A Lightweight Authentication and Communication Protocol in Vehicular Cloud Computing

Harsha Vasudev and Debasis Das
Department of CS&IS, BITS Pilani, K.K. Birla Goa Campus, Goa, India.
harshavasudev.dev@gmail.com, deba16@gmail.com

***Abstract*—With the increased number of vehicles, the traditional VANETs (Vehicular Ad-hoc Networks) experiencing new face transitions. One such concept is Vehicular Cloud Computing (VCC), where the data exchanges between vehicles or any entities, in traffic condition or in an emergency situation can be easily done. Establishing authentication and secure communication in VCC is a major problem because of the highly dynamic nature of vehicles. False messages may mislead the drivers which may cause serious issues. In this paper, we propose a lightweight protocol for authentication and secure communication in the VCC. More specifically, we design a lightweight protocol with hash operations, XORs, concatenations, etc., to ensure security. In-depth security analysis is done to check various attacks in the VCC system. In addition to that, we have done hardware implementation on a desktop computer, on a small single board computer- Raspberry Pi and verified the execution time needed. The performance analysis reveals that our protocol outperforms well in different scenarios like communication cost, storage overhead, computation time, and energy consumption.**

I. Introduction

In today's world, we are witnessing an exponential growth of traffic congestion, passenger security problems, and poor road conditions with the increased vehicle population. Different types of technologies are developed however, the accident rate and transportation issues are growing. The Intelligent Transportation Systems (ITS) is a way to solve or at least minimise traffic congestion. It includes various types of transportation media's such as air, road, rail, sea, etc. The objective of ITS is to evaluate, design, integrate, analyse information and communication technologies (ICT) to get traffic effectiveness, improve road conditions, conserve energy and time, provide security to passengers, drivers, pedestrians, etc.

The vehicular networking field has gone through smart challenging transitions in recent years. With this developing nature of safe, valuable and high-effective transportation, Vehicular Ad hoc Networks (VANETs) have transformed into intense innovation in smart transportation frameworks, where these frameworks are considered one of the powerful pillars of smart city [1]. In related with VANETs another broad area, which is getting more and more attractions in recent times is cloud computing. It is the integration of computing services, servers, networking, software, databases, storage, etc. The existing technology moving gradually towards the cloud and in the nearest future, there will be different ways to collaborate all through the Internet.

The vehicles assembled with devices (which supports wireless communication) allow the Vehicle-to-Vehicle

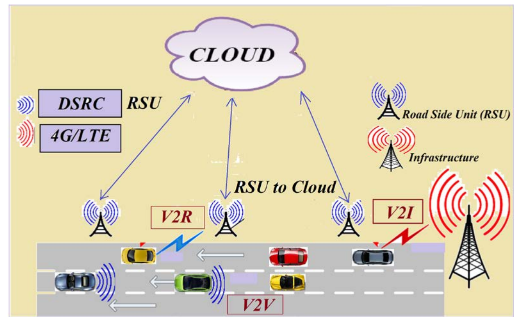

Fig. 1. An overview of VCC Communication Scenario

(V2V) and Vehicle-to-Roadside Unit (V2R)/Vehicle-to-Infrastructure(V2I) communications. It is done through Dedicated Short-Range Communication protocol (DSRC) standard [2], is the IEEE 802.11p standard for wireless communication. Whenever the vehicles are out of the communication range, the messages are forwarded by multi-hop communication. The OBU (On-Board-Unit) equipped inside every vehicle periodically broadcast traffic-related information, like direction, speed, real-time scenarios, warning messages, etc. Hence, the driver can get a clear idea of the driving environment. The integration of VANETs and Cloud computing derived an idea which can have the capability to take care of the transportation issues, traffic problems, etc., called Vehicular Cloud Computing (VCC). It is a technology which can deal with continuous message transfer with vehicles from anywhere, any time. An overview of VCC communication is shown in Fig.1.

The remaining of the paper is structured as follows. Section II reviews related works. We quoted the main motivations and contributions in Section III. In Section IV, we explain the network model, adversary model, and proposed scheme in detail. In depth security analysis is done in Section V. The hardware implementation is done in Section VI. In Section VII performance analysis is done with different perspectives such as communication cost, storage overhead, computation time, and energy consumption. Finally, we quoted the conclusion in Section VIII.

II. Related Works

Several methods proposed to ensure the security level in VCC. Recently, the concept of VCC has been defined and discussed in many research works. In 2013, Yan et al. [3] have done the analysis of different security challenges and privacy threats. Moreover, a security scheme also designed. However, it is resistant only to particular attacks. Sharma et al. [4] proposed an authentication scheme based on ECC (Elliptic Curve Cryptography) which satisfies security criteria such as privacy, confidentiality, integrity, etc. Moreover, the scheme is resistant to different types of attacks such as MIM (man-in-the-middle attack), replay attack, spoofing attack, etc. However, this scheme failed to ensure trust between vehicles. The extended three-party password-based authenticated key exchange (3PAKE) [5] is a scheme to deal with security concerns, i.e., high transmission cost, invalid service request, verification failures, DoS, and higher verification time. The main limitation of the paper is that they did not analyze different types of security attacks during the communication.

Moreover, we have done the literature review of recent authentication related research works on VANETs for doing performing analysis. Chuang et al.[6] proposed a lightweight mechanism called TEAM (Trust-Extended Authentication Mechanism) for V2V communication in VANETs. It uses the concept of a transitive trust relationship to improve the performance by reducing the storage spaces. Lack of an intruder detection system and the insider attack are the major limitations. In 2015, Li et al.[7] designed an authentication framework with conditional privacy-preservation and non-repudiation (ACPN) in VANETs. The main advantage is the reusability, which means that it can be used with other methods for performance improvements. However, the storage cost is comparatively high. In 2016, Wang et al. [8] designed a two-factor lightweight scheme (2FLIP) which removed overhead related with the CRL (Certificate Revocation List). However, the security of the scheme fully depends upon the system key from the certificate authority. MADAR [9] is a privacy-preserving authentication framework, which is resistant to denial-of-service attack. It consists of different identity-based signature schemes. However, the computation cost is very high. Li et al.[10] designed a dual authentication scheme (PPDAS) for V2V in IoV (Internet of Vehicles). However, the communication cost of the system is comparatively high.

III. Motivations and Contributions

We reviewed different types of research papers in which strong security mechanisms are implemented properly. However, these schemes take high computational time. It is the major criteria in a VCC system because of the highly dynamic nature of nodes in the vehicular system. Motivated by these reasons, we find that there is a need to have a protected lightweight data dissemination protocol for the VCC applications that can resist various attacks and provide the better user experience. Keeping focused on these aspects, we designed a lightweight scheme for message propagation for the VCC users. The main contributions included are a lightweight message propagation protocol with different cryptographic operations, authenticity between the entities involved, resistance to various known attacks, and hardware implementation on different platforms.

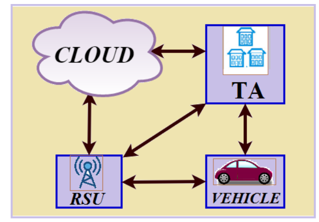

Fig. 2. Network Model

IV. Proposed Communication Protocol

In VANETs, the vehicles which are entering in one communication range communicate with the RSU and after some seconds it leaves this range. Then, it will not be able to communicate with the same RSU again at the same time because of dynamic nature. In this situation, if we are able to store some global warning messages (not going to change within a period of time such as a road work, very bad condition roads, heavy obstructions, etc.) then it will be available all the time. Here, the importance of the concept VCC comes. In our protocol, the vehicle which knows about the global warning message sends to the RSU and then to the cloud. Here, we are storing the data in the cloud where all the vehicles irrespective of the place can get the message from the corresponding RSU's because the RSU and cloud are connected through secure channel all the time.

A. Network Model

Here, a two-layer framework is considered, where the vehicles, RSU are in the low level and the TA (Trusted Authority) is in the high level (see Fig.2). The TA forwards application data to RSUs, and it works as gateways to deliver data to the lower layer (vehicles). At the communication time, whenever a vehicle requests to the nearest RSU, if the data is available then it forwards otherwise, it sends identity of the vehicle to the TA. The TA retrieves the parameters and forwards to the RSU. Then the RSU keeps one copy and performs the computations needed. After doing the computations, the RSU sends to the requested vehicle through DSRC. In our proposed protocol, the RSU acts as a high computational entity with some storage capacity.

B. Adversary Model

The network model is based on the assumption that, the computational capability of the RSU and the TA are higher than the OBU. An adversary is a person/system which can do some terrific actions, like making delay for transmissions, modifying the original messages, dropping the packets, false signature attacks, OBU compromising, etc.

C. Steps in the Proposed Lightweight Protocol

The different phases included in our protocol and its descriptions are mentioned below. Table I shows various notations used in our scheme.

TABLE I
FREQUENCY OF SPECIAL SYMBOLS

Notations	Descriptions
TA	Trust Authority
K_{TA}	The key of TA
VID_x	ID of the Vehicle x
VPW_x	Password of the Vehicle x
γ_x, ρ_x	Nonce generated by V_x
Υ_x	Nonce generated by TA for the vehicle, V_x
RID_z	Public identity of the RSU_z
K_{RSU}	Key of the RSU
ID_{RSU}	Unique identity of the RSU
σ	Nonce generated by the RSU
T_0	The RSU deployment time
$\mathscr{A}$	An attacker/hacker/adversary
$h()$	One way hash function
$\oplus$	EXOR Operation
$\|\|$	Concatenation operation

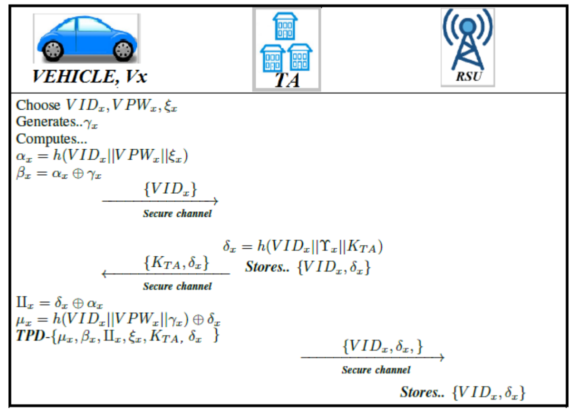

Fig. 3. Vehicle registration scenario

1) Initial Set-up:

The TA has a unique key, K_{TA} and all vehicles should be registered with TA. The TA maintains a list of identity (id) values of the vehicles, which is used at the time of registration for checking whether the vehicle is registered or not. Every RSU has a key K_{RSU} which is generated at the time of deployment. The value of K_{RSU} is calculated as, $K_{RSU} = h(ID_{RSU}||\sigma||T_0)$. Every RSU stores the values, K_{TA}, K_{RSU}, and a list of identity of vehicles with its δ_x value. The value of K_{RSU} is mainly used for the identification of RSU by the TA in any case of dispute.

2) Vehicle Registration:

The registration process is done at the time of purchase of a vehicle, because our network is ad-hoc and maintaining a TA all the time is not possible. The registration process is done through a secure channel by using the protocol like TLS (Transport Layer Security) [11] and communication done through an insecure channel. The steps required for vehicle registration scenario are as follows (see Fig.3):

- Every vehicle (say V_x) select VID_x and VPW_x. Then, V_x computes $\alpha_x = h(VID_x||VPW_x)$ and $\beta_x = \alpha_x \oplus \gamma_x$. The value VID_x sends to the TA through a secure channel.
- After receiving the value, TA generates a nonce value, Υ_x and computes $\delta_x = h(VID_x||\Upsilon_x||K_{TA})$. Then, the computed value δ_x will be sent to the vehicle through the secure channel.
- After receiving δ_x, vehicle computes the value of $\amalg_x = \delta_x \oplus \alpha_x$ and $\mu = h(VID_x||VPW_x||\gamma_x) \oplus \delta_x$. Then, the vehicle stores some values in the TPD (Tamper-Proof-Device), like $\{\mu_x, \beta_x, \amalg_x, \xi_x, \delta_x, K_{TA}\}$.

3) Authentication Module:

In the authentication module, we identify registered and unregistered drivers and take further action before starting the communication. The driver enters VID'_x and VPW'_x. The OBU computes $\alpha'_x = h(VID'_x||VPW'_x||\xi_x)$, $\gamma'_x = \alpha'_x \oplus \beta'_x$, $\delta'_x = \alpha'_x \oplus \amalg_x$ and $\mu'_x = h(VID'_x||VPW'_x||\gamma'_x) \oplus \delta'_x$. Then, the OBU checks the values μ_x and μ'_x. If the condition satisfies then the driver is authenticated. Otherwise, considered as malicious/fake driver.

4) Vehicle to RSU Message Transfer (V2R):

Whenever a vehicle knows about the situations such as heavy traffic, bad road conditions, and hazardous situations, it sends the information to nearest RSU. The V2R communication (see Fig.4) happens as follows.

- Before sending message the vehicle performs some computations like, $A_x = RID_z \oplus VID_x \oplus \rho_x \oplus \delta_x$, $B_x = \rho_x \oplus m_x$, $C_x = h(\rho_x||m_x||\delta_x||T_1)$. After that, the vehicle sends the values VID_x, A_x, B_x, C_x and T_1 to RSU.
- Initially, RSU checks the freshness of the message as, $\Delta T_1 \leq T2 - T1$, where $T2$ is the message receiving time and $T1$ is embedded with the message by the sending vehicle. If the value of $\Delta T1$ exceeds a threshold value, then RSU drops the message, otherwise continues the communication. The RSU checks whether the id value exists or not.

⋆ If the value exists, then the RSU checks the legitimacy of the message by doing the computations, such as $\rho'_x = A_x \oplus RID_z \oplus VID_x \oplus \delta'_x$, $m'_x = \rho'_x \oplus B_x$, $D'_x = h(\rho'_x||\delta_x||m'_x||T_1)$. If the values of D_x and D'_x are the same, then it is ensured that integrity of message is preserved and RSU saves the message, otherwise RSU sends the id of the vehicle, VID_x to the TA through the secure channel. The TA checks the id and sends the corresponding value of δ_x with time, T_3. The RSU receives δ_x, T_3 and saves the id and the corresponding δ_x value in its storage.

5) Batch Verification:

Whenever the RSU receives a message from a vehicle, V_x i.e., VID_x, A_x, B_x, C_x and T_1, it first checks the novelty of the message like $\Delta T_1 \leq T_2 - T_1$. If it exceeds a maximum threshold value, then rejects the message otherwise continues. Whenever the RSU receives many messages then it needs to check all these messages. In this scenario, we considered a batch verification process for minimizing the load of RSU. Let the RSU receives 'n' unique requests, which are denoted $\{VID_1, A_1, B_1, C_1$ and $T_{11}\}$,$\{VID_2, A_2, B_2, C_2$ and $T_{12}\}$,.........................,$\{VID_n, A_n, B_n, C_n$ and $T_{1n}\}$ sent from $Vehicle_1, Vehicle_2, ..., Vehicle_n$, it can verify the request's authenticity simultaneously by batch verification. Similar to single verification of one message, the RSU per-

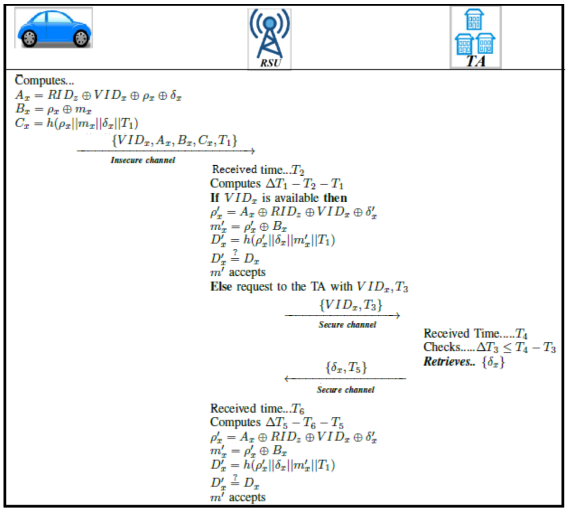

Fig. 4. Vehicle to RSU communication Phase

forms the following steps:

Step 1 : The RSU computes the values of ΔT_j and ΔT_k like $\Delta T_j \leq T_r - T_j$ and $\Delta T_k \leq T_l - T_k$ for verifying the novelty of the request, where T_r and T_l is the message receiving time of the RSU, T_j is the request sending time by the vehicle, and T_k is the sending time of the TA.

Step 2 : The Batch verification is done as follows:

• $\rho'_i = RID_z \sum_{i=1}^{n}(A_i \oplus VID_i \oplus \delta'_i)$.
• $m'_i = \sum_{i=1}^{n} \rho'_i \oplus B_i$
• $D'_i = \sum_{i=1}^{n} h(\rho'_i||T_i||m'_i||\delta_i)$.

6) Vehicle to RSU & RSU to Vehicle (V2R2V): Whenever a vehicle, say V_y wants to know about some warning messages which are recently stored in the RSU, V_y sends its identity, VID_y as a request to RSU. The RSU first verifies the legitimacy of V_y and then checks the freshness of the stored messages. After that, it sends the message to V_y. Since all the stored messages are warning message, it will be helpful to vehicles. Here, V_y is not requesting a particular query, rather than it is interested to receive any warning message that is helpful in his/her driving (see Fig.5).

• The vehicle, V_y makes a request to RSU, by sending its identity say, VID_y.

• After receiving request, the RSU performs the computations such as, $I_z = m_x \oplus K_{TA}$, $J_z = I_z \oplus m_x \oplus T_7 \oplus K_{TA}$, and $L_z = h(K_{RSU}||m_x||T_7)$. Then, it sends the values, I_z, J_z, L_z, T_7 to the requested vehicle, V_y.

• Upon receiving these values, V_y verifies the freshness of the message by the computations like $\Delta T7 \leq T_8 - T_7$, where T_4 is the message receiving time. If the value of ΔT_3 exceeds some threshold value then the message will be dropped, otherwise continues. The V_y first computes the value of message by, $m' = I_z \oplus K_{TA}$. Then, it recalculates the value of K_{RSU} and L_z like $K'_{RSU} = J_z \oplus I_z \oplus m'_x \oplus T_7$, $L'_z = h(K'_{RSU}||m'_x||T_7)$. If the value of L_z and L'_z are the same then it is ensured that message integrity preserved and V_y receives the message, 'm_x'.

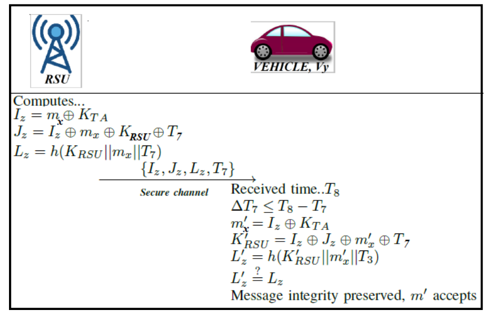

Fig. 5. V2R2V message transfer phase

V. Security Analysis

In this section, we justify how our approach is resistant to different types of attacks. Generally, there are some basic attacks, which are practised by the attackers to gain information or to access the vehicular system illegally.

A. Replay

Here, $\mathscr{A}$ makes delay or stop any request/response. In our proposed system, V_y, TA and RSU_z initially checks the validity of the received request from the sender like $\Delta T_1 \leq T_2 - T_1$, $\Delta T_3 \leq T_4 - T_3$, $\Delta T_5 \leq T_6 - T_5$, and $\Delta T_7 \leq T_8 - T_7$. If the values of $\Delta T_1, \Delta T_3, \Delta T_5$, and ΔT_7 exceeds a maximum threshold value then the entities drop the message. Hence, our proposed scheme is resistant to replay attack.

B. Modification

If an attacker, $\mathscr{A}$ wants to change data illegally, which are passed through the insecure channel then modification attack is possible. In our protocol, the parameters passed through the insecure channel are $VID_x, A_x, B_x, C_x, I_z, J_z, L_z, T_1, T_3, T_5$, and T_7. As we know VID_x, T_1, T_3, T_5, and T_7 are known to the public. The parameters C_x and L_z are the outputs of hash operation, where two of the parameters are unknown. Similarly, A_x, B_x, I_z and J_z are outputs of the hash functions, where two parameters are unknown. Hence, our system is resistant to the modification attack.

C. Password guessing

If $\mathscr{A}$ can obtain or know the correct password (VPW_x) by using a guessed password (VPW_j) with having common accessible variables, then a password guessing attack is feasible in the system. Let VPW_j be the guessed password. In order to know, whether it can be used as a password, $\mathscr{A}$ should know VID_x, VPW_x, and ξ_x. Here, if we assume that the $\mathscr{A}$ somehow guess the password, but again it needs to guess one more parameter, ξ_x. As previously mentioned guessing two parameters in polynomial time is not possible. Thus, the recommended scheme can withstand a password guessing attack.

D. Man-in-the-middle

In this type of attack, $\mathscr{A}$ enters into a communication between the entities, impersonates both entities and gets access to the data. It allows the $\mathscr{A}$ to accept and drop information meant for someone else. The $\mathscr{A}$ gets the parameters, such as VID_x, A_x, B_x, C_x and T_1. Here, VID_x and T_1 are public. The value of A_x is calculated as $A_x = RID_z \oplus VID_x \oplus \rho_x \oplus \delta_x$. In this scenario, VID_x and RID_z are known to all. However, the $\mathscr{A}$ cannot be able to guess the two parameter values ρ_x and δ_x, because ρ_x is the nonce value generated by V_x and δ_x is the output of the hash operation. Hence, it is clear that our proposed protocol can withstand the man-in-the-middle attack.

E. Plain Text Attack

In our protocol, the message m_x is used only for calculating the values of B_x and C_x. The parameters, B_x and C_x are computed using cryptographic operations ($h()$, $\oplus$, $||$). The $h()$ is an irreversible function therefore getting back the value is not possible. Additionally, the vehicle (V_x) does not send the message m_x in a simple form to other vehicles. Thus, $\mathscr{A}$ does not get any information to derive the message. Hence, our proposed protocol is resistant to a plain-text attack.

F. Impersonation

If an adversary, $\mathscr{A}$ is interested to access privileged services behalf of other users, then he or she should generate a valid login request. If $\mathscr{A}$ succeeds in this process then, a user impersonation attack is possible. In our protocol, m_x is the warning message send from either V_x to RSU_z or RSU_z to V_y. Then, the checking process happens like $D_x \stackrel{?}{=} D'_x$ and $L'_z \stackrel{?}{=} L_z$. If the message is not sent by a registered vehicle (fake message) then it will not match. Additionally, if $\mathscr{A}$ thinks to re-use any identity, then also she/he fails to impersonate a legitimate user because this request includes time-stamp and it is valid only for a limited period. Hence, it is clear that the proposed system is resistant to a user impersonation attack.

G. Collision Attack

In a collision attack, $\mathscr{A}$ tries to find two inputs generating the same hash value. As we know, $SHA-1$ is susceptible to collision attack therefore we used $SHA-2$ for all our computations. In our proposed scheme, the values α_x, μ_x, C_x, D_x, δ_x and L_z are outputs of the $SHA-2$ hash operations. Hence, our proposed scheme is resistant to collision attack.

VI. Hardware Implementation

As we know, the hardware implementation provides a better understanding of the results. In order to verify the theoretical results, we implemented our scheme on dedicated hardware devices.

A. Simulation On a Single Desktop Device (DC)

The system parameters are $Intel(R)\ Core(TM)\ i5-7200u\ CPU\ @\ M370$, 2.50 GHz , $8GB\ RAM$ with $64-bit\ Windows$ 10 $operating\ system$. The major operations are one-way hash function ($T_h()$), public key encryption/decryption (T_{PE}, T_{PD}), signing operation (T_{SG}), signing verification (T_{SV}), multiplication (T_{ML}), division (T_{DV}), addition (T_{AD}), MAC (T_{MA}), exponentiation (T_{EX}), and symmetric (AES) encryption/decryption (T_{AE}, T_{AC}). We executed all operations around 100 times and found that the time required for $||$ and $\oplus$ is negligible as compared with any other cryptographic operations. In this specification, T_{PE}, T_{PD}, $T_h()$, T_{EX}, T_{ML}, T_{AD}, T_{DV}, T_{AE}/T_{AC}, T_{MA}, T_{SG}, and T_{SV} takes 4.4063, 7.7613, 0.0020, 0.0399, 0.0268, 0.0017, 0.0012, 0.0100, 0.00967, 24.8351, and 1.8235 (in milliseconds) respectively.

B. Simulation On Raspberry Pi (RP)

In our second set up, we used RP ($BCM2708$ - $ARMv6$ - $compatible\ processor\ rev$ 7 and $8GB\ SD$ card) to simulate the IoT devices that wish to communicate with each other. We have used SHA-2 module present in the $hashlib$ module for $python$ 3.6 as our secure cryptographic hash function. Moreover, we executed each operation 100 times on the RP. In this specification, T_{PE}, T_{PD}, $T_h()$, T_{EX}, T_{ML}, T_{AD}, T_{DV}, T_{AE}/T_{AC}, T_{MA}, T_{SG}, and T_{SV} takes 866.733, 2686.533, 0.1739, 0.2448, 0.2115, 0.1736, 0.255, 1800.000, 0.1739, 709.149, and 170.574 (in milliseconds) respectively. The RP set up with a monitor is shown in Fig.6. It is imperative that the computation cost will vary according to the hardware capabilities, as demonstrated by the experiment.

VII. Performance Analysis

The performance evaluation can be done by different aspects, i.e., communication cost, storage overhead, computational time, and energy usage. In order to make the system lightweight, we used only simple operations at the same time, maintained higher security than existing schemes. The hash operation is the main function used, $h()$, more specifically SHA-2 with 256 bits (32 bytes -$E_{h()}$). Usually, a normal variable or an identity (E_{ID}), public key encryption/decryption (E_{PE}, E_{PD}), time-stamp (E_{TS}), AES encryption/decryption (E_{AE}, E_{AC}), signature (sign - E_{SI}, verify - E_{SV}), multiplication (E_{ML}), division (E_{DV}), addition (E_{AD}), and message (E_{MG}) expects 10, 128, 8, 16, 43 , 10, 10, 10, and 100 bytes respectively.

A. Communication Cost and Storage Overhead

The communication cost refers to the number of parameters (bytes) needed for transferring a single message. The RSU performs the needed action either itself or with the help of the TA (if the identity is not available in the RSU). Hence, we considered both cases and named as RSU hit (data is available in the RSU) and RSU miss (data not available in the RSU). Our scheme requires 310 bytes - RSU hit and 368 bytes - RSU miss for communication, whereas the schemes

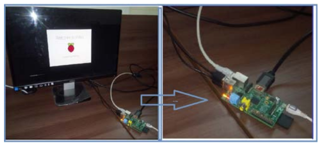

Fig. 6. Raspberry Pi set up with a monitor

TABLE II
COMMUNICATION AND STORAGE COST COMPARISON

Scheme	Communication cost	Storage cost
TEAM[6]	$18E_{h()}$	$6E_{h()}$
ACPN[7]	$10E_{ID}+7E_{TS}+7E_{PE}+4E_{MG}+7E_{SI}$	$5E_{PE}+E_{PD}$
2FLIP[8]	$E_{MG}+E_{TS}+E_{h()}$	$13E_{ID}+10E_{PE}+10E_{SI}+8E_{TS}+5E_{MG}+E_{h()}$
MADAR[9]	$13E_{ID}+10E_{PE}+10E_{SG}+8E_{TS}+5E_{MG}+E_{h()}$	$2E_{PE}$
PPDAS[10]	$3E_{ID}+6E_{TS}+17E_{h()}+6E_{PE}+4E_{ML}+3E_{MG}$	$5E_{ID}+7E_{h()}+ES_{TS}$
Proposed RSU Hit	$3E_{ID}+2E_{TS}+2E_{h()}+2E_{MG}$	$1E_{ID}+5E_{h()}$
Proposed RSU Miss	$4E_{ID}+4E_{TS}+3E_{h()}+2E_{MG}$	$2E_{ID}+6E_{h()}$

TEAM[6], ACPN[7], 2FLIP [8], MADAR[9], and PPDAS[10] requires 1152, 1393, 172, 2436, and 1730 bytes respectively. The storage cost mainly refers to the total needed memory to save different parameters. Our protocol maintains low storage overhead by storing fewer parameters in the OBU, RSU, and TA. Our scheme requires 170 bytes (RSU hit) and 210 bytes (RSU miss) for storing parameters, whereas the schemes TEAM[6], ACPN[7], 2FLIP[8], MADAR[9], and PPDAS[10] requires 384, 768, 492, 256, and 282 bytes respectively (see Table II).

B. Computational Time and Energy Consumption

The computational time is the total number of cryptographic operations required for login, authentication, and communication phases. The total execution time (in milliseconds) of competitive schemes on DC and RP is shown in Table III. The energy consumption of our scheme can be calculated as $E_{energy} = E_{CT} * E_{CP}$ and it is measured in millijoules (mJ). Here, E_{energy} is the total energy consumption power, E_{CT} is the total computational time required, and E_{CP} is the maximum CPU power. The general value of E_{CP} for wireless communication networks is 10.88 Watt [12] on DC and 2.5 Watt for RP [13].

TABLE III
COMPARISON OF COMPUTATION TIME AND ENERGY CONSUMPTION VALUES ON A DC AND RP

Scheme	Parameters	CT in DC (ms)	CT in RP (ms)	EC in DC (mJ)	DC in RP (mJ)
TEAM[6]	$22T_{h()}$	0.1792	9.3172	1.9497	23.293
ACPN[7]	$2T_{PE}+5T_{SG}+4T_{SV}+2T_{ML}+T_{DV}+T_{AD}$	140.3380	5962.3601	1536.8774	14905.9
2FLIP[8]	$7T_{h()}+T_{MA}$	0.0292	1.3912	0.3177	3.478
MADAR[9]	$12T_{SG}+12T_{SV}+2T_{ML}+T_{DV}$	319.958	10557.5264		
PPDAS[10]	$11T_{ML}+16T_{h()}+2T_{EX}+2T_{AE}+T_{PE}+T_{AC}+T_{PD}$	12.6036	8958.8708	137.1271	22397.18
Proposed	$6T_{h()}$	0.012	1.0434	0.1305	2.609

CT - Computation Time, ***EC*** - Energy consumption, ***DC*** - Desktop Computer, ***RP*** - Raspberry Pi, ***ms***- milliseconds, ***mJ*** - milliJoules.

VIII. CONCLUSION

We proposed a secure data transmission scheme for smart transportation in VCC where the vehicles, RSU, and the TA are involved. The proposed lightweight scheme uses only hash functions and maintains better security than previous schemes, TEAM[6], ACPN[7], 2FLIP[8], MADAR[9], and PPDAS[10]. The importance of cloud comes when the vehicle leaves from one communication range of RSU and enters another communication range of different RSU. In this scenario, some global information can be stored in the cloud and can be accessed from anywhere irrespective of the communication range. The implementation results on a DC and RP show that the computation time needed for RP is higher than that of a DC because of its limited processing power. The lightweight nature allows our scheme to easily incorporate into the OBU of every vehicular communication systems. This reveals that the suggested scheme can be practised to transfer data in the VCC scenario in a secure and efficient manner.

ACKNOWLEDGEMENT

This work is fully supported by Department of Science and Technology (DST), Govt. of India, Project Number: ECR/2015/000256.

REFERENCES

[1] S.Sultan, M. M. Al-Doori, H.Al Bayatti, and H. Zedan, *A comprehensive survey on vehicular ad hoc network*, Journal of network and computer applications 37, 380-392, 2014.

[2] J. B. Kenney, *Dedicated short-range communications (DSRC) standards in the United States*, Proceedings of the IEEE, 99(7), 1162-1182, 2011.

[3] G. Yan, D. Wen, S. Olariu, and M. C. Weigle, *Security challenges in vehicular cloud computing*, IEEE Transactions on Intelligent Transportation Systems, 14(1), 284-294, 2013.

[4] M. K. Sharma, R. S. Bali, and A. Kaur, *Dyanimc key based authentication scheme for Vehicular Cloud Computing*, IEEE International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 1059-1064, October 2015.

[5] R. Muthumeenakshi, T. R. Reshmi, and K. Murugan, *Extended 3PAKE authentication scheme for value-added services in VANETs*, Computers and Electrical Engineering 59, 27-38, 2017.

[6] M. C. Chuang, and J. F. Lee, *TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks*, IEEE systems journal, 8(3), 749-758, 2014.

[7] J. Li, H. Lu, and M. Guizani, *ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs*, IEEE Transactions on Parallel and Distributed Systems, 26(4), 938-948, 2015 .

[8] F. Wang, Y. Xu, H. Zhang, Y.Zhang, and L. Zhu, *2FLIP: a two-factor lightweight privacy-preserving authentication scheme for VANET*, IEEE Transactions on Vehicular Technology, 65(2), 896-911, 2016.

[9] C. Sun, J. Liu, X. Xu, and J. Ma, *A privacy-preserving mutual authentication resisting dos attacks in VANETs*, IEEE Access, 5, 24012-24022, 2017.

[10] Y. Liu, Y. Wang, and G. Chang, *Efficient privacy-preserving dual authentication and key agreement scheme for secure V2V communications in an IoV paradigm*, IEEE Transactions on Intelligent Transportation Systems, 18(10), 2740-2749, 2017.

[11] T. Dierks, and E. Rescorla, *The transport layer security (TLS) protocol version 1.2*,No. RFC 5246, 2008.

[12] D. He, C. Chen, S. Chan, and J. Bu, *Secure and efficient handover authentication based on bilinear pairing functions*, IEEE Transactions on Wireless Communications, 11(1), 48-53, 2012.

[13] J. Kiepert, *Creating a raspberry pi-based beowulf cluster*, Boise State University, 1-17, 2013.