Integrated Authentication and Key Agreement Framework for Vehicular Cloud Computing

Qi Jiang, Jianbing Ni, Jianfeng Ma, Li Yang, and Xuemin (Sherman) Shen

Abstract

VCC leverages the underutilized storage and computing resources of vehicles to collaboratively provide traffic management, road safety, and infotainment services to end users, such as drivers and passengers. It is a hybrid technology that improves the resource utilization on vehicles and is able to perform complex computing tasks that cannot be handled by a single vehicle. Despite the appealing advantages, security and privacy threats are severe in VCC due to the sharing of resources among unfamiliar vehicles. In this article, we identify security goals for the interoperability with VCC and provide an AKA framework for VCC. Specifically, we first present the research challenges and open problems for designing a reliable AKA with strong security guarantees for VCC. Then we propose an integrated AKA framework that integrates the single-server 3-factor AKA protocol and the non-interactive identity-based key establishment protocol, and evaluate its performance based on a simulated experimental platform. Finally, several interesting issues are discussed to light up the further research directions on AKA for VCC.

Introduction

A vehicular ad hoc network (VANET) enables vehicles equipped with sensing, communication, and networking capabilities to communicate with each other (V2V) or the roadside infrastructure (V2I) for information exchange [1]. It can provide a variety of promising applications, ranging from safety applications (e.g., accident reporting, congestion warning, and abnormal vehicle behavior warning) to non-safety applications (e.g., infotainment, smart parking, and advertising) [2]. Although the phenomenal growth of vehicular applications caters to drivers and passengers, a huge array of onboard capabilities are chronically underutilized.

To make full use of idle resources and extend vehicles' capability, cloud computing is utilized to revolutionize vehicular communications and applications, leading to the emergence of vehicular cloud computing (VCC) [3].

VCC is a hybrid platform that has a remarkable impact on road safety and traffic management by fully utilizing idle vehicular resources, such as computing and storage for decision making [4]. By integrating the idle onboard resources, a group of vehicles in a parking garage or on a road can form a cloud to collaboratively collect information, process data, and make decisions for improving the quality of service for both drivers and passengers. For example, travelers normally park their cars in airport parking lots while they are traveling. The airport can utilize the computing resources on vehicles to form a parking garage data center for on-demand access of end users. Similarly, vehicles stuck in congestion can generate a computing pool to perform complex simulations for traffic lights scheduling to remove congestion. End users can access the formed cloud and obtain the flexible resources on demand at the right time and place with a reasonable cost. The vehicular cloud (VC) offers an efficient way to enable the utilization of excess resources of vehicles, which are coordinated and dynamically allocated to end users. In general, a VC is dynamic and temporary due to vehicle mobility. The temporary VC is an important supplement to the conventional cloud (CC) for improving computing and storage capacities for end users. It is envisioned that VCC is able to enrich various vehicular applications and services, and improve driving and riding experiences, such as vehicular crowdsensing [5], video stream downloading, and road traffic monitoring [6].

Due to the unprecedented growth of smartphones, they can serve as important interfaces between drivers and external networks, as well as terminals responsible for storing and retrieving the information from VCC. For instance, a smartphone senses physiological conditions of the driver using biosensors and sends the collected data to VCC for intelligent processing. In the case of any accident or danger, the warning messages could be pushed to the smartphone for alarm [7]. With the involvement of smartphones, the appealing applications supported by VCC become scalable, upgradable, and inexpensive for deployment.

Although VCC is expected to play an increasingly significant role in intelligent transportation systems, inter-connectivity inevitably brings challenges and risks to drivers, one of which is security [8]. The communications between smartphones and the cloud are vulnerable to malicious attacks if countermeasures are insufficient, resulting in a range of damages on the VCC applications. First, sensitive and critical transmitting messages may suffer from unauthorized access. Second, any

Digital Object Identifier: 10.1109/MNET.2018.1700347

Qi Jiang, Jianfeng Ma, and Li Yang are with Xidian University; Jianbing Ni and Xuemin (Sherman) Shen are with the University of Waterloo.

0890-8044/18/$25.00 © 2018 IEEE

malicious behavior of drivers, such as modification and replay attacks with respect to the disseminated messages, could be fatal to data owners. Therefore, to guarantee driving safety and data reliability, it is essential to prevent adversaries from compromising VCC systems and protect the data exchanged between users and clouds (i.e., CC and VC).

To resist malicious attackers and secure data access in VCC, authentication and key agreement (AKA) mechanisms are essential and effective. AKA not only prevents unauthorized users from illegally accessing data at rest, but also enables entities to negotiate session keys for the integrity and confidentiality of transmitting data. Although some state-of-the-art AKA protocols [9, 10] have been proposed, it is difficult to migrate these protocols into VCC scenarios due to the features of VCC architecture. Specifically, a VC is a group of vehicles that is dynamic and temporary; hence, it is difficult to establish trust relationships among them. Therefore, it is important to design AKA protocols to ensure the authenticity of vehicles for VCC.

In this article, we aim to provide a comprehensive picture of designing an efficient AKA protocol for VCC and its security goals. Specifically, we first demonstrate the challenges and design goals of secure AKA protocols for VCC. Then we propose the integrated AKA framework by integrating the single-server 3-factor AKA protocol (SS-3FAKA) and the non-interactive identity-based key establishment protocol, to achieve mutual authentication and secure communications among users, VC, and CC. The performance is evaluated to demonstrate the efficiency of the new framework. Finally, we present several promising research directions to encourage more efforts on secure VCC.

Background of Vehicular Cloud Computing

VCC Architecture

Figure 1 depicts the architecture of VCC, which contains the conventional cloud (CC) and vehicular clouds (VCs) [11]. CC provides outsourced computing and storage services to end users with on-demand resource deployment and ubiquitous service access. It can be either a public cloud service, for example, iCloud, Aliyun, and Amazon EC2, or a private cloud deployed by authority, for example, the Department of Transportation and car manufacturers [7]. The CC is responsible for storing various traffic information collected by vehicles, performing data processing tasks outsourced by end users, and offering a variety of services to vehicles on road, for example, navigation, infotainment, content distribution, and driving assistance. VCs are formed by a group of vehicles on streets and roadways and in parking lots, which have plentiful and underutilized computing resources to provide services. Similar to the idle computers in CC, vehicle owners agree to rent out their excess onboard resources, which are integrated into a powerful VC through V2V and V2I communications. All the vehicles in VC form a temporary and dynamic pool of computing and storage resources, such as an airport data center, a parking lot data cloud, and a driving vehicle cloud on the road, to provide data maintenance and processing services, including traffic light scheduling, traffic information storage, and frequent congestion easing. End users can access their collected traffic data and outsource computational tasks on VCs. In addition, base stations and roadside units (RSUs) are deployed to support the management of VCs and the communications between VCs and CC.

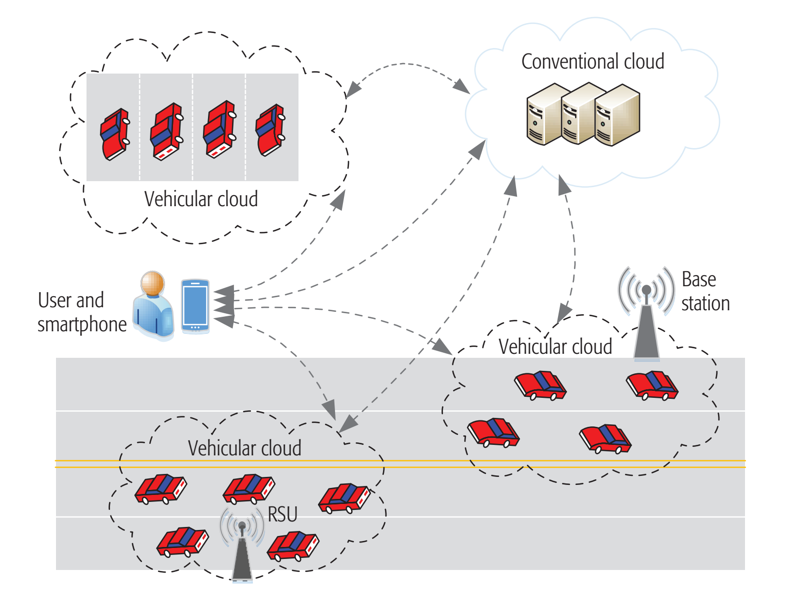

FIGURE 1. The architecture of vehicular cloud computing.

Users can access the services and resources offered by VCC using their smartphones, such as navigation services and video download applications. Through cellular communications, the smartphones are connected to the Internet and thereby enjoy the cloud services provided by CC. Due to mobility, users may access multiple VCs to enjoy different services(e.g., video stream downloading and traffic query). In addition, as important sensing devices, smartphones collect and deliver traffic and parking information to VCs for local navigation services. The communication exchange between VCs and smartphones via cellular networks improves driving safety and enriches driving experiences.

VCC Services

The interactions among users, VCs, and CC can support a variety of vehicular applications and services for users [4].

Information as a Service: VCC can provide information for safe driving, such as road conditions, advanced warnings, traffic conditions, and accident alerts, to users. These services together are recognized as information as a service.

Computing as a Service: Drivers may use their smartphones to access computation-intensive applications, such as finding the shortest path to a destination and discovering restaurants on roads. Since smartphones have limited power, VCC can provide computing services to users.

Storage as a Service: Although the storage space on smartphones continues to increase, it is common that users may require additional storage to perform their tasks. It is natural to turn to VCs, which aggregate the storage capacity of vehicles and provide storage space to users as a service.

VCC, as an integrated architecture of cloud computing and VANETs, inherits security and privacy issues from both VANETs and cloud computing. Security and privacy issues in each paradigm have been discussed. However, the integration of VANETS and cloud computing triggers new security and privacy issues, which have not attracted sufficient attention.

Network as a Service: Although a smartphone has its own network connectivity, its owner may be disconnected due to various reasons, such as weak cellular coverage in rural areas or high moving speed. In this case, the owner would rent network resources from VCs for Internet access.

Challenges of AKA for VCC

VCC, as an integrated architecture of cloud computing and VANETs, inherits security and privacy issues from both VANETs and cloud computing. Security and privacy issues in each paradigm have been discussed. However, the integration of VANETs and cloud computing triggers new security and privacy issues, which have not attracted sufficient attention.

Security Challenges of AKA for VCC

Fundamental security and privacy challenges in VCC include identity authentication of high-mobility vehicles, key management, vehicular data privacy, and location privacy [8]. A large number of security solutions have been proposed to address these challenges [8]. Nevertheless, with the involvement of multiple VCs, how to allow users to securely access VCC services needs more research effort.

Secure Access for VCC: AKA is fundamental to provide safer and more enjoyable access for VCC. To achieve identity authentication, the provider has to prove its possession of some private information or an identifiable feature to an identity authentication system. Identity authentication techniques can be classified into three categories:

- What one knows: password, personal identification number (PIN), security questions, and so on
- What one has: smart card, token, credential, and so on
- What one is: biometrics

It is widely known that each factor has its own merits and demerits. Specifically, the password can be brute forced, spied on, or even socially engineered. Smart cards may be lost, stolen, or revealed through side channel attacks. Biometrics suffer from the risk of being duplicated. For instance, fingerprints left somewhere can be lifted and used to gain illegitimate access. Moreover, biometrics cannot be updated since they are biological characteristics of individuals. Since the security and privacy threats in VCC may directly result in safety problems, secure access to VCC with strong security guarantee becomes essential and challenging for users.

Secure Authentication to Multi-VCs: Due to the vehicle mobility, a vehicle would form different VCs with various vehicles at different regions, such that these VCs are dramatically dynamic. The authentication credentials of users should be maintained on each vehicle to ensure normal service access for users, which leads to heavy storage overhead for vehicles. Furthermore, with the mobility of users, the maintenance of authentication credentials becomes extremely inefficient. Specifically, the joining and revocation of users bring about a heavy burden on the management of authentication credentials. As a result, it is quite difficult for a user to be authenticated by different VCs. Since each user may access multiple VCs to acquire different services, how to keep identity consistence during authentication to multiple VCs is quite challenging. On the contrary, if a user uses an authentication credential to access multiple VCs, the risk of service tracking becomes a huge concern for users. Therefore, during the authentication to multiple VCs, the management of authentication credentials should be taken into account in VCC.

Multi-Factor AKA: Since each authentication factor has inherent weaknesses, as discussed above, single-factor authentication schemes cannot provide the strong security guarantee required by safety applications. A straightforward approach is to increase the number of features or factors needed in authentication schemes. Thereby, designing a 3-factor AKA scheme becomes vital for users to access VCs.

According to the number of servers to be accessed, the existing 3-factor AKA protocols can be classified into two categories: SS-3FAKA [9] and multi-server 3-factor AKA (MS-3FAKA) [10]. The SS-3FAKA schemes cannot be directly used in VCC, where many VCs act as service providers to offer a plethora of services, since a user has to register with every VC repeatedly. It is not only a waste of users' time and energy to perform repeated registration, but also puts extra burden on users to maintain multiple sets of security credentials.

As a distributed computing environment, VCC is a multi-server architecture in which a user may be required to authenticate to more than one VC, such that MS-3FAKA protocols are necessary to ensure secure communications. To reduce the users' burden and guarantee secure communications, an MS-3FAKA protocol [10] has been proposed to eliminate repeated registration. However, an online registration authority is still needed to realize mutual authentication in every authentication session. Moreover, the user is required to present the authentication credentials to VCs repeatedly when accessing the VCs. To avoid the time and resource consuming interaction with online registration authority, several MS-3FAKA protocols without online registration authority have been proposed by employing public key-based cryptosystems [12, 13]. However, one common drawback in these schemes [12, 13] is that users have to deal with the management of public keys of different VCs, indicating that each user has to maintain an up-to-date public key revocation list to revoke compromised VCs. While many 3-factor AKA protocols have been developed for different service scenarios, none is in the context of VCC or matches the requirements of VCC. Therefore, 3-factor AKA for VCC has not yet been addressed, and deserves more attention.

Security Goals of AKA for VCC

To cope with the challenges of AKA in VCC, an AKA framework should satisfy the following security properties [12].

Mutual Authentication: To ensure that only a legitimate user is allowed to access resources in VCC, the user is required to present all three factors to corroborate his/her identity. Clouds also need to be authenticated by users to prevent rogue services. In doing so, the users can have secure access to VCC services with strong security guarantees.

Key Agreement: The session keys should be created for the data transmission process to achieve the confidentiality and integrity of transmitting data. Thus, privacy leakage and data corruption can be avoided during transmission on communication channels.

User Anonymity: To prevent users from being identified, the AKA framework should be able to hide users' real identities, indicating that an adversary cannot learn data sources even if it captures authentication transcripts on channels, except their intended counterparts.

User Untraceability: The AKA framework should be capable of providing user untraceability, a stronger notion than user anonymity, to protect a user's actions from being traced by adversaries through identifying the sources of the transmitting data or linking the authentication transcripts in different sessions to the same user.

Single Sign-on: The user can securely access multiple VCs using only a single set of credentials obtained from the CC during registration. In addition, the user can be relieved of the burden of public key management. This goal is to counteract the challenge of secure access to multiple VCs.

An Integrated AKA Framework in VCC

In this section, we propose our integrated AKA framework to address the security challenges in VCC.

Framework Overview

To achieve the above security goals, we propose an integrated AKA framework for VCC, in which there are three roles: U_i, VC_j, and CC. CC acts as a trusted third party and is responsible for setting system parameters, generating private keys for VC_j, and issuing a smart card to each registered user U_i. All users and VCs build trust relationships with CC in the registration phase.

U_i can access multiple services from both CC and VC_j. To access the services offered by CC, the user needs to present the required credentials to authenticate to CC. After the completion of the authentication to CC, U_i can request a ticket for the service offered by VC_j from CC, who creates a ticket for the user without interacting with VC_j. Then U_i can access VC_j directly with the obtained ticket without further involvement of CC.

There are three key ingredients in our integrated framework: a non-interactive identity-based key establishment protocol, the SS-3FAKA protocol, and the ticket concept. The non-interactive identity-based key establishment protocol [14] is used to non-interactively establish a shared key between two clouds, which in turn facilitates the distribution of a service ticket. The SS-3FAKA protocol can achieve a high security level to ensure secure access to CC. In addition, the concept of a ticket is leveraged to allow single sign-on, such that a user can securely access multiple VCs with only one set of security credentials registered with CC without repeated registration with VC_j.

Our framework is suitable for VCC, since the user only needs to register with CC, and it is flexible in terms of accessing multiple dynamic and temporary VCs. Moreover, our framework hides the complexity of public key management from a user's perspective.

Our framework is suitable for VCC, since the user only needs to register with CC, and it is flexible in terms of accessing multiple dynamic and temporary VCs. Moreover, our framework hides the complexity of public key management from a user's perspective.

Single-Server 3-Factor AKA Protocol

Our integrated AKA framework is designed based on an SS-3FAKA protocol. A typical SS-3FAKA protocol [9] involves two entities, the user U and the remote server S, and consists of five phases: initialization, registration, login and authentication, password and biometric change, and revocation and re-registration [9].

Initialization (SS-3FAKA.Init): S selects system parameters, generates the private-public key pair, and publishes the public parameters.

Registration (SS-3FAKA.Reg): When U registers on S, U selects the identifier ID and the password PW, provides the biometric sample B, and submits a value derived from ID and (PW, B) to S through a secure channel. S issues a smart card storing the secret key derived from ID. U stores the related values in the card after obtaining it.

Login and Authentication (SS-3FAKA.Auth): U attaches the smart card to a terminal and enters (ID, PW, B). The terminal interacts with the card and sends the login request MSG_1 to S. S checks the legitimacy of U and sends a login response MSG_2 to U.

Password and Biometric Change (SS-3FAKA.Change): U updates the password and the biometric sample periodically.

Revocation and Re-registration (SS-3FAKA.Revoc): U revokes his/her account and re-regis ters without changing his/her identity.

Our Integrated AKA for VCC

Our framework consists of eight phases: system setup, VC registration, user registration, user authentication to CC, ticket request, user authentication to VC, password change, and smart card revocation, as shown in Fig. 2.

In the system setup phase, CC first selects its master private key and public parameters, computes the corresponding public key, and publishes its public key and public parameters. In the VC registration phase, VC_j submits its identity, which serves as the public key, and obtains the corresponding private key generated by CC. In the user registration phase, CC issues a smart card to U_i. In the phase of authentication between U_i and CC, the SS-3FAKA protocol is executed between U_i and CC. U_i can request a service ticket from CC through the established secure channel in the ticket request phase. In the phase of authentication between U_i and VC_j, U_i presents the ticket to VC_j to establish a secure channel between U_i and VC_j. Finally, the password change and smart card revocation phases can be invoked to change the user password and revoke a lost/stolen smart card without the demand for user identity changing.

System Setup: CC first initializes the parameters by performing the steps in SS-3FAKA.Init. CC then generates (p, G_1, G_2, e), where G_1 is a cyclic additive group of the prime order q, P is a generator of G_1, and G_2 a cyclic multiplicative group of the same order, $e{:}G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing [14], and p is a large prime. CC picks $s \in Z_p$, outputs its master secret and public key pair $(s, S = sP)$, and publishes the public key parameters (P, S, H_1, H_2, h), where $H_1 : \{0, 1\}^* \rightarrow G_1$ serves as identity mapping function, $H_2{:}G_2 \rightarrow \{0, 1\}^l$, $h : \{0, 1\}^* \rightarrow \{0, 1\}^l$, and l is the bit length of hash function output (e.g., $l = 160$). CC also selects an identity ID_{CC}, and computes its secret key $k_{CC} = sH_1(ID_{CC})$.

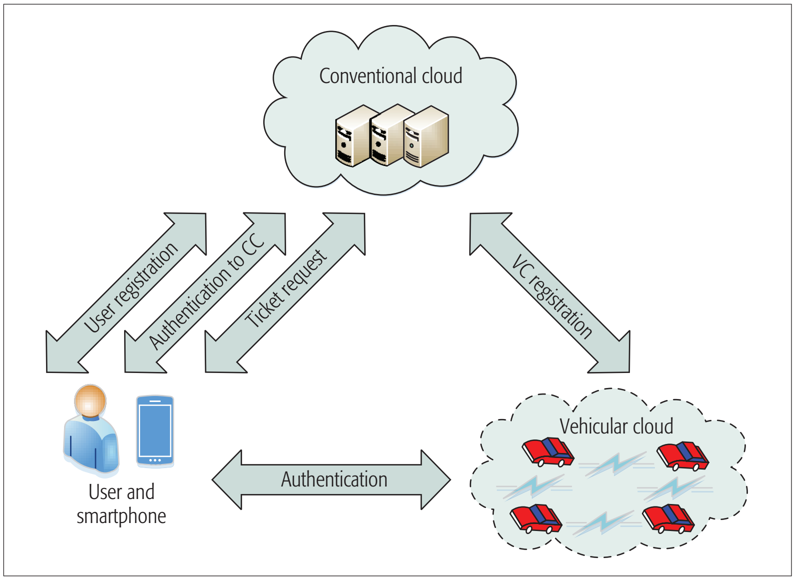

FIGURE 2. AKA framework.

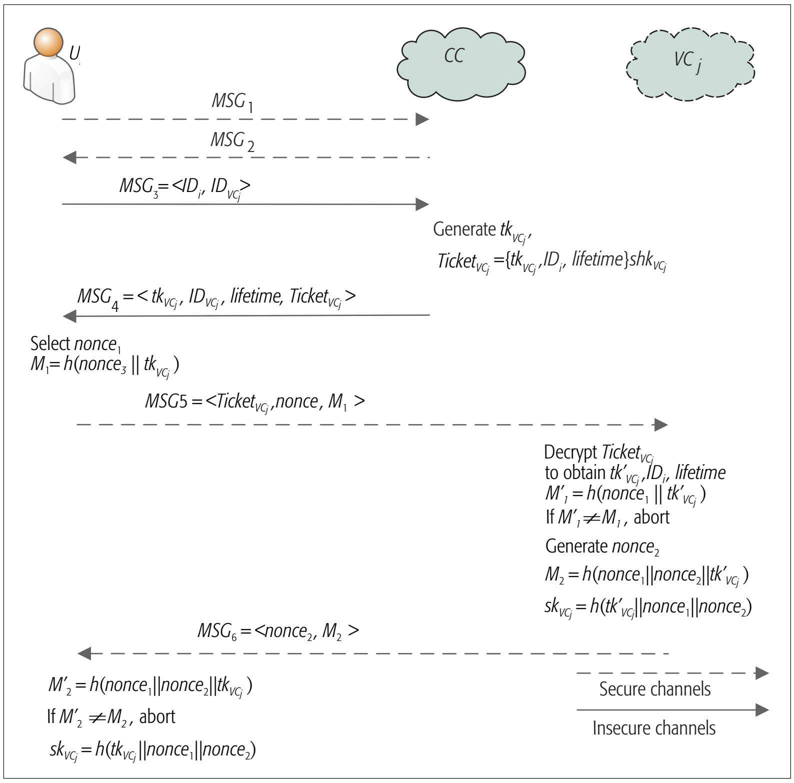
U
CC
VC j
MSG_1
MSG_2
MSG_3=<ID_i, ID_{VC_j}>
Generate tk_{VC_j},
$Ticket_{VC_j}$ ={tk_{VC_j},ID_i, lifetime}shk_{VC_j}
MSG_4 =< tk_{VC_j}, ID_{VC_j}, lifetime, $Ticket_{VC_j}$ >
Select $nonce_1$
M_1= h($nonce_3$ || tk_{VC_j})
MSG5 =<$Ticket_{VC_j}$,nonce, M_1 >
Decrypt $Ticket_{VC_j}$
to obtain tk'_{VC_j},ID_i, lifetime
M'_1 = h($nonce_1$ || tk'_{VC_j})
If $M'_1 \neq M_1$, abort
Generate $nonce_2$
M_2 = h($nonce_1$||$nonce_2$||tk'_{VC_j})
sk_{VC_j} = h(tk'_{VC_j}||$nonce_1$||$nonce_2$)
MSG_6 =<$nonce_2$, M_2 >
M'_2 = h($nonce_1$||$nonce_2$||tk_{VC_j})
If $M'_2 \neq M_2$, abort
sk_{VC_j} = h(tk_{VC_j}||$nonce_1$||$nonce_2$)
Secure channels
Insecure channels

FIGURE 3. The flowchart of our framework.

VC Registration: Each VC registers with CC to obtain an identity-based public and private key pair as follows:

- To build the trust relationship with CC, VC_j first sends the identity ID_{VC_j} to CC.
- CC computes $k_{VC_j} = sH_1(ID_{VC_j})$ (i.e., the private key of VC_j) and $shk_{VC_j} = H_2(e(k_{CC}, H_1(ID_{VC_j}))) = H_2(e(H1(ID_{CC}), H_1(ID_{VC_j}))s)$ (i.e., the key established non-interactively between CC and VC_j). Then CC sends k_{VC_j} to VC_j through a secure channel.
- When receiving its private key, VC_j checks if $e(S, H_1(ID_{VC_j})) = e(P, k_{VC_j})$ holds. If yes, VC_j computes $shk_{VC_j} = H_2(e(H_1(ID_{CC}), k_{VC_j})) = H_2(e(H_1(ID_{CC}), H_1(ID_{VC_j}))^s)$, and stores k_{VC_j} and shk_{VC_j} in the secure memory.

Note that shk_{VC_j} can be used to build secure communication between CC and VC_j, and achieve ticket distribution without the interaction with VC_j.

User Registration: In this phase, the operations are the same as those in SS-3FAKA.Reg.

Authentication between User and CC: In this phase, the operations are the same as those in SS-3FAKA.Auth, and the session key sk_{CC} is shared between U_i and CC after the operations.

Ticket Request: After U_i has been authenticated by CC, U_i can request a ticket from VC_j through a secure channel, as shown in Fig. 3.

- Step 1: U_i first sends a ticket request MSG_3 =<ID_i, ID_{VC_j}> to CC.
- Step 2: Upon receiving the request, CC generates a temporary key tk_{VC_j}, defines the validity period of the ticket *lifetime*, and $Ticket_{VC_j} = \{tk_{VC_j}, ID_i, lifetime\}_{shk_{VC_j}}$, and sends $MSG_4 = <tk_{VC_j}, ID_{VC_j}, lifetime, Ticket_{VC_j}>$ to U_i. Here, $\{M\}_K$ denotes the cipher text of message M encrypted by a key K.

Authentication between User and VC: U_i can authenticate VC_j using the obtained ticket (Fig. 3):

- Step 1: U_i generates a one-time random number $nonce_1$ and computes $M_1 = h(nonce_1 \,||\, tk_{VC_j})$ and sends VC_j the ticket authentication request $MSG_5 = <Ticket_{VC_j}, nonce_1, M_1>$.
- Step 2: VC_j decrypts the ticket to obtain $(tk'_{VC_j}, ID_i, lifetime)$. VC_j first verifies whether the ticket is valid. If it is expired, VC_j rejects the authentication request; otherwise, it computes $M'_1 = h(nonce_1 \,||\, tk'_{VC_j})$. If M’1 = M_1, VC_j aborts this session; otherwise, U_i is authorized to access its resources and services. VC_j generates a nonce $nonce_2$, computes $M_2 = h(nonce_1 \,||\, nonce_2 \,||\, tk'_{VC_j})$ and the session key $sk_{VC_j} = h(tk'_{VC_j} \,||\, nonce_1 \,||\, nonce_2)$, and sends the message $MSG_6 = <nonce_2, M_2>$ to U_i.
- Step 3: U_i computes $M'_2 = h(nonce_1 \,||\, nonce_2 \,||\, tk_{VC_j})$. If $M'_2 = M_2$, U_i computes $sk_{VC_j} = h(tk_{VC_j} \,||\, nonce_1 \,||\, nonce_2)$, which is the session key between U_i and VC_j; otherwise, U_i terminates this session.

The subsequent phases (i.e., password and biometric change and revocation and re-registration) are the same as those in the SS-3FAKA protocol.

Remarks

We show that our framework is secure through inspection against security goals given the above and compare our framework with three state-of-the-art schemes.

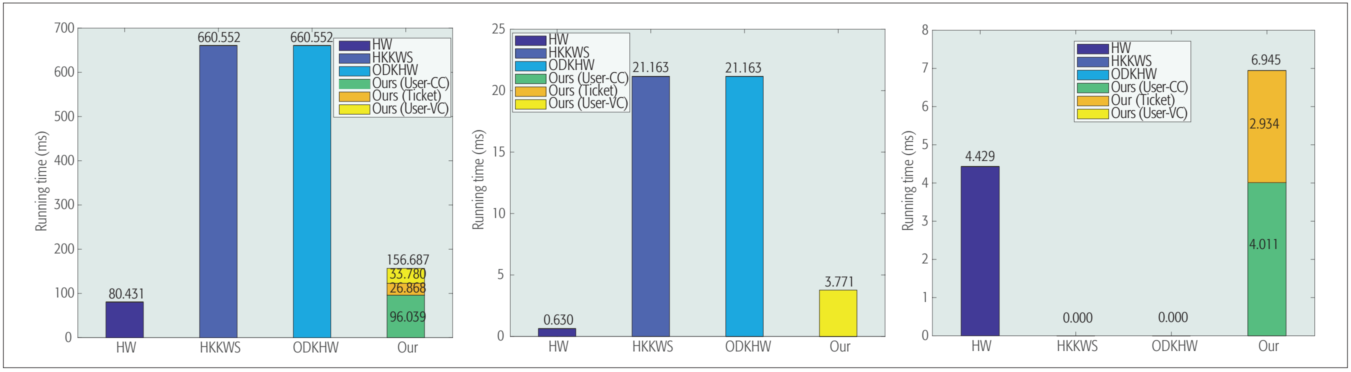

FIGURE 4. Computational cost comparison: left: time cost on users; middle: time cost on VCs; right: time cost on CC.

Operation	U_i	VC_j	CC
T_{bp}	361.282	5.562	4.115
T_{ecm}	0.537	0.050	0.041
T_{eca}	0.601	0.051	0.045
T_e	200.670	2.097	1.695
T_m	0.731	0.007	0.006
T_s	13.434	1.587	0.978
T_h	11.260	0.728	0.483

TABLE 1. Time costs of cryptographic operations (milliseconds).

Remarks on Security: Our framework achieves the essential security goals.

Mutual Authentication: VC_j authenticates U_i through the validation of $M_1 = h(nonce_1 \,||\, tk_{VC_j})$, where tk_{VC_j} is a secret value only known by them. U_i has to provide all three factors (i.e., password, smart card, and biometric) to compute a valid value M_1. U_i authenticates VC_j through the validation of $M_2 = h(nonce_1 \,||\, nonce_2 \,||\, tk'_{VC_j})$, which can by computed by a legitimate VC_j knowing the ticket key tk'_{VC_j}. Hence, U_i and VC_j are mutually authenticated.

Session Key Agreement: The session key $sk_{VC_j} = h(tk_{VC_j} \,||\, nonce_1 \,||\, nonce_2)$ is established between U_i and VC_j, which is computed from the secret value tk_{VC_j} only known by U_i and VC_j. Thus, our framework supports session key agreement.

User Anonymity and Untraceability: In the ticket request phase, U_i's identity ID_i is transferred through the secure channel established in the phase of authentication between U_i and CC. In the phase of authentication between U_i and VC_j, ID_i is concealed in $Ticket_{VC_j}$. One way for an adversary to get ID_i is to use the system master secret s of CC to compute shk_{VC_j}. It is infeasible to do so because s is the system master secret. Hence, our framework supports user anonymity and untraceability.

Single Sign-on: In our framework, both U_i and VC_j only need to build the trust relationship with CC in the registration phase. The SS-3FAKA protocol allows U_i to access CC. After the secure channel is established, U_i can request a ticket from CC and access VC_j directly using the obtained ticket without involving CC, thus achieving single sign-on. Moreover, U_i is relieved of the complexity of public key management. Specifically, U_i only maintains his/her own security credentials without the need to maintain a public key revocation list.

Our framework also provides other important security features, including no password exposure, biometric privacy protection, and so on. With these desirable properties, our framework can properly support secure access for multiple VCs in VCC.

In summary, our framework addresses the challenges of AKA for VCC through the provision of 3-factor authentication of users and single sign-on. With 3-factor authentication of users, users can have secure access to VCs with strong security guarantee. With single sign-on, users can freely access multi-VCs without repeated registration.

Performance Evaluation

We evaluate the performance of our framework in terms of computational and communication overhead. The following notations are used to denote the running time of cryptographic operations. The bit XOR operation is neglected due to its extremely short running time.

T_{bp}: The running time of bilinear pairing
T_{ecm}: The running time of elliptic curve multiplication
T_{eca}: The running time of elliptic curve point addition
T_e: The running time of exponentiation
T_m: The running time of multiplication
T_s: The running time of symmetric encryption/decryption
T_h: The running time of the hash function

The simulation platform of the VCC system consists of a smartphone (Huawei Mate 7 with a Hisilicon Kirin 925 2.45 GHz processor, 3 GB memory, and Google Android 4.4.2 OS), a laptop simulating VCs (Apple Macbook Pro with an Intel I7-4460S 3.1 GHz processor, 16 GB memory, and the MacOS 10.12.4 OS), and a desktop simulating CC (Dell Alienware with an Intel I7-6700k 4.0 GHz processor, 32 GB memory, and Windows 10 64-bit OS). The pairing operation is implemented using the JPBC Library, and the other operations are implemented using the standard Java library.

Table 1 lists the running time of the involved operations on this platform.

In [10], the time costs of user, VC, and CC are $3T_{ecm} + 7T_h = 80.431$ ms, $2T_{ecm} + 5T_h = 0.630$ ms, and $2T_{ecm} + 9T_h = 4.429$ ms, respectively. In [12], the time costs of user and VC are $3T_{ecm} + T_{eca} +$

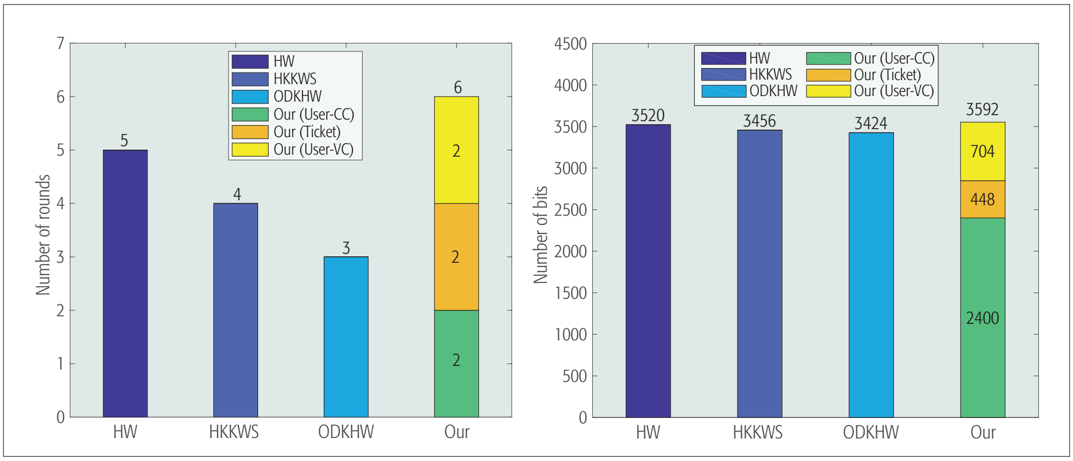

FIGURE 5. Communication cost comparison: left: rounds in authentication; right: binary length of messages.

Our integrated AKA framework cannot be applied to achieve authentication between the user and VC from different security domains. Therefore, multi-domain integrated AKA framework is important to fit the real scenario, in which multiple cloud service providers cooperatively provide cloud services.

$3T_e + 5T_h$ = 660.522 ms and $2T_{bp} + T_{ecm} + T_{eca} + 3T_e + 5T_h$ = 21.163 ms, respectively. In [13], the time costs of user and VC are $3T_{ecm} + T_{eca} + 3T_e + 5T_h$ = 660.522 ms and $2T_{bp} + T_{ecm} + T_{eca} + 3T_e + 5T_h$ = 21.163 ms, respectively. In our framework, the authentication process is divided into three phases, that is, user authentication to CC (User-CC), ticket request (Ticket), and user authentication to VC (User-VC). The SS-3FAKA [9] is used to instantiate the integrated 3FAKA framework. In the phase of User-CC, the time costs of user and CC are $3T_{ecm} + 2T_s + 6T_h$ = 96.039 ms and $3T_{ecm} + 2T_s + 4T_h$ = 4.011 ms, respectively. In the phase of Ticket, the time costs of user and CC are $2T_s$ = 26.868 ms and $3T_s$ = 2.934 ms, respectively. Besides, in the phase of User-VC, the time costs of user and VC are $3T_h$ = 33.780 ms and $T_s + 3T_h$ = 3.771 ms, respectively. Figure 4 depicts the comparison results in computational overhead.

The computational burden of the user and VC in our framework is much lower than that of [12, 13]. Although our framework incurs more computational burden on CC than [12, 13], it is not an issue since CC is dedicated computing infrastructure. Compared to [10], our framework is efficient as the online interaction with CC is avoided.

To compare communication cost, we assume that the binary length of identity, nonce, ticket key, ticket lifetime, hash output, and an elliptic curve point is 32, 96, 128, 32, 160, and 1024 bits, respectively. The messages MSG_1, ..., MSG_6 have lengths of 1216, 1184, 64, 384, 448, and 256 bits, respectively. The total communication cost of our framework is 3552 bits. As shown in Fig. 5, the total communication cost of our framework is slightly higher than that of [10, 12, 13]. However, the communication cost of our framework can be greatly reduced as the authentication between a user and CC, which only needs to be executed once in a relatively long period.

In summary, our framework has better efficiency in computation and communication than the existing schemes [10, 12, 13].

Future Directions

The proposed integrated AKA framework resolves the security challenges of secure access to multiple VCs with strong security guarantee, but the following directions still need further efforts to ensure secure data communications and privacy protection in VCC.

Multi-Domain AKA

In reality, there are more than one conventional cloud service provider, indicating that not all users and vehicular clouds are registered with the same conventional cloud. Our integrated AKA framework cannot be applied to achieve the authentication between the user and VC from different security domains. Therefore, a multi-domain integrated AKA framework is important to fit the real scenario, in which multiple cloud service providers cooperatively provide cloud services to users.

Continuous Authentication

Our framework provides login-time verification of user identity and cannot detect any change of user identity in an authentication session. One strategy to mitigate this limitation is to shorten the valid period of the authentication session and re-authenticate the user periodically, but this strategy greatly degrades user experience. Continuous authentication is a promising approach to address the above issue by implicitly and constantly verifying a user's identity based on soft biometrics, such as user behavior and biomedical signals. If any change in identity is identified, the current access session will be locked. Therefore, continuous authentication is an important complement to our framework and deserves more attention in VCC.

Data Privacy

Authentication provides the first line of defending against external adversaries to guarantee data security and privacy. However, it cannot ensure data privacy against the honest-but-curious cloud servers in VCC, where the onboard computing units act as distributed servers to perform the tasks. Although a variety of solutions have been proposed to ensure data privacy in cloud computing, such as searchable encryption and secure multi-party computation [15], these schemes are not suitable for VCC to prevent the onboard units from learning the data contents in a dynamic and distributed environment due to their efficiency. It is expected that the scheme for VCC should be efficient in terms of computation and storage consumption. Therefore, it is essential to design efficient data privacy solutions for vehicles in VCC.

Conclusions

In this article, we have introduced the architecture of VCC and presented the challenges of designing the efficient AKA protocol in VCC to secure the interactions between users and VCs. We have proposed an integrated AKA framework that caters for the scalability and flexibility required in VCC. The framework can support single sign-on, such that a user is able to securely access multiple VCs without registering with each VC repeatedly. The performance analysis demonstrates that our framework provides firm security while ensuring acceptable computational cost and low communication overhead. Finally, several interesting future directions have been discussed.

Acknowledgment

This work was supported in part by the National Science Foundation of China (61672413, U1405255, 91638204, 61671360, 61672415, 61772548), and in part by the China 111 Project (B16037).

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.

Authentication provides the first line of defending against external adversaries to guarantee data security and privacy. However, it cannot ensure data privacy against the honest-but-curious cloud servers in VCC, where the onboard computing units act as distributed servers to perform the tasks.

[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.

Biographies

Qi Jiang (jiangqixdu@gmail.com) received his B.S. degree from Shaanxi Normal University, Xi’an, China, in 2005, and his Ph.D. degree from Xidian University, Xi’an, China, in 2011, both in computer science. He is currently an associate professor with the School of Cyber Engineering, Xidian University. His research interests are in the areas of security and privacy in the Internet of Things and applied cryptography.

Jianbing Ni [S’16] (j25ni@uwaterloo.ca) received his B.E. degree and M.S. degree from the University of Electronic Science and Technology of China in 2011 and 2014, respectively. He is currently working toward a Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests are applied cryptography and information security, with current focus on cloud computing, mobile crowdsensing, smart grid, and big data.

Jianfeng Ma (jfma@mail.xidian.edu.cn) received his M.S. degree and Ph.D. degree from Xidian University in 1992 and 1995, respectively. He is currently a professor with the School of Cyber Engineering, Xidian University. He has published over 150 journal and conference papers. His research interests include applied cryptography, wireless network security, data security, and mobile security.

Li Yang (yangli@xidian.edu.cn) received his Ph.D. degree from Xidian University in 2010. He is currently an associate professor with the School of Computer Science and Technology, Xidian University. His research interests are in the areas of security and privacy in mobile Internet, cloud security, and trusted computing.

Xuemin (Sherman) Shen [M’97, SM’02, F’09] (sshen@uwaterloo.ca) received his B.Sc. degree from Dalian Maritime University, China, in 1982, and his M.Sc. and Ph.D. degrees from Rutgers University, Newark, New Jersey, in 1987 and 1990, respectively, all in electrical engineering. He is a professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo. He was the Associate Chair for Graduate Studies from 2004 to 2008. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He was a recipient of the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo; the Premier’s Research Excellence Award in 2003 from the Province of Ontario; and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He served as the Technical Program Committee Chair/Co-Chair for ACM MobiHoc ’15, IEEE INFOCOM ’14, IEEE VTC-Fall ’10; Symposia Chair for IEEE ICC ’10; Tutorial Chair for IEEE VTC-Spring ’11 and IEEE ICC ’08; and Technical Program Committee Chair for IEEE GLOBECOM ’07. He also serves/has served as the Editor-in-Chief for *IEEE Network*, *Peer-to-Peer Networking and Application*, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.