Performance Analysis of Cryptography Methods for Secure Message Exchanging in VANET

M. Alimohammadi, and A. A. Pouyan

Abstract—Vehicular Ad-hoc Networks (VANET) are becoming more popular as the accident statistics increase. VANET as a comprehensive system provides many of safety applications to save people lives, eliminate accidents and damage to the vehicles and people and also prevent wasting time due to busy traffic and time consuming services for drivers. From one side, safety applications due to their high sensitivity should be resistant against various attacks and on the other side, privacy of drivers including location and identifier information should be preserved in the network. So for preventing many of attacks and also preserving privacy of drivers, many protocols need an infrastructure for key distribution, revocation and secure exchange of the messages containing private information. Protocols for secure communications in this infrastructure use some sufficient cryptography based methods. Some security mechanisms used for encrypting and authenticating V2V and V2I messages comes with overhead in terms of computation and communications. Therefore, for feasibility and better performance of cryptographic based protocols, we should investigate operation of different cryptography based methods. Selected methods due to high mobility nodes in the network or high speed vehicles, should have a very little processing time, have a small key length, do not increase the length of created message as much as possible and have an acceptable level of safety over the key lifetime. This paper provides a comprehensive comparative analysis between most common symmetric and asymmetric key cryptography algorithms on the basis of the parameters: speed; block size and key size. Simulation programs are implemented using Openssl library and c programming on Ubuntu 12.04.

Index Terms— Cryptography based protocols, Privacy, Public key methods, Secure message exchanging, Symmetric key encryption, VANET.

1 INTRODUCTION

High mobility nodes, fast topology changing and predictable vehicle movements are make VANET different from other types of ad hoc networks. In vehicular network, communications are vehicle to vehicle (V2V communications) and vehicle to infrastructure (V2I communications). All of communications are through two devices called Roadside Unit (RSU) as an infrastructure installed at the road side and onboard unit (OBU) installed on each vehicle. Various wireless standards are being developed for communicating in VANET, such as Dedicated Short Range Communication (DSRC) standard [1]. VANET provide many applications varying from safe driving to driver assistance and Internet access. Many applications need to cryptosystems for preventing attacks and exchanging some private messages consist of location and vehicle identifier information (e.g. plate number). In this network, attackers are divided into two categories: insider attackers that are authentic members of VANET communicating with other vehicles, and outsider attackers that are not recognized by other members. Many of attacks performed by either insider or outsider attackers can be prevented or detected by cryptography-based methods [2-7]. Some of attacks are:

- Message eavesdropping: it is for extracting private information belonging to another vehicle and then vehicle tracking or impersonating. This attack violates privacy of drivers. Encryption of important messages containing identifier and location information of vehicles can resolve this problem.
- Message manipulation: it is for creating traffic jam or accident. Broadcasting public key of vehicles issued by a trusted party (Certificate Authority) and then signing exchanging messages eliminate this attack.
- Wrong message injection: it is for changing result of voting applications. For outsider attackers, vehicle authentication is a normal defense mechanism and for insider attackers, authentication mechanisms can be helpful with other mechanisms for attack detection. With vehicles authentication, each vehicle is not capable to send dummy messages more than once and this attack can affect on the network performance only if malicious vehicles cooperate (collaborating attack). After detection, Public Key Infrastructure (PKI) helps RSUs to track and revoke attacker (malicious vehicle).
- Sybil attack: attacker tries to forge some identities belonging to real ones in the network or bogus identities made by the attacker. It is for decreasing network performance, traffic safety violation, routing disruption and for other benefits of attacker. There are some of mechanisms for Sybil attack detection that cryptography based methods for the reason of high detection rate, low overhead for increasing number of exchanging messages in the network, providing secure message exchanging and acceptable time for attack detection is a suitable mechanism.
- Dropping legitimate packets: in order to not sending warning messages to the vehicles that cause to accident for vehicles approaching toward the accident place or traffic jamming in traffic management application. It is hard but different mechanisms can be used for detecting some attackers. (PKI) helps RSUs to track and revoke attacker after detecting (malicious vehicle).
- Replaying packets reporting an event after that event has been expired to create malfunctions for other vehi-

cles. Messages will expire after a limited period of time included in the message. Message signature helps to verify message lifetime.

All of the cryptography based methods can be done with efficient cryptography methods. For securing messages we can use symmetric or asymmetric key based methods. But symmetric key based methods are applicable for longer messages and asymmetric/public key based methods are applicable for short messages such as message digest. In this paper we investigate many of common methods for symmetric and asymmetric key based methods to select more efficient method that consume a little time (processing time affects on network scalability), has small key length, do not increase the length of created message (longer message consume more bandwidth) and has an acceptable level of safety.

2 Encryption Methods

All of encryption methods are include the: 1. symmetric key encryption and 2. asymmetric or public key encryption methods. Both two methods have advantages and disadvantages which are outlined below [8,9].

2.1 Symmetric Key Encryption

In symmetric key encryption, that is also known as secret-key or private-key encryption, there is only one key that both sender and receiver share it to encrypt and decrypt messages. This key should be shared between sender and receiver before secure message exchanging. Some of methods are include AES (Advanced Encryption Standard) and DES (Data Encryption Standard).

Advantages of Symmetric key encryption methods are:

- They are simple**:** This type of encryption is easy and all things have to do is specify and share the secret key and then encrypt and decrypt messages in a little time.
- Fast: Symmetric key encryption is much faster than asymmetric key encryption.
- They use less computer resources: they do not require a lot of computer resources in comparison with public key encryption.
- They have short key size.

Disadvantages of Symmetric key encryption methods are:

- Needing to secure channel for exchanging secret key.
- Too many keys: A new key is necessary for communication with every different party. Management and ensuring the security of the symmetric keys becomes problematic. Management of the keys is difficult as numbers of trading partners increases, especially when multiple keys exist for each trading partner.
- Message authenticity cannot be proved: Both sender and receiver use the same key, so it is not possible to verify the message have come from a particular user.

2.2 Symmetric Ke Encryption Algorithms

SEED: SEED is a 128-bit symmetric key block cipher developed by the Korea Information Security Agency (KISA) and described in RFC 4269 [10]. It used popularity in Korea because 40-bit encryption was not considered strong enough, so the Korea Information Security Agency developed its own standard.

The features of SEED are outlined as follows:

- The Feistel structure with 16-round
- 128-bit input/output data block size
- 128-bit key length
- A round function that is strong against known attacks
- Two 8x8 S-boxes
- Mixed operations of XOR and modular addition

Camellia: Camellia was jointly developed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000 [11,12]. It possesses the security level and processing capability equivalent to AES. Camellia is characterized by its suitability for both software and hardware implementations on common 32-bit processors as well as 8-bit processors (e.g., smart cards, cryptographic hardware, and embedded systems). Camellia's application in IPsec is described in RFC 4312 and application of OpenPGP in RFC 5581.It has high level of security. The design goals of Camellia are: High level of security and efficiency on multiple platforms. The features of Camellia are outlined as follows:

- The Feistel structure with either 18 rounds (when using 128-bit keys) or 24 rounds (when using 192 or 256-bit keys)
- 128-bit input/output data block size
- 128, 192, and 256-bit key sizes
- Using 8x8 S-boxes.

CAST-128: The algorithm was created in 1996 by Carlisle Adams and Stafford Tavares using the CAST design procedure. CAST-128 (described in RFC-2144 document [13]) is a popular 64-bit block cipher allowing key sizes up to 128 bits. The name CAST stands for Carlisle Adams and Stafford Tavares, the inventors of CAST. It is used in some applications as the default cipher in some versions of GPG and PGP. It has also been approved for Canadian government use by the Communications Security Establishment. One of the positive characteristics in this method is inmunity against differential and linear cryptanalysis attacks; standard cipher algorithm on last versions of PGP [17]. The features of CAST-128 are:

- The Feistel structure with either 12 or 16 round.
- 64-bit input/output data block size
- A key size between 40 to 128 bits (but only in 8-bit increments). The full 16 rounds are used when the key size is longer than 80 bits.
- Using large 8×32-bit S-boxes.

Blowfish: Blowfish is a symmetric-key block cipher, designed in 1993 by Bruce Schneier. Blowfish provides a good encryption rate in software. It is a fast, compact, and simple block encryption algorithm with variable length key allowing a tradeoff between speed and security. Blowfish is unpatented and license-free, and is available free for all applications. Blowfish is known to be susceptible to attacks on reflectively weak keys [14,15]. This means Blowfish users must carefully select keys as there is a class of keys known to be weak. Though it suffers from weak keys problem, but no attack is

known to be successful against it [16]. In this method, key dependent S-boxes and subkeys, generated using cipher itself, makes analysis very difficult and provided key is large enough, so brute-force key search is not practical, especially given the high key schedule cost [17]. It is invulnerable against differential related-key attacks [18]. The features of blowfish are:

- The Feistel structure with 16 round.
- 64-bit input/output data block size
- A variable key size between 32 bits up to 448 bits making it ideal for both domestic and exportable use.
- Using large key-dependent S-boxes (similar to CAST-128, uses fixed S-boxes).

AES: The Advanced Encryption Standard is a block cipher standard invented by Joan Daemen and Vincent Rijmen and developed by NIST, the US National Institute of Standards and Technology. AES is a variant of Rijndael with a fixed block size. AES ciphers use a 128-bit block and 128, 192 or 256-bit keys. The larger block size helps resist birthday attacks while the large key size prevents brute force attacks [19]. It is efficient in both software and hardware. It was selected through an open competition involving hundreds of cryptographers during several years. For brute-force attack, AES is definitely more secure than DES due to the larger-size key, for statistical attacks, numerous tests have failed to do statistical analysis of the ciphertext, and for differential and linear attacks, There are no differential and linear attacks on AES as yet. The main features of AES are:

- AES does not use a Feistel network. It uses 10, 12, or 14 rounds.
- 128-bit input/output data block size
- 128, 192, and 256-bits key sizes. The key size depends on the number of rounds.
- AES uses one S-box which takes in 8 bits and outputs 8 bits.

DES: Data Encryption Standard was the first encryption standard published by NIST (National Institute of Standards and Technology). This method is not regarded as a secure method for the reason of its short key length (56-bit key size). So DES has been replaced by the AES.

For brute-force attack, by regarding to weakness of short cipher key and also the key complement weakness in DES (We have used an arbitrary key and plaintext to find the corresponding ciphertext. If we have the key complement and the plaintext, we can obtain the complement of the previous ciphertext) [20], it can be broken using 2^{55} encryptions. For differential Cryptanalysis, it has been revealed that the designers of DES already knew about this type of attack and designed S-boxes and chose 16 rounds to make DES specifically resistant to this type of attack [20]. For linear Cryptanalysis, Linear cryptanalysis is newer than differential cryptanalysis. DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis. S-boxes are not very resistant to linear cryptanalysis. It has been shown that DES can be broken using 2^{43} pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely [20]. The features of DES are:

- The Feistel structure with either 16 round.
- 64-bit input/output data block size
- A key size of 56 bits.
- Using 8 S-boxes which each S-boxes maps 6 bits to 4-bits

2.3 Asymmetric/Public Key Encryption

This method uses two keys: public key and private key. The public key is made publicly available and for secure message exchanging, receiver public key is used to encrypt messages by sender. The private key is secret and is used to decrypt received messages. Applications of public key cryptosystems are: 1. Secrecy, encryption/decryption of messages, 2. Digital signature, sign message with private key, and 3. Key exchange, share secret session keys. An example is RSA public key method. It is better to use large keys to avoid brute force attacks, but public key algorithms are less efficient with larger keys. So public key cryptography mainly used for key management and signatures.

Advantages of asymmetric key encryption methods are:

- Convenience: they have not the problem of key distributing. The unique allocation of private and public keys are for each user that allows them to conduct secure exchanges of information without first needing to devise some way to secretly swap keys. Anyone broadcasts its public key and kept its private key secret.
- Public key cryptography simplifies the management of symmetric keys to the point whereby a symmetric key can be used not only for each trading partner, but for each exchange between trading partners.
- They can provide a method for authentication with digital signatures: these methods allow using digital signatures which enable the message receiver to verify message is truly from claimed sender. The sender cannot deny sending this message. Digital signatures also allow the receiver to detect if the message was altered in sender or in channel by the other nodes. Authentication via secret-key systems requires the sharing of some secret information and sometimes requires a trustworthy third party as well. As a result, a sender can repudiate a previously authenticated message by claiming that the shared secret was somehow compromised by one of the parties. For example, the Kerberos secret-key authentication system involves a central database that keeps copies of the secret keys belonging to all users; an attack on the database would allow widespread forgery. But public-key authentication prevents this type of repudiation.

Disadvantages of asymmetric key encryption methods are:

- Public keys should be authenticated by a trusted party: No one can be sure that a public key belongs to the person it specifies.
- Slow: Public key encryption is slow compared to symmetric encryption.
- Need to larger key size than symmetric-key encryption.
- Their security is based on some hard mathematical problems.

- Use more computer resources: they require more computer resources compared to symmetric key encryption methods.

2.4 Asymmetric/Public Key Encryption Algorithms

RSA: RSA is one of the first and probably best known public-key schemes. It was developed in 1977 by R.Rivest, A.Shamir and L. Adleman [21]. The security of RSA is based on the effort to factorize the big numbers of modulus. It is computationally infeasible to determine decryption key given only algorithm and encryption key. RSA key generation is as follow:

1. Choose two big prime numbers p and q.
2. Compute $n = p.q$.
3. Compute Euler value => $\varphi(n) = (p - 1)(q - 1)$.
4. Choose at random the e value such that $1 < e < \varphi(n)$ and e and n are co-prime ($gcd(e, \varphi(n))=1$).
5. Compute a value for d such that $(d.e \bmod \varphi(n)) = 1$ and $0<d<n$.
6. Public key is (e, n) and private key is (d,n).

For encrypting a message m, we should convert the message into message blocks $m_1, m_2, ..., m_n$ (each block is consisting of 1 to k characters or bytes). Then each block m_i is mapped to an integer value p_i with an arbitrary rule (p_i codes are selected as $0 < p_i < n$). For encryption goal, we use receiver public key to encrypt message in sender side as $c_i = p_i^{\,e} \bmod n$ (m must be smaller than the modulus n). Public keys should be broadcasted before secure message exchanging in the network with issued certificates by a Center of Authority (CA) that is reliable for all of existing nodes. After receiving message, receiver can decrypt encrypted message using its private key as $p_i = c_i^{\,d} \bmod n$.

There is a problem for using RSA in message exchanging in VANET. Message size should be small (smaller than number of key bytes-11). It means we cannot use RSA for encrypting long messages in the network. On the other hand, many of messages that should transmit securely in vehicle to vehicle (V2V) or vehicle to roadside unit (V2R) communications, are long messages (usually they are consist of message signature or other security information in addition to the message). Setting a large key is not a good idea for it, because VANET is sensitive to time and with increasing the key size, decryption and specially encryption will take more time. But we would be able to encrypt long messages with RSA the same way as it is done with block ciphers. It encrypts the messages in blocks and binds the blocks with an appropriate chaining mode. But, this is not the usual way to do it and we won't find support for it (RSA chaining) in the available libraries. There is another way for encryption long messages. Since RSA is quite slow, the usual way to encrypt large messages is using hybrid encryption. In hybrid encryption we use a fast symmetric encryption algorithm for encrypting the data with a random key. The random key is the same secret key that encrypted with RSA and send along with the symmetric key encrypted data.

In RSA algorithm, to prevent Brute-force attack, we should choose a large d (it makes algorithm slower). Mathematical attacks are possible by: 1. Factoring n into its two prime factors, 2. Determining $\Phi(n)$ directly, without determining p or q, and 3. Determining d directly, without determining $\Phi(n)$. Factoring n is considered fastest approach, hence used as measure of RSA security. To prevent chosen ciphertext attack, we should use padding (Optimal Asymmetric Encryption Padding) to the message.

Elliptic Curve Cryptography (ECC): ECC Uses elliptic curve arithmetic (instead of modular arithmetic in RSA). Its security is equivalent to RSA and it is used for encryption/decryption, key exchange and digital signatures. It is being implemented in smaller devices like cell phones. ECC encryption systems are based on the idea of using points on a curve to define the public/private key pair. A full description is expressed in [22]. ECC security relies on the difficulty of the Elliptic Curve Discrete Logarithm problem (ECDLP), which means smaller key size yield equivalent levels of security. Algorithms based on ECDLP are [23]:

- Elliptic Curve Diffie-Hellmann Key Agreement (ECDH) that is used to exchange the secret keys securely via a non secure channel.
- Elliptic Curve Menezes-Qu-Vanstone (ECMQV): the key agreement algorithm allows two parties to agree on a common secret value with user authentication.
- Elliptic Curve Integrated Encryption Scheme (ECIES) is a hybrid encryption scheme which provides semantic security against chosen plain text and chosen cipher text attacks. ECIES uses following functions. It has following encryption and authentication functions.
  1. Key Agreement function
  2. Key Derivation Function
  3. Encryption algorithm
  4. Hash function
- Elliptic Curve Digital Signature Algorithm (ECDSA) that is a digial signature algorithm.
- Elliptic Curve Nyberg-Rueppel (ECNR) that is a signature scheme used in a number of standards, defined by IEEE 1363-2000.
- Elliptic Curve Pinstov Vanstone Signature (ECPVS) is a digital signature scheme offering partial message recovery. The size of signatures created using ECPVS is smaller than other schemes (e.g., RSA).

For VANET secure message exchanging, we need to ECIEC for public key encryption and ECDSA for digital signature generation.

3 Experimental Results

2.3 Comparison between ECC Method and RSA

We compared encryption and decryption times for different key lengths of RSA method in Figure 1 and 2. As is shown in the figures, increasing the key length strongly affects the decoding time. So RSA is not sufficient for encrypting long messages.

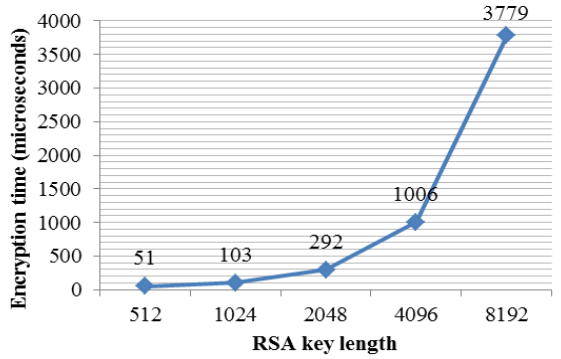

Fig. 1. Encryption time for different key sizes.

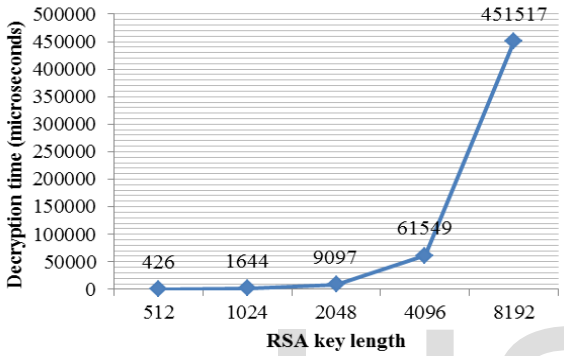

Fig. 2. Decryption time for different key sizes.

ECC provides similar functionality to RSA but requires to less computing power and memory and has smaller keys (better performance) compared with RSA for longer messages. A comparison of security levels for three public key algorithms based on reported data on [24] is shown in figure 3. In this figure, a MIPS-year is computing time of one year on a machine performing one Million Instructions Per Second. The size of selected key pairs for the RSA and ECC cryptosystems are in Table 1. It is evidence that key pairs are shorter for the ECC than RSA.

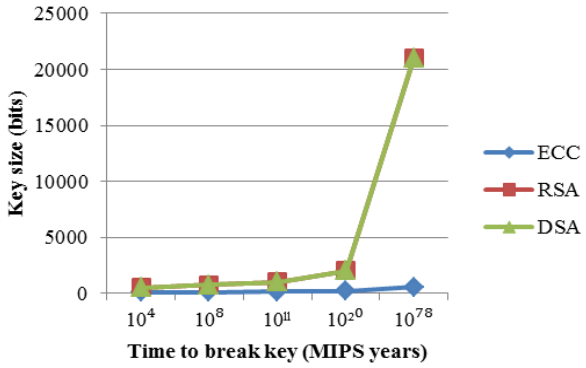

Fig. 3. Comparison of security levels for ECC, RSA and DSA.

One of the problems in public key cryptography methods is message length. Encrypting long messages with public-key methods requires combined schemes, because final message have a long length and it consumes a lot of time, which is not applicable and sufficient for VANETs. Each public key method can either encrypts session key (i.e. AES key) (data itself is encrypted with that key), or it is used for signing a short message such as message digest (signature application). For public key encryption and signature generation, we can use both ECC and RSA methods. A full comparison for using two methods for signing exchanging messages is expressed in [25]. So we compare only public key encryption ECC method with the name of ECIES, expressed in [26], and RSA method in Table 2. We compared ECIES-224 with RSA-2048 because they have the same security levels as is stated in [26]. Increasing keys length in RSA method has a huge effect than ECIES, on decryption time (it is used more than encryption). It means for short messages that need to keys less than 1024 bits in RSA, using RSA is effective for processing time and for longer messages we should use ECIES method for increasing performance and scalability in the network.

TABLE 1
SPACE REQUIREMENT [24]

	Public key (bits)	Private key (bits)
1024-bit RSA	1088	2048
160-bit ECC	161	160

TABLE 2
TIME COMPARISON BETWEEN TWO ENCRYPTION METHODS WITH THE SAME SECURITY LEVELS (IN MICROSECONDS)

	ECIES-224	RSA-2048
EN	3029	292
DE	2479	9097

2.3 Comparison Symmetric Key Based Methods

Experimental results are given in Figures 4 and 5 for the selected six encryption algorithms. We used different size of the messages for testing the common selected methods with 56 bits key size for DES method and 128 bits key size for other methods. The results show blowfish has the minimal encryption time and Camellia has the maximum encryption time. On the other side, Camellia has the minimum decryption time and Seed algorithm has the maximum decryption time.

In vehicular networks messages containing road events are signed once by sender vehicle and are verified by all of vehicles in sender radio range (V2V communication). So verification repeats more than once. Encryption usually performs once and Camellia has a very high encryption time compared with other algorithms, on the other hand blowfish is the best algorithm for encryption and the best algorithm for decryption after Camellia with a very small difference to it (up to 8 microseconds for 2000 bits message). Also Blowfish has not any known security weak points so far. So these make it an excellent candidate for a standard symmetric key encryption algorithm for using in VANET.

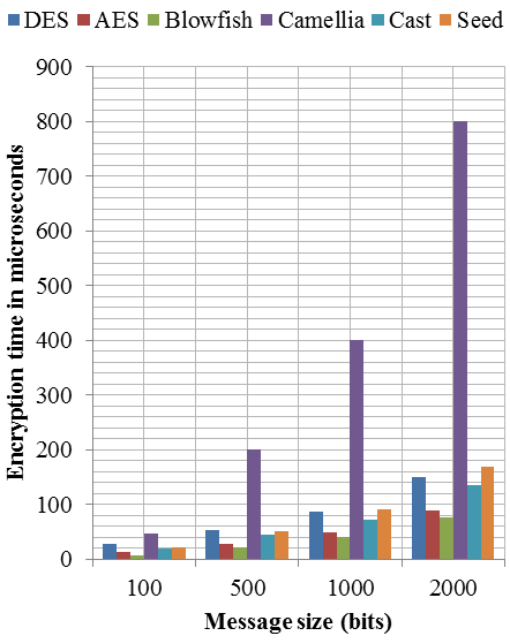

Fig. 4. Encryption time for different algorithms and different message sizes.

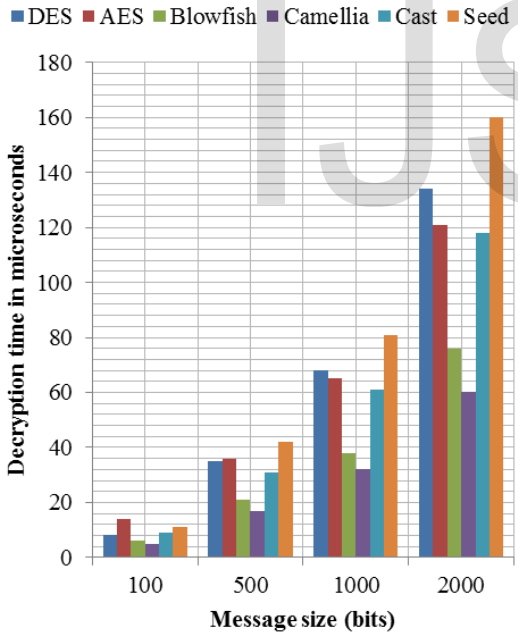

Fig. 5. Decryption time for different algorithms and different message sizes.

7 Conclusion

Vehicular networks provide many useful applications for avoiding dangerous crashes, warning the driver about weather, road, traffic, and other hazardous driving conditions, improving traffic flow with traffic management. In order to take full advantage of this network, the communications must be secured with all of security requirements. Many attacks in this network can be prevented or detected using cryptography methods. Due to the high speed of vehicles, they have limited opportunities to communicate with each other. So the response time for selected encryption method must be minimal and the security level must be acceptable considering the key lifetime. This paper provided evaluation of symmetric and asymmetric encryption algorithms for using in this network. A comparison has been conducted that blowfish is found to be the best encryption method for symmetric key based methods. For encrypting message by public key methods, if messages are too short, RSA at 1024 bits, consumes less time with a high security level, otherwise ECIES is the best choice.

References

[1] ASTM E2213-03, "Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems — 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer PHY) Specifications", *ASTM International*, July 2003.

[2] M. Rahbari, M. A. J. Jamali, "Efficient detection of sybil attack based on cryptography in VANET", International journal of network security & its applications (IJNSA), pp.185-195, 2011.

[3] R. Lu, Doctoral dissertation, *Security and Privacy Preservation in Vehicular Social Networks* , University of Waterloo, 2012.

[4] C. Zhang, Doctoral dissertation, *On Achieving Secure Message Authentication for Vehicular Communications*, University of Waterloo, 2010.

[5] T. W. Chim, S. M. Yiu, L. C. K. Hui, & V. O. K. Li, Grouping-enabled and privacy-enhancing. *Information Systems Security*, vol. *7, no.* 1, pp. 60-96, 2011.

[6] S., Qi, Y. Chang, H. Zhu, J. Zhao, & X. Shen, Footprint: Detecting Sybil Attacks in Urban Vehicular Networks. *Parallel and Distributed Systems, IEEE Transactions on*, vol. *23, no.* 6, pp. 1103-1114, 2012.

[7] S. Park, B. Aslam, D. Turgut, & C. C. Zou, Defense against Sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support. Security and Communication Networks, vol. 6, no. 4, pp. 523–538, 2013.

[8] Chukwumah Ezeobika M.D., Advantages and Disadvantages of Symmetric and Asymmetric Key Encryption Methods, available on: http://voices.yahoo.com/comparing-symmetric-asymmetric-key-encryption-6329400.html.

[9] M. Blumenthal, "Encryption: Strengths and Weaknesses of Public-Key Cryptography," Villanova University, Villanova, Computing Research Topics CSC 3990, 2007, (lecture note).

[10] RFC standard available on: http://tools.ietf.org/html/rfc4269

[11] RFC standard available on: http://tools.ietf.org/html/rfc3713

[12] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, T. Tokita, Camellia: A 128-bit block cipher suitable for multiple platforms. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 41–54. Springer, Heidelberg, 2001.

[13] RFC standard available on: http://www.faqs.org/rfcs/rfc2144.html

[14] T. Gonzalez, A Reflection Attack on Blowfish, JOURNAL OF LATEX CLASS FILES, vol. 6, no. 1, 2007.

[15] O. Kara and C. Manap, "A New Class of Weak Keys for Blowfish", In Alex Biryukov, editor, *Fast softeare Encryption*, *14th International Workshop*, FSE 2007, vol. 4593 of *Lecture Notes in Computer Science*, pages 167-180. Spring-Verlag, 2007.

[16] N. Kumar, J. Thakur, A. Kalia, PERFORMANCE ANALYSIS OF SYMMETRIC KEY CRYPTOGRAPHY ALGORITHMS: DES, AES and

BLOWFISH, An International Journal of Engineering Sciences, vol. 4, pp. 28-37, 2011.

[17] Nguyen Nam Hong, Cryptography lecture notes, chapter 07, Contemporary Symetric Ciphers, available on: http://namhongthanhloc.webs.com/cryptography.htm.

[18] M. Ebrahim, S. Khan and U. B. Khalid, Symmetric Algorithm Survey: A Comparative Analysis, *International Journal of Computer Applications,* vol. 61, no. 20, pp. 12-19, January 2013.

[19] Glossary for the Linux FreeS/WAN project available on: http://www.freeswan.org/freeswan_trees/freeswan-2.01/doc/glossary.html

[20] Quantum Information and Network Security Laboratory, Lecture note on: http://islab.csie.ncku.edu.tw/course/slide/ch_06.ppt

[21] W. Stallings, Cryptography and Network Security, chapter 9, Fourth Edition.

[22] L. Delgrossi, & T. Zhang, Cryptographic Mechanisms. *Vehicle Safety Communications: Protocols, Security, and Privacy*, John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 167-208, 2012.

[23] C. Endrodi, Efficiency Analysis and Comparison of Public Key Algorithms; Technical Report; SEARCH Laboratory, Budapest, Hungary, 2002.

[24] H. Pietilainen, "Elliptic Curve Cryptography on Smart Cards", Masters Thesis, Faculty of Information Technology, University of Helsinki, 2000.

[25] N. Jansma and B. Arredondo, "Performance Comparison of Elliptic Curve and RSA Digital Signatures" Technical Report, University of Michigan College of Engineering, 2004 .

[26] V. G. Martínez, L. H. Encinas, and C. S. Ávila,"A Survey of the Elliptic Curve Integrated Encryption Scheme", Journal Computer Science & Engineering, vol. 2, no. 2, August 2010.