium 4 (3 GHz) using the Microsoft cryptographic application programming interface (CryptoAPI) and the widely used Crypto++ library [15]. A program was developed to test the algorithms on a sample data using the indicated libraries several times and the timings averages were listed in table 3.

It is well known that Elliptic Curve Cryptography (ECC) is a new competent to RSA. It has smaller key size but achieving similar security to RSA, 160 bit in ECC is equivalent to 1024 bit RSA. Smaller keys and signatures mean low communication cost. Also it has faster signature generation than RSA but RSA outperforms in verification operations.

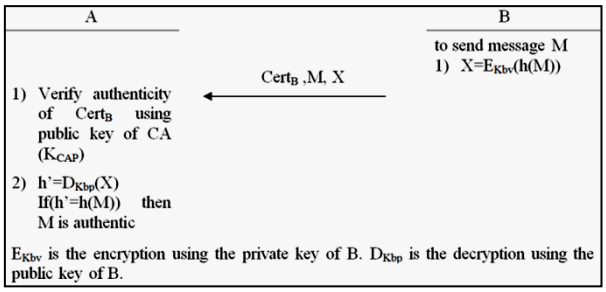


Fig 6: Message Authentication Using Certificates

Table 1. TTS and NESSIE round 2 candidates signature schemes on a 500MHz Pentium III [13]

Scheme	Signature (bits)	Pub. Key (Byte)	Priv. Key (Byte)	Key Setup (sec)	Signing (Sec)	Verifying (Sec)
RSA-PSS	1024	128	320	2.7	84 m	2.0 m
ECDSA	326	48	24	1.6 m	1.9 m	5.1 m
ESIGN	1152	145	96	0.21	1.2 m	0.74 m
QUARTZ	128	7100 0	3900	3.1	11	0.24 m
SFLASH$^{v}_{2}$	259	1540 0	2400	1.5	2.8 m	0.39 m
TTS/2	224	8600	1300	5.3 m	35 μ	0.13 m
TTS/4	224	8600	1300	5.3 m	36 μ	0.13 m

Table 2 Time for signature operations with different signature schemes on a StrongARM CPU @206 MHZ [14]

Year	Level of Security (key size[bit])		Time for Signature Generation [ms]		Time for Signature Verification [ms]	
	ECDSA	RSA	ECDSA	RSA	ECDSA	RSA
1999	113	512	2.8	13.7	7.5	1.3
2006	131	704	3.8	32.4	11.5	2.5
2015	163	1024	5.7	78.0	17.9	4.3
2026	193	1536	7.6	251.9	25.0	9.7
2039	233	2240	10.1	731.8	37.3	20.4

Table 3 RSA and ECC timings on Pentium 4 (3 GHz)

Algorithm	Operation	Time (msec)
RSA 1024 using CryptoAPI	Signature	3.6
	Verification	0.175
ECC 160 using Crypto++	Encryption	17.2
	Decryption	11
RSA 1024 using Crypto++ with e = 17	Signature	3.9
	Verification	0.25

3.1 Proposed Certificate Caching

In vehicular networks the movement is significantly more than other environments which means that the communicating parties change frequently. So the internal cached certificates may be used for few times only and the internal cache suffers from one miss at first time communication.

To improve the effectiveness of certificate cache, it is possible to make use of the BSs to act as certificate cache servers. So each BS will contain a list of certificates belonging to vehicles that are in the near region. The list may be constructed by the contribution of BSs in same neighborhood.

There are two possible ways to send the list to vehicles passing by: either the list is sent individually upon the request of the vehicle or it can be broadcasted to all vehicles in transmission range periodically. The broadcast option is preferred because it requires less communication overhead on the BS. The list may be sent individually on request when the vehicle density is very low, in which the time interval between two vehicles passing through the BS transmission range is more than the time interval between two broadcasts.

3.2 List Integrity

In case the certificates are cached inside the vehicle itself after being verified in the first time, then there is no need to verify the signature of the CA on the certificates in subsequent uses (assuming the cache is secure), this represents the performance enhancement. When the BS sends the list of cached certificates, the vehicle must have a way to verify the authenticity of the certificates. So if the vehicle verifies the CA's signature on each certificate then there will be no usefulness of the list, since the number of verifications will be the same as when the vehicle gets the certificates from their owners. Instead, the BS digitally signs the whole list after verifying the included certificates, and since the BS is a trusted party then all certificates in the list could be verified by verifying the BS' signature on the list, this is shown in figure (7). By this way the vehicle will make one verification to verify the authenticity of all certificates in the list.

Assuming two vehicles A and B, when A receives a message from B whose certificate is in the list (consequently in the internal cache), then figure (6) will be applied except step 1 for A. But if the certificate is not found in the cache then all steps are carried out.

The performance enhancement on the vehicles side depends on the hit ratio that the list will achieve. This hit ratio is the number of certificates to be verified that are found in the list to the total number of certificates to be verified. The hit ratio can be increased by increasing the size of the list, and it is also affected by the protocol or policy used to construct the list. But larger list means higher communication on both the vehicles and BS.

3.3 List Update

The list must be updated periodically such that new certificates are added. Also old certificates must be removed from the list, because the basic idea of the list is to contain the most recent certificates that have high probability to be used by vehicles. Another reason to remove old certificates is to keep the size of the list within some acceptable range.

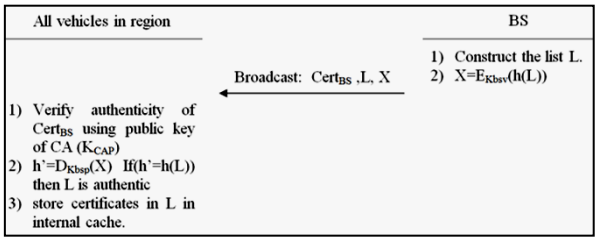


Fig 7: Distribution of Verified Certificates' List

As new vehicles pass, the BS records their certificates after verifying them. Then at some fixed intervals the BS adds the new certificates to the list and removes the old ones. Assuming the list keeps certificates for some time window *W* seconds (lifetime of certificate in the list), and that the update period is every *T* seconds. Then each update will add the new certificates in period *T* and remove the certificates that their lifetime in list has ended (greater than *W*). Since the removal of certificates from the list is based on their lifetime in the BS cache, the BS must store the certificates along with the time they were recorded.

Each update is a reconstruction of the list which means a digital signature is recomputed for the new list. So the BS will have to do a digital signature every T seconds, then the load of signature is increased as T is decreased. Also it must verify each new vehicle's certificate before being added, this load is increased as the number of new vehicles passing is increased.

If we take 17.9 msecs as a time for verification, then each BS can make about 55 verifications per second. This means that the BS will be busy all the time making verifications only and neglecting its other duties. So the verification of vehicle's certificates may be a bottleneck which the next section addresses.

In addition to communication between vehicles, any vehicle may send or receive messages from servers. These servers have fixed locations so their certificates may be included in the lists of BSs that are in neighborhood, but they are not removed after T seconds from the list.

The list can also include the certificates of nearby BSs, thus vehicles will not have to verify the certificates of the next BS, and so it directly verifies the next list. The addition of servers and BS certificates reduces the load on the client vehicles.

3.4 Cooperation between Base Stations

It is possible to reduce the load on the BSs, produced by the continuous verification of vehicles' certificates, by cooperation between the BSs. The basic idea is that instead of vehicle's certificate being verified by each BS the vehicle passes through; one BS verifies the certificate and notifies other BSs of its authenticity.

According to figure (8) we can see that vehicles (V1,V3,V5,V7) will be new vehicles for B33, which means that B33 will have to get their certificates and verify them before being added to its list. But since each vehicle of them have been previously verified by another BS (V1 was verified by B34, V3 by B43, V5 by B32, and V7 by B23) then to reduce the load on B33, the surrounding BSs send the certificates of corresponding vehicles encrypted symmetrically. This requires that BSs communicate with each other using symmetric encryption which requires shared keys between each two.

In fact B23 may have a set of vehicles, other than V7, going towards B33, then it should send a list of certificates encrypted to B33. In the same manner, B43, B34, and B32 will send to B33 lists of vehicles' certificates in their region which are going towards B33. On the other hand, B33 will send lists of vehicles' certificates to B34, B43, B32, and B23. Figure (9) shows the messages sent from B33 to B23 for cooperation. In this way when a vehicle's certificate is verified by any BS, it is not needed to be verified by other BSs again, since the BS that made the verification will notify the surrounding BSs using symmetric encryption and so on. This obviously reduces the number of verification processes resulting in lower load.

The taken scenario assumes that there is an equal distribution of vehicles among the BS regions. Also the probability of movement is 25% in each direction, this means that 25% of vehicles in B33 will go to B23, 25% to B43, 25% to B32, and 25% to B34. The equations in figure (10) assume that each BS can verify 10% of the available vehicles in its region every T period, and the 10% are distributed in the four directions as 2.5% per direction. It is obvious that B11 will cooperate with B12 and B21 which means that it will benefit from two directions, while B33 will be cooperating with four BSs.

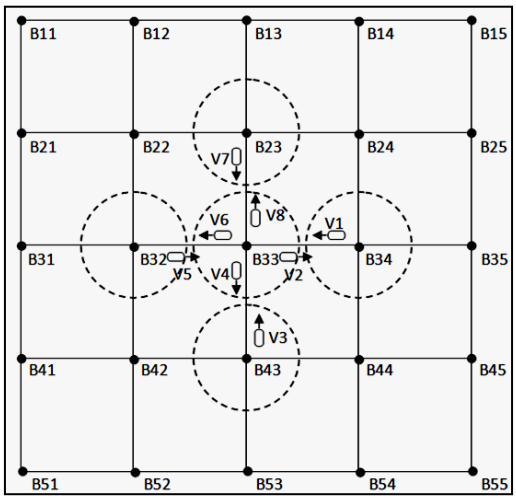


Fig 8: Vehicles Movement With Respect to BSs

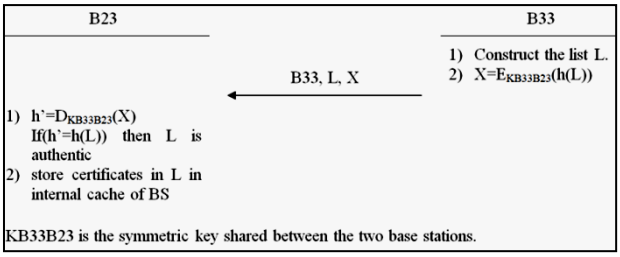


Fig 9: Exchanged Messages for Cooperation between BSs

Figures (11)(12)(13)(14) and (15) show how the number of vehicles' certificates that have already been verified is increased periodically. From the numbers we see that B33 is the most beneficial of the cooperation. So after 20 T B33 approaches 100% verified certificates, so it will not have to verify certificates since it will get all the required certificates from nearby BSs.

B11=10+0.25×B21+0.25×B12
B21=10+0.25×B31+0.25×B22+0.25×B11
B31=10+0.25×B41+0.25×B32+0.25×B21
B41=10+0.25×B51+0.25×B42+0.25×B31
B51=10+0.25×B52+0.25×B41

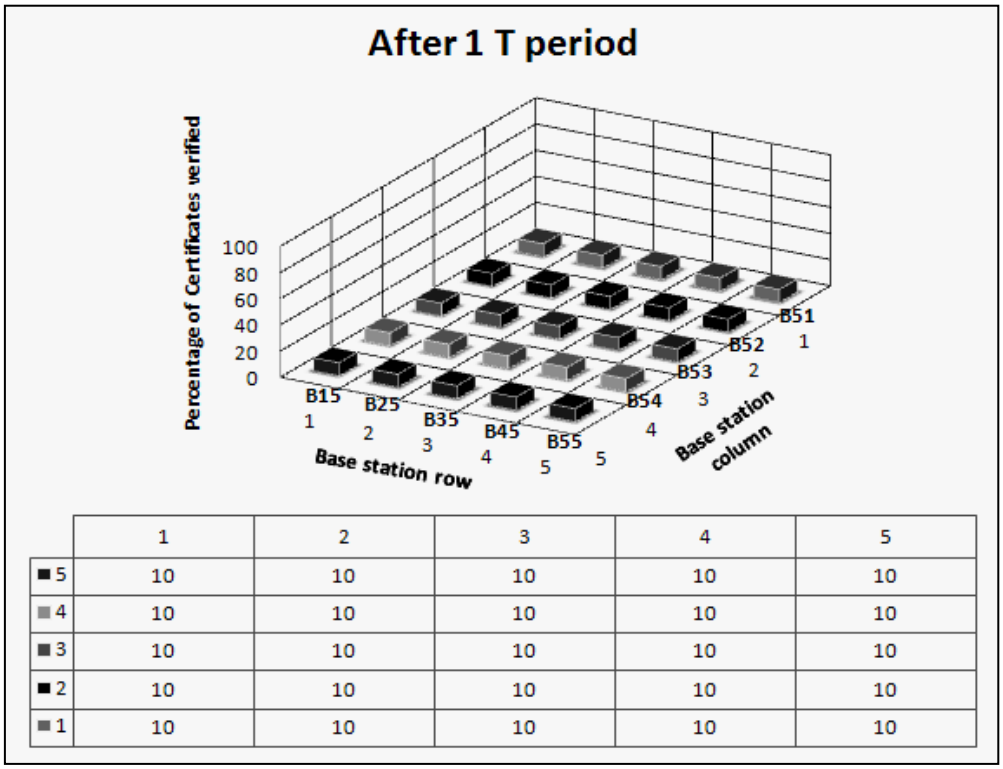
B12=10+0.25×B11+0.25×B22+0.25×B13
B22=10+0.25×B21+0.25×B32+0.25×B23+0.25×B12
B32=10+0.25×B31+0.25×B42+0.25×B33+0.25×B22
B42=10+0.25×B41+0.25×B52+0.25×B43+0.25×B32
B52=10+0.25×B51+0.25×B53+0.25×B42

B13=10+0.25×B12+0.25×B23+0.25×B14
B23=10+0.25×B22+0.25×B33+0.25×B24+0.25×B13
B33=10+0.25×B32+0.25×B43+0.25×B34+0.25×B23
B43=10+0.25×B42+0.25×B53+0.25×B44+0.25×B33
B53=10+0.25×B52+0.25×B54+0.25×B43

B14=10+0.25×B13+0.25×B24+0.25×B15
B24=10+0.25×B23+0.25×B34+0.25×B25+0.25×B14
B34=10+0.25×B33+0.25×B44+0.25×B35+0.25×B24
B44=10+0.25×B43+0.25×B54+0.25×B45+0.25×B34
B54=10+0.25×B53+0.25×B55+0.25×B44

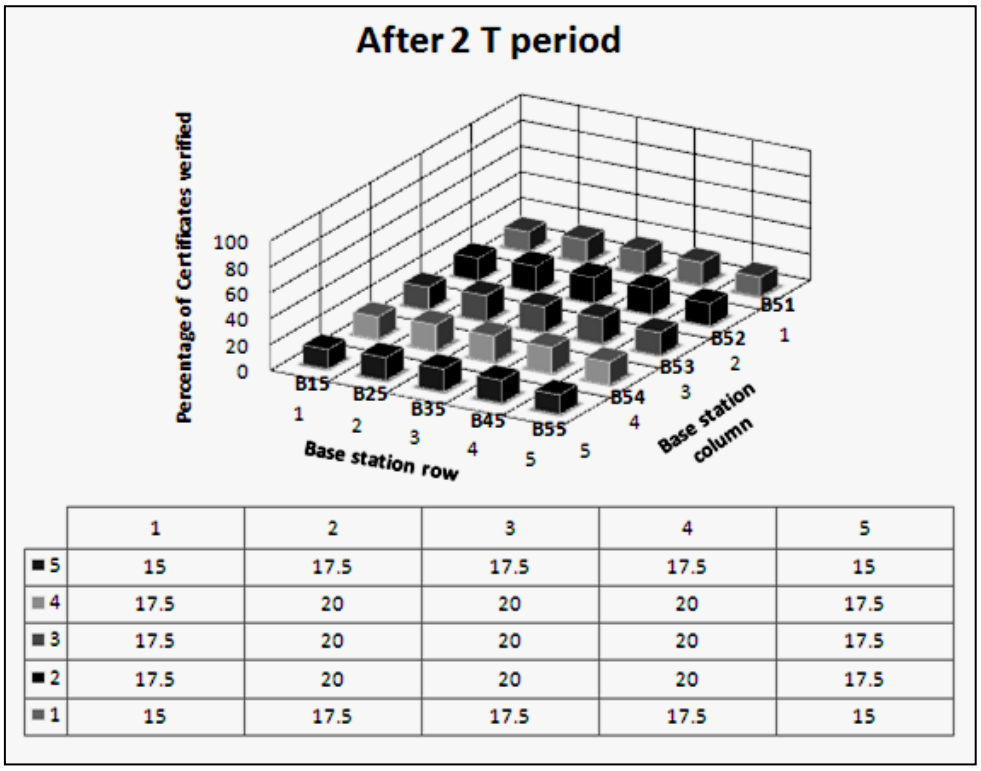
B15=10+0.25×B14+0.25×B24
B25=10+0.25×B24+0.25×B34+0.25×B15
B35=10+0.25×B34+0.25×B44+0.25×B25
B45=10+0.25×B44+0.25×B54+0.25×B35
B55=10+0.25×B54+0.25×B45

Fig 10: Equations of verified certificates progress



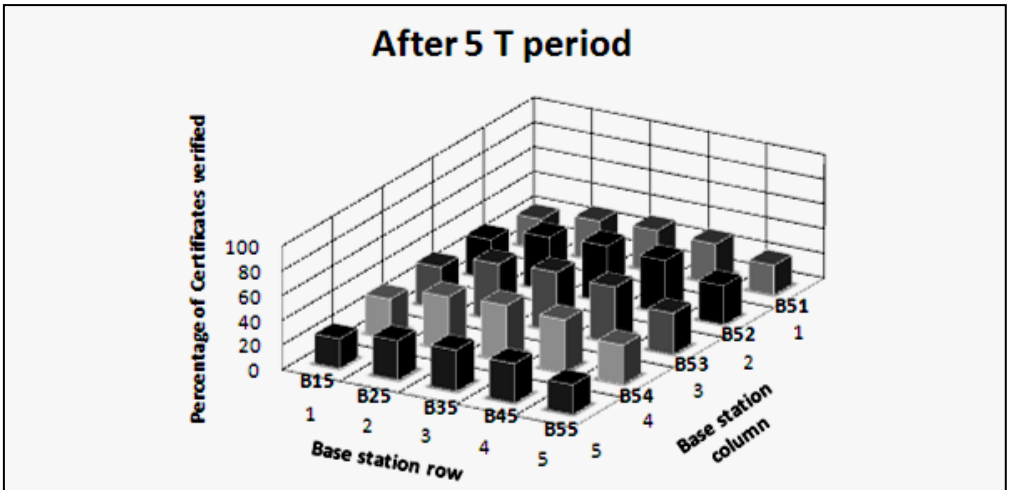
	1	2	3	4	5
5	10	10	10	10	10
4	10	10	10	10	10
3	10	10	10	10	10
2	10	10	10	10	10
1	10	10	10	10	10

Fig 11: Percentages of Certificates that are Verified for Each BS after 1 Updates



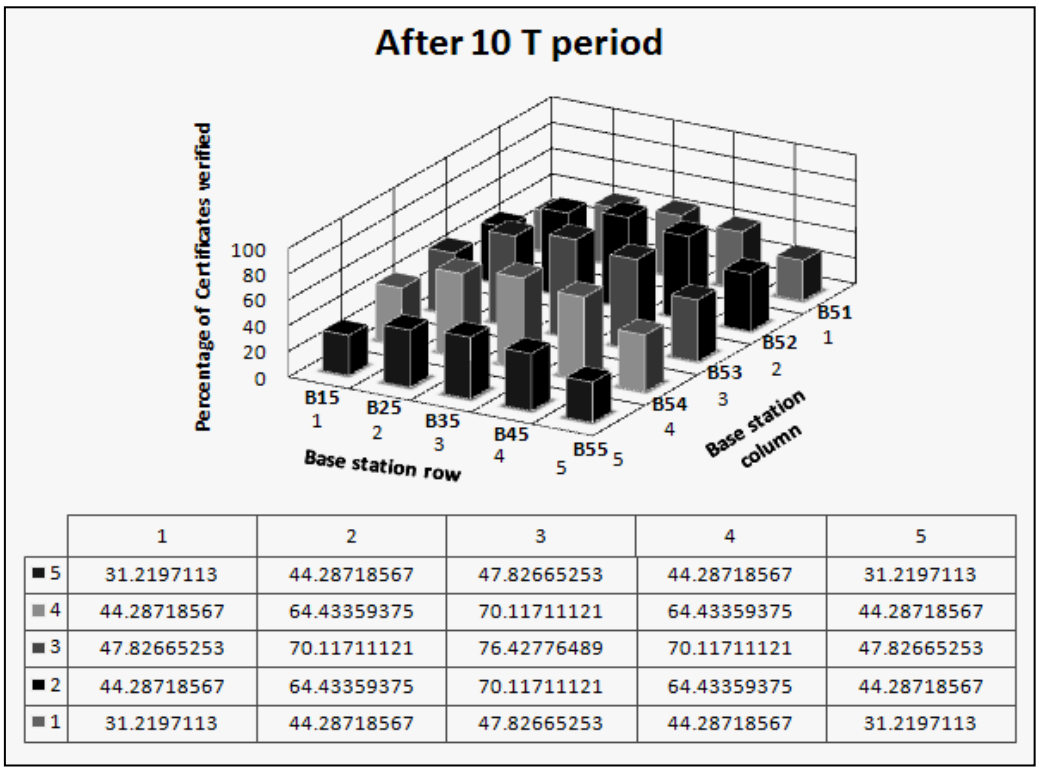
	1	2	3	4	5
5	15	17.5	17.5	17.5	15
4	17.5	20	20	20	17.5
3	17.5	20	20	20	17.5
2	17.5	20	20	20	17.5
1	15	17.5	17.5	17.5	15

Fig 12: Percentages of Certificates that are verified for Each BS after 2 Updates



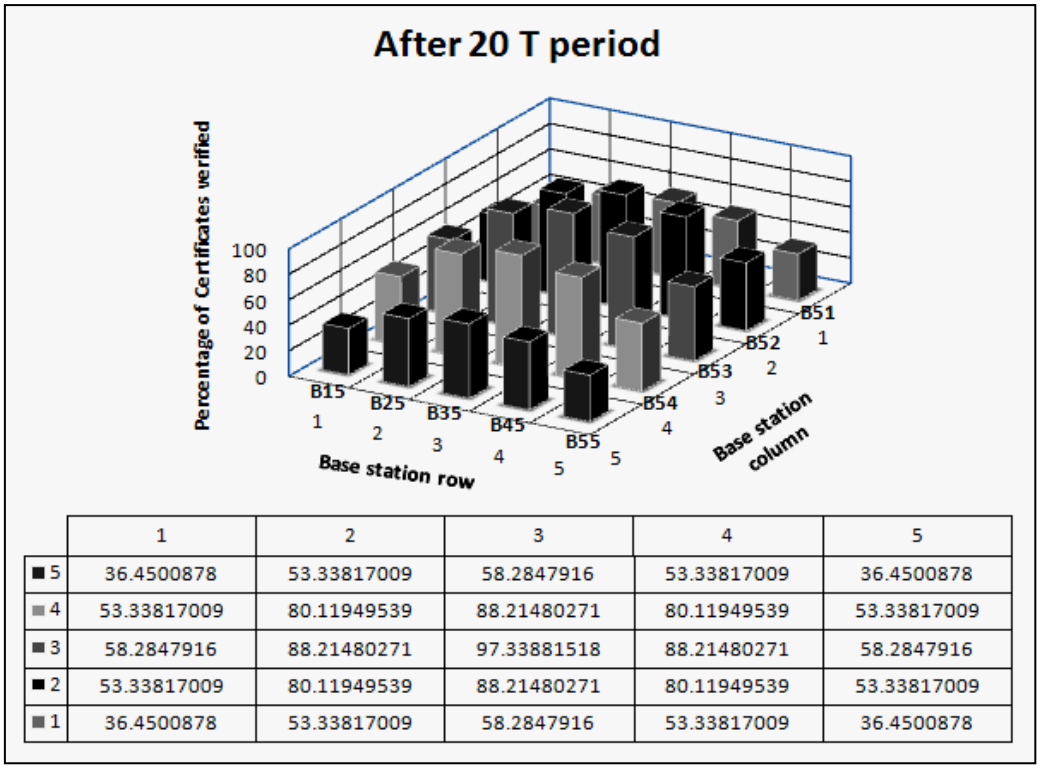
	1	2	3	4	5
5	23.90625	31.6796875	33.359375	31.6796875	23.90625
4	31.6796875	42.8125	45.1953125	42.8125	31.6796875
3	33.359375	45.1953125	47.8125	45.1953125	33.359375
2	31.6796875	42.8125	45.1953125	42.8125	31.6796875
1	23.90625	31.6796875	33.359375	31.6796875	23.90625

Fig 13: Percentages of Certificates that are verified for Each BS after 5 Updates



	1	2	3	4	5
5	31.2197113	44.28718567	47.82665253	44.28718567	31.2197113
4	44.28718567	64.43359375	70.11711121	64.43359375	44.28718567
3	47.82665253	70.11711121	76.42776489	70.11711121	47.82665253
2	44.28718567	64.43359375	70.11711121	64.43359375	44.28718567
1	31.2197113	44.28718567	47.82665253	44.28718567	31.2197113

Fig 14: Percentages of Certificates that are verified for Each BS after 10 Updates



	1	2	3	4	5
5	36.4500878	53.33817009	58.2847916	53.33817009	36.4500878
4	53.33817009	80.11949539	88.21480271	80.11949539	53.33817009
3	58.2847916	88.21480271	97.33881518	88.21480271	58.2847916
2	53.33817009	80.11949539	88.21480271	80.11949539	53.33817009
1	36.4500878	53.33817009	58.2847916	53.33817009	36.4500878

Fig 15: Percentages of Certificates That are Verified for Each BS After 20 Updates

The above results are affected by the mobility of vehicles; faster change of locations may degrade the percentages. Also the density of vehicles is a great factor; more density means that there are more certificates to be verified by BSs. We assumed that each BS can verify 10% of the certificates of vehicles in its region every T period, so when the number of vehicles increases then the percentage decreases and the results decrease too (or the progress is slower). Also when the density increases the BSs will have to serve more vehicles so the time left for verifications is less which decreases the results more.

4. CONCLUSIONS AND FUTURE WORK

Certificate revocation overhead can be substantially reduced for the cases where all certificates of a certain vehicle are revoked by making the certificates related to each other by a secret. This can be useful if the revealing of certificates correlation after vehicle revocation is not considered a problem. In fact the problem arises when the certificates are related to their owner which is not the case here.

On the other hand the distribution of certificates between vehicles can be enhanced by making the base stations act as a trusted certificates cache servers. This role for the base stations adds a high load on them which can be reduced by making them cooperate with each other.

The caching and cooperation have been checked for the case of equal probability of movement in each direction, the results change according to the movement pattern and distribution, this will be tested in future works since it requires building a vehicle movement simulator to get the results. Also more details of the cooperation protocol must be developed and tested.

The same factors above, mobility and density, increase the importance of the caching since more vehicles means more verifications between vehicles. So the one verification of the list constructed by BSs can replace the many verifications that may be done by vehicles without caching.

5. REFERENCES

[1] Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society, “IEEE Std 1609.2-2006, IEEE standard for wireless access in vehicular environments (WAVE), resource manager”, 2006.

[2] Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society, “IEEE Std 1609.2-2006, IEEE standard for wireless access in vehicular environments (WAVE), security services for applications and management messages”, 2006.

[3] Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society, “IEEE Std 1609.3-2007, IEEE standard for wireless access in vehicular environments (WAVE), networking services”, 2007.

[4] Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society, “IEEE Std 1609.4-2006, IEEE standard for wireless access in vehicular environments (WAVE), multi-channel operation”, 2006.

[5] Raya M. and Hubaux J.P. “The Security of Vehicular Ad Hoc Networks”, In Proceedings of SASN 2005, November 2005.

[6] Raya M., Papadimitratos P., and Hubaux J.P. “Securing Vehicular Communications”, In IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, October 2006

[7] Parno B. and Perrig A., "Challenges in Securing Vehicular Networks", Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV), November 14-15, 2005.

[8] Armknecht F., Festag A., Westhoff D., Zeng K., "Cross-layer Privacy Enhancement and Non-repudiation in Vehicular Communication", 4th Workshop on Mobile Ad-Hoc Networks (WMAN), Bern, Switzerland, March 2007.

[9] Raya M., Jungels D., Papadimitratos P., Aad I. and Hubaux J.P. "Certificate Revocation in Vehicular Networks", Laboratory for Computer Communications and Applications (LCA), EPFL, Tech. Rep. LCA-REPORT, 2006.

[10] David A. Cooper, "A More Efficient Use of Delta-CRLs", 2000 IEEE Symposium of Security and Privacy, pp 190-202, 2000.

[11] Mezzour G., "Credentials Revocation in Vehicular Networks: Design and Evaluation", Report, Semester project and Miniproject IC-71 Security and Cooperation in Wireless Networks, Fall 2007.

[12] NIST, "FIPS Publication 180-2: Secure Hash Standard", August 1, 2002.

[13] Chen J. and Yang B. "A more secure and efficacious TTS signature scheme", ICISC, Lecture Notes in Computer Science, Volume 2971, pp: 320 – 338, 2004.

[14] Rao W. and Gan Q. "The Performance Analysis of Two Digital Signature Schemes Based on Secure Charging Protocol", Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, Volume: 2, pages: 1180-1182, Sept. 2005.

[15] Wei Dai. "Crypto++TM Library 5.1 – a Free C++ Class Library of Cryptographic Schemes", Available at: http://www.eskimo.com/~weidai/cryptlib.html

ABBREVIATIONS

AES	Advanced Encryption Standard
BS	Base Station
CA	Certificate Authority
CRL	Certificate Revocation List
DES	Data Encryption Standard
ECDSA	Elliptic Curve Digital Signature Algorithm
GID	Global Identification
MANET	Mobile Ad-hoc Network
MD5	Message Digest algorithm 5
OBU	On-Board Unit
PKI	Public Key Infrastructure
RSA	Rivest, Shamir, and Adleman algorithm
RSU	Road Side Unit
SHA	Secure Hash Algorithm
TPD	Tamper Proof Device
VANET	Vehicular Ad-hoc Network