Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6870742, 9 pages
https://doi.org/10.1155/2018/6870742

Research Article

Concurrently Deniable Group Key Agreement and Its Application to Privacy-Preserving VANETs

Shengke Zeng [1,2] and Yong Chen [1,3]

[1] *School of Computer and Software Engineering, Xihua University, Chengdu, China*
[2] *Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[3] *School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China*

Correspondence should be addressed to Shengke Zeng; zengshengke@gmail.com

Received 5 December 2017; Accepted 4 March 2018; Published 15 April 2018

Academic Editor: Ning Zhang

VANETs need secure communication. Authentication in VANETs resists the attack on the receipt of false information. Authenticated group key agreement (GKA) is used to establish a confidential and authenticated communication channel for the multiple vehicles. However, authentication incurs privacy leakage, that is, by using digital signature. Therefore, the deniability is deserved for GKA (which is termed as DGKA) due to the privacy protection. In the DGKA protocol, each participant interacts with intended partners to establish a common group session key. After this agreement session, each participant can not only be regarded as the intended sender but also deny that it has ever participated in this session. Therefore, under this established key, vehicles send confidential messages with authentication property and the deniability protects the vehicles privacy. We present a novel transformation from an unauthenticated group key agreement to a deniable (authenticated) group key agreement without increasing communication round. Our full deniability is achieved even in the concurrent setting which suits the Internet environment. In addition, we design an authenticated and privacy-preserving communication protocol for VANETs by using the proposed deniable group key agreement.

1. Introduction

Vehicular ad hoc networks (VANETs) [1] refer to the peer-to-peer networks formed by roadside units and adjacent vehicles for sharing information, including traffic information (the speed and flow of vehicles, etc.) and warning information. VANETs provide a safe and comfortable driving environment for the drivers, which avoids the congestion and traffic accidents. VANETs should provide secure communication in case false information is inserted into the network. Besides that, VANETs should also have privacy issues as vehicles are reluctant to expose the sensitive information while sharing their own traffic information.

The group key agreement (GKA) provides a secure channel for the vehicles communication. GKA protocol [2] allows a group of participants to establish a common session key for a secure communication channel over an insecure network by agreement. However, key agreement without authentication incurs man-in-middle attack. In order to handle this problem, the authentication is necessary. However, authentication binds the identity, which causes the privacy leakage. In many cases, the participants do not want the third party to know their involvements in some key agreements. In other words, they want to have the capacity to deny that they have ever participated in some sessions after the key agreement execution. Hence, the deniable GKA (DGKA) protocol was presented by introducing deniability into GKA protocol. Bohli and Steinwandt first formalized the DGKA protocol in [3]. In the DGKA protocol, it is not feasible to convince a third party that these participants in a group key agreement session have been involved in the conversation from the communication transcript. In other words, each participant can deny its involvement to the third party.

1.1. Related Work. The deniability is formalized by introducing a simulator that can simulate the conversation transcript without secrets. Therefore, the participants can deny this as someone else would produce this indistinguishable conversation transcript. If this simulator can be run by anyone, it is denoted by the full deniability. Deniable authentication was first introduced by Dolev et al. in [4] and formally studied by Dwork et al. in [5]. The general technique to realize the deniable authentication is that the sender uses its secret (i.e., the private key) to generate a value v_s. If the receiver produces a value v_r which equals v_s by using a related witness, the receiver is convinced of the sender's authentication. In order to simulate the transcript (for the deniability), this witness has to be revoked. Thus, the early works such as [5, 6] require more rounds to revoke the witness upon the receipt of the committed v_s (i.e., COM(v_s)). In this way, the simulator run by anyone can extract this witness to simulate the transcript by rewinding steps. However, the deniability does not hold in the concurrent scenario due to the rewinding. Therefore, the timing assumption is necessary to be considered to realize concurrent deniability, such as [5, 6].

However, the timing assumption is farfetched for the Internet which is a fully concurrent environment. Some related works have to handle this problem by avoiding rewinding. Di Raimondo et al. [7] showed that the plaintext-awareness [8] of the underlying encryption can extract the witness for the simulation without rewinding steps. Jiang's work [9] depends on the public random oracle to extract the witness and therefore the rewinding steps are not necessary. Yao and Zhao [10, 11] proposed the deniable Internet key exchange based on the knowledge of exponent assumption (KEA). By the KEA assumption, the witness can be extracted and the transcripts are perfectly simulated. Tian et al. [12] made use of the selectively unforgeable but existentially forgeable signature to simulate the transcript. Zeng et al. [13] presented a multireceiver encryption under KEA assumption and used it as a building block to propose a concurrently deniable ring authentication. Jiang [14] made use of a moderate encryption to avoid rewinding to construct the concurrently deniable key exchange. These approaches achieve the deniability without rewinding steps; thus the simulation even in the concurrent scenario is normal. However, as we see above, these works suffer the limitations such as the strong assumptions, inefficiency, or random oracles.

Deniable authentication has been applied to many occasions nowadays [13, 15, 16] and was first introduced into the two-party key exchange protocol [17] in [18]. Mao and Paterson [18] informally defined the deniable key exchange (DKE) protocol and obtained its deniability by using identity-based techniques. Later, how to use the approach in [19] to design a DKE protocol was discussed and a concrete approach was proposed in [20], where the technique based on the public information was used to derive a symmetric key for authentication. Following this work, a series of DKE protocols were proposed in [10, 21, 22].

When a two-party DKE protocol is extended to a group setting, there may be some troubles [23, 24]. If there exists malicious insiders, the use of a common symmetric key for deniable authentication is infeasible as the malicious participants may impersonate other participants [3]. A solution provided in [3] is to make use of Schnorr's zero-knowledge identification scheme [25]. This approach needs 4 rounds to complete the establishment of session key and its efficiency was improved by Zhang et al. [26], which reduced the communication round to 3. Some approaches [27–29] transform the passively secure group key establishment to an actively secure one by adding one more round and the deniability was achieved as well.

DGKA protocols can be applied to VANETs to provide the security and privacy protection for vehicles. In recent years, wireless networks (WN) have achieved rapid development [30], and their security issues have been extensively studied [31–33]. As a kind of WN, the security of VANETs should be taken seriously due to the high risk. Some related schemes for secure communication in VANETs have been presented [34–36]. Huang et al. [35] proposed a communication scheme based on GKA protocol that the roadside unit generated session key for adjacent vehicles in batches. This scheme could effectively reduce the cost of computation and communication. In [36], a representative selected from the adjacent vehicles was arranged to communicate with the roadside unit, thus making that the security of other vehicles guaranteed. Nevertheless, the public verification in these works leaks the privacy of vehicles. Hence, the deniable group key agreement is necessary to apply to privacy-preserving communication for VANETs.

1.2. Contribution. We focus on the full deniability of the authenticated group key agreement. We provide a generic transformation from unauthenticated GKA to DGKA without increasing any additional communication rounds. Moreover, our deniability does not require rewinding steps; thus it holds even in the concurrent environment such as Internet. We also do not depend on any strong assumptions to reach the full deniability. The contribution of this work is as follows.

(1) We present a generic transformation from an unauthenticated GKA (named as DB protocol [37]) to a deniable (authenticated) GKA. The existing works achieve the full deniability by the rewinding steps, KEA assumption (which is strong), or the public random oracles. It results in inefficiency or insecurity (strong assumption). Our approach does not resort to these ways. We do not require that the underlying primitive is PA secure and the random oracles are not necessary.

(2) Our work achieves the concurrent deniability without timing constraint. In concurrent setting, the adversary can open and schedule sessions arbitrarily. Indeed, our simulation does not require extracting the witness by rewinding steps. Thus, the concurrent session attack (i.e., adversaries schedule the executions or delay messages in arbitrary ways) does not work in our scheme.

(3) We realize the optimal communication complexity. Our transformation does not increase the round of the unauthenticated one (original DB) although it realizes the property of privacy-preserving authentication in GKA, while the related works such as [3, 26]

Round 1: Participant U_i performs the following steps:
(1) Choose $x_i \in Z_q^*$ and compute $X_i = g^{x_i}$.
(2) Broadcast message (U_i, X_i).
Round 2: Upon receiving messages (U_{i-1}, X_{i-1}) and (U_{i+1}, X_{i+1}), each U_i does as following:
(1) Compute $Y_i^L = X_{i-1}^{x_i}$, $Y_i^R = X_{i+1}^{x_i}$, $Y_i = Y_i^R / Y_i^L$.
(2) Broadcast message (U_i, Y_i).
Session Key Generation: Upon receiving all messages $(U_j, Y_j)_{j\in\{1,\ldots,n\}, j\neq i}$, each U_i carries out the following steps:
(1) Compute orderly $\widehat{Y}_{i+1}^R = Y_{i+1} \cdot Y_i^R$, $\widehat{Y}_{i+2}^R = Y_{i+2} \cdot \widehat{Y}_{i+1}^R, \ldots, \widehat{Y}_{i+(n-1)}^R = Y_{i+(n-1)} \cdot \widehat{Y}_{i+(n-2)}^R$.
(2) Check $Y_i^L \stackrel{?}{=} \widehat{Y}_{i+(n-1)}^R$. If it is true, continue; Otherwise, abort.
(3) Generate the session key $\mathrm{sk} = \widehat{Y}_1^R \cdot \widehat{Y}_2^R \cdot \ldots \cdot \widehat{Y}_n^R = g^{x_1x_2 + x_2x_3 + \cdots + x_nx_1}$.

Algorithm 1: DB-GKA protocol without authentication.

have to increase the additional rounds to obtain the deniability.

(4) We also design a privacy-preserving communication protocol for VANETs using the proposed DGKA protocol. In this communication protocol, the vehicles share their information without leaking any identity privacy and without leaving any evidence in the transcript of authentication.

Organization. This paper is organized as follows. Section 2 introduces some preliminaries which are the building blocks in our protocol. Section 3 describes the adversarial model and related security definitions of the DGKA. We propose an efficient DGKA protocol with 2 rounds in Section 4. The security of DGKA protocol is proven and the performance is analyzed in Section 5. We design a privacy-preserving protocol for VANETs in Section 6. Section 7 concludes this work.

2. Preliminaries

We show the notations and introduce the building blocks in this section.

2.1. Notations. The notations used in this paper are listed in Notations in DGKA Protocol.

2.2. DB-GKA Protocol. Our deniable group key agreement (DGKA) protocol is developed on the basis of Dutta-Barua (DB) GKA protocol [37], which is a 2-round unauthenticated GKA protocol. It is a variant of [38]. We now review the original DB-GKA protocol [37]. Each participant U_i chooses x_i as its short-term private key, computes $X_i = g^{x_i}$, and broadcasts X_i in the first round. In Round 2, upon the receipt of messages (X_{i-1}, X_{i+1}), U_i computes $Y_i = f(X_{i-1}, X_{i+1})$ and broadcasts it. Finally, each U_i generates the common session key sk with the received Y_i and its secret x_i. The concrete DB-GKA protocol is presented in Algorithm 1. The security of DB-GKA protocol has been proven in [37].

2.3. Ring Signature with 2 Members. Our deniable group key agreement protocol provides the deniability based on the ring signature with 2 members. Now we introduce the syntax and the security properties of the ring signature with 2 members.

The ring signature scheme was used to sign a message privately. Given a valid ring signature σ with respect to a message M and a set of public keys $\mathcal{PK} = \{\mathrm{PK}_1, \ldots, \mathrm{PK}_n\}$, any verifier cannot decide which member in set $\mathcal{PK}$ is the actual signer.

We consider the ring signature with n members where $n = 2$. The syntax of the ring signature is as follows.

(1) A probabilistic key generation algorithm KGen: given the security parameter κ, output the key pair $(\mathrm{PK}_i, \mathrm{SK}_i)$ for U_i $(i = 1, 2)$. That is, $(\mathrm{PK}_i, \mathrm{SK}_i) \leftarrow \mathrm{KGen}(1^\kappa)$.

(2) A probabilistic ring signing algorithm RSig: given a message M, two public keys $(\mathrm{PK}_1, \mathrm{PK}_2)$, and a private (signing) key of U_k $(k \in \{1, 2\})$, output the ring signature σ. That is, $\sigma \leftarrow \mathrm{RSig}(M, (\mathrm{PK}_1, \mathrm{PK}_2); \mathrm{SK}_k)$.

(3) A deterministic verification algorithm RVer: given the ring signature σ, the message M, and the two public keys $(\mathrm{PK}_1, \mathrm{PK}_2)$, determine whether σ is valid with respect to $(M, \mathrm{PK}_1, \mathrm{PK}_2)$. That is, check $1 \stackrel{?}{=} \mathrm{RVer}(\sigma, M, \mathrm{PK}_1, \mathrm{PK}_2)$.

The properties of a secure ring signature with 2 members contain the *unconditional anonymity* and *unforgeability* as follows.

(i) *Unconditional Anonymity*. The distributions of the two ring signatures $\sigma_1 \leftarrow \mathrm{RSig}\,(M, (\mathrm{PK}_1, \mathrm{PK}_2); \mathrm{SK}_1)$ and $\sigma_2 \leftarrow \mathrm{RSig}(M(\mathrm{PK}_1, \mathrm{PK}_2); \mathrm{SK}_2)$ are `statistic,` identical. It implies that, given a ring signature σ with respect to $(M, \mathrm{PK}_1, \mathrm{PK}_2)$, no one can decide the signer although the private keys $(\mathrm{SK}_1, \mathrm{SK}_2)$ are revealed.

(ii) *Unforgeability*. A forger without the signing key SK_1 or SK_2 forges a ring signature $\widehat{\sigma}$ with respect to $(M, \mathrm{PK}_1, \mathrm{PK}_2)$. The probability that $1 \leftarrow \mathrm{RVer}(\widehat{\sigma}, M, \mathrm{PK}_1, \mathrm{PK}_2)$ is negligible.

3. Model of Deniable Group Key Agreement Protocol

3.1. Syntax. The syntax of the deniable group key agreement (DGKA) protocol is as follows. Let $\mathcal{U} = \{U_1, \ldots, U_n\}$

denote the set of n potential participants who would like to build a common session key to communicate securely. Each participant $U_i \in \mathscr{U}$ has a private/public key pair $(\mathrm{SK}_i, \mathrm{PK}_i)$ and the public keys are authenticated and can be accessed by any member. The DGKA protocol may be executed among any subsets of $\mathscr{U}$ at any time. At the end of this execution, the common session key is built. Each participant is convinced of the identity of his partners. In addition, all of them can also deny the involvement in this conversation of this session.

3.2. Security Model. We formalize the underlying adversarial behaviors in this subsection.

(i) Execute($\mathrm{pid}_i^{l_i}$): this query models the passive attacks in which the adversary can only eavesdrop the execution of protocol among the participants in $\mathrm{pid}_i^{l_i}$ and outputs the transcript of the session $\mathrm{sid}_i^{l_i}$. The transcript consists of the messages that are exchanged during the honest execution of the protocol.

(ii) Send(d, i, l_i, M): this query models the active attacks which the adversary can arbitrarily eavesdrop, delay, modify, and insert on any message M. The output of this query is the reply generated by instance $\pi_i^{l_i}$. When $d = 0$, the query initializes the execution of the instance $\pi_i^{l_i}$.

(iii) Reveal(i, l_i): if instance $\pi_i^{l_i}$ has successfully accepted the session key $\mathrm{sk}_i^{l_i}$, then $\mathrm{sk}_i^{l_i}$ is returned. Otherwise, NULL is returned.

(iv) Corrupt(i): the long-term private key of participant U_i is returned, and the future action will be fully taken by adversary. This query implies that there exists the malicious insiders.

(v) Test(i, l_i): the query is allowed only once. The queried instance $\pi_i^{l_i}$ must be fresh and $\mathrm{sk}_i^{l_i}$ is not NULL. Furthermore, this session as well as its partnered session should not be issued when a Corrupt query or Reveal query occurs. When the Test query occurs, a bit $b \in \{0, 1\}$ is randomly chosen. The session key $\mathrm{sk}_i^{l_i}$ is returned if $b = 1$; otherwise a random value of the same length is returned if $b = 0$.

(vi) Response: the adversary outputs a guess bit b'. We say that the adversary wins the game if $b' = b$. Let $\mathrm{Succ}_{\mathscr{A}}$ denote the event that the adversary wins the game and $\mathrm{Adv}_{\mathscr{A}}$ denote the advantage of the adversary by $\mathrm{Adv}_{\mathscr{A}} = |\Pr[\mathrm{Succ}_{\mathscr{A}}] - 1/2|$.

Freshness. An instance $\pi_i^{l_i}$ is fresh if none of the following happens: (1) A Reveal(i, l_i) query or a Reveal(j, l_j) query happens, where $\pi_i^{l_i}$ is partnered with $\pi_j^{l_j}$. (2) A Corrupt(j) query happens, where $U_j \in \mathrm{pid}_i^{l_i}$.

Partnering. The instances $\pi_i^{l_i}$ and $\pi_j^{l_j}$ are said to be partnered if $\mathrm{sid}_i^{l_i} = \mathrm{sid}_j^{l_j}$ and $\mathrm{pid}_i^{l_i} = \mathrm{pid}_j^{l_j}$.

Communicational Networks. We assume that our protocol is executed in the broadcasting channel; thus the adversaries can arbitrarily eavesdrop, delay, modify, and insert any message.

A secure DGKA protocol should satisfy the *correctness*, *deniability*, *authentication*, and *secrecy*.

Correctness. This property states that the protocol will establish a session key without adversarial interference. The DGKA protocol is said to be correct if for any pair of instances $\pi_i^{l_i}$ and $\pi_j^{l_j}$ ($i, j = 1, \dots, n$ and $i \neq j$), which have been accepted with $\mathrm{sid}_i^{l_i} = \mathrm{sid}_j^{l_j}$ and $\mathrm{pid}_i^{l_i} = \mathrm{pid}_j^{l_j}$, the condition $\mathrm{sk}_i^{l_i} = \mathrm{sk}_j^{l_j} \neq \mathsf{NULL}$ holds.

Deniability. This deniability states that the adversary cannot convince anyone that the honest participants have indeed joined in some sessions. Let $\mathscr{A}_d$ be the adversary that violates the deniability. We use the *simulation* paradigm to formally define the deniability. We construct a simulator $\mathscr{S}$ that is a probabilistic polynomial time (PPT) Turing machine. The simulator $\mathscr{S}$ can answer all queries from the adversary $\mathscr{A}_d$, and its inputs only involve the public information and the long-term private keys of the corrupted participants. Let $\mathtt{View}_{\mathscr{S}}$ denote the outputs of the adversary $\mathscr{A}_d$ after interacting with the simulator $\mathscr{S}$. Let $\mathtt{View}_{\mathtt{R}}$ denote the outputs of the adversary $\mathscr{A}_d$ in the real world. The protocol is said to be deniable if, for any PPT adversary $\mathscr{A}_d$ and the distinguisher $\mathscr{D}$ with unbounded computation, there exists a simulator $\mathscr{S}$, such that $|\Pr[\mathscr{D}(\mathtt{View}_{\mathscr{S}}) = 1] - \Pr[\mathscr{D}(\mathtt{View}_{\mathtt{R}}) = 1]| = \mathrm{negl}(\kappa)$.

Authentication. The authentication of the protocol guarantees that the received messages of the participants come from the intended participants. If an adversary $\mathscr{A}_a$ that may even be a malicious insider can impersonate an uncorrupted participant U_i and succeed to accomplish the protocol, then we say the adversary violates the authentication of DGKA protocol. We use Forge to denote the event that the adversary succeeds in cheating the honest participants. The protocol is said to be authenticated if $\Pr[\mathrm{Forge}] \approx \mathrm{negl}(\kappa)$ for any PPT adversary.

Secrecy. The secrecy of the protocol states that the session key is known only to participants but is random to outsiders. Formally, let $\mathscr{A}$ be the adversary that violates the secrecy and $\mathrm{Succ}_{\mathscr{A}}$ denote the success of $\mathscr{A}$ in the Test query, who decides the session key from a random value successfully. We say the protocol meets the secrecy if $\Pr[\mathrm{Succ}_{\mathscr{A}}] \approx 1/2 + \mathrm{negl}(\kappa)$.

4. Our Deniable Group Key Agreement Protocol

We construct the deniable GKA protocol based on Dutta-Barua (DB) GKA protocol [37], which is elaborated in Section 2. Our DGKA protocol achieves the deniable authentication by employing a ring signature with 2 members. We first give the high level description of our DGKA protocol.

Let $(\mathrm{SK}_i, \mathrm{PK}_i)$ denote the private/public key pair for the participant U_i and n is the number of the participants of this session.

Round 1: Participant U_i performs the following steps:

(1) Choose $x_i, t_i \in Z_q^*$ and compute $X_i = g^{x_i}, T_i = g^{t_i}$.

(2) Broadcast message $M_i^1 = (U_i, X_i, T_i)$.

Round 2: Upon the receipt of all messages $\{M_j^1\}_{j\in\{1,\ldots,n\}, j\neq i}$, U_i parses X_{i-1}, X_{i+1} and $\{T_j\}_{j\in\{1,\ldots,n\}, j\neq i}$. Next, U_i executes the following operations:

(1) Compute $Y_i^L = X_{i-1}^{x_i}, Y_i^R = X_{i+1}^{x_i}, Y_i = Y_i^R/Y_i^L, T = \prod_{j=1}^{n} T_j$.

(2) Generate a two-member ring signature on the message $M = (X_1, \ldots, X_n, Y_i)$: $\sigma_i = \mathrm{RSig}(M, (\mathrm{PK}_i, T); \mathrm{SK}_i)$.

(3) Broadcast message $M_i^2 = (U_i, Y_i, \sigma_i)$.

Session Key Generation: Upon the receipt of all messages $\{M_j^2\}_{j\in\{1,\ldots,n\}, j\neq i}$, each U_i carries out the following steps:

(1) Compute orderly $\widehat{Y}_{i+1}^R = Y_{i+1} \cdot Y_i^R, \widehat{Y}_{i+2}^R = Y_{i+2} \cdot \widehat{Y}_{i+1}^R, \ldots, \widehat{Y}_{i+(n-1)}^R = Y_{i+(n-1)} \cdot \widehat{Y}_{i+(n-2)}^R$. Check $Y_i^L \stackrel{?}{=} \widehat{Y}_{i+(n-1)}^R$. If it is true, continue; Otherwise, abort.

(2) Check $1 = \mathrm{RVer}(\sigma_i, M, \mathrm{PK}_j, T)$ hold or not for $j = 1, \ldots, n$ and $j \neq i$. If it fails to any participant, abort; Otherwise, continue.

(3) Generate the session key $\mathrm{sk} = \widehat{Y}_1^R \cdot \widehat{Y}_2^R \cdot \ldots \cdot \widehat{Y}_n^R = g^{x_1x_2+x_2x_3+\cdots+x_nx_1}$.

Algorithm 2: Our deniable group key agreement protocol.

Considering a ring signature scheme with two members: a real participant and a logic entity. In the first round, each participant follows DB-GKA protocol to generate X_i. Besides that, each one also produces another group element T_i. The product of each T_i is viewed as the public key of the logic entity. Therefore, in the second round, each participant gathers all T_i to result T. Then each one uses its own public key PK_i and the logic public key T to form a ring to generate a ring signature on the message $(X_1, \ldots, X_n, Y_i)$ with its signing key SK_i. The corresponding private key of the logic public key T is unknown to any participant and the third party. Thus, a valid ring signature implies that the authentication to $(X_1, \ldots, X_n, Y_i)$ can be completed only by the participant U_i. The authentication is achieved. On the other hand, the simulator can simulate the value T by its random choice of the exponent t to get $T = g^t$. Then, the simulator produces a ring signature $\sigma' = \mathrm{RSig}(M, (\mathrm{PK}_i, T); t)$ with the "private key" of T. By the unconditional anonymity property of the ring signature, the two distributions of $\sigma = \mathrm{RSig}(M, (\mathrm{PK}_i, T); \mathrm{SK}_i)$ and $\sigma' = \mathrm{RSig}(M, (\mathrm{PK}_i, T); t)$ are statistic, identical, where the former one is the real transcription. Therefore, the simulation is perfect and the deniability is achieved. Since the rewinding steps are not necessary in the simulation, the deniability can also hold in the concurrent setting. We give a detailed description of our protocol in Algorithm 2.

Remark 1. The ring signature is with 2 members. One is the participant U_i, and the other one is a logic entity whose public key is $T = \prod_{j=1}^{n} T_j$. Obviously, the private key of T is $t = \sum_{j=1}^{n} t_j$ and it is unknown to anybody. In the real conversation, U_i uses its private key SK_i to generate the ring signature σ. Since σ is only bounded to 2 public keys and one of the public key is logic with unknown secret, the partner can be convinced of U_i's signing. The authentication is completed. Meanwhile, in the simulation, the simulator simulates t (as no secret value is required) to produce the ring signature. Obviously, this simulation is perfect without any rewinding steps; concurrent deniability is realized.

5. Security and Performance

In this section, we analyze the security and performance of our protocol. Since the verification of *correctness* of our protocol is straightforward, in what follows we will prove that our protocol meets the other three properties: *deniability*, *authentication*, and *secrecy*, which have been presented in the security model. Then we give the performance comparisons of the related deniable key agreements regarding the communication round and the deniability.

5.1. Security

5.1.1. Deniability. This property states that all the participants can deny the fact that they have joined in the generation of the session key. We use the *simulation* fashion to prove that our protocol satisfies the deniability. That is, if a simulator without any participant's secret can simulate the transcript and the simulated transcript is indistinguishable from the real one, then we say the deniability is proven. Formal proof is presented as follows.

Theorem 2. *The DGKA protocol is concurrently deniable if the underlying ring signature is secure.*

Proof. In order to prove our protocol satisfying the deniability, we have to show the real view and the simulated view are indistinguishable. Formally, we construct a simulator $\mathcal{S}$, whose inputs involve the public information and the long-term private keys of the corrupted participants. $\mathcal{A}_d$ is an adversary that violates the deniability of the protocol. Use $\mathtt{View}_{\mathrm{R}}$ to denote the view of $\mathcal{A}_d$ in the real conversation and $\mathtt{View}_{\mathcal{S}}$ to denote the view of $\mathcal{A}_d$ in the simulated setting performed by $\mathcal{S}$. We show that any distinguisher $\mathcal{D}$

with unbounded computation cannot distinguish $\mathtt{View}_{\mathrm{R}}$ and $\mathtt{View}_{\mathscr{S}}$.

With the inputs of $\{\mathrm{PK}_i\}$ and the long-term private keys of the corrupted participants, $\mathscr{S}$ simulates the Send, Corrupt, and Reveal queries for $\mathscr{A}_d$ as follows.

(i) $\mathrm{Send}(0, i, l_i, M)$: $\mathscr{S}$ normally performs protocol and answers the query as it does not require any secrets. $\mathscr{S}$ randomly chooses $x_i, t_i \in Z_i^*$ to compute X_i, T_i, respectively. Then, $\mathscr{S}$ broadcasts message $M_i^1 = (U_i, X_i, T_i)$ and records $\mathrm{stat}_i^{l_i} = (x_i, X_i, t_i, T_i)$.

(ii) $\mathrm{Send}(1, i, l_i, M)$: $\mathscr{S}$ checks if U_i has been corrupted.

  (a) If U_i has been corrupted, $\mathscr{S}$ with the known private key SK_i simulates M_i^2 normally.

  (b) If U_i is uncorrupted, $\mathscr{S}$ retrieves x_j, t_j from $\mathrm{stat}_j^{l_j}$ to compute Y_j (where $j = 1, \ldots, n$), $t = \sum_{j=1}^{n} t_j$, and $T = g^t$. Then $\mathscr{S}$ produces a ring signature $\sigma_i = \mathrm{RSig}(M, (\mathrm{PK}_i, T); t)$. $\mathscr{S}$ updates $\mathrm{stat}_i^{l_i} = (x_i, t_i, X_i, T_i, Y_i, \sigma_i)$.

(iii) $\mathrm{Send}(2, i, l_i, M)$: $\mathscr{S}$ normally answers the query no matter whether U_i has been corrupted or not as there is no secret required.

(iv) $\mathrm{Reveal}(i, l_i)$: $\mathscr{S}$ computes the session key $\mathrm{sk}_i^{l_i}$ according to the protocol and returns it to $\mathscr{A}_d$.

(v) $\mathrm{Corrupt}(i)$: $\mathscr{S}$ returns the private key SK_i of participant U_i and the fact that U_i is corrupted is marked.

Now, we argue that $\mathtt{View}_{\mathrm{R}}$ and $\mathtt{View}_{\mathscr{S}}$ are perfectly identical. It is obvious that $\mathscr{S}$ does not introduce any difference from the view of real one when $\mathrm{Send}(0, i, l_i, M)$, $\mathrm{Send}(2, i, l_i, M)$, $\mathrm{Reveal}(i, l_i)$, and $\mathrm{Corrupt}(i)$ are asked. Let us consider $\mathrm{Send}(1, i, l_i, M)$. In the real transcript, $\mathrm{Send}(1, i, l_i, M)$ is performed using U_i's private key SK_i. In the simulation, this oracle is answered using t, which is the private key of the logic party (whose public key is $T = g^t$). This is a ring signing with U_i and the logic party. Since the underlying ring signature scheme with two members is secure, it implies that the unconditional anonymity property holds. If $\mathrm{Send}(1, i, l_i, M)$ introduces any difference, which means the ring signature under SK_i and the ring signature under t can be distinguishable, obviously, it breaks the unconditional anonymity of this ring signature scheme. It is a contradiction. □

5.1.2. Authentication. Authentication states that each U_i can ensure that the message it received is authenticated by the intended partner. This property can prevent the man-in-middle attack which exists in the unauthenticated key agreement protocol. In our protocol, we apply the ring signature with two members to preserve the authentication. Indeed, the generated ring signature σ_i is bounded to two public keys PK_i and T. Due to the unforgeability of the ring signature, anyone who knows SK_i or t can generate a valid signature. Given a valid σ_i, the partner is convinced that $M = (X_1, \ldots, X_n, Y_i)$ which is used to generate the common session key is signed by U_i as t is unknown to anyone. Obviously, our protocol is authenticated due to the unforgeability of the underlying ring signature scheme.

5.1.3. Secrecy. This property ensures the security of the session key. That is, any member without participating in the session cannot obtain the session key. Obviously, our DGKA protocol satisfies the secrecy if DB-GKA protocol produces the session key securely. It is easy to see that our DGKA protocol equals the original DB-GKA protocol only except that we provide a ring signing on $(X_1, \ldots, X_n, Y_i)$ in DGKA. We denote the game $\mathtt{G}_0$ as the environment of DB-GKA protocol and the game $\mathtt{G}_1$ as the environment of our DGKA. Let Forge be the event that $\mathscr{A}$ succeeds in forging a valid message after Round 2. The difference between the games $\mathtt{G}_0$ and $\mathtt{G}_1$ is that the challenger in $\mathtt{G}_1$ would stop the simulation when the event Forge occurs. However, Pr[Forge] is negligible as the authentication property states. Therefore, the secrecy of our DGKA protocol can be reduced to the secrecy of DB-GKA, which is proven in [37].

5.2. Performance. The obvious advantage of our construction is the optimal communication round. We transfer the unauthenticated DB-GKA protocol to the deniable GKA without increasing round. While other related DGKA protocols are more than 2 rounds.

One DGKA protocol [3] is based on Schnorr's zero-knowledge identification scheme; the participants make the commitments in Round 1. Next, an unauthenticated GKA protocol is executed in Rounds 2 and 3. The deniable authentication is achieved in Round 4. It needs 4 rounds to complete the protocol. Similarly, in [26], the participants also make commitments in Round 1. At the same time, the participants begin to execute the unauthenticated GKA protocol in this round. Finally, the deniable authentication is executed in Round 3. It is easy to see that the deniable authentication depends on the generated session key in [3, 26]. This is the reason that these two protocols require more rounds than the unauthenticated GKA to realize the deniable authentication.

Our protocol makes use of the unconditional anonymity of the ring signature to achieve the concurrent deniability. This ring signature is bounded to 2 members. The one is the actual participant and the other one is a logic party. This logic public key is accumulated by all participants with its own secret in Round 1. Then each participant uses the logic public key and its own public key to form a ring and signs the elements which are used to generate the common session key in Round 2. Obviously, the deniability is no longer dependent on the session key. Therefore, our work does not increase the communication round of the unauthenticated GKA.

We also focus on the concurrent deniability. However, both [3, 26] depend on the rewinding steps to simulate the transcript. Therefore, the deniability cannot hold in the concurrent setting. Some other deniable authentication protocols or deniable key exchange protocols which realize the concurrent deniability depend on the strong assumptions/primitives, such as KEA assumption, public random

Table 1: Comparisons of deniable key agreement protocols.

Scheme	Scale	Round	Concurrency	RO	Deniability realization
[3]	group	4	×	√	Rewinding
[26]	group	3	×	-	Rewinding
[10]	2-party	2	√	√	KEA assumption
[22]	2-party	2	√	√	Public RO
Proposed	group	2	√	-	Ring signature

oracle, or timed commitment/encryption to extract the witness for the simulations. Compared with them, our DGKA protocol is not restricted to these limitations.

The comparisons of the related protocols with deniability are listed in the Table 1.

6. A Privacy-Preserving Communication Protocol for VANETs

In this section, we design a privacy-preserving communication protocol for VANETs by using the proposed deniable group key agreement protocol. Our protocol guarantees the secure communication between vehicles and vehicles and vehicles and roadside unit. VANETs are composed of Trusted Authority (TA), roadside unit (RSU, which is the infrastructure), and On-board Units (OBUs, with which vehicles are equipped). Our security model for privacy-preserving VANETs is as follows.

(i) Authentication: in the VANETs environment, RSU and OBUs should ensure that only legitimate (certificated by TA) vehicles can join this networks. Similarly, RSU should be also authenticated by vehicles in order to prevent pseudo base stations.

(ii) Anonymity: OBUs receive the information without knowing the sender identity, but only to confirm that this message is from an authenticated group.

(iii) Privacy: the conversations among the OBUs do not leave any paper trail. This "off-the-record" property prevents the shared information from being maliciously used.

(iv) Secrecy: during the process of communication, the sent messages are only known to receivers but are random to any third parties.

Our privacy-preserving communication protocol for VANETs is mainly divided into three steps. The first step is to initialize a group of VANETs. Then, OBUs and RSU in this group authenticate mutually to generate a session key. Finally, they communicate with each other with this session key under an authenticated and privacy-preserving environment.

Let U_i be one of vehicles and U_R be RSU. $(\mathrm{SK}_i, \mathrm{PK}_i) = (s_i, g^{s_i})$ denote the private/public key pair for vehicle U_i; $(\mathrm{SK}_0, \mathrm{PK}_0) = (s_0, g^{s_0})$ denote the private/public key pair for U_R. $H : \{0,1\}^* \rightarrow \{0,1\}^l$, where l is the length of a message. A detailed protocol is given as follows.

Initialization Step. The members of a group of VANETs are decided.

(i) U_R randomly chooses id as the session ID and forms a group $\mathscr{R}$ by using its public key PK_0 and the public keys of adjacent vehicles $\{\mathrm{PK}_i\}$. Finally, broadcast message $V_{\mathrm{init}} = \mathrm{id} \parallel \mathscr{R}$.

Authentication Step. The identities of members are authenticated.

(i) *Round 1 (OBUs and RSU)*. Choose x_i, t_i, and compute $X_i = g^{x_i}$, $T_i = g^{t_i}$. Broadcast message $V^1_{\mathrm{auth}} = \mathrm{id} \parallel \mathrm{PK}_i \parallel X_i \parallel T_i$.

(ii) *Round 2 (OBUs and RSU)*. Compute Y_i and σ_i as the proposed DGKA protocol (described in Algorithm 2). Broadcast message $V^2_{\mathrm{auth}} = \mathrm{id} \parallel Y_i \parallel \sigma_i$.

(iii) *Key Generation (OBUs and RSU)*. Authenticate the identities of other members and get the session key sk as in Algorithm 2.

Communication Step. With this session key sk, all the members in this group $\mathscr{R}$ can communicate securely. There are two cases in this step, including broadcast from U_R or U_i to all members and communication from U_R to U_i, U_i to U_R, or U_i to U_j.

(i) Broadcast (one-to-many):

(a) RSU or OBUs send message m_b: compute $v = H(\mathrm{id}, \mathrm{sk})$ and $e = m_b \oplus v$. Broadcast message $V_{\mathrm{bro}} = \mathrm{id} \parallel e$.

(b) RSU and OBUs recover m_b: compute $v = H(\mathrm{id}, \mathrm{sk})$ and $m_b = e \oplus v$.

(ii) Communication (one-to-one):

(a) RSU or OBUs send m_c to U_i: choose $r \leftarrow Z_q^*$ and compute $R = g^r$, $v = H(\mathrm{id}, R, {\mathrm{PK}_i}^r, \mathrm{sk})$, and $e = m_c \oplus v$. Broadcast message $V_{\mathrm{com}} = \mathrm{id} \parallel e \parallel \mathrm{PK}_i \parallel R$.

(b) U_i recovers m_c: compute $v = H(\mathrm{id}, R, R^{s_i}, \mathrm{sk})$ and $m_c = e \oplus v$.

By employing the proposed DGKA protocol, each receiver can identify the source of the received information without knowing the actual sender by using the session key sk. Moreover, this session key sk can be simulated by anyone; the vehicles involved in the above communication can deny this. There is no paper trail; thus the vehicle privacy is protected.

7. Conclusions

This paper presents a 2-round fully deniable group key agreement protocol. We provide a novel approach to transfer an unauthenticated GKA to a deniable GKA without increasing round. The transcript simulation does not require the rewinding steps; thus our deniability also holds even in the concurrent setting. We also design a privacy-preserving communication protocol for VANETs using the proposed DGKA protocol.

Notations in DGKA Protocol

κ: The security parameter
G: A multiplicative group of prime order q
g: The generator of group G
U_i: The ith participant
PK_i: U_i's public key
SK_i: U_i's private key
$\pi_i^{l_i}$: A session of U_i called an instance—a participant may have many instances and denotes the instance l_i of U_i as $\pi_i^{l_i}$
$\mathrm{sid}_i^{l_i}$: The session ID of instance $\pi_i^{l_i}$
$\mathrm{pid}_i^{l_i}$: A set containing the identities of the participants in the group with whom $\pi_i^{l_i}$ intends to establish a session key, including U_i
$\mathrm{stat}_i^{l_i}$: The current state of instance $\pi_i^{l_i}$
$\mathrm{sk}_i^{l_i}$: The common key generated by instance $\pi_i^{l_i}$ after the protocol finished
$\mathrm{negl}(\kappa)$: A negligible function for the security parameter κ.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61402376, U1433130), the Ministry of Education "Chunhui Plan" (Z2016150), and the National Key R&D Program of China (2017YFB0802300, 2017YFB0802000).

References

[1] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.

[2] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A conference key distribution system," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 28, no. 5, pp. 714–720, 1982.

[3] J. Bohli and R. Steinwandt, "Deniable Group Key Agreement," in *Progress in Cryptology - VIETCRYPT 2006*, vol. 4341 of *Lecture Notes in Computer Science*, pp. 298–311, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[4] D. Dolev, C. Dwork, and M. Naor, "Nonmalleable cryptography," *SIAM Journal on Computing*, vol. 30, no. 2, pp. 391–437, 2000.

[5] C. Dwork, M. Naor, and A. Sahai, "Concurrent zero-knowledge," *Journal of the ACM*, vol. 51, no. 6, pp. 851–898, 2004.

[6] M. Di Raimondo and R. Gennaro, "New approaches for deniable authentication," *Journal of Cryptology*, vol. 22, no. 4, pp. 572–615, 2009.

[7] M. Di Raimondo, R. Gennaro, and H. Krawczyk, "Deniable authentication and key exchange," in *Proceedings of the CCS 2006: 13th ACM Conference on Computer and Communications Security*, pp. 400–409, Alexandria, Va, USA, November 2006.

[8] M. Bellare and A. Palacio, "Towards plaintext-aware public-key encryption without random oracles," in *Advances in cryptology—ASIACRYPT 2004*, vol. 3329 of *Lecture Notes in Comput. Sci.*, pp. 48–62, Springer, Berlin, 2004.

[9] S. Jiang, "Deniable authentication on the internet (extended abstract)," in *Information security and cryptology*, vol. 4990 of *Lecture Notes in Comput. Sci.*, pp. 298–312, Springer, Berlin, 2008.

[10] A. C.-C. Yao and Y. Zhao, "Privacy-preserving authenticated key-exchange over internet," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 125–140, 2014.

[11] A. C.-C. Yao and Y. Zhao, "OAKE: A new family of implicitly authenticated Diffie-Hellman protocols," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*, pp. 1113–1128, Germany, November 2013.

[12] H. Tian, X. Chen, and W. Susilo, "Deniability and forward secrecy of one-round authenticated key exchange," *The Journal of Supercomputing*, vol. 67, no. 3, pp. 671–690, 2014.

[13] S. Zeng, Y. Chen, S. Tan, and M. He, "Concurrently deniable ring authentication and its application to LBS in VANETs," *Peer-to-Peer Networking and Applications*, vol. 10, no. 4, pp. 844–856, 2017.

[14] S. Jiang, "Timed encryption with application to deniable key exchange," *Theoretical Computer Science*, vol. 560, no. part 2, pp. 172–189, 2014.

[15] L. Harn and J. Ren, "Design of fully deniable authentication service for E-mail applications," *IEEE Communications Letters*, vol. 12, no. 3, pp. 219–221, 2008.

[16] W. Susilo and Y. Mu, "Non-interactive deniable ring authentication," in *Information security and cryptology—ICISC 2003*, vol. 2971 of *Lecture Notes in Comput. Sci.*, pp. 386–401, Springer, Berlin, 2004.

[17] W. Diffie, W. Diffie, and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[18] W. Mao and K. G. Paterson, "On the plausible deniability feature of internet protocols," Tech. Rep., 2002.

[19] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in cryptology—EUROCRYPT 2001 (Innsbruck)*, vol. 2045 of *Lecture Notes in Comput. Sci.*, pp. 453–474, Springer, Berlin, 2001.

[20] C. Boyd, W. Mao, and K. G. Paterson, "Deniable authenticated key establishment for Internet Protocols," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 3364, pp. 255–271, 2005.

[21] M.-H. Lim, S. Lee, Y. Park, and S. Moon, “Secure deniable authenticated key establishment for Internet protocols,” in *Proceedings of the 2nd International Conference on Information Security and Assurance, ISA 2008*, pp. 3–6, Republic of Korea, April 2008.

[22] S. Jiang and R. Safavi-Naini, “An efficient deniable key exchange protocol, in 12th International Conference on Financial Cryptography and Data Security,” in *Proceedings of the 12th International Conference on Financial Cryptography and Data Security*, vol. 2008, pp. 47–52.

[23] D. R. Brown, “Deniable authentication with RSA and multicasting,” *IACR Cryptology*.

[24] J. Katz and J. S. Shin, “Modeling insider attacks on group key-exchange protocols,” in *Proceedings of the CCS 2005 - 12th ACM Conference on Computer and Communications Security*, pp. 180–189, USA, November 2005.

[25] C.-P. Schnorr, “Efficient identification and signatures for smart cards,” in *Advances in cryptology*, vol. 435 of *Lecture Notes in Comput. Sci.*, pp. 239–252, Springer, New York, 1990.

[26] Y. Zhang, K. Wang, and B. Li, “A deniable group key establishment protocol in the standard model,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6047, pp. 308–323, 2010.

[27] Y.-Z. Zhang, H.-X. Xu, and B. Li, “Generic construction of deniable group key establishment from group key establishment,” *Tongxin Xuebao/Journal on Communication*, vol. 32, no. 3, pp. 143–149, 2011.

[28] K. Neupane, R. Steinwandt, and A. Suárez Corona, “Scalable deniable group key establishment,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7743, pp. 365–373, 2013.

[29] Y. Chen, L. Lu, S. Zeng, and M. He, “Deniable authentication for multiparty protocol,” *Chinese Journal of Network and Information Security*, vol. 2, no. 6, pp. 71–78, 2016.

[30] N. Zhang, P. Yang, S. Zhang et al., “Software defined networking enabled wireless network virtualization: Challenges and solutions,” *IEEE Network*, vol. 31, no. 5, pp. 42–49, 2017.

[31] D. Chen, N. Zhang, and Z. Qin, “S2M: a lightweight acoustic fingerprints based wireless device authentication protocol,” *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 88–100, 2017.

[32] Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[33] Q. Wang, D. Chen, and N. Zhang, “LACS: A Lightweight Label-Based Access Control Scheme in IoT-Based 5G Caching Context,” *IEEE Access*, 4027 pages, 2017.

[34] S. Zeng, M. He, M. Xia, X. Li, Y. Chen, and S. Tan, “Privacy-preserving location-based service based on deniable authentication,” in *Proceedings of the 9th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2016*, pp. 276–281, chn, December 2016.

[35] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, “ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, 2011.

[36] C.-Y. Chow, M. F. Mokbel, and X. Liu, “A peer-to-peer spatial cloaking algorithm for anonymous location-based service,” in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS ’06)*, pp. 171–178, ACM, November 2006.

[37] R. Dutta and R. Barua, “Provably secure constant round contributory group key agreement in dynamic setting,” *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 54, no. 5, pp. 2007–2025, 2008.

[38] M. Burmester and Y. Desmedt, “A secure and efficient conference key distribution system,” in *Advances in Cryptology — EUROCRYPT’94*, vol. 950 of *Lecture Notes in Computer Science*, pp. 275–286, Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.