International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 2, Mar-Apr-2019, ISSN (Online): 2395-566X

Providing Security and Privacy for communication in Vehicular Cloud Computing

Patil Nilochana
Pati17cse@cmrit.ac.in
Department of Computer Science and Engineering
C.M.R Institute of Technology
Bangalore, India

P. Kavitha
kavitha.p@cmrit.ac.in
Department of Computer Science and Engineering
C.M.R Institute of Technology
Bangalore, India

Abstract- **Vehicular cloud computing (VCC) uses unused vehicle storage and computing resources to collaboratively provide end users such as drivers and passengers with traffic management, road safety and infotainment services. It is a hybrid technology that enhances the use of vehicle resources and is capable of performing complex computing tasks that a single vehicle cannot handle. Despite the appealing benefits of sharing resources among unfamiliar vehicles, there are severe threats to security and privacy in VCC. In this paper, we identify security goals for VCC interoperability and provide an authentication key agreement (AKA) framework for VCC. Specifically, to design a reliable AKA with strong safety guarantees for VCC, we present the research challenges and open issues first. Then we propose a secure and efficient message access control and authentication scheme in VCC. For message sharing, a hierarchical, Cipher Policy Attribute Based Encryption (CP-ABE) technique is used to ensure that vehicle whose persistent or dynamic attributes satisfy access policies can access the broadcast message with equipped cloud. Finally, to illuminate the further directions of research on AKA for VCC, several interesting issues are discussed.**

Keywords **– Vehicular Cloud Computing, Message access control, CP-ABE, Authentication Key Agreement.**

I. INTRODUCTION

A vehicular ad hoc network (VANET) allows vehicles equipped with sensing, communication and networking capabilities to communicate with each other for the exchange of information Vehicle to Vehicle (V2V) or roadside infrastructure Vehicle to Infrastructure(V2I) [1]. It can provide a variety of promising applications, ranging from safety applications (such as accident reporting, congestion warning, and anomalous vehicle behavior warning) to non-safety applications (such as infotainment, smart parking, and advertising) [2].

Despite the phenomena-al growth of vehicle applications supports drivers and passengers, a wide range of onboard capabilities is chronically underused. Cloud computing is used to revolutionize vehicular communications and applications in order to make full use of idle resources and increase vehicle capacity, leading to vehicular cloud computing (VCC) [3]. VCC is a hybrid platform with a significant impact on road safety and traffic management by making full use of idle vehicle resources such as computing and storage of decisions [4].

By embedding idle resources, a group of vehicles in a parking garage or on the road can create a cloud to collect information, collaboratively process data and make decisions to improve service quality for drivers and passengers alike. Travelers, for example, usually park their cars in parking lots at the airport while travelling. The airport can use vehicle computing resources to create an end-user on-demand access parking garage data center. Similarly, congested vehicles can generate a computing pool to perform complex simulations to remove congestion by scheduling traffic lights. End users can access the formed cloud at the right time and get the flexible resources on demand at a reasonable cost. The Vehicular Cloud (VC) provides an efficient way for end-users to use excess vehicle resources that are coordinated and dynamically assigned. A VC is generally dynamic and temporary due to vehicle mobility. The temporary VC is an important addition to the conventional cloud (CC) to improve the storage and computing capabilities of end users. It is expected that VCC will be able to enrich various vehicle applications and services and improve driving and riding experiences such as crowd sensing of vehicles [5], downloading video streams and traffic monitoring [6].

Because of the unprecedented growth of smart phones, they can serve as important interfaces between drivers and external networks, as well as terminals responsible for storing and retrieving VCC information. A smart phone, for example, uses biosensors to senses the physiological conditions of the driver and for smart processing sends the data collected to VCC. In case of an accident or hazard, the warning messages could be pushed to the smart phone for alarm [7]. With the involvement of smart phones, the attractive applications supported by VCC become scalable, upgradable and inexpensive to deploy.

While VCC is expected to play an increasingly important role in smart transport systems, interconnectivity inevitably presents drivers with challenges and risks, one of which is safety [8]. If countermeasures are insufficient, smart phone cloud communications are vulnerable to malicious attacks, resulting in a range of VCC applications damage. First, sensitive and critical messages may be affected by unauthorized access.

© 2019 IJSRET

Second, any conduct or behavior that is malicious, such as modifying and replaying attacks on disseminating messages, may be fatal to data owners. Therefore, preventing adversaries from compromising VCC systems and protecting data exchanged between users and clouds (i.e., CC and VC) is essential in order to ensure security and reliability of data. Mechanisms for authentication and key agreement (AKA) are essential and effective in resisting malicious attackers and ensuring access to data in VCC.

Not only does AKA prevent unauthorized users from illegally accessing data at rest, it also allows entities to negotiate session keys for integrity and confidentiality of data transmission. Although some state of the art, AKA protocols [9, 10] have been proposed, it is difficult to migrate these protocols to VCC scenarios due to the characteristics of VCC architecture. In particular, a VC is a dynamic and temporary group of vehicles; therefore, it is difficult to establish trust relationships between them. It is therefore important to design AKA protocols to ensure the authenticity of VCC vehicles.

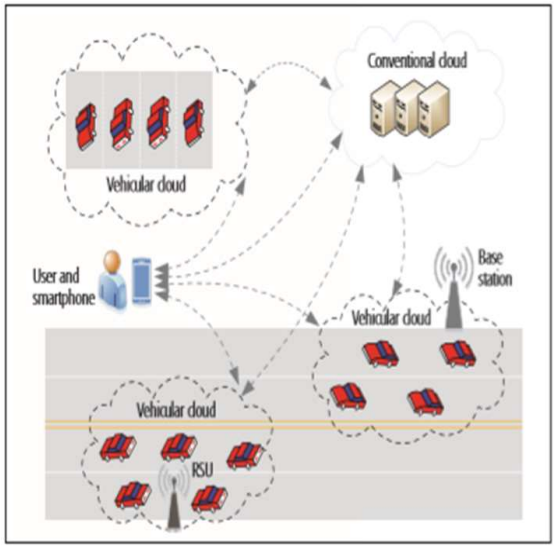

Fig. 1 Architecture of vehicular cloud computing.

In this paper, we aim to provide a comprehensive overview of the design of an effective AKA protocol for VCC and its safety goals. The challenges and design goals of VCC's secure AKA protocols are demonstrated first in particular. Then we propose the integrated AKA framework to achieve mutual authentication and secure communication between users, VC, and CC by integrating the AKA 3-factor single-server protocol (SS-3FAKA) and the key establishment protocol based on non-interactive identity.

The performance is evaluated to demonstrate the efficiency of the new framework. Finally, we are presenting several promising directions for research in order to encourage more efforts on secure VCC.

II. RELATED WORK

Figure 1 shows the conventional cloud (CC) and vehicle cloud (VC) VCC architecture [11]. CC provides outsource-d computing and storage services for end-users with on-demand resource deployment and omnipresent service access. It can be either a public cloud service such as icloud, Aliyun, and Amazon EC2, or an authority-deployed private cloud such as the Department of Transportation and automotive manufacturers [7]. The CC is responsible for storing various traffic information collected from vehicles, performing outsourced data processing tasks for end-users, and providing on road vehicles with a variety of services such as navigation, infotainment, content distribution, and driving support.

VCs are formed on the streets and roads by a group of vehicles and in parking lots that have plenty of computing resources to deliver services. Similar to idle CC computing, vehicle owners agree to rent out their onboard excess resources, which through V2V and V2I communications are integrated into a powerful VC. All VC vehicles form a temporary, dynamic computer and storage resource pool such as an airport data centre, data cloud parking, and on road vehicle driving cloud to provide data maintenance and processing services including traffic light scheduling, traffic information storage, and frequent congestion relief services. End users are able to access their collected traffic data and outsource computational tasks on VCs. In addition, base stations and roadside units (RSUs) are used to support VC management and VC and CC communication.

Users can access the services and resources of VCC such as navigation services and video download apps using their smart phones. The smart phones are connected via cellular communications to the Internet and thus benefit from CC's cloud services. Because of mobility, users can access multiple VCs to enjoy various services (e.g. video stream downloading and traffic query). Also important sensing devices are smart phones that collect and deliver traffic and parking information to VCs for local navigation services. The exchange of communication between VCs and smart phones via cellular networks enhances driving safety and enriches driving experiences.

VCC Services: Interactions between users, VCs and CCs can support a range of user-friendly applications and services for vehicles [4]. Information as a Service: VCC can provide users with information about safety driving, such as road conditions, advanced warnings, traffic conditions and alerts for accidents. These services are recognized as information about the service together. Computing as a Services Drivers can use their smart phones to access computation-intensive applications,

© 2019 IJSRET

such as finding the shortest route to a destination and discovering restaurants on roads. VCC can provide computing services to users due to the limited power of smart phones. Storage as a Service.

While storage space for smart phones continues to grow, users may need additional storage to perform their tasks. Turning to VCs is natural, adding storage capacity for vehicles and providing storage space as a service for users. Network as a Service: While a smart phone has its own network connectivity, for various reasons such as low cellular coverage in rural areas or high-speed movement, its owner may be disconnected. In this case, the owner would rent to the Internet, access network resources from VCs.

VCC inherits security and privacy issues from both VANETs and cloud computing as a cloud computing and VANET integrated architecture. In each paradigm, security and privacy issues have been discussed. Integrating VANETs and cloud computing, however, triggers new security and privacy issues that have not attracted enough attention.

Security Challenges of AkA for Vcc: The core security and privacy challenges faced by VCC include authentication of vehicle identity, key management, vehicle data privacy, and location privacy [8]. To address these challenges, a large number of security solutions have been proposed [8]. Nevertheless, the involvement of multiple VCs requires more research effort to enable users to securely access VCC services.

Secure Access for VCC: AKA is key to making access to VCC safer and more pleasant. To achieve identity authentication, the provider must demonstrate its possession of any private information or an identifiable feature in an identity authentication system. Authentication techniques for identity can be divided into three categories: What you know: password, personal identification number (PIN), security issues, etc. • What you have: smart card, token, credential, etc. • What one is: biometrics. Each factor is widely known to have its own merits and demerits. Specifically, the password can be brute forced, spying on, or even socially engineered. By side channel attacks, smart cards may be lost, stolen, or revealed. Biometrics are at risk of duplication. For example, it is possible to lift fingerprints left somewhere and use them to gain illegal access. In addition, biometrics cannot be updated as they are individual biological characteristics. Secure access to VCC with a strong security guarantee becomes essential and challenging for users as security and privacy threats in VCC can directly result in security issues.

Secure Authentication to Multi-VCs: Because of the mobility of vehicles, a vehicle would form various VCs with different vehicles in different regions, making these VCs dramatically dynamic. User's authentication credentials should be kept in each vehicle to ensure normal user service access, leading to heavy overhead storage for vehicles. In addition, maintaining authentication credentials becomes extremely inefficient with user mobility. In particular, joining and revoking users will result in a heavy burden on authentication credentials management. As a result, being authenticated by different VCs is quite difficult for a user. The way to maintain identity consistency in multiple VCs during authentication is quite challenging as each user can access multiple VCs to acquire different services. On the contrary, if a user uses an authentication credential to access multiple VCs, a major concern becomes the user's risk of tracking service. Therefore, the management of authentication credentials should be considered in VCC during authentication to multiple VCs. Multi-factor AKA: Due to the inherent weaknesses of each authentication factor, as discussed above, single-factor authentication schemes cannot provide stronger security guarantee required by security applications. To increase the number of features or factors required in authentication schemes is a straightforward approach. Thus, it becomes vital for users to access VCs to design a 3-factor AKA scheme. The existing 3-factor AKA protocols can be classified into two categories, depending on the number of servers to be accessed: SS-3FAKA [9] and multi-server 3-factor AKA (MS-3FAKA) [10].

The SS-3FAKA schemes cannot be used directly in VCC, where many VCs act as service providers to provide a multitude of services as a user has to register repeatedly with each VC. Repeated registration is not only a waste of time and energy for users, but it also places additional burden on users to maintain multiple sets of security credentials. As a distributed computing environment, VCC is a multi-server architecture that requires a user to authenticate to more than one VC, thus requiring MS-3FAKA protocols to ensure secure communications. A MS-3FAKA protocol [10] has been proposed to eliminate repeated registration in order to reduce the burden of users and ensure secure communications. However, each authentication session still requires an online registration authority to carry out mutual authentication. In addition, when accessing the VCs, the user is required to repeatedly present authentication credentials to VCs.

In order to avoid time consuming and resource-consuming interaction with online registration authority, multiple MS-3FAKA protocols without online registration authority were proposed using public key-based cryptosystems [12, 13]. However, one common drawback in these schemes [12, 13] is that users have to handle the management of the public keys of different VCs, indicating that each user must maintain an up-to-date public key revocation list to revoke compromised VCs. While many 3 factor AKA protocols have been

© 2019 IJSRET

developed for various service scenarios, none is in the VCC context or corresponds to VCC requirements. As a result, VCC's 3 factor AKA has not been addressed yet and deserves more attention.

III. SYSTEM ARCHITECTURE

We are proposing our integrated AKA framework in this section to address the security challenges in VCC. System Overview: We propose an integrated AKA framework for VCC to achieve the above safety goals, in which there are three roles: Ui, VCj, and CC. CC acts as a trusted third party and is responsible for setting system parameters, generating VCj's private keys and issuing a smartcard to every registered Ui user. During the registra tion-on phase, all users and VCs build trust relationships with CC. Ui can access multiple CC and VCj services. The user must present the necessary credentials to authenticate to CC in order to access the services offered by CC. Ui can request a ticket for the service offered by VCj from CC after completing the CC authentication, which creates a ticket for the user without interacting with VCj. Then Ui can directly access VCj with the ticket obtained without additional CC involvement. Our integrated framework contains three key ingredients: a non-interactive key established protocol based on identity, the CP-ABE AKA protocol, and the ticket concept. The non-interactive identity-based key establishment protocol [14] is used to establish a shared key between two clouds in a non-interactive manner, which in turn facilitates a service ticket distribution. To ensure secure access to CC, the CP-ABE AKA protocol can achieve a high level of security. In addition, the concept of a ticket is leveraged to allow single sign-on, so that a user can securely access multiple VCs with only one set of CC registered security credentials without repeated VCj registration.

Our framework is suitable for VCC, as users only need to register with CC and are flexible to access multiple dynamic and temporary VCs. In addition, from the perspective of a user, our framework hides the complexity of public key management. Our integrated AKA framework is designed on the basis of a CP-ABE AKA protocol. A typical CP-ABE AKA protocol [9] consists of two entities, user U and remote server S, comprising five phases: Initialization, registration, login and authentication, change of password and biometrics and re-registration[9].

Initialization (CP-ABE AKA.Init): S selects system parameters, generates the public-private key pair and publishes the public parameters. Registration (CP-ABE AKA.Reg): U selects the identifier ID and password PW when U registers on S, provides the biometric sample B and sends a value derived from ID and (PW, B) to S via a secure channel. S issues a smart card that stores the ID's secret key. U stores the card's related values after it has be-en obtained. Login and Authentication (CP-ABE AKA. Auth): U attaches and enters the smart card (ID, PW, B) to a terminal. The terminal interacts with the card and sends MSG1 to S for the login request. S monitors U's legitimacy and sends an MSG2 login response to U. Password and Biometric Change (CP-ABE AKA. Change): U periodically updates the password and biometric sample. Revocation and Registration (CP-ABE AKA. Revocation): Without changing his / her identity, U revokes his / her account and re-registration.

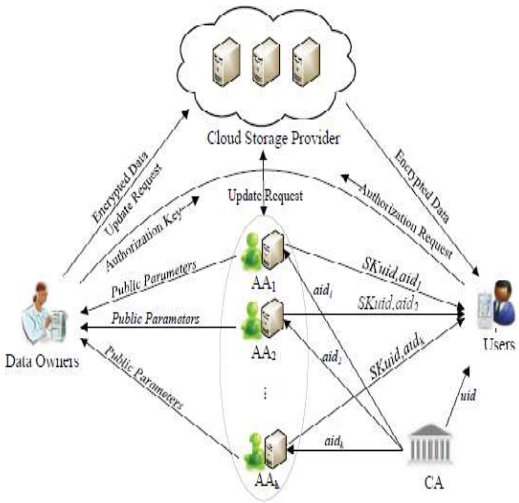

Fig. 2: System Model of TFDAC - MACS

IV. IMPLEMENTATION OF INTEGRATED AKA FOR VCC

Our framework consists of eight phases: system setup, VC registration, user registration, CC user authentication, request for tickets, VC user authentication, change of password and revocation of smart cards as shown in Fig. 2. CC first selects its private master key and public parameters in the system setup phase, calculates the corresponding public key, and publishes its public key and public parameters. VCj submits its identity as the public key in the VC registration phase and obtains the corresponding private key generated by CC. CC issues a smart card to Ui during the user registration phase. The CP-ABE AKA protocol is executed between Ui and CC during the authentication phase between Ui and CC. In the ticket request phase, Ui can request a service ticket from CC via the established secure channel. Ui presents the ticket to VCj in the authentication phase between Ui and VCj

© 2019 IJSRET

to set up a secure channel between Ui and VCj. Finally, it is possible to invoke the password change and smart card revocation phases to change the user password and revoke a lost / stolen smart card without changing the user identity requirement.

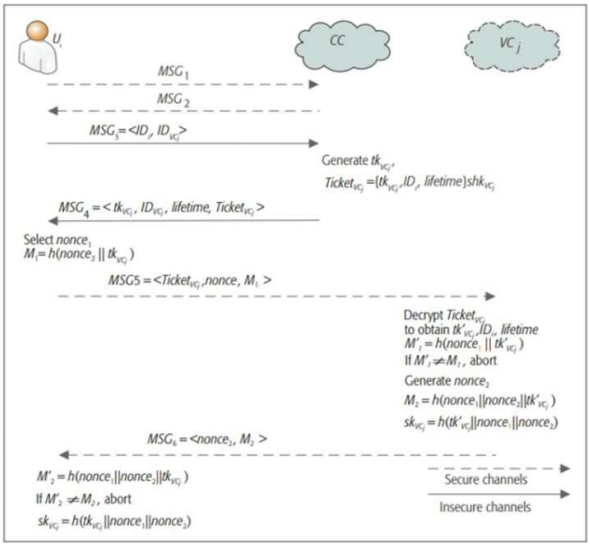

Fig. 3 Flow chart of proposed framework.

1. System Setup-

The first steps in CP-ABE AKA.Init are initialized by CC. CC then generates (p, G1, G2, e), where G1 is a cyclic additive group of prime order q, P is a generator of G1, and G2 is a cyclic multiplicative group of the same order, e: G1 below G1 belongs to a bilinear pairing [14], and p is a large prime group. CC picks s = Zp, outputs its master secret and public key pair (s, S = sP), and publishes the public key parameters (P, S, H1, H2, h), where H1:{0, 1 } * = G1 serves as identity mapping function, H2:G2 { 0, 1}l, h:{0, 1 } * { 0, 1}l, and l is the bit length of the hash function output (e.g., l = 160). CC also selects an IDCC identity and calculates its KCC = sH1 (IDCC) secret key.

2.VC Registration-

To obtain an identity based public and private key pair, e ach VC registers with CC as follows: • VCj first sends the IDVCj identity to CC in order to build the trust relationship with CC. • CC calculates kVCj = sH1 (IDVCj) (i.e. VCj private key) and shkVCj = H2 (e (KCC, H1(IDVCj))) = H2 (e (H1 (IDCC), H1 (ID VCj)) s) (i.e. non-interactively established key between CC and VCj). Then CC will send kVCj via a secure channel to VCj. • VCj checks if e(S, H1(IDVCj)) = e(P, kVCj) holds when it receives its private key. If so, VCj will calculate shkVCj = H2 (e(H1(IDCC), kVCj)) = H2(e(H1(IDCC), H1(IDVCj))s), and store k VCj and shk VCj in the safe memory. Note that shk VCj can be used to build secure communication between the CC and VCj and to achieve distribution of tickets without VCj interactive. User Regi strat-ion: The operations are the same as in CP-ABE AKA. Reg during this phase.

User CC authentication: The operati-ons in this phase are the same as those in SS-3FAKA.Auth, and after the operations the session key skCC is shared between Ui and CC. Ticket Request: Ui may request a ticket from VCj via a secure channel after Ui has been authenticated by CC, as shown in Fig. 3. • Step 1: Ui sends MSG3 = < IDi, IDVCj > to CC for the first time. • Step 2: Upon receiving the application, CC generates a temporary key tkVCj, defines the ticket lifetime validity period and TicketVCj = {tkVCj, IDi, lifetime} shkVCj, and sends MSG4 = < tkVCj, IDVCj, lifetime, TicketVCj > to Ui. Here, {M} K denotes message M cipher text encrypted with key K. User-VC authentication: Ui can use the ticket obtained to authenticate VCj (Fig. 3):

- Step 1: Ui generates a single nonce1 random number and calculates M1 = h (nonce1) and sends the MSG5 = < TicketVCj, nonce1, M1 > ticket authentication request to VCj.
- Step 2: VCj will decrypt the ticket (tk'VCj, IDi, lifetime). VCj first checks the validity of the ticket. If expired, VCj will reject the request for authentication; otherwise, M'1 = h (nonce1) will be calculated. If M'1 = M1, this session is aborted by VCj ; otherwise, Ui may access its resources and services. VCj generates M2 = h (nonce1) and t nonce2.
- Step 3: M'2 = h(nonce1 tkVCj) is calculated. If M'2 = M2, Ui calculates skVCj = h(tkVCj), which is the session key between Ui and VCj, otherwise this session is terminated by Ui. The following phases (i.e. password and biometric change and revocation and re-registration) are the same as in the SS-3FAKA protocol.

V. PERFORMANCE EVALUATION

We assess the execution of our system as far as computat io-nal and correspondence overhead.

1. **Encryption-** The Enc (M, T, PK) algorithm is a randomized algorithm that takes the message to be encrypted (M) as an input, the access structure T that needs to be met, and the public parameters (PK) for the CT ciphertext output. We can say that the encryption algorithm embeds the ciphertext access structure so that only those users with T-compliant attributes can decrypt and retrieve it.

2. Decryption- The Dec (CT, SK, PK) decryption algorithm takes the ciphertext CT, the user secret keys SK and the public parameters PK as its input, and it sends out the encrypted message (M) if and only if the A attributes embedded in SK satisfy the access structure T used to encrypt the ciphertext CT, i.e. if T (A) = 1 then sends out the message M otherwise, it sends out.

© 2019 IJSRET

We discuss the efficiency of our scheme in terms of message encryption time and message decryption time. Fig. 4 shows the computation cost of message encryption time and Fig. 5 shows the computation cost of message decryption time. For the message decryption phase, the vehicles use policies to decrypt the encrypted message recursively, In any case, the correspondence cost of our structure can be significantly diminished as the validation between a user and CC, which just should be executed once in a generally extensive stretch. In synopsis, our structure has better effectiveness in calculation and correspondence than the existing schemes [10, 12, 13].

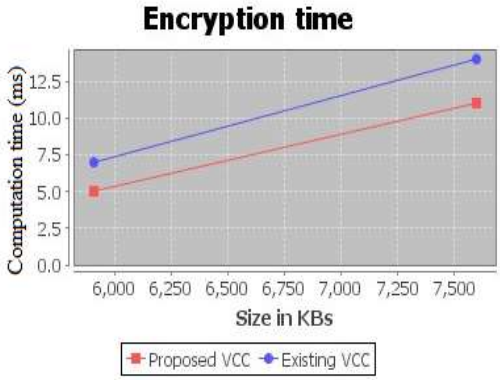

Fig. 4: Computation cost of message encryption time

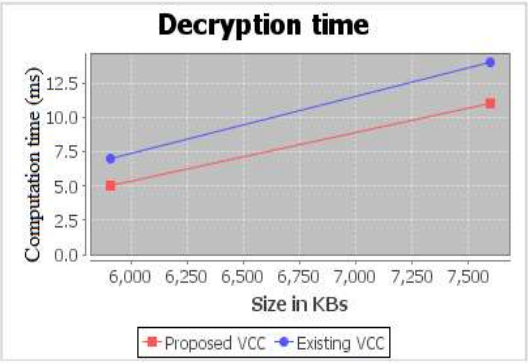

Fig. 5: Computation cost of message decryption time

VI. CONCLUSION

In this paper, we introduced VCC's architecture and presented the challenges of designing VCC's efficient AKA protocol to secure user-VC interactions. We have proposed an integrated AKA (CP-ABE) framework that provides the scalability and flexibility that VCC requires. The framework can support single sign-on, so a user can access multiple VCs securely without repeatedly registering with each VC. The performance analysis shows that, while ensuring acceptable computational costs and low overhead communication, our framework provides firm security. Finally, there has been a discussed several interesting future directions.

REFERENCES

[1] N. Lu et al., "Connected Vehicles: Solutions and Challenges," IEEE Internet Things J., vol. 1, no. 4, 2014, pp. 289–99.
[2] K. Zheng et al., "Software-Defined Heterogeneous Vehicular Network: Architecture and Challenges," IEEE Network, vol. 30, no. 4, July/Aug. 2016, pp. 72–80.
[3] R. Yu et al., "Toward Cloud-Based Vehicular Networks with Efficient Resource Management," IEEE Network, vol. 27, no. 5, Sept./Oct. 2013, pp. 48–55.
[4] M. Whaiduzzaman et al., "A Survey on Vehicular Cloud Computing," J. Network Comp. Appl., vol. 40, Apr. 2014, pp. 325–44.
[5] J. Ni et al., "Security, Privacy and Fairness In Fog-Based Vehicular Crowdsensing," IEEE Commun. Mag., vol. 55, no. 6, June 2017, pp. 146–52.
[6] K. Zhang et al., "Security and Privacy in Smart City Applications: Challenges and Solutions," IEEE Commun. Mag., vol. 55, no. 1, Jan. 2017, pp. 122–29.
[7] H. Abid et al., "V-Cloud: Vehicular Cyber-Physical Systems and Cloud Computing," Proc. ISABEL, 2011, article no. 165. [8] G. Yan et al., "Security Challenges in Vehicular Cloud Computing," IEEE Trans. Intell. Transp. Sys., vol. 14, no. 1, 2013, pp. 284–94.
[9] Q. Jiang et al., "A Privacy Preserving Three-Factor Authentication Protocol for e-Health Clouds," J. Supercomput., vol. 72, no. 10, 2016, pp. 3826–49.
[10] D. He and D. Wang, "Robust Biometrics-Based Authentication Scheme for Multiserver Environment," IEEE Systems J., vol. 9, no. 3, 2015, pp. 816–23.
[11] H. Zhang, Q. Zhang, and X. Du, "Toward Vehicle-Assisted Cloud Computing for Smartphones," IEEE Trans. Vehic. Tech., vol. 64, no. 12, 2015, pp. 5610–18.
[12] D. He et al., "Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services," to appear, IEEE Systems J.
[13] V. Odelu et al., "Provably Secure Authenticated Key Agreement Scheme for Distributed Mobile Cloud Computing Services," Future Generation Comp. Sys., vol. 68, 2016, pp. 74–88.
[14] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems Based on Pairing," Proc. CICS, 2000, pp. 26–28. [15] H. Li et al., "Engineering Searchable Encryption of Mobile Cloud Networks: When QoE Meets QoP," IEEE Wireless Commun., vol. 22, no. 4, Aug. 2015, pp. 74–80.

© 2019 IJSRET