Security and Communication Networks

Research Article

Trajectory as an Identity: Privacy-Preserving and Sybil-Resistant Authentication for Internet of Vehicles

Jiangtao Li[1,2], Zhaoheng Song[1], Yufeng Li[1,2], Chenhong Cao[1,2] and Yuanhang He [3]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai, China

[2]Purple Mountain Laboratories, Nanjing, China

[3]No. 30 Research Institute of China Electronics Technology Group Corporation, Chengdu, China
Emails: lijiangtao@shu.edu.cn (Jiangtao Li), songzh@shu.edu.cn (Zhaoheng Song), liyufeng_shu@shu.edu.cn (Yufeng Li), caoch@shu.edu.cn (Chenhong Cao), kevin0319@163.com (Yuanhang He)

Abstract
With the advancement of the 5G network, the Internet of Vehicles (IoV) is becoming more and more attractive for academic researchers and industrial. A main challenge of IoV is to guarantee the authenticity of messages and protect the privacy of drivers simultaneously. The majority of privacy-preserving authentication schemes for IoV adopt pseudonyms or group signatures to achieve a balance between security and privacy. However, defending the Sybil attacks in these schemes is challenging. In this work, we propose a novel privacy-preserving authentication scheme for announcement messages which utilizes the trajectories of vehicles as their identifies. When an authenticated message is verified, the verifier is convinced that the message is generated by a vehicle that has a unique masked trajectory. Meanwhile, the real trajectories of vehicles are kept private. Especially, our scheme achieves Sybil attack resistance without the limitation of trajectory length even when the attacker is allowed to use cloud services.

Keywords: Privacy-preserving; Authentication; Internet of Vehicles

1. Introduction

Internet of Vehicles (IoV) is made possible with the advancement of 5G network, which achieves lower latency and supports higher mobility [1, 2]. As a special type of Internet of Things (IoT) [3], IoV allows vehicles to communicate with surrounding vehicles, Road Side units (RSUs), and infrastructures. By providing road condition warnings, cooperative perception, and driving assistant services, IoV is able to improve traffic efficiency and road safety [4]. However, vehicle-to-vehicle communications in IoV do not adopt any security mechanism in the network layer [5, 6]. For privacy concerns, the IP address of each vehicle is also dynamically changed. Without a security mechanism, a malicious vehicle in IoV may send false data or misleading messages to other vehicles and infrastructures. For instance, a vehicle may claim a fake traffic jam by broadcasting an announcement message in order to achieve a better driving condition for himself. Hence, it is necessary to authenticate the messages broadcast in IoV.

Adopting signature schemes is a commonly used way to guarantee the authenticity of messages in IoV [7]. If a signature is verified, a vehicle is convinced that the message is sent by the claimed sender and not tempered in transmission. However, due to the openness of wireless communication, an attacker can violate vehicle privacy by simply collecting messages broadcast in IoV [8]. Pseudonym mechanism is a classical way to achieve a trade-off between security and privacy in IoV. Under the pseudonym mechanism, each vehicle will generate multiple certified identities (i.e., public/private key pairs). A vehicle needs to use a fresh identity for each message to keep its identity private. This will bring the vehicle extra storage cost. Besides, the pseudonym mechanism is vulnerable to Sybil attacks. A vehicle can use multiple identities simultaneously to pretend to be multiple vehicles.

One approach to overcome the drawback of the pseudonym mechanism is employing the group signature instead of traditional signatures. In group signatures, any group member is allowed to generate signatures on behalf of the group. This way, vehicles' identities will not be leaked during message authentication. To defend Sybil attacks, the researchers also propose a new primitive named message linkable group signature [9]. If an attacker broadcasts the same message twice, this behavior can be efficiently detected. It is obvious that the message linking technique is useful only when the attacker simply sends the same message repeatedly in launching Sybil attack. If an attacker tries to send different messages with the same semantic, this Sybil attack defence technique will fail.

Several works have been proposed to deal with the Sybil attacks which are launched with different messages. One solution is to use the cryptographic puzzle to prevent a vehicle from impersonating multiple vehicles since the computational resource of each vehicle is limited [10]. However, if a vehicle is allowed to outsource the puzzle to the cloud, this solution will not work. Later, Baza *et al.* [11] propose to consider both the locations (trajectories) [12] and cryptographic puzzles in detecting Sybil attacks. In [11], only vehicles who are proved to have traversed some RSUs are regarded valid vehicles. Whereas, their work does not consider the situation where vehicles may collude with each other. Furthermore, when the vehicle is allowed to use cloud service, the Sybil attack can only be efficiently detected if the trajectory length is long enough.

1.2 Contribution

To cope with the above-mentioned problems. We propose to use trajectory as an identity of a vehicle to achieve privacy-preserving and Sybil resistant authentication for IoV.

In the proposed scheme, when a vehicle needs to broadcast an announcement message, it will sign the message with a freshly generated identity. Then the signed message is broadcast together with the trajectory of the vehicle. In fact, the trajectory can be viewed as a certificate of the freshly generated identity. When an announcement message is verified, it is established that the message is signed by the vehicle that is confirmed to have a valid trajectory in IoV. Besides, since the identities of RSUs are masked, the real trajectory of vehicles will not be revealed and hence the privacy of vehicles is preserved.

Especially, when the vehicle is allowed to use cloud services, our proposal is also secure against Sybil attacks without the trajectory length limitation. A vehicle in IoV needs to solve a series of computational puzzles before it is able to send a valid trajectory request to the RSU. Since the computational puzzles are bound with the secret keys of vehicles, so rational vehicles will not choose to outsource the secret key and hence the computational puzzles. Besides, benefits from the sequential aggregate signatures used in the scheme, the scheme also achieves efficient RSU verification.

1.1 Related Work

Existing privacy protection authentication schemes for IoV can generally be categorized into two types: pseudonym-based ones and group signature-based ones. In the pseudonym-based authentication scheme, each vehicle can hold multiple pseudonymous identities [13]. The vehicle can use a new pseudonymous identity to send messages at regular intervals or after sending several messages according to its own privacy needs. No entity except a trusted authority can distinguish whether any two pseudonymous identities belong to the same vehicle. However, the anonymity of these schemes is achieved at the cost of frequently changing pseudonymous identities [14]. Under the traditional Public Key Infrastructure (PKI) architecture [15], the trusted authority needs to issue a certificate for each pseudonymous public key, so the vehicle needs to store a large number of pseudonymous certificates and their corresponding keys in advance. This brings extra storage costs to vehicles.

In order to reduce the cost storage and communication of vehicles caused by pseudonym management, privacy-preserving authentication schemes based on group signature are proposed [16, 17]. In these schemes, when the vehicle needs to change its pseudonym, the vehicle can use its group private key to issue a certificate to the new pseudonym, so there is no need to store a large number of pseudonyms. Later, Hao *et al.* [18] also proposes a distributed group model [19] for group signature-based authentication schemes in IoV. Under this model, an RSU acts as a group administrator to manage the group and distribute certificates and keys to vehicles entering its area. When the vehicle leaves the communication range of the current RSU and enters the next RSU, the vehicle will be automatically revoked. Recently, Lin *et al.* [20] and Mollah *et al.* [21] also propose to use blockchain and smart contracts in realizing identity management and improve existing privacy protection authentication schemes. However, traditional group signature-based solutions are not secure against Sybil attacks.

In order to defense Sybil attacks, Wu *et al.* [9] proposed the concept of message linkable group signature. In the message linkable group signature-based schemes, if a vehicle generates more than two signatures for the same message, it can be efficiently detected. Chen *et al.* [22] also propose to use direct anonymous authentication schemes and one-time anonymous signatures to realize message linkability which is similar to the effect of [9]. However, the message linking techniques are only useful for event-driven announcement messages in IoV. For the announcement messages which allow the vehicle to send different messages with the same semantics, message linking techniques will not play the role of defending Sybil attacks.

Another method to realize Sybil detection is anonymous trajectory-based authentication. In [12], each vehicle needs to request a trajectory from the RSU when it passing through. Compared with the message linkable solutions, this method may prevent an attacker from sending Sybil messages with the same semantics. At the same time, in order to achieve anonymity, RSU needs to use a zero-knowledge signature that can be linked in a short time to protect the privacy of the vehicle's location. However, the vehicle may generate multiple false trajectories and send them to the RSU to achieve a denial of service (DOS) attack. Hence, Liu *et al.* [10] propose to use cryptographic puzzles to resist DOS attacks and Sybil attacks in the IoV. Recently, Baza *et al.* [11] combined the idea of the cryptographic puzzle with the trajectory proof method in [12] and proposed a solution to detect Sybil attacks. However, this work does not consider the situation where vehicles may collude with each other. When the vehicle is allowed to use cloud service, the Sybil attack can only be efficiently detected if the trajectory length is long enough.

1.3 Organization

The rest of the paper is organized as follows. We describe the system architecture and design goals in Section 2. Section 3 gives some of the cryptographic primitives used in our scheme. We present our scheme in Section 4 and analyze its security and efficiency in Section 5. Finally, we conclude the paper in Section 6.

2. System Architecture and Design Goals

In this section, we describe the system models and design goals of this work.

2.1 System Architecture

As shown in Figure 1, the IoV scenario considered in this work consists of the following entities:

- **Road Side Units (RSUs):** RSUs in the system are equipped with computation and communication modules. It is responsible for issuing partial authenticated trajectories for vehicles.
- **Vehicles:** Vehicles in the system have the capability of communicating with surrounding vehicles and RSUs. When it encounters an event, e.g., traffic accident, it is able to send an announcement message to the announcement manager. We note that the vehicles might be malicious. They may send fake messages or launch Sybil attacks.
- **Trusted Authority (TA):** The trusted authority can be the Department of Motor Vehicles (DMV). It is responsible for issuing certificates for vehicles and initializing the system.
- **Announcement Manager:** The announcement manager is an authority that collects and verifies the announcement messages sent by the vehicles. It is supposed to be a semi-honest authority, i.e., it will perform all steps of the scheme honestly, however, it is curious about the privacy of vehicles.

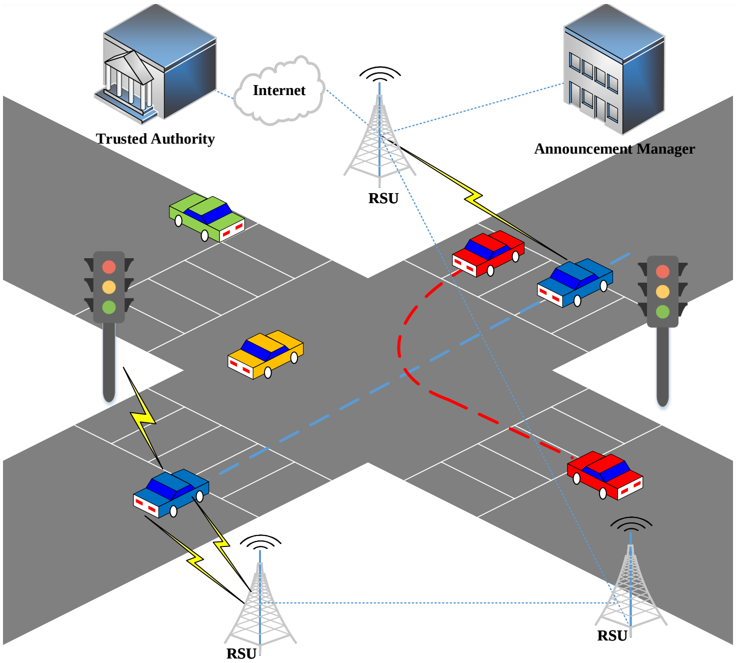


Figure 1: System Architecture

2.2 Design Goals

A privacy-preserving Sybil resistant announcement scheme for IoV should achieve the following security and efficiency requirements.

- **Privacy-Preserving Authentication:** It is required that all announcement messages should be authenticated. Besides, the real trajectories of vehicles should be kept private.
- **Sybil Attack Resistant:** If a vehicle tries to launch a Sybil attack, i.e., pretend to be multiple vehicles, it can be efficiently detected. Especially, the scheme should be secure, even if the vehicle is allowed to use cloud service to launch such an attack.
- **Resisting RSU Compromise:** In the system, the authenticity of messages should be kept even if some RSUs are compromised by the attacker.
- **Efficient RSU Verification:** The RSU is responsible for verifying the trajectory requests. The request verification should be efficiently processed. Besides, the RSU should also be able to handle the trajectory requests efficiently even under Denial-of-Service (DoS) attacks.

3. Preliminary

In this section we describe several cryptographic primitives that will be used in our proposed scheme.

3.1 Sequential Aggregate Signature

Sequential aggregate signature (SeqAS) allows a signer to add his signature on a different message to the current signature sequentially. The aggregated signature has the same size of the previous signature. A signature will be able to pass the verification if and only if all the aggregated signatures are generated correctly.

Definition 1 (Sequential Aggregate Signature) *A sequential aggregate signature (SeqAS) scheme consists of four algorithms* AggSetup, AggKeyGen, AggSign, *and* AggVerify, *which are defined as follows:*

- AggSetup*: The setup algorithm takes as input a security parameter and outputs public parameters.*
- AggKeyGen*: The key generation algorithm takes as input a security parameter, and outputs a public key PK and a private key SK.*
- AggSign$_{SK}(\alpha', \boldsymbol{M}; M)$*: The aggregate signing algorithm takes as input an aggregate-so-far $\alpha_{'}$ on messages* ***M****=($tr_1, ...M_k$), a message M, and a private key SK, it outputs a new aggregated signature α.*
- AggVerify$(\alpha, \boldsymbol{M}; PK)$*: The aggregate verification algorithm takes as input an aggregate signature α on messages* ***M****=($tr_1, ..., M_l$), it outputs $\top$ or $\bot$ which indicates the signature is valid or not.*

In this paper, we adopt the SeqAS signature scheme proposed by Lee *et al.* [23].

3.2 Threshold Signatures without a Trusted Dealer

A threshold signature is a distributed multi-party signature protocol. In a (θ, n)-threshold signature, a valid signature will be reconstructed only when no less than θ signers sign on the same message.

Definition 1 (Threshold Signature without a Trusted Dealer) *A (θ, n)-threshold signature scheme without trusted dealer consists of the following algorithms:*

- TSetup : *This can be an interactive algorithm run by the players $P_1, P_2, \ldots, P_n$. The algorithm outputs a public key PK. The private outputs of each player P_i is x_i such that $(x_1, \ldots, x_n) \to SK$, where SK is the secret key corresponding to PK.*
- TSign : *This algorithm generates the signature share. A player P_i takes a message m and its private key x_i as input and output a signature share σ_i.*
- Reconstruct : *This algorithm takes as a collection of θ signature shares, if all signature shares are valid, it outputs a message signature pair (m, σ); else, it outputs $\perp$.*
- TVerify : *This algorithm takes a message signature pair as input and outputs $\top/\perp$ which indicates the signature is valid or not.*

It is easy to construct a secure threshold signature algorithm using BLS short signature [24] and the distributed key generation algorithm for discrete-log based cryptosystems [25].

3.3 Message Authentication Code

A message authentication code (MAC) is a cryptographic primitive that can prevent an adversary from modifying the messages sent from one entity to another.

Definition 2 (Message Authentication Code) *A message authentication code consists of the following three algorithms:*

- Gen*: It outputs a uniformly distributed key k.*
- MAC*: It takes a message m and the secret key k as input, and outputs a tag t.*
- Vrfy *It takes a message m and the secret key k and a tag t as input and outputs $\top/\perp$ which indicates the MAC is valid or not.*

For every secret key k, it holds that $\mathsf{Vrfy}_{\mathsf{k}}(m, \mathsf{MAC}_k(m)) = \top$.

3.4 Computational Puzzle

A computational puzzle is a computational problem, which is used to verify if an entity has an estimated computational power. The hardness of a computational puzzle can be adjusted by a hardness parameter. A specific computational puzzle is usually represented by a random seed. To solve a computational puzzle, an entity needs to consume some computational resources. However, given a solution, it should be easy to verify the correctness of the solution. The hash-based puzzle is one of the famous computational puzzles which has been used in the proof of work mechanism.

4. Privacy-Preserving and Sybil Attack Resistant Announcement

In this section, we describe the proposed privacy-preserving and Sybil attack resistant announcement scheme. We first give a high level description of our scheme. Then we explain the scheme in detail.

4.1 High Level Description

In the scheme, we use the trajectory as the identity of a vehicle to achieve privacy-preserving authentication of announcement messages. The trajectory of a vehicle contains a number of RSUs and timestamps pairs which indicate that the vehicle encounters an RSU at a specific time. For privacy concerns, the identity of each RSU is masked as a time-varying tag. To get the trajectory authenticated by an RSU, a vehicle needs to solve a series of computational puzzles generated by one of its neighboring RSUs. This is the first line of defending a vehicle from generating multiple trajectories. Besides, considering that a vehicle may use cloud service to bypass this computational puzzle defense line, a vehicle is required to sign on the intermediate computational puzzle results. When a vehicle obtains a verified trajectory, it is able to send an authenticated announcement message. Finally, before accepting an announcement message, we also adopt a Sybil detection method to prevent vehicles from colluding with each other to launch a Sybil attack. The details of our scheme are described in the following subsections.

4.2 System Initialization

In this stage, RSUs run the TSetup algorithm of the (θ, n)-threshold signature scheme without a trusted dealer. At the end of this algorithm, it generates a public key PK. Each RSU R_k holds a secret key x_k. Besides, each vehicle holds the public keys corresponding to the secret key shares of the neighboring RSUs.

Besides, TA runs the Gen algorithm of message authentication code to generate the secret key MK and the $\mathsf{AggSetup}$ algorithm of the sequential aggregate signature (SeqAS) to generate the public parameters. MK is distributed to all the RSUs in the system. Each vehicle v_i generates an initial public/private key pairs $(PK_1^{v_i}, SK_1^{v_i})$ using the $\mathsf{AggKeyGen}$ algorithm of SeqAS, and obtains certificates for the public key from the TA so that vehicles can anonymously authenticate to the RSUs. Getting these certificates from the TA can be done during the annual vehicle registration at the DMV.

4.3 Trajectory Requests

In this subsection, we explain the trajectory requests stage by an example illustrated in figure 2 where a vehicle travels along 4 RSUs. It is assumed that the threshold θ of signature scheme is set to be 3. The vehicle v_i requests a trajectory proof as follows:

1) When v_i encounters the first RSU R_1, v_i interacts with R_1 as follows:

   - It computes a signature $\alpha_1^{v_i} = \mathsf{Aggsign}_{SK_1^{v_i}}(t_0)$, where t_0 is the time-stamp, then sends the request
   $$Ticket_0 := t_0 || \alpha_1^{v_i} || C_{TA}(PK_1^{v_i}),$$
   to R_k, where $C_{TA}(PK_1^{v_i})$ is a certificate of vehicle v_i which was generated by the TA;
   - On receiving the request from v_i, R_1 verifies the signature $\alpha_1^{v_i}$. If $\alpha_1^{v_i}$ is valid, it computes $\mathsf{Tag}_1 = \mathsf{MAC}_{MK}(ID_{R_1} || t_1)$, where MK is the shared secret key among all RSUs, ID_{R_1} is the identity of RSU R_1,t_1 is the current timestamp. Let $tr_1 = (PK_1^{v_1} || t_1 || \mathsf{Tag}_1)$, it then computes $\sigma_{R_1}(tr_1) = \mathsf{TSign}_{sk_{R_1}}(tr_1)$ and sends back
   $$\mathcal{T}_{R_1} := tr_1 || \sigma_{R_1}(tr_1).$$

2) When v_i receives $\mathcal{T}_{R_1}$, and travels from R_1 to R_2, it generates a new public/private key pairs $(PK_2^{v_i}, SK_2^{v_i})$ as its fresh pseudonym. It needs to solve a series of computational puzzle before

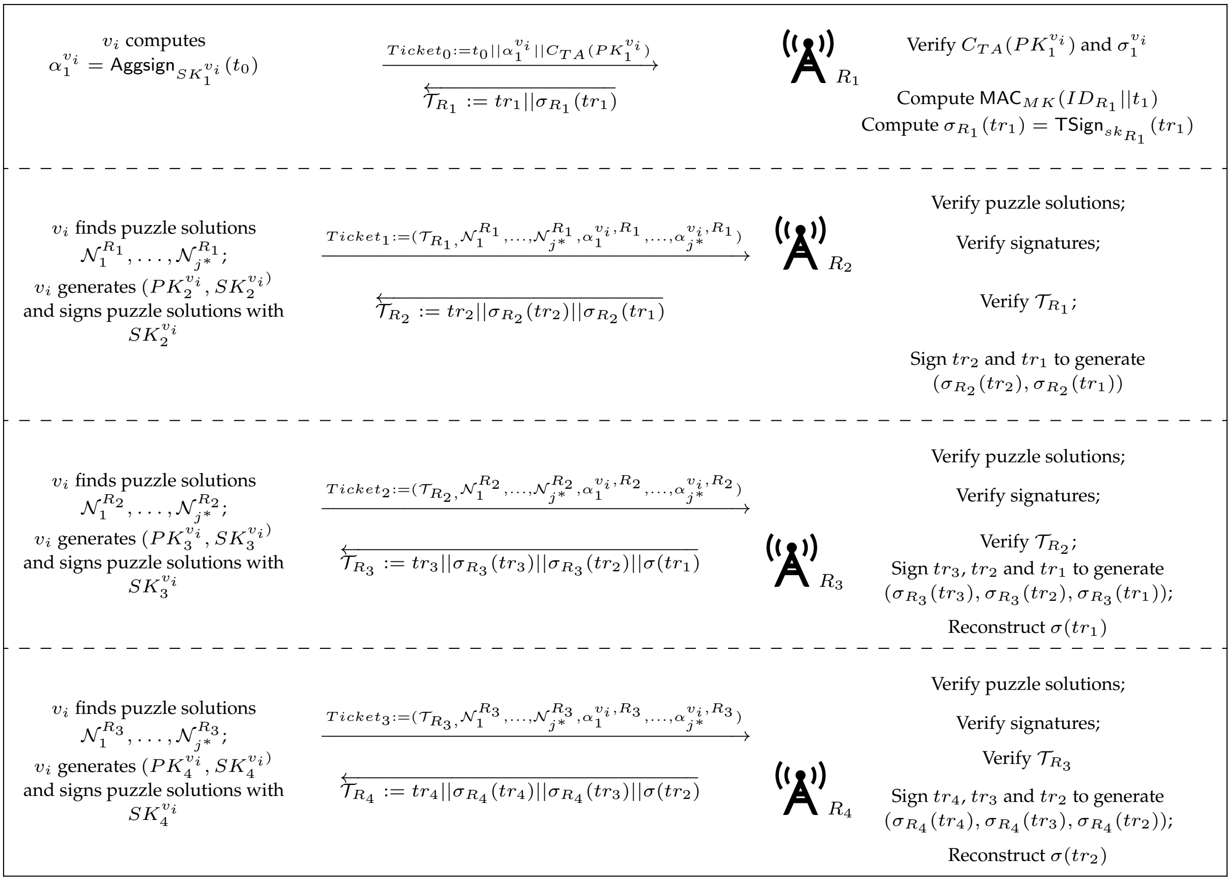

v_i computes
$\alpha_1^{v_i} = \mathsf{Aggsign}_{SK_1^{v_i}}(t_0)$
$Ticket_0 := t_0 || \alpha_1^{v_i} || C_{TA}(PK_1^{v_i})$
$\mathcal{T}_{R_1} := tr_1 || \sigma_{R_1}(tr_1)$
R_1
Verify $C_{TA}(PK_1^{v_i})$ and $\sigma_1^{v_i}$
Compute $\mathsf{MAC}_{MK}(ID_{R_1}||t_1)$
Compute $\sigma_{R_1}(tr_1) = \mathsf{TSign}_{sk_{R_1}}(tr_1)$
v_i finds puzzle solutions
$\mathcal{N}_1^{R_1}, \ldots, \mathcal{N}_{j^*}^{R_1}$;
v_i generates $(PK_2^{v_i}, SK_2^{v_i})$
and signs puzzle solutions with
$SK_2^{v_i}$
$Ticket_1 := (\mathcal{T}_{R_1}, \mathcal{N}_1^{R_1}, \ldots, \mathcal{N}_{j^*}^{R_1}, \alpha_1^{v_i, R_1}, \ldots, \alpha_{j^*}^{v_i, R_1})$
$\mathcal{T}_{R_2} := tr_2 || \sigma_{R_2}(tr_2) || \sigma_{R_2}(tr_1)$
R_2
Verify puzzle solutions;
Verify signatures;
Verify $\mathcal{T}_{R_1}$;
Sign tr_2 and tr_1 to generate
$(\sigma_{R_2}(tr_2), \sigma_{R_2}(tr_1))$
v_i finds puzzle solutions
$\mathcal{N}_1^{R_2}, \ldots, \mathcal{N}_{j^*}^{R_2}$;
v_i generates $(PK_3^{v_i}, SK_3^{v_i})$
and signs puzzle solutions with
$SK_3^{v_i}$
$Ticket_2 := (\mathcal{T}_{R_2}, \mathcal{N}_1^{R_2}, \ldots, \mathcal{N}_{j^*}^{R_2}, \alpha_1^{v_i, R_2}, \ldots, \alpha_{j^*}^{v_i, R_2})$
$\mathcal{T}_{R_3} := tr_3 || \sigma_{R_3}(tr_3) || \sigma_{R_3}(tr_2) || \sigma(tr_1)$
R_3
Verify puzzle solutions;
Verify signatures;
Verify $\mathcal{T}_{R_2}$;
Sign tr_3, tr_2 and tr_1 to generate
$(\sigma_{R_3}(tr_3), \sigma_{R_3}(tr_2), \sigma_{R_3}(tr_1))$;
Reconstruct $\sigma(tr_1)$
v_i finds puzzle solutions
$\mathcal{N}_1^{R_3}, \ldots, \mathcal{N}_{j^*}^{R_3}$;
v_i generates $(PK_4^{v_i}, SK_4^{v_i})$
and signs puzzle solutions with
$SK_4^{v_i}$
$Ticket_3 := (\mathcal{T}_{R_3}, \mathcal{N}_1^{R_3}, \ldots, \mathcal{N}_{j^*}^{R_3}, \alpha_1^{v_i, R_3}, \ldots, \alpha_{j^*}^{v_i, R_3})$
$\mathcal{T}_{R_4} := tr_4 || \sigma_{R_4}(tr_4) || \sigma_{R_4}(tr_3) || \sigma(tr_2)$
R_4
Verify puzzle solutions;
Verify signatures;
Verify $\mathcal{T}_{R_3}$
Sign tr_4, tr_3 and tr_2 to generate
$(\sigma_{R_4}(tr_4), \sigma_{R_4}(tr_3), \sigma_{R_4}(tr_2))$;
Reconstruct $\sigma(tr_2)$

Figure 2: Example of Trajectory Requests with 4 RSUs

submitting a trajectory request. In detail, let $\mathcal{N}_0 = \mathcal{T}_{R_1}, \alpha_0^{v_i} := \mathcal{T}_{R_1}$, and $j = 1$. It performs the following operations repeatedly until it arrives at the communication radius of R_2:

a) Find a nonce $\mathcal{N}_j$ such that $H(\mathcal{N}_j || PK_2^{v_i} || \mathcal{N}_{j-1}) < \beta$;
b) Compute a signature using the AggSign algorithm: $\alpha_j^{v_i} = \mathsf{AggSign}_{SK_2^{v_i}}(\alpha_{j-1}^{v_i}, \mathcal{N}_j)$;
c) $j = j + 1$.

Suppose $j = j^*$ when v_i arrives at the range of R_1, it sends the ticket

$$Ticket_1 := (\mathcal{T}_{R_1}, \mathcal{N}_1^{R_1}, \ldots, \mathcal{N}_{j^*}^{R_1}, \alpha_1^{v_i, R_1}, \ldots, \alpha_{j^*}^{v_i, R_1})$$

to R_2.

When R_2 receives the ticket, R_2 verify the ticket as follows:

a) Compute $\hat{j} = \frac{t^* - t_1}{\Delta}$, where t^* is the current timestamp and Δ is the estimated time for a vehicle to solve one computational puzzle. If $j^* \geq \hat{j}$, go to the next step; or else, return $\perp$. the estimated number of puzzles v_i can solve when traveling from R_1 to R_2,
b) If $H(\mathcal{N}_j || PK_2^{v_i} || \alpha_{j-1}^{v_i}) < T$ holds for all $j \in 1, \ldots, j^*$, go to the next step; or else, return $\perp$.
c) If $\mathsf{AggVerify}(\alpha_{j^*}^{v_i}, \mathcal{N}_0, \ldots, \mathcal{N}_{j^*}; PK_2^{v_i})$=1, go to the next step; or else, return $\perp$.
d) If $\mathsf{TVerify}(tr_1, \sigma_{R_1}(tr_1)) = \top$, and $\mathsf{Tag}_1 = \mathsf{MAC}_{MK}(ID_{R_1}||t_1)$; or else, return $\perp$.

If the ticket is valid, it sends back the partial trajectory

$$\mathcal{T}_{R_2} := tr_2||\sigma_{R_2}(tr_2)||\sigma_{R_2}(tr_1)$$

where

$$tr_2 = (PK_2^{v_i}||t_2||\mathsf{Tag}_2||t_1||\mathsf{Tag}_1),$$

and $\mathsf{Tag}_2 = MAC_{MK}(ID_{R_2}||t_2)$.

3) When the vehicle v_i moves on and reach the communication radius of R_3, it computes the new ticket

$$Ticket_2 := (\mathcal{T}_{R_2},\mathcal{N}_1^{R_2},\ldots,\mathcal{N}_{j^*}^{R_2},\alpha_1^{v_i,R_2},\ldots,\alpha_{j^*}^{v_i,R_2})$$

similar to the procedure when traveling from R_1 to R_2. When R_3 receives the ticket, it verifies the ticket. If the ticket is valid, it sends back the partial trajectory

$$\mathcal{T}_{R_3} := tr_3||\sigma_{R_3}(tr_3)||\sigma_{R_3}(tr_2)||\sigma(tr_1)$$

where

$$tr_3 = (PK_3^{v_i}||t_3||\mathsf{Tag}_3||t_2||\mathsf{Tag}_2||t_1||\mathsf{Tag}_1),$$

and $\mathsf{Tag}_3 = MAC_{MK}(ID_{R_3}||t_3)$.

We note that, to compute $\sigma(tr_1)$, R_3 needs to compute $\sigma_{R_3}(tr_1)$, and then performs the Reconstruct algorithm of the threshold signature scheme described in Section 3.2.

4.4 Message Announcement and Trajectory Update

When a vehicle encounters at least θ RSUs, a vehicle is able to generate/update a trajectory and send an announcement message with its secret key.

When considering the example that $\theta = 3$, after v_i receives $\mathcal{T}_{R_3}$, it is able to retrieve the signature $\sigma(tr_1)$ on message $tr_1 = (PK_1^{v_i}||t_1||\mathsf{Tag}_1)$. This message signature pair indicates that the vehicle with public key $PK_1^{v_i}$ reaches a RSU (masked as Tag_1) at time t_1. Hence, $tr_1, \sigma(tr_1)$ is an authenticated trajectory,v_i is able to compute the signature $\alpha^{v_i}(M) = \mathsf{AggSign}(M)$ and broadcast the announcement message

$$Annou := (M, \alpha^{v_i}(M); tr_1, \sigma(tr_1); \hat{t}),$$

where $\hat{t}$ is the timestamp.

When v_i reaches R_4, it is able to obtain

$$\mathcal{T}_{R_4} := tr_4||\sigma_{R_4}(tr_4)||\sigma_{R_4}(tr_3)||\sigma(tr_2)$$

and hence retrieve $tr_2, \sigma(tr_2)$ as its new authenticated trajectory which indicates that the vehicle with public key $PK_2^{v_i}$ reaches a RSU (masked as Tag_1) at time t_1 and another RSU (masked as Tag_2 at time t_2).

We note that, if a vehicle keeps extending one trajectory, it is possible for the announcement manager to distinguish whether two announcement messages are generated by the same vehicle or not. If a vehicle does not wish to link two messages, it just needs to drop the old trajectory and start trajectory request from scratch.

4.5 Announcement Verification and Sybil Attacks Detection

On receiving an announcement message $Annou := (M, \alpha^{v_i}(M); tr_k, \sigma(tr_k); \hat{t})$, the announcement manager verifies the message as follows:

a) If $\mathsf{TVerify}(tr_k, \sigma(tr_k)) = 1$, go to the next step; or else, return $\perp$.

b) If $\mathsf{AggVerify}(M, \alpha^{v_i}(M))$, return $\top$; or else, return $\perp$.

If an announcement message passes the above verification, the announcement manager will search all existing trajectories to finish the Sybil attack detection. Similar to the method in [12], the following two types of trajectories will be regarded as Sybil trajectories:

a) *Small windows size trajectories.* Since it is impossible for a vehicle to travel from one RSU to another in a time much shorter than an estimated time. Any trajectory contains a window size shorter than a certain time will be regarded as a Sybil trajectory.

b) *Long trajectories within a time period.* Since a vehicle is not able to encounter a large number of RSUs within a fixed time. Any trajectory that is longer than an estimated length will be regarded as a Sybil trajectory.

If a trajectory survives the Sybil attack detection and the signature verification, this announcement message will be accepted by the announcement manager.

5. Analysis and Evaluation

5.1 Security Analysis

We show that our scheme achieves privacy-preserving authentication and Sybil attack resistant via the following two theorems.

Theorem 1 *Our scheme achieves privacy-preserving authentication property defined in Section 2.2.*

Proof: We show our scheme achieves privacy-preserving authentication in the following two aspects:

Firstly, a message will be accepted by the announcement manager if and only if both $(m_k, \sigma(m_k))$ and $M, \alpha^{v_i}(M)$ are valid message signature pairs. The first signature guarantees that the trajectory m_k are signed by at least θ RSUs. The second signature guarantees that the announcement message is signed by the vehicle with public key $PK_k^{v_i}$ who is the owner of the trajectory m_k. Combining the unforgeability of SeqAS signatures and threshold signatures, we have that the announcement messages are authenticated.

Secondly, the real trajectories of vehicles are hidden to both eavesdroppers and curious announcement managers. Since vehicles will generate a fresh pseudonym when it travels from one RSU to another, it is not possible to link different pseudonyms for both eavesdroppers and announcement manager. Since all the identities of RSUs are masked as a time-varying tag, it is not possible to link two vehicles who passed by the same location at different time. Hence, it is not possible for vehicles to reveal the real trajectories of vehicles.

Overall, we have that our scheme achieves privacy-preserving authentication. □

Theorem 2 *Our scheme achieves Sybil attack resistant property defined in Section 2.2.*

Proof: Firstly, it is difficult for a vehicle to generate two valid trajectory requests to a specific RSU. when a vehicle travels from one RSU to another, it has to solve a series of computational puzzles. Due to the security of the hash functions, the computational hash puzzle cannot be solved with less than an estimated time. Besides, since the initial random seed of the puzzle i.e., $\mathcal{N}_0 := \mathcal{T}_{R_1}$ is generated by the previous RSU, hence the hash puzzle cannot be pre-computed. Furthermore, all the subsequent random seeds depend on solutions to previous hash puzzles, hence also the hash puzzle should be computed sequentially. This guarantees that it is difficult for a vehicle to generate two valid trajectory requests to a specific RSU.

Secondly, a rational attacker will not choose to outsource the computational puzzle. Since every solution to a computational puzzle should be signed before it proceeds to the next computational puzzle. Hence, if a vehicle chooses to outsource the computational puzzle to the cloud service, it has to provide its secret key to the cloud. Or else, the vehicle has to communicate frequently with the cloud service, and the communication delay might be even longer than the time to solve one computational puzzle.

Overall, the proposed scheme is secure against Sybil attacks. □

In the following, we briefly argue that our scheme also satisfies the property of resisting RSU compromise and efficient RSU verification.

Resisting RSU compromise: The security of the threshold signature scheme guarantees that the signature is unforgeable even if the adversary has $\theta - 1$ secret keys. Hence, even if $\theta - 1$ RSUs are compromised by an attacker, it is not possible to generate a forged trajectory.

Efficient RSU verification: When verifying the validity of a trajectory request, the RSU needs to verify the correctness of the solution to the hash puzzle and the intermediate signatures. It is obvious that the puzzle verification is fast since only hash computation is required. Hence, the computational puzzle also guarantee the efficiency of RSU verification even under the DoS attacks. Besides, since aggregate signature is adopted in our scheme, multiple signatures generated by the same vehicle can be verified efficiently.

5.2 Theoretical Comparison

In table 1, we compare the efficiency and security of proposed scheme with the result of Wu *et al.* [9] and Baza *et al.* [11]. For the efficiency, we compare the signature size and and computational cost of message announcement and verification. T_{Exp} and $T_{Pairing}$ refer to the running time of exponential computation in $\mathbb{G}_1$ and pairing computation. From the benchmark result, the running time of $T_{Pairing}$ is around 10 times of T_{Exp}. The signature size and signature generation cost of our scheme is comparable with related works. When verifying multiple signatures, the computational cost of our proposal is lower than that of [9] and [11]. Besides, compared with existing result, our scheme achieves Sybil attack resistance without requiring long enough trajectories.

Table 1: Theoretical Comparison

	Signature	**Sign**	**Verify n signatures**	**PPA**[†]	**Sybil**	**RSU Compromise**
[9]	160 Bytes	$6T_{Exp}$	$3nT_{Pairing}$	✓	✗	✗
[11]	35 Bytes	T_{Exp}	$nT_{Pairing}$	✓	✓[‡]	✗
This work	**96 Bytes**	$\mathbf{4T_{Exp}}$	$\mathbf{2(n-1)T_{Exp} + 5T_{Pairing}}$	✓	✓	✓

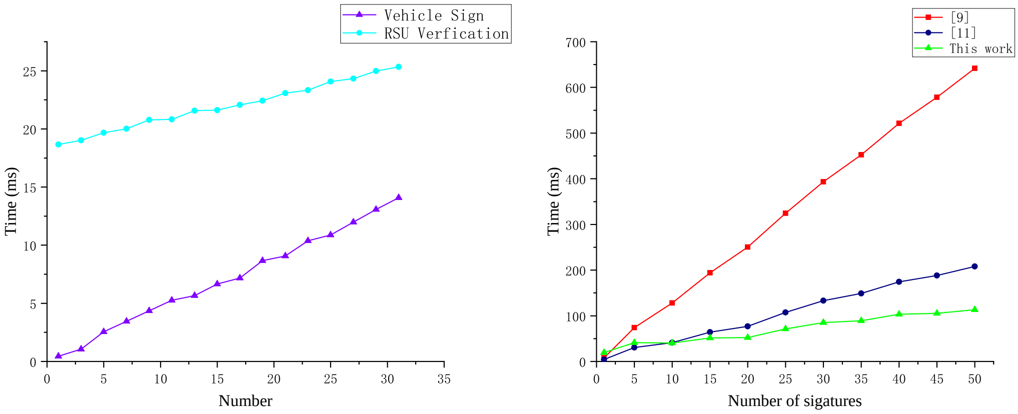
[†] PPA – privacy-preserving authentication; Sybil – Sybil attack resistance; RSU Compromise – security against RSU compromise attacks.

[‡] It achieves Sybil attack resistance with the restriction of long enough trajectories.

5.3 Efficiency Evaluation

To evaluate the efficiency of our scheme, we first perform a simulation to evaluate the computational cost of signature generation and verification. Besides, we compare the signature verification cost of our proposal with that of Wu *et al.*[9] and Baza *et al.*[11]. The simulations were run on a Linux machine with an Intel Core i7-4790 processor @ 3.6 GHz. The bilinear group is implemented by a BN Curve with 128 bits security level. We use the MIRACL library to implement the cryptographic algorithms of our scheme. We simulate the stages where a vehicle requests a trajectory and the RSU verifies the signature of vehicles. In the trajectory request generation phase, we only evaluate the signature generation time since the time for solving a puzzle is almost fixed.

Figure 3(a) shows the computational cost of generating and verifying signatures of our scheme. When the number of puzzles a vehicle needs to solve varies from 1 to 31, the signature generation time grows from 0.45 milliseconds to 14 milliseconds. Meanwhile, the cost of signature verification grows much slower than the signature generation time. When the number of puzzles grows from 1 to 31, the signature verification time for RSU grows from 18.6 ms to 26.3 ms. Figure 3(b) compares the computation cost for verifying n signatures of our proposal with that of Wu *et al.*[9] and Baza *et al.*[11]. It is shown that when the number of signatures is larger than 10, our proposal has a lower computational cost than the result of [9] and [11].



(a) The cost of sign and verify

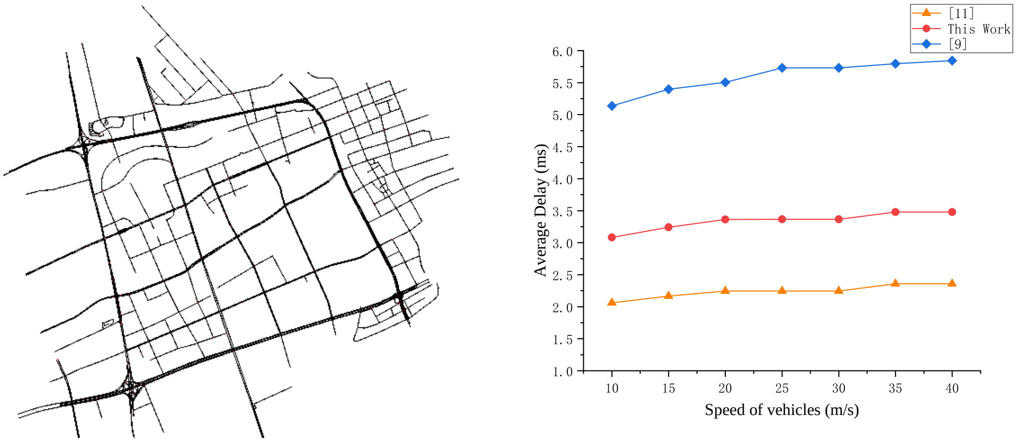
(b) Comparison

Figure 3: The computational costs

We then use MOSAIC to simulate the communication delay of the proposed scheme and compares our result with that of [9] and [11]. The road scenario is shown in Figure 4(a) which is an area of 5 × 10 km^2 in Shanghai, China. The average speed of vehicles grows from 10 m/s to 40 m/s. The number of vehicles is set to be 100. The time for each simulation is 400 s. Similar to [9], the communication delay is computed as the average delay of each message for each vehicle. Figure 4(b) shows the simulation result. At the cost of realizing a stronger security guarantee, the communication delay of our proposal is slightly higher than [11]. However, the highest delay of our proposal in the simulation is less than 4 ms.

6. Conclusion

Privacy-preserving authentication and Sybil attack defense are major challenges in IoV. In this work, we propose to use verified trajectories as the identities of vehicles to achieve privacy-preserving authentication and Sybil attack resistance. Since all the trajectories are masked, the privacy of vehicles is preserved.



(a) The road scenario

(b) The simulation result

Figure 4: The communication delay

Furthermore, with the help of trajectories, Sybil attacks can be efficiently detected. Benefit from our proposed puzzle chains, our scheme is secure against Sybil attacks in the presence of cloud service assistant attackers. Finally, efficient evaluation shows that the computational overhead of our scheme is acceptable.

Acknowledgement

This work was supported by Shanghai Sailing Program (21YF1413800,20YF1413700), and National Science Foundation of China (No. 62002213).

References

[1] Shanzhi Chen, Jinling Hu, Yan Shi et al. "Vehicle-to-everything (v2x) services supported by lte-based systems and 5g". *IEEE Communications Standards Magazine*, vol. 1, no. 2, 70–76, 2017.

[2] Feng Li, Kwok-Yan Lam, Zhengguo Sheng et al. "Agent-based spectrum management scheme in satellite communication systems". *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, 2877–2881, 2021.

[3] Kwok-Yan Lam, Sananda Mitra, Florian Gondesen and Xun Yi. "Ant-centric iot security reference architecture–security-by-design for satellite-enabled smart cities". *IEEE Internet of Things Journal*, 2021.

[4] Anupam Chattopadhyay, Kwok-Yan Lam and Yaswanth Tavva. "Autonomous vehicle: Security by design". *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] Lili Miao, John Jethro Virtusio and Kai-Lung Hua. "Pc5-based cellular-v2x evolution and deployment". *Sensors*, vol. 21, no. 3, 843, 2021.

[6] Shanzhi Chen, Jinling Hu, Yan Shi and Li Zhao. "Lte-v: A td-lte-based v2x solution for future vehicular network". *IEEE Internet of Things journal*, vol. 3, no. 6, 997–1005, 2016.

[7] Shanzhi Chen, Jinling Hu, Yan Shi, Li Zhao and Wen Li. "A vision of c-v2x: technologies, field testing, and challenges with chinese development". *IEEE Internet of Things Journal*, vol. 7, no. 5, 3872–3881, 2020.

[8] Xiuhua Li, Luxi Cheng, Chuan Sun et al. "Federated-learning-empowered collaborative data sharing for vehicular edge networks". *IEEE Network*, vol. 35, no. 3, 116–124, 2021.

[9] Qianhong Wu, Josep Domingo-Ferrer and Ursula González-Nicolás. "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications". *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, 559–573, 2009.

[10] Jian Liu, Jiangtao Li, Lei Zhang et al. "Secure intelligent traffic light control using fog computing". *Future Generation Computer Systems*, vol. 78, 817–824, 2018.

[11] Mohamed Baza, Mahmoud Nabil, Mohamed Mohamed Elsalih Abdelsalam Mahmoud et al. "Detecting sybil attacks using proofs of work and location in vanets". *IEEE Transactions on Dependable and Secure Computing*, 2020.

[12] Shan Chang, Yong Qi, Hongzi Zhu, Jizhong Zhao and Xuemin Shen. "Footprint: Detecting Sybil Attacks in Urban Vehicular Networks". *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, 1103–1114, June 2012.

[13] Jie Cui, Jing Zhang, Hong Zhong and Yan Xu. "Spacf: A secure privacy-preserving authentication scheme for vanet with cuckoo filter". *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, 10283–10295, 2017.

[14] Shehzad Ashraf Chaudhry. "Designing an efficient and secure message exchange protocol for internet of vehicles". *Security and Communication Networks*, vol. 2021, 2021.

[15] Ikram Ali and Fagen Li. "An efficient conditional privacy-preserving authentication scheme for vehicle-to-infrastructure communication in vanets". *Vehicular Communications*, vol. 22, 100228, 2020.

[16] Rongxing Lu, Xiaodong Lin, Haojin Zhu, P-H Ho and Xuemin Shen. "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications". In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1229–1237. IEEE, 2008.

[17] Xiaodong Lin, Xiaoting Sun, Pin-Han Ho and Xuemin Shen. "Gsis: A secure and privacy-preserving protocol for vehicular communications". *IEEE Transactions on vehicular technology*, vol. 56, no. 6, 3442–3456, 2007.

[18] Yong Hao, Yu Cheng and Kui Ren. "Distributed key management with protection against rsu compromise in group signature based vanets". In *IEEE GLOBECOM 2008-2008 IEEE global telecommunications conference*, pages 1–5. IEEE, 2008.

[19] Yipin Sun, Zhenqian Feng, Qiaolin Hu and Jinshu Su. "An efficient distributed key management scheme for group-signature based anonymous authentication in VANET". *Security and Communication Networks*, vol. 5, no. 1, 79–86, 2012.

[20] Chao Lin, Debiao He, Xinyi Huang, Neeraj Kumar and Kim-Kwang Raymond Choo. "Bcppa: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks". *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[21] Muhammad Baqer Mollah, Jun Zhao, Dusit Niyato et al. "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey". *IEEE Internet of Things Journal*, vol. 8, no. 6, 4157–4185, 2020.

[22] Liqun Chen, Siaw-Lynn Ng and Guilin Wang. "Threshold anonymous announcement in vanets". *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, 605–615, 2011.

[23] Kwangsu Lee, Dong Hoon Lee and Moti Yung. "Aggregating cl-signatures revisited: Extended functionality and better efficiency". In *International Conference on Financial Cryptography and Data Security*, pages 171–188. Springer, 2013.

[24] Dan Boneh, Ben Lynn and Hovav Shacham. "Short signatures from the weil pairing". In *International conference on the theory and application of cryptology and information security*, pages 514–532. Springer, 2001.

[25] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk and Tal Rabin. "Secure distributed key generation for discrete-log based cryptosystems". *Journal of Cryptology*, vol. 20, no. 1, 51–83, 2007.