Analysis of TESLA Protocol in Vehicular Ad Hoc Networks Using Timed Colored Petri Nets

Mohammad Hossein Jahanian, Farshad Amin, Amir Hossein Jahangir
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
{jahanian, f_amin}@ce.sharif.edu, jahangir@sharif.edu

Abstract— **Authentication is an essential security requirement in safety-related vehicle-to-vehicle applications in VANETs. TESLA is one of the most popular broadcast source authentication protocols proposed and standardized for this purpose. Having strict time constraints and being prone to GPS synchronization errors make the analysis of this protocol challenging. In this paper, we utilize a timed model checking approach based on timed colored Petri nets to model and verify TESLA considering its time-sensitive behaviors. In this way, we show how neglecting timing aspects in protocol design and in protocol modelling can lead to successfully launched attacks and erroneous analyses respectively, and how its refinement can help improve the protocol's security. Our work extends the problem of analyzing basic TESLA done in previous related works, to one in which TESLA is modelled and verified in a loose synchronization setting, which is the case in VANETs. This new problem definition led to finding new attacks that are directly rooted in the sending times of packets. We show what changes need to be made to TESLA so that its security property be preserved in a loose synchronization condition.**

Keywords— Vehicular ad hoc networks; TESLA; Model checking; Synchronization; Timed colored Petri nets

I. INTRODUCTION

Vehicular Ad Hoc Network (VANET) is a special type of Mobile Ad Hoc Network (MANET) in which mobile nodes are moving vehicles [1]. VANETs use Dedicated Short Range Communication (DSRC) technology, based on the IEEE 1609 family of standards (called WAVE) to enable wireless transmission of safety and non-safety related messages between vehicles or between a vehicle and the Road-Side Units (RSUs) [2, 3]. Each vehicle is assumed to be equipped with network devices, sensors and a Global Positioning System (GPS) device [4]. The main goal of the deployment of VANETs is enhancing driving safety, i.e. decreasing accidents and life loss rates [2]. For this purpose, safety applications are introduced.

Vehicle-to-Vehicle (V2V) communication pattern plays an important role in safety applications by enabling the transmission and delivery of safety-related messages, i.e. event-driven alerts and periodic beacons [2]. These messaging schemes have certain security requirements which are critical due to the nature of safety applications, and must be addressed and paid attention to. These security requirements include authentication, integrity, privacy, non-repudiation and availability. Confidentiality is generally not required since safety information is publicly broadcast and should be readable to everyone [5].

Authentication protocols provide authenticity and integrity of data by preventing impersonation and modification attacks. To achieve this for safety messaging in VANET, one-way broadcast source authentication schemes are needed. Different authentication protocols have been proposed for VANETs.

Timed Efficient Synchronous Loss-Tolerant Authentication (TESLA) protocol has been shown to be a low-delay authentication protocol, and has been introduced as a potential protocol to be used in VANETs. The reason for that is that TESLA uses synchronous key encryption schemes which is verified by the receiver in less time than the time consumed for asynchronous digital signatures. Therefore, TESLA is suitable for real-time nature of VANET safety applications. TESLA employs message authentication codes (MACs), hash chain of keys and a strict timing schedule regarding when each key should be disclosed, while requiring loosely synchronized clocks between principals [6, 7].

In this paper, we verify TESLA protocol based on a timed model checking approach using Timed Colored Petri Nets as the modelling language and CPN Tools as the model checker. In this analysis, we also investigate the impact of loose time synchronization caused by vehicles' GPS synchronization, which is the specific synchronization method in VANETs, on the correctness of the protocol. We show how a time-related attack occurs on this protocol and propose a security improvement based on the addition of awareness of loose synchronization to the sender.

The organization of the paper is as follows. In Section II we provide an overview of the related work. We describe details of TESLA protocol and timed colored Petri net modelling concepts in Sections III and IV, respectively. In Section V, we present our proposed timed model checking method for analysis of TESLA. Finally, the paper is concluded in Section VI.

II. RELATED WORK

Verification of security cryptographic protocols is a way of answering the question: "Is this protocol correct and secure?".

For time-sensitive protocols, time should be explicitly dealt with in the verification process. Timestamp and freshness, timeouts and retransmissions, time schedules and time synchronization are timing issues that can affect the security of a protocol. One important work in the field of timed analysis of security protocols has been done in [8]. The authors of the paper propose a method based on Timed Automata and the model checker UPPAL to verify security protocols such as Needham-Shroeder and Yahaolom, considering timeouts and retransmissions. In [9], the authors propose a real-time process algebra to verify security properties of protocols. The author in [10] uses colored Petri nets and the model checker CPN Tools [11], to analyze time-sensitive security protocols, namely the Wide Mouthed Frog (WMF) protocol.

The basic TESLA protocol, as introduced in [7] has been the subject of some formal verification works. In [12], which is the first attempt in verification of TESLA, a theorem proving approach using TAME was employed. In [13], TESLA was analyzed using CSP/FDR model checking. In this approach, data independence techniques and reduction strategies were used to design finite and small state spaces. In [14], the model checker MCMAS-X along with a temporal epistemic logic were proposed to verify TESLA. In [15], verification of TESLA was done through a theorem proving approach using TOTS/CafeOBJ. To our knowledge, TESLA, as specifically standardized for VANET in [6], has never been subject to formal verification. Also, the impact of time synchronization errors in TESLA has not been explicitly addressed before.

III. TESLA Protocol

In this section, we present the informal description of TESLA, as detailed in [6] which belongs to VANETs. Note that this version of TESLA employs the basic ideas of the original version of TESLA in [7] with some differences about the initial phase and packet formats. The message payloads in VANET's TESLA contain safety-related information.

TESLA uses hash chains in which some secret is iteratively linked to an initial commitment, using a one-way hash algorithm such as SHA-256. Time is divided into separate time slots (ideally 10 milliseconds) and each element in the hash chain represents a (synchronous) key. Each key is valid only in one time slot for the purpose of creating MACs by the sender, and must be disclosed to the recipients in the next time slot (or based on implementation choice, *b* time slots later). All data transmissions and key disclosures take place through broadcast.

For a receiver, each key is validated if it is verified by the hash function and the previous key that has been validated. In other words, to accept a key K_{i+1}, the condition $K_i=H(K_{i+1})$ must hold, in which K_{i+1} and K_i are the current and previous keys respectively, and H denotes the one-way hash function.

As for the first key, K_1, it must be committed to a digital signature by the sender, using its certified asynchronous key pairs. So although TESLA works mainly based on synchronous cryptography, it is not completely needless of a pre-deployed public key infrastructure (PKI).

Since TESLA principals work according to a strict time schedule, synchronization of the clocks must be assured. Synchronization can be either tight or loose. Loose synchronization must be tolerated by TESLA, without degradation of its functionality and security. As stated in [6], time synchronization in VANETs is achieved through the GPS system. Due to occasional loss of GPS signals, vehicles' clocks can have small (or perhaps large) deviations from the global clock. According to [6], this skew can be up to 1.5 milliseconds.

TESLA is secure if and only if:

a) Integrity and authenticity of a message *m* is verified using MAC and hash functions and,

b) Each key is disclosed in the network only after the corresponding message has been sent.

IV. Timed Colored Petri Nets

Petri net was first introduced by Carl Adam Petri and has been considered as a practical graphical modeling language to describe various concurrent systems. A Petri net N is a directed, bipartite graph consisting of two kinds of nodes, namely places and transitions, which represent conditions (or data holders) and events (or computation steps), respectively [16]. The edges in the graph are called arcs which represent flow relations between places and transitions. Tokens represent data items which can reside in places and can move from place to place if they are fired by transitions. Firing occurs if preconditions of a transition hold (or enough needed data inputs are available). The arrangement of tokens in places is called a marking (or state), which can be represented by an adjacency matrix. Initial token values in each place determine the initial marking (initial state).

Two kinds of properties can be investigated through a Petri net: behavioral properties and structural properties. Behavioral properties are those which are dependent on marking and include reachability, boundedness and liveness. In this analysis, we deal mostly with reachability. We say a marking M_n is reachable from the initial marking M_0 if a sequence $\sigma = M_0\, t_1\, M_1\, t_2 \ldots\, t_{n-1}\, M_n$ exists where t_i is the *i*-th occurring transition and M_i is the *i*-th reachable marking. The problem of finding whether a certain marking is reachable is called the reachability problem, which is a decidable one [16]. Petri Net has a formal definition which is the basis of the soundness of analyses done by it. This formal definition is presented in [16].

What came above is called basic Petri net and is not much convenient to be used to model many complex real world systems. Therefore, extensions were proposed to basic Petri net to achieve high level Petri nets. The extension we use in this analysis is timed colored Petri nets.

In a colored Petri net, each token has a distinct color which represents its data type. Each place only holds certain colors and each transition inputs and outputs tokens of certain colors [17]. In [18], thirteen reasons are mentioned why colored Petri net is a powerful and suitable modelling tool to be used for various systems. In a timed colored Petri net, in addition to a color, each token has also a timestamp. This timestamp determines when a token is available to be used. A global system clock in the model is accessible. Therefore, the arrangement of tokens in places, in addition to timing aspects (tokens' timestamps and the global clock) are called a timed marking.

For automatic verification, a software tool supporting Petri net based model description, property specification and state space (reachability) analysis is needed. Our choice is CPN Tools, which has the features of editing, simulating and analyzing, for timed colored Petri nets [19]. So it seems to be a capable model checker. CPN Tools uses a version of standard ML (metalanguage), for the declarations and specifications of properties through queries.

V. Timed Model Checking of TESLA

In this section, we investigate the security correctness of TESLA and determine attack success scenarios in two synchronization conditions: tight synchronization and loose synchronization. In loose synchronization case, we analyze the impact of sender awareness in protocol security. To verify requirements, we define a "timed authentication" property and show why this kind of property definition is important.

The verification approach we choose is timed model checking, in which the validity of each property is checked in the whole state space in an exhaustive search manner, and each state is timed. Time in our model is defined explicitly, quantitatively and discretely.

To achieve our analysis, we envision four different scenarios and apply the following procedure for each scenario:

- modelling phase
    - defining color sets (types), variables, constants and functions
    - designing graphical Petri net models using places, transitions, tokens and arcs
    - determining initial marking of the system model
    - conducting a number of step-by-step executions (simulations) to find early modelling errors quickly
    - specifying verifiable properties and defining them as ML queries
- running phase
    - generating the complete state space automatically by CPN Tools and check properties in that state space
- analyzing phase
    - collecting the verification results
    - extracting property violation traces (counterexamples, if any) and propose improvement solutions (if needed)

Model design is done in a bottom-up way using a two-level hierarchical Petri net in which three low-level models, namely sender, receiver and intruder models, and one high-level model, namely the network model exist. Sender and receiver work exactly according to what is expected by them from a protocol perspective. In our analysis, cryptographic algorithms are considered to be perfect and no entity can encrypt or decrypt without having the appropriate keys.

In Table I, a summary of the piggybacked version of TESLA as in [6] is presented, with the simplifying fact that the sender transmits one and only one packet during each time slot. In this scheme, data and keys are sent together in one packet, rather than separately.

As shown in Table I, packet formats sent by the sender are categorized as follows: one packet of type one, one packet of type two (the first two packet types are used for initialization phase) and the remaining packets of type three. In Table I, i denotes packet number, P_i the i-th packet, m_i the i-th message content (payload), k_i the i-th key, *MAC* the message authentication code generating function, *Sig* the digital signature generating function and *S_PrK* the sender's certified private key. On reception of each packet P_i, the receiver first completes the verification of the previous packet P_{i-1}'s authenticity, and then buffers P_i to be processed and verified in the next time slot (In our model, the receiver waits for P_i only for one time slot). Each failure in verification of a packet results in the abortion and session disconnection of the protocol and therefore, no later packet will be accepted (this is a TESLA condition). In our analysis, timeouts, retransmissions and channel delays are not considered.

Each packet P_i must be sent within a certain timeslot T_i and the assurance by the receiver of this fact is one of the conditions of the acceptance of that packet (time verification). t_i is an arbitrarily chosen instant within the timeslot T_i and denotes the exact sending time of P_i. For t_i, the following conditions hold:

$$t_i \in [(i-1)T, iT[\tag{1}$$

$$t_i = (i-1)T + delay \tag{2}$$

where *delay* is the offset chosen in $[0,T[$. We consider T=10 time units as the (logical) duration of each timeslot.

A. Scenario 1- Tight Synchronization

In this scenario, we assume different principals' clocks are precisely synchronized. This assumption is similar to that of previous analyses of TESLA in [12-15]. After defining the declarations of color sets, variables, constants and functions using ML to be used in model descriptions, we design the models.

The sender and the receiver work exactly as expected in Table I. The offset delay for each packet is chosen in a random, non-deterministic manner in the interval [0,10[.

TABLE I. Summary of TESLA packets

Type	**Packet format**	**Sending time**
One	$P_1:[m_1,MAC(m_1,k_1)]$	$t_1 \in T_1$
Two	$P_2:[m_2,MAC(m_2,k_2), k_1,Sig_{S\_PrK}\{k_1\}]$	$t_2 \in T_2$
Three	$P_i:[m_i,MAC(m_i,k_i), k_{i-1}]$ $(i \geq 3)$	$t_i \in T_i$

1) Model Description

Due to the limitation of space, the sender and the receiver models are omitted from this paper. Only the intruder model and the top-level network model are displayed in Fig. 1 and Fig. 2 respectively.

The intruder is the central part of the model, being the main cause of non-determinism and state space branching the model. The intruder is modelled according to the general Dolev-Yao model [20], and resides between the two ordinary principals and is able to intercept, capture, destroy and modify originally sent packets, start new sessions and create new packets to be sent to the receiver, while not being able to reverse MAC and hash functions and perform cryptanalysis. This intruder has no initial knowledge of keys, MACs, messages and signatures and can only know them after their disclosures in the network.

The goal of this active intruder is to impersonate the sender for the receiver, i.e. receiver R thinks he received message M from sender S while actually R has received M from intruder I, and modify the message content. To achieve this goal, the only practical thing the intruder can do, is to capture one of the packets P_i, keep it (and not pass it) until it receives the next packet P_{i+1}, and then using the disclosed key K_i in P_{i+1} create forged packet $P_ix=[mx, MAC(mx,K_i),K_{i-1}]$, s.t. mx is intruder's message content in place of original m_i by sender, and send P_ix together with P_{i+1} to the receiver. If the receiver accepts the authenticity of the two packets, then the security property of the protocol is violated.

2) Property Specification

To verify the correctness of TESLA, we define a property named timed authentication, as the product of the conjunction of two other properties: origin authentication and time constraint.

Definition 1. Receiver only accepts and considers those messages as authentic that are actually created and sent by the sender (origin authentication).

Based on this property, none of the messages created by the intruder should be accepted by the receiver. Therefore, in the state space, any marking including the message content mx in *result* (place holder for approved messages) in the receiver model, means the violation of this property and is therefore an insecure state.

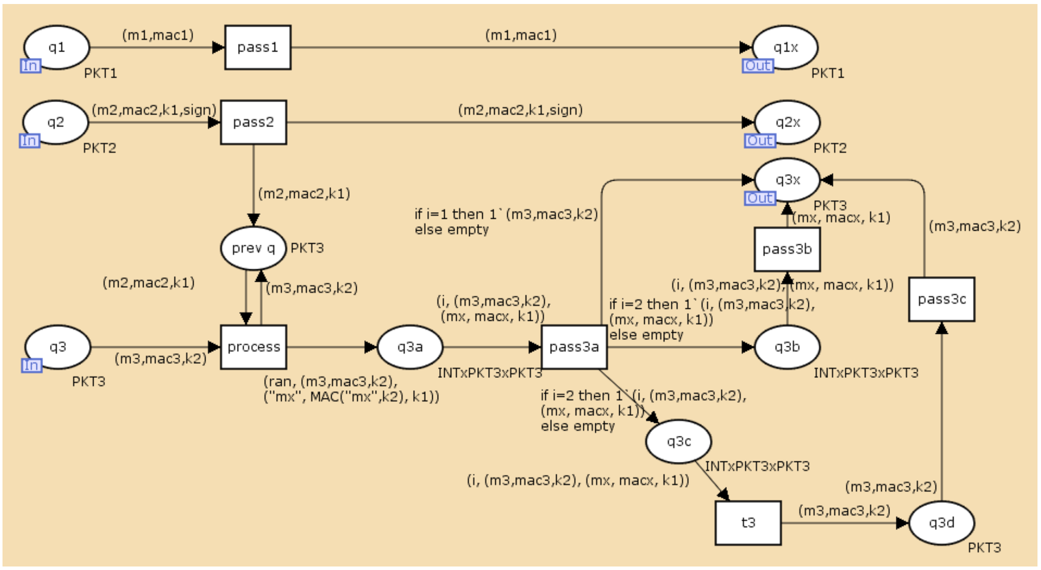

Fig. 1. Intruder Petri net model

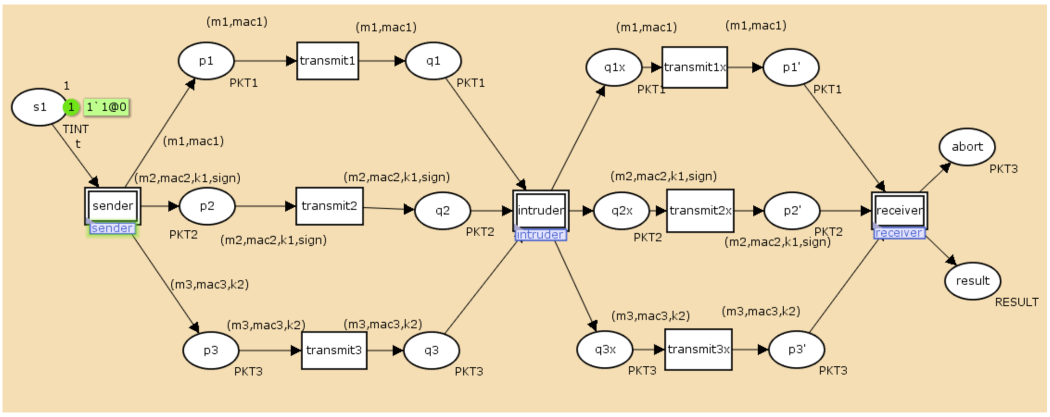

Fig. 2. Top-level (network) Petri net model

Definition 2. Receiver only accepts and considers those messages as authentic if the corresponding keys are not disclosed before the message was sent (time constraint).

Based on this property, for each $i>0$, the sending time of P_{i+1} must be later than reception time of P_i.

Definition 3. Origin authentication and time constraint properties always hold (timed authentication).

3) Verification

To model check properties, we define each by writing a CPN ML query function and apply it to the state space which has been automatically generated by CPN Tools. Doing that, we have observed that both sub-properties and therefore the main property of timed authentication in this scenario are satisfied.

B. Scenario2- Loose Synchronization with Unaware Sender and Active Intruder

In the remaining of this section, we examine that slight differences between principals' clocks due to GPS synchronization errors (e.g. the 1.5 milliseconds mentioned in [6]), can lead to not only degradation of communication functionality, but the occurrence of attacks.

Unlike Scenario 1, in Scenarios 2-4, different principals in the model do not exactly share the system global time. The receiver's time in this situation may have a skew relative to the sender. We set an upper bound on the error (skew) E, namely 2 time units, so the actual time error e can be randomly chosen from the values in {-2, -1, 0, 1, 2} since the receiver's clock may be at most 2 time units ahead or behind the sender's clock. Duration of each timeslot T is still set to 10 time units as before. The intruder is always aware of the existence of loose synchronization in the network and knows E but not e. The sender may be either aware or unaware of this fact. We analyze different scenarios in the remainder of this section.

Scenarios 2 and 3 differ based on the activity of the intruder which can be either active or passive. In each remaining scenario, slight changes to one or two of the low-level models are made. Properties and verification methods are exactly the same as explained in Scenario 1.

In Scenario 2, we assume principals are communicating in a loosely synchronized situation, while the sender is unaware of this fact and is sending messages at any instant, the same as in Scenario 1. It is almost obvious that this will lead to some correct packets being wrongly rejected if they are sent in the edges of a timeslot. Meanwhile, our purpose is to investigate its impact on attacks, i.e. whether this unawareness will lead to accept forged messages by the receiver. Sender and intruder model in this scenario are exactly the same as those in Scenario 1. Receiver model is enriched to take into the account time difference behaviors.

By model checking this scenario, we found out that both sub-properties were violated. So, the intruder successfully launched its attack in some situations and therefore the timed authentication property was not fulfilled.

C. Scenario3- Loose Synchronization with Unaware Sender and Passive Intruder

The change we apply in this scenario in comparison to Scenario 2, is that the intruder sends exactly the same m_i to the receiver, instead of changing it to mx (Fig. 3). We call this a passive intruder. The purpose of this scenario is only to show the vitality of defining the timed authentication property.

We verified the model of this scenario and observed that while the second sub-property, i.e. time constraint, was violated as in Scenario 2, the first property, i.e. origin authentication was indeed satisfied, since no mx was actually created by the intruder to be probably accepted by the receiver. This shows the necessity of specifying the timed authentication property. If we had defined TESLA's security property as merely a conventional authentication property, i.e. untimed one, this scenario would have passed the verification successfully and we would not have discovered this problem. However, it is clearly a vulnerability because although the intruder does not actually do any impersonation or modification, but in some situations he deduces the key before the receiver gets the corresponding message. He can hand the key to another intruder for other malicious purposes.

D. Scenario 4- Loose Synchronization with Aware Sender

By looking at the attack traces, i.e. sequence of states starting from the initial marking and ending in one of the insecure markings, we can recognize the timing pattern leading to the possibility of attack success in Scenario 2.

In this way, we observed that the intruder only had a chance of success in situations where $t_{i+1} \in [iT, iT+e[$, i.e. if a packet is sent in the starting edge of a timeslot and the actual skew is so that sender and receiver consider that particular instant originating from different timeslots. Hence, if we prevent this event, we can prevent the attack. That is the main idea of our time-related security improvement solution to TESLA.

To refine the protocol, we limit the duration within each timeslot in which the sender can send packets (Fig. 4). Note that cutting the beginning edge is for security reasons while cutting the ending edge has only functionality reasons.

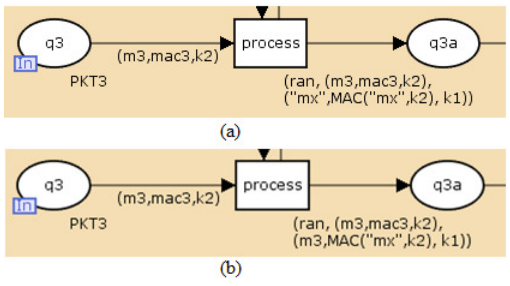

Fig. 3. Intruder models: a) active b) passive

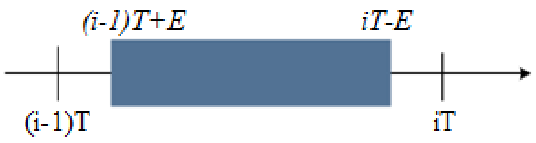

Fig. 4. Permitted sending intervals (sender's clock)

TABLE II. SUMMARY OF VERIFICATION SCENARIOS

Scenario	Synchronization	Sender awareness	Intruder	Origin auth. property	Time cons. property	Timed auth. property	Verification result
1	Tight	-	Active	✓	✓	✓	Correctness proved
2	Loose	Unaware	Active	×	×	×	Attack found
3	Loose	Unaware	Passive	✓	×	×	Vulnerability discovered
4	Loose	Aware	Active	✓	✓	✓	Correctness proved

After changing the sender model to support the changes made above, using the same receiver and active intruder models from Scenario 2, and model checking the properties, we observed that both sub-properties, and therefore the main property of timed authentication hold and TESLA is now secure in the loose synchronization environment.

VI. CONCLUSION

In this paper, we analyzed and verified the TESLA protocol in VANET by considering its timeliness through a timed model checking approach using timed colored Petri nets and CPN Tools model checker. We showed how global time and clock skews due to GPS loose synchronization can be modelled. We also specified a special timed authentication property and showed why this way of defining the property is necessary to correctly verify TESLA, rather than the conventional authentication property specification.

By model checking, we concluded that if there is loose synchronization in the network and the upper bound on clock skew is previously known (which usually is, according to [6]), by adding awareness to the sender of this situation and limiting its choices of sending instants, we can prevent timely attacks that could occur successfully otherwise. A summary of conclusions of various scenarios is presented in Table II.

REFERENCES

[1] M. Jerbi, S. M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, "An improved vehicular ad hoc routing protocol for city environments," Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, June 2007, pp. 3972-3979.

[2] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," Journal of Network and Computer Applications, Vol. 37, January 2014, pp. 380-392.

[3] S. Olariu, and M. C. Weigle, "Vehicular networks: from theory to practice," Chapman and Hall/CRC, 1st Edition, March 2009.

[4] A. Festag, et al., "Vehicle-to-vehicle and road-side sensor communication for enhanced road safety," Proceedings of ITS World Congress and Exhibition, New York, USA, November 2008.

[5] J. M. De Fuentes, A. I. González-Tablas, and A. Ribagorda, "Overview of security issues in vehicular ad hoc networks," Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts, IGI-Global, 2011.

[6] F. Ahmed-Zaid, et al., "Vehicle safety communications - applications (VSC-A) final report: appendix volume 3 security," U.S. Department of Transportation, National Highway Traffic Safety Administration, Rep. No. DOT HS 811 492D, September 2011.

[7] A. Perrig, R. Canetti, J.D. Tygar, and D. Xiaodong Song, "Efficient authentication and signing of multicast streams over lossy channels," Proceedings of the 2000 IEEE Symposium on Security and Privacy, IEEE Computer Society Washington, DC, USA, 2000.

[8] R. Corin, S. Etalle, P. H. Hartel, and A. Mader, " Timed model checking of security protocols," Proceedings of the 2004 ACM workshop on Formal methods in security engineering, Washington DC, USA, October 2004, pp. 23-32.

[9] R. Gorrieri, E. Locatelli, and F. Martinelli, "A simple language for real-time cryptographic protocol analysis," Proceedings of the 12th European Symposium on Programming, ESOP 2003 Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003 Warsaw, Poland, April 2003, pp. 114-128.

[10] M. Xu, "Considering time in formal analysis of security protocols using colored Petri nets," International Conference on Embedded Software and Systems Symposia, (ICESS Symposia 2008), Sichuan, China, July 2008, pp. 63-68.

[11] CPN Tools, http://cpntools.org/

[12] M. Archer, "Proving correctness of the basic TESLA multicast stream authentication protocol with TAME," Workshop on Issues in the Theory of Security, Portland, Oregon, USA, January 2002 pp.14-15.

[13] P. Broadfoot, and G. Lowe, "Analysing a stream authentication protocol using model checking," Proceedings of the 7th European Symposium on Research in Computer Security, Zurich, Switzerland, October 2002, pp. 146-162.

[14] A. Lomuscio, F. Raimondi, and B. Wozna, "Verification of the TESLA protocol in MCMAS-X," Proceedings of Concurrency, Specification and Programming (CS&P), Humboldt University Press, Wandlitz, Germany, 2006, pp. 255-267.

[15] I. Ouranos, K. Ogata, and P. Stefaneas, "Formal analysis of TESLA protocol in the timed OTS/CafeOBJ method," 5th International Symposium (ISoLA 2012), Heraklion, Crete, Greece, October 2012, pp. 126-142.

[16] T. Murata, "Petri nets: properties, analysis and applications," Proceedings of the IEEE, Vol.77, No.4, April 1989, pp. 541-580.

[17] K. Jensen, "Coloured Petri nets and the invariant-method," Theoretical Computer Science, Vol. 14, No. 3, 1981, pp. 317-336.

[18] K. Jensen, "An introduction to the theoretical aspects of coloured Petri nets," of A Decade of Concurrency, Lecture Notes in Computer Science, Springer-Verlag, 1994, pp. 230-272.

[19] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri nets and CPN Tools for modelling and validation of concurrent systems," International Journal on Software Tools for Technology Transfer (STTT), Vol. 9, No. 3, May 2007, pp. 213-254.

[20] D. Dolev, and A. C. Yao, "On the security of public key protocols," Information Theory, IEEE Transactions on, Vol. 29, No. 2, March 1983, pp. 198-208.