

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/324047352>

EPAF: An Efficient Pseudonymous-Based Inter-vehicle Authentication Framework for VANET

Chapter in Communications in Computer and Information Science · March 2018

DOI: 10.1007/978-981-10-8890-2_18

CITATIONS

0

READS

53

5 authors, including:



Fei Wang

Chinese Academy of Sciences

67 PUBLICATIONS 2,355 CITATIONS

SEE PROFILE



Yong-Jun Xu


Chinese Academy of Sciences

243 PUBLICATIONS 4,188 CITATIONS

SEE PROFILE



EPAF: An Efficient Pseudonymous-Based Inter-vehicle Authentication Framework for VANET

Fei Wang¹ , Yifan Du^{1,2}, Yongjun Xu¹, Tan Cheng¹,
and Xiaoli Pan³

¹ Institute of Computing Technology, Chinese Academy of Sciences,
Beijing, China

{wangfei, duyifan, xyj, chengtan}@ict.ac.cn

² University of Chinese Academy of Sciences, Beijing, China

³ 93868 Troops, People's Liberation Army, Beijing, China
lavenderpanpan@163.com

Abstract. Road users are now able to retrieve safety information, computing task results and subscribing content through various vehicular ad hoc network (VANET) services. Most commonly used services are safety beacon, cloud computation, and content subscription. Road users concern more about data security than ever. Privacy preserving authentication (PPA) is one main mechanism to secure inter-vehicle messages. However, for historical reasons, PPA for three services are different and therefore hard to be unified and not lightweight enough. To improve the flexibility and efficiency of PPA for various VANET services, it is necessary to securely authenticate messages preserving privacy for individual service, but also to unify PPA processes of various services in one vehicle. Here we propose an Efficient Pseudonymous-based Inter-Vehicle Authentication Framework for various VANET services. Our novel framework employs three methods. Method No. 1 consists of a decentralized certificate authority (CA), which allows vehicles to communicate only if vehicles registering themselves. Method No. 2 adopts a three-stage mutual authenticating process, which adapts to different communicating models in various services. Method No. 3 we design a universal basic module that requires only lightweight hashing and MAC operations to accomplish the signing and verifying processes. To analyze the security performance of our EPAF, we use automated tool under symbolic approach. Our results strongly suggest that EPAF is secure, robust and adaptable in vehicular safety, as well as in content and cloud computation services. To analyze the performance of EPAF, we calculate benchmarks and simulate the network. Our results strongly suggest that EPAF reduces computation cost by 370–3500 times, decreases communication overhead by 45.98%–75.53% and CA need not to manage CRL compared with classical schemes. In conclusion our framework provides insights into how data privacy can be simultaneously protected using our EPAF, while also improving communication and computing speed even in high traffic density.

Keywords: VANET · Privacy preserving · Authentication framework
Cloud computation

1 Introduction

Various services now depend on vehicular networks like traditional safety service, arising content subscribe/publish service [1] and vehicular cloud computation service [3]. In safety service, VANET is supposed to collect and disseminate useful information through vehicle-to-vehicle (V2V) and vehicle-to-roadside unit (V2R) based on dedicated short-range communications (DSRC), supporting applications like Forward Collision Warning (FCW) and Blind Spot Warning (BSW) [2]. In content service, it on one hand receive content through 4G/5G, on the other store and forward content (road map around, media, POIs) through opportunistic communication. In computation service, a temporary vehicular computation cloud is formed on the fly by dynamically integrating resources from a cluster of vehicles like hundreds in parking lots, helping nearby users especially ones in cheaper cars to accomplish heavy computation tasks and gain actual benefits [6]. No matter what services VANET take on, security issues are inevitable.

Three types of security requirements are considered in this paper. (1) Basic type like resilience to eavesdropping, forgery and modification due to wireless communication. (2) Common type includes service data privacy preserving, unlinkability for multiple anonymous messages and tracking a vehicle (implied by level 3 privacy in [7]) and conditional traceability. (3) Dedicated type for various services. The first is core service data, apart from private identity information and locations, safety service focuses on vehicle motion status and road events, content service focuses on subscribe/publish information, computation service focuses on request, result and vehicle reputation. Unlinkability for identity and location is fundamental. But subscribe/publish information in content service, request/reply and reputation in computation service should be preserved. Apart from above, the performance should be redundant to adapt to arising different services.

Privacy preserving authentication (PPA) keeps an astonishing idea to ensure the security of VANET [3–10]. Some are based on public key infrastructure (PKI) and employs traditional digital signature technique to authenticate messages. Main downsides of such schemes are three: (1) Vulnerable to effortless Denial of Service (DoS) attack. (2) Collapse of scheme caused by high packet loss ratio. (3) Heavy burden on trusted authority (TA) performing certificate updating and revoking. We observe new trends in VANET. On-vehicle computer computes much faster, which leads to that road infrastructure needs not to be responsible for security tasks. As the vehicular network services are developing, the integration of hardware and software for PPA is inevitable rather than separated for different services. Lastly, urban three-dimensional traffic system incurs the need of enough redundant performance.

In this paper, we proposed an Efficient Pseudonymous-based Inter-Vehicle Authentication Framework for various VANET services (EPAF). For each vehicle a telematics device (TD) and a tamper proof device (TPD) are equipped acting as a distributed security proxy. Lightweight basic modules are decoupled and designed, each of which requires only several extreme lightweight operations to accomplish signing and verification. Moreover, the framework is able to adapt to one-way dissemination or a request/reply routine in various services. As we know, EPAF is the first strong-privacy-preserving and dos-resilient authentication framework compatible with

various services. Even compared with PPA scheme for safety service, hundreds of times efficiency redundancy is assured.

Followings are the advantages of EPAF:

- Level 3 privacy and strong privacy preservation: EPAF is able to guarantee level 3 privacy. Moreover, TD and TPD devices act like proxy, which leads to strong privacy that adversary is unable to pry into real identities of vehicles even if all RSUs are compromised.
- Reduced certificate overhead: In EPAF, a dynamic pseudo identity and a short MAC is carried within message. All CRL related overhead is eliminated and our EPAF achieves a decrease of 45.98%–75.53% in communication compared with other schemes.
- Compatible with various services: Through a mutual authentication mechanism, EPAF is able to satisfy security requirements which is compatible with one-direction message dissemination or bi-direction request/reply service. EPAF's integrity and unlinkability are compatible with different service data.
- Redundant performance efficiency: In user, message or service authentication, EPAF employs only hash operations coupled with MAC generation to accomplish the signing verification of service message. Subsequently achieving a significant increase of nearly 370–3500 times in computation compared with even safety schemes. This makes EPAF efficient enough for various services even in large traffic density.

Rest of this paper are as follows. Section 2 presents the related work about privacy authentication in VANET. In Sect. 3 system model and adversary model is defined. Then Sect. 4 gives full details of EPAF. In Sect. 5 we analyze the security of the scheme using symbolic approach. Section 6 gives performance analysis of EPAF, and Sect. 7 concludes the paper and look into the future work.

2 Related Work

Bulk of research work has been proposed to improve conditional privacy preservation for VANET in last decades, which is considered as candidate framework design references [5–9, 15, 18–21].

Pseudonymous-based schemes like BP [5], ECPP [7] and PTVC [6] link and update many pairs of private key and pseudonymous certificate to a pseudo identity. However, these schemes suffer from high overhead of certificate management and time window between certificate update (e.g. 1 min). In group based schemes, group members hide their real identities through a group identity. In schemes like [8, 9], CRL item checking needs two paring calculations, which is 104 times high than a string comparison in computation overhead, which makes computational cost for authentication too high to adapt to complex service handshake. Pervasive road side units (RSUs) are usually needed to maintain group, which makes group leader bottleneck. Hybrid schemes are ones which combine pseudonymous authentication protocol, digital signature, MAC and other authentication techniques to make a tradeoff [9]. In [8], group signature CRL checking is still expensive. TESLA++ [9] provides fast authentication and

non-repudiation and data integrity. However, it is unable to provide privacy preservation and conditional traceability. Batch verification based schemes are based batch verification. In RAISE and succeeding schemes [15], RSU was utilized as aggregator to verify messages from vehicles. The approach utilizes the IBE cryptography for generating secret keys for pseudo identities and thus avoids the use of certificates. Total computation overhead of vehicles are significantly reduced, but a vehicle still need store and wait for aggregation message from RSU. As for batch verification based ones, on one hand conditional traceability is not effective for replying messages shared back to vehicles, on the other the verification delay are inevitable and hard to deal.

The mentioned schemes have common bottlenecks of relying on infrastructure and not supporting mutual service message authentication. High overhead of signing, verification or certificate management by centered architecture is inevitable.

3 System Model

In this section, system model (network model and attack model), and design goals are presented.

3.1 Network Model

We consider a hybrid VANET scenario in which safety, content and computation services are supported. We divided the network model into two parts: common model for entities and service processing model for services.

In common model, TA is fully trusted by others. It has nearly unlimited computation and storage resources and accomplishes tasks as (1) RSUs and vehicle registration, (2) system key management, (3) conditional tracing. RSU communicates with TA directly through wired channel. It has large storage and powerful communication capability of 1 km to 3 km. It is responsible for message forwarding and distributed RSU aided key updating.

Every vehicle is equipped with an OBU as shown in Fig. 1. In safety service, OBU gathers information from vehicle sensors (e.g., GPS, forward, speed) and Event Data Recorder, packs safety beacon and broadcasts it through dedicated channel [2]. In content service, OBU acquires and provides content. For computation service, OBU helps generate requests and receive results. Telematics Device (TD) and Tamper Proof Device (TPD) cooperate with each other to ensure the security of the services. TPD is hard to hack into and used to store cryptographic materials and process cryptographic operations. Time synchronization is assured for all OBUs.

Different messages exchanging process are as followings:

Safety Service:

$V_i \rightarrow V_j$: *<id, timestamp, motion attribute, events>*

V_j : *consume and make driving decision*

Content Service:

$V_i \rightarrow V_j$: *<id, timestamp, motion attribute, subscribe info>*

V_j : *consume and make routing decision*

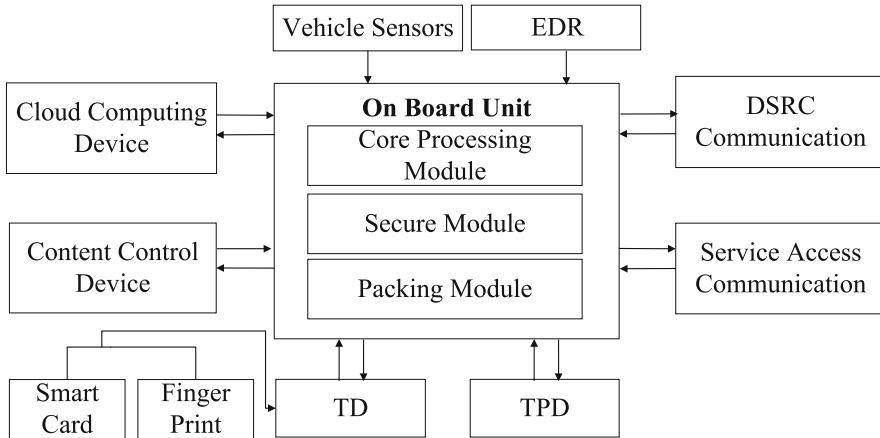


Fig. 1. OBU functional model

Set up a secure channel between V_i and V_j

$V_j \rightarrow V_i$: $\langle id, timestamp, motion attribute, publish info \rangle$

Computation Service:

$V_i \rightarrow V_j$: $\langle id, timestamp, motion attribute, computation request, trust threshold \rangle$

$V_j \rightarrow V_i$: $\langle id, timestamp, motion attribute, computation reply, reputation value \rangle$

In cloud computation service [6], End user (EU) needs to locate the high-reputation computing units (CUs) firstly. V_i sets a threshold trust level, and broadcasts the computation request. After receiving the V_i 's requests, V_j verifies the request, calculates the proof and reply to V_i . If the reputation satisfies the trust level requirement, vehicle (EU) outsource its data through secure channel and receive results eventually.

The above processes show that, to achieve a compatible PPA framework, the direct and simple way is bi-direction authentication.

3.2 Adversary Model

Attack model is divided into common and dedicated.

Common Attack Model

Adversary controls communication channel, monitor all the on-the-fly data through these channel and tamper the message. Eavesdropping, RSU or vehicle compromising, privacy prying, identity impersonation and DoS attack (through jamming, injection or high density traffic) are possible. One hypothesis is that materials are kept safe in TPDs.

Special Attack Model for Safety Service

An adversary would forge safety beacon to induce the legitimate vehicles to accept false or harmful messages without being detected, thus abusing the VANET to maximum its gains (e.g., cheating a clear path, snooping users' location).

Special Attack Model for Content Service

Subscribe information is forged to disturb opportunistic networking, or to help adversary obtain more information without being billed. Adversary might also forge high quality content information and cheat on credit. Sub./Pub. message link attack might happen because of embedded interest.

Special Attack Model for Cloud Computation Service

Reputation spoofing attack is one severe attack when CUs impersonate as other CUs and provide fake reputation to obtain more data. Adversary is able to use reputation in several messages to track a vehicle. Vehicular cloud computation is a hot topic. However, in this paper we focus only on privacy-preserving authentication, thus only reputation spoofing attack and reputation message link attack are considered.

3.3 Design Goals

First are basic security goals for wireless communication: *Resilience to forgery or modification* is that every message should be authenticated to ensure that its source legitimate and payload unaltered. Any forged or modified messages shall be detected by vehicle. As service is different, core messages are different. *Non-repudiation* includes three meanings which are (1) not claiming to be other vehicle; (2) not cheating about their position and service data; (3) not denying the actions and the time of generating and sending messages.

Second are goals concerning V2V communications: *Identity privacy preserving (Authentication and Anonymity)* is fundamental because of the broadcasting nature in VANET. Privacy leaking must be prevented in which binding between real identities and information of VANET. *Unlinkability* is part of level 3 privacy. It means that adversaries are never able to find common properties in multiple messages and link them to one particular vehicle. Considering various services, the meanings are different. Location privacy violation problem might be incurred without unlinkability. Subscribe and publish interest privacy leaking also happens. For computation service, reputation linking is considered in this paper. *Strong privacy preservation* is also necessary, which means with all RSUs compromised, the adversary cannot obtain vehicles' private information. *Conditional traceability* means that TA is able to retrieve a vehicle's real identity when the message is in dispute.

Third are goals to achieve efficiency redundancy and flexibility redundancy like DoS resilience and separate user to one vehicle support.

4 Proposed EPAF Framework

4.1 Overview

The proposed framework is based on three methods: (1) Decentralization is implemented by TD and TPD devices which stores secret information like system key, initial pseudo identity, one-way-function result of user's password and help to verify user's or vehicle's identity and to keep or update passwords. (2) To achieve good extensibility

for various services, a three stage mutual authentication is designed, which includes user authentication, message authentication and service authentication. (3) EPAF divides modules into four types of basic function modules: modules of registration, modules on TD, modules on TPD for message authentication, modules on TPD for service authentication. This decoupling aims to give a lightweight adaptable pseudonymous-based protocol structure without implementation details. MAC and one-way hash operations are used to implement the modules. Revocation and conditional tracing are also designed.

4.2 Framework Workflow and Modules

In registration and initialization phase, after **System Initialization** of CA, user of the vehicular services drives the vehicle to CA, and uploads his password $pw_{i,u}$ (usually in form of biometrics features like fingerprint and iris scan), identity of vehicle and necessary information of vehicle through **Info. Upload** module. **Info. Upload** module does one-way function to $pw_{i,u}$ and obtain $\gamma_{i,u}$, then uploads $\langle ID_i, \gamma_{i,u}, Info_i \rangle$ to CA. Through **Pseudo Identity and Param. Generation** module, CA picks initial pseudo identity PID_i and $TDID_i$ for both TPD and TD devices. Then relevant secure parameters are calculated. CA writes $\langle PID_i, k_m, ts_{key}, [param.] \rangle$ to TPD device and $\langle TDID_i, ID_i, [param.] \rangle$ to TD device.

In order to handle different application situations, we propose a direct and simple three stage authentication scheme, structure of which is shown in Fig. 2.

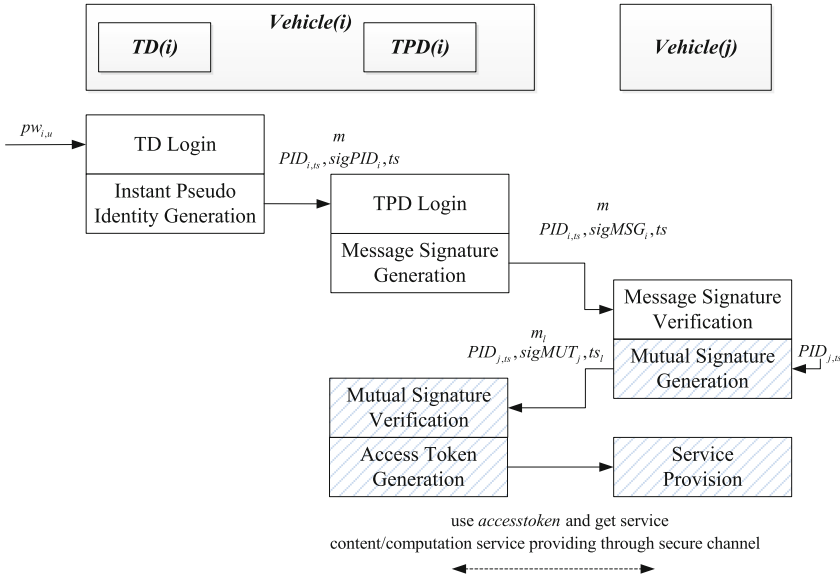


Fig. 2. Three stage authentication phase

User Authentication and TPD Login

User authentication stage is performed by **TD Login** module in telematics device. When new messages come, instant pseudo identity $PID_{i,u}$ and the signature $sigPID_i$ are generated using PID_i by **Instant Pseudo Identity Generation** module. Afterwards, the $\langle PID_{i,ts}, sigPID_i, ts \rangle$ is delivered to tamper proof device. Two core modules **TPD Login** and **Message Signature Generation** are performed in TPD. With assistance of PID_i pre-installed in TPD, $sigPID_i$ is verified.

Message Authentication Stage

After the verification of **TPD Login module**, **Message Signature Generation** module would generate the signature of message using $PID_{i,ts}$, pre-stored k_m , and current timestamp ts . Then $\langle m, PID_{i,ts}, sigMSG_i, ts \rangle$ is broadcasted to nearby vehicles. When message is received by another vehicle, **Message Signature Verification** module is performed by $TPD(j)$. If the verification process returns true, message is valid and is available to be consumed.

Service Authentication Stage

This stage aims to support bi-direction communication service and verification of vehicular service provider. **Mutual Signature Generation** module on $TPD(j)$ is performed to take service link information, $PID_{i,ts}$, ts_l as input and generate mutual signature for service link information. Then $\langle PID_{j,ts}, sigMUT_i, ts_l, m_l \rangle$ is sent back to $Vehicle(i)$. **Mutual Signature Verification** is performed on $TPD(i)$ to verify the identity and the service information from $Vehicle(j)$. If the verification is passed, **Access Token Generation** is performed to output a service *accesstoken* for $Vehicle(j)$, which would sent it to $Vehicle(j)$. After the *accesstoken* is verified valid, $Vehicle(j)$ enters into **Service Provision** module and the vehicles communicate through specific secure channels.

4.3 Core Module Implementation

To explain the module implementation of EPAF, we use two vehicles. The correlated modules are shown in Figs. 3 and 4. Each module implementation is shown as followings:

System Initialization and Info. Upload

Suppose \mathbf{G} be a cyclic additive group of order q , $P \in \mathbf{G}$ a generator of \mathbf{G} and $e: \mathbf{G} \times \mathbf{G} \rightarrow V$ be a bilinear map which satisfies following conditions [12]: Bilinear, $e(x_1 + x_2, y) = e(x_1, y)e(x_2, y)$ and $e(x, y_1 + y_2) = e(x, y_1)e(x, y_2)$; Non-degenerate, There exists $x \in \mathbf{G}$ and $y \in \mathbf{G}$ such that $e(x, y) \neq 1$. CA randomly picks integer $\alpha \in \mathbb{Z}_q^*$ as private key for the vehicular network system, and computes $\beta = \alpha P$ as public key. CA computes $S_{IDCA} = \alpha H(ID_{CA})$ as its identity secret key and generates system key $k_m = \{k_m^1, k_m^2\}$, where $k_m^1 \in \{0, 1\}^a$, a is the key length of $Enc_k(\cdot)$; $k_m^2 \in \{0, 1\}^b$, b is the key length of $h_k^1(\cdot)$. CA publishes $\{\beta, ID_{CA}\}$, and keeps α, k_m, S_{IDCA} secret.

$Vehicle_i$ along with its user firstly submit real identity ID_i , $\gamma_{i,u} = h(pw_{i,u})$ and $Info_i$ (e.g., engine serial number, date of manufacture, vehicle owner, service registration information) to **Info. Upload** through secure channels. **Info. Upload** then outputs $\langle ID_i, \gamma_{i,u}, Info_i \rangle$.

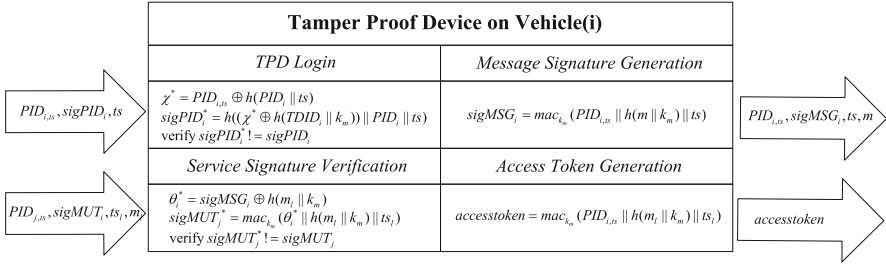


Fig. 3. Modules on TPD(i) of Vehicle(i)

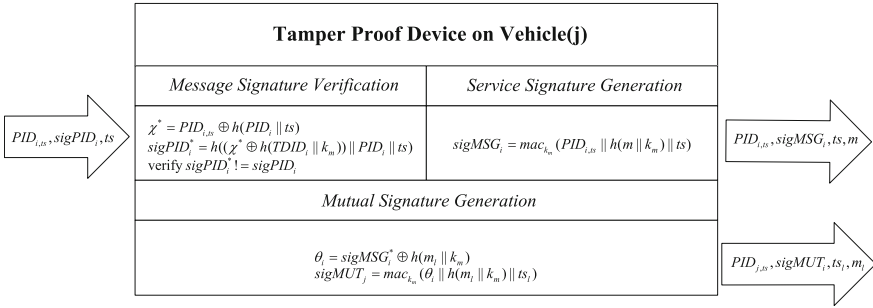


Fig. 4. Modules on TPD(j) of Vehicle(j)

Pseudo Identity and Param. Generation

CA checks the correctness of input $\langle ID_i, \gamma_{i,u}, Info_i \rangle$ (usually with help of national vehicle management department and the VANET service provider). Then CA randomly picks $PID_i \in \mathbf{Z}_q^*$ and $TDID_i$ for $Vehicle_i$ and $TD_i \langle ID_i, TDID_i, PID_i, Info_i \rangle$ is then inserted into user and vehicle information table. CA computes parameters like (Table 1):

$$pS_i = h(ID_i \parallel TDID_i \parallel PID_i) \oplus h(TDID_i \parallel k_m), pU_{i,u} = h(ID_i \parallel \gamma_{i,u} \parallel PID_i) \oplus h(TDID_i \parallel k_m),$$

$$pV_{i,u} = h(\gamma_{i,u} \oplus PID_i), pK_{i,u} = PID_i \oplus h(TDID_i \oplus \gamma_{i,u}).$$

Here $pV_{i,u}$ is employed as a user verifier to authenticate driver's identity, $pK_{i,u}$ is employed as a password keeper and $pU_{i,u}$ is used to update the user password through TPD device. Moreover, if user changes the password $pK_{i,u}$ would be updated and all values of $pK_{i,u}$ are kept in a table in TPD device for message tracing use, we call it **pK-table**.

Finally, CA saves $\langle ID_i, TDID_i, PID_i, Info_i \rangle$ to a user information table, and writes $\{TDID_i, ID_i, pS_i, \langle pV_{i,u}, pK_{i,u} \rangle\}$ to TD_i , and preloads $\{PID_i, k_m, ts_{key}, \langle pU_{i,u}, pK_{i,u} \rangle\}$ on TPD_i .

Table 1. Notations used in proposed scheme

Symbol	Description	Symbol	Description
CA	Certificate authority	ID_{CA}	Identity of CA
k_m	System key	$Vehicle_i$	The i^{th} vehicle
ts_{key}	Timestamp of current system key being updated	$h(.)$	Hash function $h : \{0, 1\}^* \times V \rightarrow \mathbf{Z}_q^*, \mathbf{Z}_q^* = \{x \in \{1, \dots, q-1\} \mid \gcd(x, q) = 1\}$
$Info_i$	Vehicle information of $Vehicle_i$	$h_k^1(.)$	Hash function $h_k^1 : \{0, 1\}^* \rightarrow \{0, 1\}^n$
ID_i	Real identity of $Vehicle_i$	$TDID_i$	Identity of TD_i
$pw_{i,u}$	Biological password of driver u of $Vehicle_i$	$PID_{i,ts}$	Dynamic pseudo identity of $Vehicle_i$ at ts
$Enck(.)$	Encryption function using k as key, like AES	$Dec_k(.)$	Decryption function using k as key, like AES
$H(.)$	Hash function $H: \{0, 1\}^* \rightarrow \mathbf{G}^*$, $\mathbf{G}^* = \mathbf{G} \setminus \{0\}$	$mac_k(.)$	MAC using k as a key, such as HMAC [11]
PID_i	Initial pseudo identity of $Vehicle_i$	\parallel	Message concatenation operation

TD Login

User firstly plugs the TD_i into the $Vehicle_i$ and input $pw_{i,u}$. Then $\langle pV_{i,u}, pK_{i,u} \rangle$ is used to verify user as shown in Fig. 4. If the driver is legitimate, the restored PID_i is stored in memory until TD_i is unplugged.

Instant Pseudo Identity Generation

Figure 4 gives the generation process of $PID_{i,ts}$ and $sigPID_i : PID_{i,ts} = h(ID_i \parallel TDID_i \parallel PID_i) \oplus h(PID_i \parallel ts)$, $sigPID_i = h(pS_i \parallel PID_i \parallel ts)$. Then TD_i sends $\{PID_{i,ts}, sigPID, ts\}$ to TPD_i .

TPD Login

TPD_i would verify the legitimacy of TD_i , if TD_i is legitimate, then OBU is free to use TPD_i to perform further action.

Message Signature Generation

Every time a new message payload m is generated, TD_i redoes the **TPD login** to update dynamic pseudo identity $PID_{i,ts}$. If the **TPD login** is passed, TPD_i would generate: $sigMSG_i = mac_{km}(PID_{i,ts} \parallel h(m \parallel k_m) \parallel ts)$ and packs the message like $\{PID_{i,ts}, sigMSG_i, ts, m\}$ as module output.

Message Signature Verification

After $Vehicle_j$ receives a packet $\{PID_{i,ts}, sigMSG_i, ts, m\}$ from $Vehicle_i$, TPD_j on $Vehicle_j$ would calculate $sigMSG_i^* = mac_{km}(PID_{i,ts} \parallel h(m \parallel k_m) \parallel ts)$ to verify the legitimacy of the message. If $sigMSG_i^* \neq sigMSG_i$ returns false, $Vehicle_j$ then accepts the message for application or launches **Mutual Signature Generation** in content or computation service.

Mutual Signature Generation

Mutual signature is generated by vehicle which replies the service request like: $sigMUT_j = mac_{km}(\theta_i || h(m_l || k_m) || ts_l)$, $\theta_i = sigMSG_i^* \oplus h(m_l || k_m)$. The generation process is just simple like **Message Signature Generation** module.

Mutual Signature Verification

TPD_i uses $\{PID_{j,ts}, sigMUT_j, ts_l, m_l\}$ as input and computes $\theta_i^* = sigMSG_i \oplus h(m_l || k_m)$, $sigMUT_j^* = mac_{km}(\theta_i^* || h(m_l || k_m) || ts_l)$ to verify service signature.

Through $accesstoken = mac_{km}(PID_{i,ts} || h(m_l || k_m) || ts_l)$, access token is generated and is used in future service acquisition. $Vehicle_j$ would verify the access token, if it returns true, then the vehicles enter service provision through secure dedicated channel. The Service provision processing is potential to be realized in many different ways and the corresponding discussion is not included in this paper.

4.4 Revocation and Conditional Tracing Phases

Pseudonym Revocation

In a decentralized framework, it is hard to revoke an invalid vehicle which is judged invalid. EPAF only needs CA to broadcast one revocation message $\{PID_i, sg_{rev}\}$ to all vehicles, in which sg_{rev} is the signature of PID_i calculated by $sg_{rev} = Sign_{SIDCA}(PID_i)$. If $Vehicle_i$ receives the revocation message and verify the source legitimacy of it, TPD_i deletes all the secret materials preloaded in registration phase including $\{PID_i, k_m, ts_{key}, <pU_{i,u}, pK_{i,u}>\}$. Once the telematics device is plugged in the vehicle, the corresponding preload secret materials $\{TDID_i, ID_i, pS_i, <pV_{i,u}, pK_{i,u}>\}$ would also be erased.

Conditional Tracing for Vehicle and User

Message tracing process provides the capability of tracing messages in services. Take $\{PID_{i,ts}, sigMSG_i, ts, m\}$ as an example. CA selects $<ID_i, TDID_i, PID_i, Info_i>$ where $PID_{i,ts} == h(ID_i || TDID_i || PID_i) \oplus h(PID_i || ts)$, from user and vehicle information table. Therefore, $Vehicle_i$ is found and located through $Info_i$. Through pK -table stored on TPD_i , the authority is able to trace the user on vehicle using evidence of $pU_{i,u}$ when the message is being sent.

5 Security Analysis

Preliminaries about symbolic approach is given in this chapter. Then we implement core phases of EPAF using ProVerif and compare the security properties of schemes.

5.1 Preliminaries

The computational approach and the symbolic approach are two major methods to analyze the cryptographic protocols employed in last two decades. Symbolic approach is amenable enough to realize in automatous way due to its algebraic structure. Many automated tools are introduced for the symbolic approach. For example, ProVerif is a

tool for applied spi calculus. Yet problems exist: (1) the computational soundness is unclear; (2) the number of participants has to be fixed. (3) the time complexity increases exponentially along with the number of participants. Recently, Canetti et al. [16] has proposed the universally composable symbolic analysis (UCSA) approach, in which it is proved that the security is unrelated with the number of sessions. However, it is only able to deal with two-party cryptographic protocols. Later in [17], the UCSA approach is extended to deal with arbitrary number of participants. Moreover, according to Theorem 2 in [17], symbolic approach implies computational approach. Some important keywords of the pi calculus are as followings:

Query attacker: M means that the attacker may have M in phase (M is not secret).
query $ev:f(x_1, \dots, x_n) \implies ev:f'(x_1, \dots, x_n)$ is non-injective agreement: it is true when, if the event f has been executed, then the event f' must have been executed before f .
choice $\langle \text{term} \rangle, \langle \text{term} \rangle$: it tries to reconstructs a trace until a program point at which the process using the first argument of choice behaves differently from the one using the second. If a trace is reconstructed, it means the attacker is able to distinguish the first argument from the second one. **!<process>**: it means the replication executes an unbounded number of copies of $\langle \text{process} \rangle$ in parallel: $\langle \text{process} \rangle \mid \langle \text{process} \rangle \mid \langle \text{process} \rangle \mid \dots$

5.2 Experiment and Analysis

In this chapter, we compare the security features of EPAF framework with classical BP, GSIS, VAST and PTVc.

Resilience to Forgery or Modification of Message

The messages in the framework is protected by MAC. The proposed scheme is able to detect the forged or modified messages with the assistance of tamper proof device. Results in [19] show that if “event endAuthV2V(PID_{i,ts}, sigMSG_i, ts)” has been executed, then “event beginAuthV2V(PID_{i,ts}, sigMSG_i, ts)” must have been executed. Thus the adversary is unable to forge or modify $\{PID_{i,ts}, sigMSG_i, ts, m\}$ or $\{PID_{j,ts}, sigMUT_j, ts_1, m_1\}$.

Non-repudiation

Each message is integrated with instant pseudo identity, which is generated from ID_i , PID_i , $TDID_i$ and timestamp by *Instant Pseudo Identity Generation* module. Non-repudiation is guaranteed because an adversary is never able to deny the action nor time of message.

Identity Privacy Preserving

$PID_{i,ts}$ or $PID_{j,ts}$ is utilized to preserve the ID_i . User need to pass *TD Login* module on TD and pass *TPD Login* module on TPD to access the vehicular network. Thus identity privacy is preserved even if the telematics device is stolen. As shown in [19], the adversary is unable to obtain any information about ID_i and ID_j .

Unlinkability

In EPAF, $PID_{i,ts}$ differs as time changes, adversary is unable to launch replay attack nor link numerous messages to one vehicle. Moreover, in core modules, key operations of

MAC generation and message authentication are accomplished without knowing the real identity. We use keyword “*choice[PID_i_ts,r0]*”, “*choice[PID_j_ts,r0]*” in to test the anonymity. The result is “RESULT Observational equivalence is true (bad not derivable)”, as shown in [19], which means $PID_{i,ts}$ is unable to be distinguished from a random number $r0$. To test the unlinkability, we use “!” before the processes and the result is still true, which means no matter how many commutation processes are running, none of information about vehicle’s identity will be revealed. Thus the proposed scheme achieves level 3 privacy: authentication, anonymity, unlinkability.

Mutual Authentication

In proposed framework, a three stage mutual authentication mechanism is provided. In message authentication stage, beacon safety message itself or service request message (Subscribe information message or computation request message) is being verified. In service authentication stage, the service reply message (Publish information or computation reply) is being verified to achieve the PPA along with extendibility for various services.

Compatible with Different Services

EPAF achieves the compatibility which focus on different core data in message integrity, unlinkability and a mutual authentication service hand shake. These are the simplest model to provide unified privacy preserving authentication service. In contrast, BP, GSIS and VAST are designed for safety and PTVC is designed for computation. Other schemes including batched based schemes, hybrid schemes, k-anonymity based schemes and cloud assisted schemes are only able to adapt for one kind of vehicular services and hard to extend.

Apart from above security feature analysis, EPAF also achieves strong privacy preservation and conditional traceability which are fundamental in PPA. As is shown in

Table 2. Security comparison

Schemes		Properties				
		BP	GSIS	VAST	PTVC	EPAF
Data integrity		✓	✓	✓	✓	✓
Non-repudiation		✓	✓	✓	✓	✓
Level3 privacy	Authentication	✓	✓	✓	✓	✓
	Anonymity	✓	✓	✗	✓	✓
	Unlinkability	✗	✓	✗	✓	✓
Strong privacy preserving		✗	✗	✗	✗	✓
Conditional traceability	Vehicle	✓	✓	✗	✓	✓
	User	✗	✗	✗	✗	✓
Mutual authentication		✗	✗	✗	✓	✓
Service compatible		✗	✗	✗	✗	✓
Efficient revocability		✗	✗	✗	✗	✓
Resist to DoS	Computation	✗	✗	✓	✓	✓
	Memory	✓	✓	✓	✗	✓

Table 2, EPAF achieves all the issued security properties and is more practical and extendable than their schemes.

6 Performance Analysis

6.1 Authentication Overhead

Communication overhead of one message consists of attached certificate and signature. For PTVC, request and reply message are 87 bytes and 91 bytes (1 timestamp for request and 2 for reply, proof of reputation is not included), shown in Table 3. In EPAF it includes MAC, pseudo identity and a timestamp. It is evident that EPAF significantly decreases communication overhead by 45.98%–75.53% compared with other schemes.

Table 3. Communication overhead for one message

Schemes	BP	GSIS	VAST	PTVC	EPAF
Request overhead (byte)	105	192	145	87	47
Reply overhead (byte)	–	–	–	91	47

In Table 4 and Fig. 5, it illustrates that EPAF is the second most efficient for request and the most efficient for reply. EPAF significantly reduces request signing cost by near 2000 times compared with other schemes, reply signing by 1800 times compared with PTVC.

Table 4. Message signing cost

Schemes	BP	GSIS	VAST	PTVC	EPAF
Request overhead(s)	T_{mul}	$3T_{par} + T_h$	$T_{mul} + T_{mac}$	$T_h + T_{EXP}^*$	$7T_h + T_{mac}$
Reply overhead(s)	–	–	–	$3T_{mul} + 3T_{mod} + 3T_h$	$2T_h + T_{mac}$

*Note: T_{EXP} represents uncertain time cost for exponent computation.

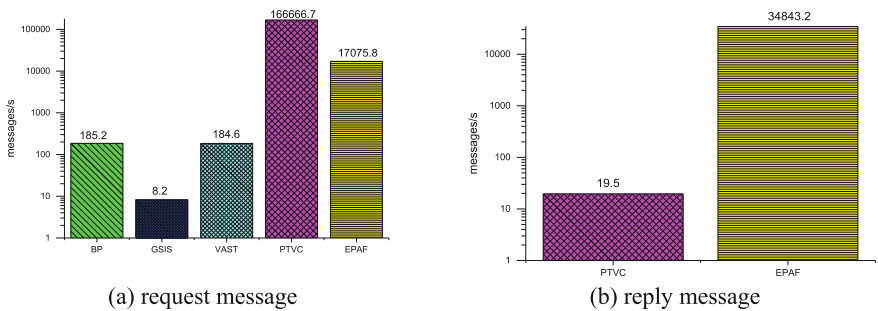


Fig. 5. Message signing speed

Message verification includes CRL checking, certificate verification and signature verification for BP, GSIS, VAST and PTVC. BP, VAST and PTVC perform CRL checking through string comparison, computation cost is able to be ignored. In GSIS, each CRL item needs two paring operations, which makes total cost $2N_{cri}T_{par}$. In PTVC, both request and reply message verification need one T_{par} . In comparison, EPAF only needs light MAC and hash operations to accomplish verification and achieves an efficient verification speed as shown in Fig. 6 (Table 5).

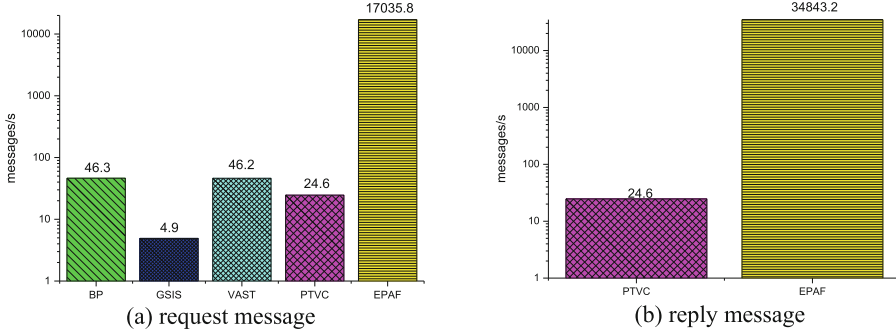


Fig. 6. Message verification speed

Table 5. Message verification cost

Schemes	BP	GSIS	VAST	VAST*	EPAF
CRL checking	0	$2N_{cri}T_{par}$	0	0	—
Certificate verification	$2T_{mul}$	0	$2T_{mul}^*$	0	—
Request signature verification	$2T_{mul}$	$5T_{par} + T_h$	$2T_{mul}^* + 2T_{mac}$	$T_{par} + 2T_h$	$T_h + T_{mac}$
Reply signature verification	—	—	—	$T_{par} + 2T_h$	$2T_h + T_{mac}$
Total	$4T_{mul}$	$2N_{cri}T_{par} + 5T_{par} + T_h$	$4T_{mul}^* + 2T_{mac}$	$2T_{par} + 4T_h$	$3T_h + 2T_{mac}$

*Note: In VAST, certificate and digital signature verification is only performed when non-repudiation is necessary. In this paper, we focus on VAST needing non-repudiation because of service concern.

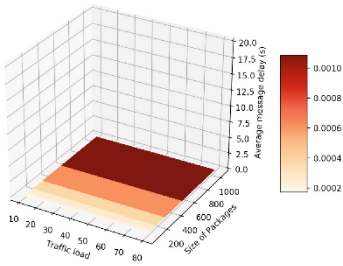
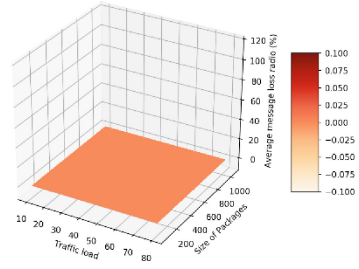
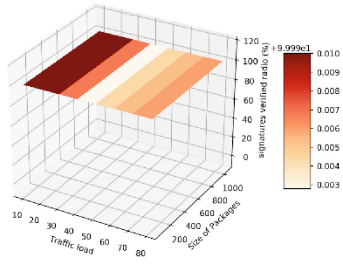
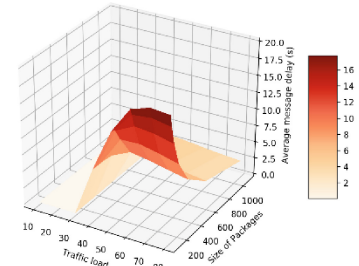
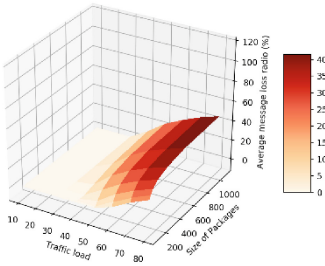
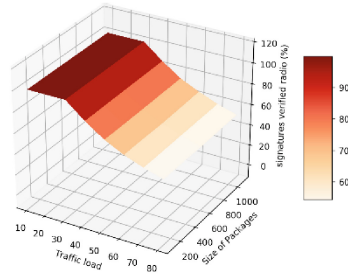
6.2 Simulations

In this subsection, we use Opportunistic Networking Environment (ONE [13]) to run simulations. We import a part from real map (northeast corner of area surrounded by the No. 2nd Ring Road of Beijing). Parameters are in Table 6.

In simulation, EPAF and PTVC both need request and reply messages and the metrics for each type are the average message delay, average message loss ratio and

Table 6. Simulation configuration

Parameter	Values
Communication range	4000 m
Simulation time	100 s
Channel bandwidth	6 Mbps
Wait time	0–5 s
Buffer size	1M bytes
Broadcast interval	0.3 s
Speed	[20 km/h, 100 km/h] \pm 10 km/h

(a-1) $avgD_{msg}$ of EPAF request(a-2) $avgLR$ of EPAF request(a-3) $avgPerSV$ of EPAF request(b-1) $avgD_{msg}$ of PTVC request(b-2) $avgLR$ of PTVC request(b-3) $avgPerSV$ of PTVC request**Fig. 7.** Traffic load and message size's impact on performance

percentage of signature verified, which are represented as $avgDmsg$, $avgLR$ and $avgPerSV$ and stated as same as in our previous work [14].

In Fig. 7 we compare the performance of EPAF with PTVC under different traffic load (vehicles in communication range) and message sizes. Because reply performance is nearly the same as request in both PTVC and EPAF, only figures for request performance are listed.

EPAF achieves low and stable $avgDmsg$ below 0.002 s for both request and reply message as shown in Fig. 7(a-1). However, PTVC's $avgDmsg$ increases dramatically along with traffic load increasing shown in (b-1). When size of packages increases, PTVC's $avgDmsg$ decreases because vehicles' buffer space is filled rapidly, older messages are dropped and are not count. The trend in (b-2) is able to prove it.

Figure 7(a-2) shows, as traffic load growing larger and messages growing bigger, $avgLR$ of EPAF is stable at nearly 0%, even with traffic load 80 and size of packages 1000 bytes. For PTVC, with traffic load above 40, $avgLR$ increases dramatically as shown in (b-2). Apparently, EPAF achieves good performance in high traffic load and large package size.

Comparisons for $avgPerSV$ are shown in (a-3) and (b-3). $avgPerSV$ for both EPAF request and reply messages keeps near 100% at all configuration. For PTVC, $avgPerSV$ decreases as traffic load growing larger. It is lower than 60% when traffic load is above 80. This is unacceptable in real use.

It is evident that EPAF is DoS resilient and significantly increases availability of PPA, which turns out to have potential extendibility to support various VANET services.

7 Conclusion

In this paper, we proposed an efficient pseudonymous-based inter-vehicle authentication framework for various VANET services. Security analysis based on ProVerif proves that EPAF achieves all designing security features, including 3 level privacy, strong privacy preserving, mutual authentication and other security features. Performance evaluation shows that EPAF has advantage in communication, message signing/verification speed and achieves a significant increase of nearly 370–3500 times in computation compared with safety schemes. This makes EPAF DoS resilient in complex scenarios.

To the best of our knowledge, EPAF is the first PPA framework which achieves both necessary security features and DoS resilience for various VANET services. It would work as a design reference in more vehicular services like navigation, data fusion and unmanned driving, and be implemented using other cryptographic methods. Proposing a unified privacy preserving authentication framework for various services is a new topic. We will focus on the common security problems in different scenarios and make the EPAF framework to a more adaptable version, while maintaining the applicable efficiency.

References

1. Su, Z., Hui, Y., Yang, Q.: The next generation vehicular networks: a content-centric framework. *IEEE Wirel. Commun.* **24**(1), 60–66 (2007)
2. Harding, J., Powell, G., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., Lukuc, M., Simons, J., Wang, J.: Vehicle-to-vehicle communications: readiness of V2V technology for application. NHTSA, Washington, DC, Technical report, DOT-HS-812-014 (2014)
3. Chim, T.W., Yiu, S.M., Hui, L.C., Li, V.O.: VSPN: VANET-based secure and privacy-preserving navigation. *IEEE Trans. Comput.* **63**(2), 510–512 (2014)
4. Wang, M., Liu, D., Zhu, L., Xu, Y., Wang, F.: LESPP: lightweight and efficient strong privacy preserving authentication scheme for secure VANET communication. *Computing* **98**, 1–24 (2014)
5. Raya, M., Hubaux, J.-P.: Securing vehicular Ad hoc networks. *J. Comput. Secur.* **15**(1), 39–68 (2007)
6. Huang, C., Lu, R., Zhu, H., Hu, H., Lin, X.: PTVC: achieving privacy-preserving trust-based verifiable vehicular cloud computing. In: 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, pp. 1–6 (2016)
7. Lu, R., Lin, X., Zhu, H., Ho, P.-H., Shen, X.: ECPP: efficient conditional privacy preservation protocol for secure vehicular communications. In: Proceedings of the INFOCOM 2008, pp. 1903–1911 (2008)
8. Lin, X., Sun, X., Ho, P.-H., Shen, X.: GSIS: a secure and privacy-preserving protocol for vehicular communications. *IEEE Trans. Veh. Technol.* **56**(6), 3442–3456 (2007)
9. Zhang, L., Wu, Q., Solanas, A., Josep, D.-F.: A scalable robust authentication protocol for secure vehicular communications. *IEEE Trans. Veh. Technol.* **59**(4), 1606–1617 (2010)
10. Studer, A., Bai, F., Bellur, B., Perrig, A.: Flexible, extensible, and efficient VANET authentication. *J. Commun. Netw.* **11**, 589–598 (2009)
11. Bellare, M., Canetti, R., Krawczyk, H.: Message authentication using hash functions: the HMAC construction. *RSA Laboratories' CryptoBytes* **2**(1), 12–15 (1996)
12. Scott, M.: Efficient Implementation of Cryptographic Pairings. http://www.pairing-conference.org/2007/invited/Scott_slide.pdf
13. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (2009)
14. Wang, F., Xu, Y., Zhang, H., Zhang, Y., Zhu, L.: 2FLIP: a two-factor lightweight privacy-preserving authentication scheme for VANET. *IEEE Trans. Veh. Technol.* **65**(2), 896–911 (2016)
15. Horng, S.J., Tzeng, S.F., Li, T., Wang, X., Huang, P.H., Khan, M.K.: Enhancing security and privacy for identity-based batch verification scheme in VANET. *IEEE Trans. Veh. Technol.* **66**(4), 3235–3248 (2017)
16. Canetti, R., Herzog, J.: Universally composable symbolic analysis of mutual authentication and key-exchange protocols. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 380–403. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_20
17. Zhang, Z., Zhu, L., Liao, L., Wang, M.: Computationally sound symbolic security reduction analysis of the group key exchange protocols using bilinear pairings. *Inf. Sci.* **276**(20), 93–112 (2012)
18. Du, X., Chen, H.H.: Security in wireless sensor networks. *IEEE Wirel. Commun. Mag.* **15**(4), 60–66 (2008)

19. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: Secure and efficient time synchronization in heterogeneous sensor networks. *IEEE Trans. Vehi. Technol.* **57**(4), 2387–2394 (2008)
20. Du, X., Xiao, Y., Guizani, M., Chen, H.H.: An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw.* **5**(1), 24–34 (2007)
21. Xiao, Y., Rayi, V., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. *J. Comput. Commun.* **30**(11–12), 2314–2341 (2007)