PBA: Prediction-Based Authentication for Vehicle-to-Vehicle Communications

Chen Lyu, Dawu Gu, Yunze Zeng, and Prasant Mohapatra

Abstract—In vehicular networks, broadcast communications are critically important, as many safety-related applications rely on single-hop beacon messages broadcast to neighbor vehicles. However, it becomes a challenging problem to design a broadcast authentication scheme for secure vehicle-to-vehicle communications. Especially when a large number of beacons arrive in a short time, vehicles are vulnerable to computation-based Denial of Service (DoS) attacks that excessive signature verification exhausts their computational resources. In this paper, we propose an efficient broadcast authentication scheme called Prediction-Based Authentication (PBA) to not only defend against computation-based DoS attacks, but also resist packet losses caused by high mobility of vehicles. In contrast to most existing authentication schemes, our PBA is an efficient and lightweight scheme since it is primarily built on symmetric cryptography. To further reduce the verification delay for some emergency applications, PBA is designed to exploit the sender vehicle's ability to predict future beacons in advance. In addition, to prevent memory-based DoS attacks, PBA only stores shortened re-keyed Message Authentication Codes (MACs) of signatures without decreasing security. We analyze the security of our scheme and simulate PBA under varying vehicular network scenarios. The results demonstrate that PBA fast verifies almost 99 percent messages with low storage cost not only in high-density traffic environments but also in lossy wireless environments.

Index Terms—VANETs, broadcast communication, signatures, DoS attacks, prediction-based authentication

1 Introduction

Vehicular ad hoc networks (VANETs) have recently attracted extensive attentions as a promising approach to enhancing road safety, as well as improving driving experience. By using a Dedicated Short-Range Communications (DSRC) [1] technique, vehicles equipped with wireless On-Board Units (OBUs) can communicate with other vehicles and fixed infrastructure, e.g., Road-Side Units (RSUs), located at critical points of the road [2]. Therefore, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are regarded as two basic types of communications in VANETs.

Once VANETs become available, numerous safe, commercial and convenient services can be deployed through a variety of vehicular applications. These applications mostly rely on vehicles' OBUs to broadcast outgoing beacon messages and validate incoming ones. The broadcast beacons often contain information about position, current time, speed, direction, driving status, etc. For example, by frequently broadcasting and receiving beacons, drivers are better aware of obstacles and collision scenarios. They may act early to avoid any possible damage, or to assign a new route in case of a traffic accident in the existing route. However, before implementing these attractive applications, particularly safety-related ones, we must first address and resolve VANET-related security issues [3], [4], [5].

To secure vehicular networks, an authentication scheme is indispensable to ensure messages are sent by legitimate vehicles and not altered during transmissions. Otherwise, an attacker can easily disrupt the normal function of VANETs by injecting bogus messages. Therefore, vehicles should broadcast each message with a digital signature. However, the current VANET signature standard [6] using Elliptic Curve Digital Signature Algorithm (ECDSA) would cause high computational overhead on the standard OBU hardware, which has limited resources for cost constraints. Prior work has shown that one ECDSA signature verification requires 20 milliseconds on a typical OBU with a 400 MHz processor [7]. When a large number of signed messages are received in a short time period, an OBU cannot process them before their dedicated deadline. In this paper, we define this attack as *computation-based DoS attacks*. Even without any malice, the computation-based DoS attacks can be easily initiated in a high-density traffic scenario. For example, when traffic-related messages (beacons) are sent 10 times per second as suggested by the DSRC protocol [1], [6], a vehicle is overwhelmed with more than five neighbors within its radio range. To defend against such attacks, most existing schemes [8], [9], [10] make use of the technology of identity-based batch verification [11] or aggregate signature [12] built on asymmetric cryptography to improve the efficiency of verification. In their schemes, the computational cost is mainly dominated by a few operations of pairing and a number of operations of point multiplication over the elliptic curve [13]. It is affordable for RSUs, but expensive for OBUs to verify the messages [14]. Furthermore, if attackers inject false beacons, the receiver is hard to locate them so that these schemes are also vulnerable to the computation-based DoS attacks [15]. Therefore, designing an effective

- *C. Lyu and D. Gu are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {chen_lv, dwgu}@sjtu.edu.cn.*
- *Y. Zeng and P. Mohapatra are with the Department of Computer Science, University of California, Davis, CA 95616 USA. E-mail: {zeng, pmohapatra}@ucdavis.edu.*

Manuscript received 10 Nov. 2014; accepted 5 Jan. 2015. Date of publication 2 Feb. 2015; date of current version 20 Jan. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2015.2399297

authentication scheme under high-density traffic scenarios is a big challenge for V2V communications.

In this paper, we propose an effective broadcast authentication scheme: Prediction-Based Authentication (PBA) to defend against computation-based DoS attacks for V2V communications. Unlike most of existing schemes based on asymmetric cryptography [8], [9], [10], [15], [16], [17], [18], [19], [20], our PBA is primarily implemented on symmetric cryptography, whose verification is more than 22 times faster than ECDSA. In addition, PBA resists packet losses naturally. Similar to mobile wireless networks, packet losses are common in VANETs. Especially, Bai et al. have shown that the packet loss rate can reach 30 percent in a benign network, and nearly 60 percent in a congestion network [21]. We design our PBA on the TESLA scheme [22], [23], [24], which is proposed to secure lossy multicast streams with hash chains. With TESLA signatures piggyback, PBA operates smoothly even when the packet loss rate is high.

PBA also aims at improving the efficiency of authentication. Certain vehicular applications may require receivers to verify urgent messages immediately. To support instant verification, we exploit the property of predictability of a future beacon, constructing a Merkle Hash Tree (MHT) [25] to generate a common public key or predication outcome for the beacon. With the prediction outcome known in advance, receivers can instantly verify the incoming beacon. Furthermore, we examine the storage overhead brought by our authentication scheme. If a mechanism brings a large storage burden, an attacker would initiate *memory-based DoS attacks* where an OBU is overwhelmed by storing a large number of unverified signatures. To defend against such attacks, PBA records shortened re-keyed MACs instead of storing all the received signatures.

We design PBA with an objective of providing effective, efficient, scalable broadcast authentication and also non-repudiation in VANETs. To the best of our knowledge, prior authentication schemes for V2V communications either lack non-repudiation, or fail to operate in high packet loss or high-density traffic scenarios. The main contributions of this work are:

- First, we analyze the security requirements for broadcast authentication in VANETs, and design a lightweight authentication scheme called PBA for V2V communications. Without the participation of RSUs or other vehicles, PBA is a distributed scheme and operated independently.
- Second, PBA is designed to minimize the computational cost and storage overhead of authentication. Lightweight MAC and hash operations are mostly performed in PBA to defend against computation-based DoS attacks. To reduce the storage overhead, PBA exploits a local secret key to construct new shortened MACs of signatures without sacrificing security.
- Third, PBA enables instant verification. With the predictability of a vehicle's position, we construct a MHT to commit all the possible results of the vehicle's movements between successive two beacons. Signature verification can be instantly performed based on prediction outcomes from MHTs integrated into beacons in advance.

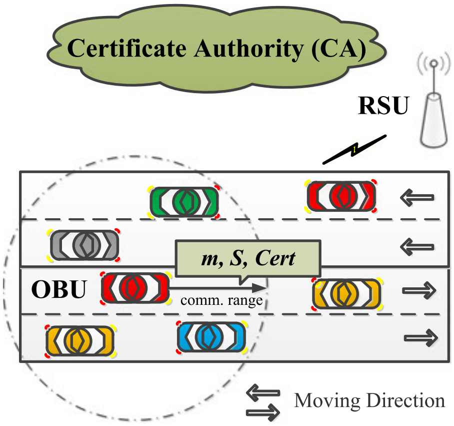


Fig. 1. Typical VANET scenario. A vehicle's OBU will periodically broadcast a beacon 10 times per second.

- Finally, analytical and empirical validations are done to evaluate our PBA scheme. We prove PBA is secure, and use Markov chains to analyze the effect of packet losses on the authentication delay and storage cost. Extensive simulations also indicate that PBA achieves excellent performance while incurring low delay and storage cost.

The rest of the paper is organized as follows. Section 2 introduces background on VANET settings and cryptographic primitives. Section 3 describes the security requirement and threat model. In Section 4, we present the construction of PBA. A detailed analysis of PBA is provided in Section 5. In Section 6, we present our evaluation results. Section 7 summarizes related work on authentication in VANETs. Finally, we conclude our work in Section 8. A preliminary version of parts of this paper was reported in [26].

2 Background

In this section, we provide an overview of the VANET setting and the basic TESLA scheme.

2.1 VANET Setting

We divide VANET messages into two types based on the distance that they are going to spread, which means these packets are either single-hop beacons or multi-hop traffic data. For secure multi-hop traffic data, the standard ECDSA scheme [6] performs well when messages are sent infrequently. In this paper, we focus on the single-hop relevant applications, where vehicles periodically exchange beacons with nearby vehicles that are within the radio range.

As shown in Fig. 1, based on the IEEE 1609.2 standard, vehicles will periodically broadcast beacon information (e.g., position, velocity and time) 10 times per second to avoid the traffic accidents and react to unsafe situations. These information can be obtained from on-board devices such as GPS sensors, which could support nanosecond-level timing accuracy and meter-level positioning accuracy [7].

In the IEEE 1609.2 standard, a Public Key Infrastructure (PKI) is required for key management in VANETs. Each vehicle is equipped with a pair of ECDSA keys: a private key for signing and a public key for verification. These keys would be issued by a Certificate Authority (CA). Each key

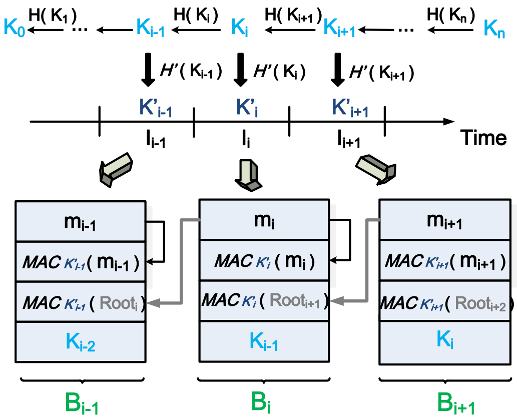


Fig. 2. Chained keys generation.

pair will be stored in the vehicle's OBU, with tamper-resistant property to defend against the compromising attack.

A VANET beacon often contains a message body m, the sender's signature S, and the public key certificate of the sender $Cert$. The creation time is included in m which could help receivers determine the message's deadline. S ensures that the sender is accountable for this message, and thus prevents drivers from releasing malicious information. $Cert$ is used to announce the sender's public key and identify the sender's legality.

2.2 TESLA

TESLA is an efficient scheme based on symmetric cryptography [22], [23], [24]. It makes use of one-way hash chains with delayed disclosure of keys to achieve source authentication. For TESLA to operate securely, the sender and the receiver should be loosely time synchronized, which means that the synchronization does not need to be precise, but the receiver requires to know an upper bound on the sending time [23].

Consider the chain of length n with the values $K_1, \dots, K_n$ for time intervals $I_1, \dots, I_n$ (as shown in Fig. 2). TESLA can generate this chain by randomly selecting the last value K_n and repeatedly applying a one-way hash function H to derive the previous values: $K_i = H(K_{i+1})_{\forall i \in \{0,\dots,n-1\}}$. The beginning of the chain, K_0 serves as a commitment to the entire chain and allows anybody to authenticate the following values of the chain. Moreover, TESLA uses a second hash function H' to derive the key K'_i: $K'_i = H'(K_i)$, which is used to compute the MACs of the messages for each time interval.

To authenticate a message for an interval I_i, a sender broadcasts the message m_i with a MAC of the message using the sender's key for this interval (K'_i). The key K'_i remains secret for the future $d-1$ intervals, so recipients need to store the entire message and MAC until the sender broadcasts the key. After d intervals, the sender discloses the key. Then, receivers check the key by recovering the commitment with iteratively invoking the hash function. If it is valid, they apply the verified key to check the stored MAC.

TESLA can guarantee the receiver never accepts a message as an authentic message unless it was actually sent by the sender [23]. As a lightweight authentication scheme, TESLA also tolerates arbitrary packet loss. However, a drawback of TESLA is that the receiver has to buffer packets one disclosure delay before it can authenticate them. Moreover, TESLA does not provide non-repudiation, since the receiver cannot convince a third party that the message arrived from the claimed sender.

3 Security Requirement and Threat Model

In this section, we will discuss the desirable security requirements of a broadcast authentication scheme in VANETs, and describe the potential attacks against those requirements.

3.1 Security Requirement

An efficient authentication scheme should guarantee timely message authenticity and non-repudiation. Meanwhile, it should resist packet losses and DoS attacks for relevant applications in VANETs. Here, we discuss each of these properties in detail.

Timely authentication. With the authentication mechanism, receivers can ensure that a message was sent by a valid vehicle and it has not been modified during the transmission. Furthermore, timely signature verification is essential since each message has an expiration time by which the receiver should verify it. In VANETs, single-hop relevant applications usually have a shorter deadline.

Non-repudiation. The property of non-repudiation allows a receiver to prove to a third party that the sender is accountable for generating the message. If the broadcast mechanism lacks non-repudiation, an adversary can claim it to be another party that created the message. Non-repudiation usually implies authentication, so the receiver can identify the sender and detect the manipulation of bogus packets.

Packet losses resistant. Packet losses are common in wireless networks, especially in VANETs. When a packet is lost during the transmission, it should have little influence for the receiver to verify other subsequent packets.

DoS attacks resistant. Given the relatively expensive nature of signature verification, attackers may initiate computation-based DoS attacks that broadcasting a number of invalid signatures overwhelms the receivers' computational resources. If an authentication scheme brings large storage overhead, attackers may initiate memory-based DoS attacks which overwhelm the receivers' memory resources by broadcasting a number of invalid malicious messages. An authentication mechanism should have low computational and memory cost such that other applications can be operated normally in VANETs.

3.2 Threat Model

An attacker may pretend to be another entity, generate or modify a packet, or block future packets to prevent authentication. We assume that an attacker can modify a series of packets from a sender without signatures. If the sender broadcasts the signature for the last few packets, the attacker can intercept the signature so that receivers are unable to authenticate packets.

We consider both computation-based and memory-based DoS attacks, which are caused by one or more colluding attackers broadcasting invalid signatures or a number of legitimate vehicles sending valid message signatures within the radio range. We consider packet losses are caused by the poor quality of communication channels (e.g., high mobility of vehicles). We do not consider flooding attacks where attackers flood a high volume of beacons to block the communication, because receivers can quickly identify them. To protect the privacy of vehicles, pseudonym-based scheme [5], [27] could be exploited that OBUs periodically change public keys in our scheme. Jamming attacks [28], [29], [30] are out of the scope of this paper.

4 The PBA Scheme

This section presents PBA, which makes use of both ECDSA signatures and TESLA-based scheme to authenticate beacons. Similar to the TESLA scheme, PBA also requires loose time synchronization. In VANETs, it is naturally supported since messages sent by GPS-equipped vehicles are time-stamped with nanosecond-level accuracy.

By looking into beacons, we find that the information in a beacon except a vehicle's position is almost deterministic based on its previous beacons. As also mentioned in [7], the entropy of beacons is relatively low from the sender vehicle's point of view. Given the past trajectory, a vehicle's future position can be predicted as the vehicle's movement is mainly restricted by the road topology and speed limit. We mainly use this fact to construct our PBA scheme. We will next describe how it authenticates beacons.

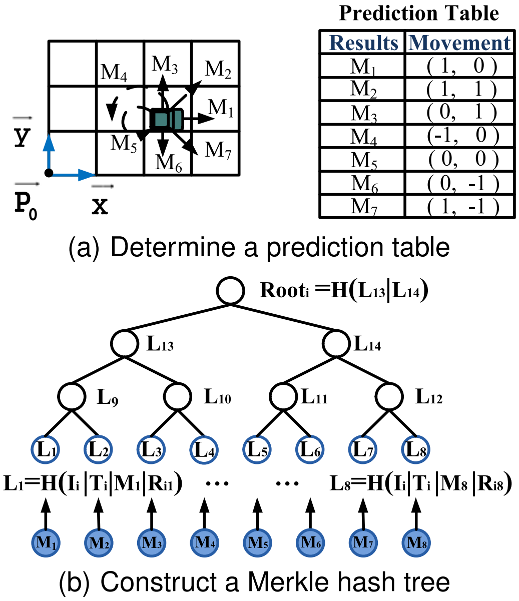
4.1 Protocol Overview

Our PBA includes the process of generating a signature by a sender and verifying the signature by a receiver. We introduce them separately.

First, each vehicle splits its timeline into a sequence of time frames. Each time frame is also divided into a sequence of beacon intervals, which we remark $I_0, I_1, \ldots, I_n$. In a time frame, to send the first beacon B_0 for I_0, a vehicle will perform four steps: *chained keys generation*, *position prediction*, *Merkle hash tree construction*, and *signature generation*. To send other beacons in that time frame, the vehicle only operates the last three steps.

1) *Chained keys generation.* At the beginning of a time frame, each vehicle generates n chained private keys for the next n beacons. It uses one interval worth of private key for authentication as the TESLA scheme. In the following description, we call these private keys TESLA keys.
2) *Position prediction.* At each beacon interval, each vehicle predicts its position broadcast in the next beacon. To do so, vehicles model all the possible results of movements between two consecutive beacons based on information of the past trajectory, as shown in Fig. 3a.
3) *Merkle hash tree construction.* After position prediction, the vehicle will construct one interval worth of a public key and private keys. These private keys are associated with the results of movements. We propose a MHT, which ties these pre-computed keys together and then generates a single public key or prediction outcome for all the possible movements.

   As illustrated in Fig. 3b, $Root_i$ is the prediction outcome for all the results of movements from I_{i-1} to I_i.



$\vec{y}$
$\vec{P_0}$
$\vec{x}$
M_1
M_2
M_3
M_4
M_5
M_6
M_7

Prediction Table

Results	Movement
M_1	(1, 0)
M_2	(1, 1)
M_3	(0, 1)
M_4	(-1, 0)
M_5	(0, 0)
M_6	(0, -1)
M_7	(1, -1)

(a) Determine a prediction table

$Root_i = H(L_{13}|L_{14})$
L_{13}
L_{14}
L_9
L_{10}
L_{11}
L_{12}
L_1 L_2 L_3 L_4 L_5 L_6 L_7 L_8
$L_1 = H(I_i|T_i|M_1|R_{i1})$ ··· ··· $L_8 = H(I_i|T_i|M_8|R_{i8})$
M_1 M_2 M_3 M_4 M_5 M_6 M_7 M_8

(b) Construct a Merkle hash tree

Fig. 3. Example of Merkle hash tree construction. Each leaf node in a tree corresponds to one entry in the prediction table, and the inner node is the hash of the two children.

4) *Signature generation.* After position prediction and MHT construction, a vehicle signs the commitment of the hash chain and the prediction outcome from MHT using ECDSA signatures, and broadcasts it along with the first beacon B_0 in the time frame. For the rest of beacons such as $B_1, B_2, \ldots, B_n$, the vehicle signs the message and the prediction outcome from MHT using the TESLA keys assigned in the intervals $I_1, I_2, \ldots, I_n$ (shown in Fig. 4).

After receiving a beacon, a vehicle will perform the following two steps:

1) *Self-generated MAC storage.* To reduce the storage cost of unverified signatures, the receiver only records a shortened re-keyed MAC. When the receiver keeps the used key secret, PBA provides security guarantees according to the size of beacon interval and network bandwidth.
2) *Signature verification.* For the first beacon, the receiver verifies the ECDSA signature. To verify the following signed B_i, the receiver will get the corresponding TESLA key, and reconstruct the prediction outcome from MHT (shown in Fig. 4). If a matching MAC of prediction outcome is found in the memory, the receiver authenticates the beacon instantly. Otherwise, the receiver authenticates it with the later TESLA key.

4.2 Chained Keys Generation

Before sending any beacon, a vehicle first generates n chained keys for signing and a commitment K_0 like the TESLA scheme, as shown in Fig. 2.

As we mentioned before, the drawback of the TESLA scheme is that the receiver needs to buffer packets some intervals before it can authenticate them. This might not be practical for certain single-hop relevant applications where timing is usually critical. We modify the basic TESLA

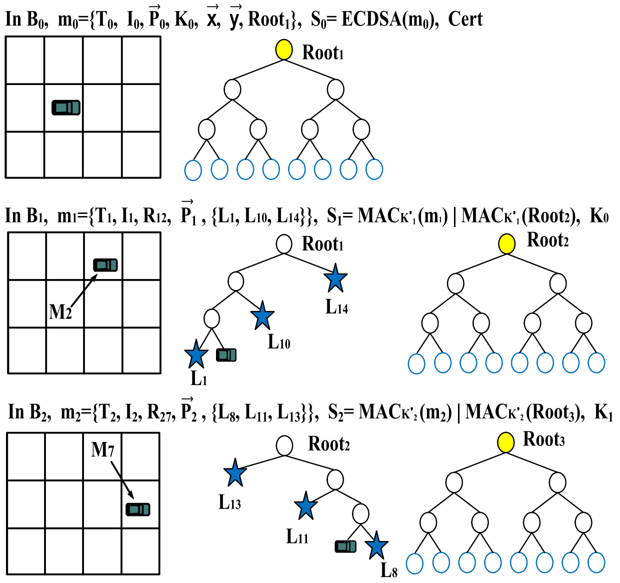

In B0, m0={T0, I0, P0, K0, x, y, Root1}, S0= ECDSA(m0), Cert
Root1
In B1, m1={T1, I1, R12, P1 , {L1, L10, L14}}, S1= MACK'1(m1) | MACK'1(Root2), K0
M2
Root1
L14
L10
L1
Root2
In B2, m2={T2, I2, R27, P2 , {L8, L11, L13}}, S2= MACK'2(m2) | MACK'2(Root3), K1
M7
Root2
L13
L11
L8
Root3

Fig. 4. Signature broadcast and verification. To verify B_1 or B_2, the receiver gets the TESLA key K_0 or K_1, rebuilds the root of MHT with the information in B_1 or B_2, and then checks whether the root matches the one signed in B_0 or B_1.

scheme to support instant authentication, which allows the receiver to verify packets as soon as they arrive.

In our TESLA-based scheme, the sender predicts the next interval's message m_{i+1} in the interval I_i, and gets the prediction outcome $Root_{i+1}$ (we show the detailed process of prediction outcome construction in Sections 4.3 and 4.4). To construct the beacon packet B_i, the sender picks the TESLA key K_i for I_i, and appends the MAC over m_i and $Root_{i+1}$ with K_i', respectively. As illustrated in Fig. 2, the beacon B_i is shown as: $m_i \mid MAC_{K_i'}(m_i) \mid MAC_{K_i'}(Root_{i+1}) \mid K_{i-1}$, where the last item means the disclosed TESLA key. Here, the notion $\mid$ stands for message concatenation.

We now briefly present how our TESLA-based scheme works. In Fig. 2, when the beacon B_i with the disclosed key K_{i-1} arrives at a receiver, it allows the receiver to verify the beacon B_{i-1} sent in interval I_{i-1}. B_{i-1} carries the prediction outcome $Root_i$ for m_i. Therefore, the message m_i can be immediately verified with $Root_i$ and K_{i-1}.

Dealing with packet losses. If certain previous beacon, such as B_{i-1}, is lost or dropped due to the poor quality of wireless channel, we cannot immediately authenticate the incoming beacon B_i. However, we are able to authenticate it with the original TESLA signature $MAC_{K_i'}(m_i)$, where the TESLA key K_i is disclosed in or after interval I_{i+1}.

4.3 Position Prediction

As position is the main source of uncertainty in beacons, we discuss how the sender vehicle predicts its own future positions.

For every two consecutive beacons, such as B_{i-1} and B_i, PBA requires the sender to model all the possible results of the distance vector differences or movements between them. The output of this step is a prediction table PT_i in which each entry represents one possible movement between I_{i-1} and I_i. Inspired by the work [7], [26], we also use a local coordinate to express the sender's future positions.

We place the origin of this local coordinate at the beginning position $\vec{P_0}$ of the current time frame. A pair of orthogonal vectors (i.e., $\vec{x}$ and $\vec{y}$) are also required, the scalar of which can be chosen according to a desired level of positioning accuracy. Then, every future position $\vec{P_i}$ could be represented as $\vec{P_i} = \vec{P_0} + a_i\vec{x} + b_i\vec{y}$, where a_i and b_i are rounded to integers. Hence, the movement from the interval I_{i-1} to I_i is:

$$\vec{M_i} = \vec{P_i} - \vec{P}_{i-1} = (a_i - a_{i-1})\vec{x} + (b_i - b_{i-1})\vec{y}, \tag{1}$$

which can be briefly given by a pair of integers $(a_i - a_{i-1}, b_i - b_{i-1})$.

As shown in Fig. 3a, the prediction table PT_i collects all the possible results of $\vec{M_i}$. Here, we are not interested in accurately modeling the mobility of a vehicle given the past trajectory, which is orthogonal to our work. In this work, we would like to design a broadcast signature scheme working with an arbitrary prediction model.

4.4 Merkle Hash Tree Construction

Given the prediction table, the vehicle needs to generate a single public key (or prediction outcome) for all the possible movements. It first generates private keys, which are associated with the results of movements in PT_i. Then, a MHT structure is proposed to tie these keys together and generate a single public key or prediction outcome for all the movements.

A MHT structure is a binary tree structure where each leaf is assigned a hash value and an inner node is assigned the hash value of its children. As shown in Fig. 3b, for an entry M_k in PT_i (which shows that the vehicle will move to location $\vec{P}_{i-1} + \vec{M_k}$ with a certain probability in interval I_i), there is a leaf labeled as $L_k = H(I_i|T_i|M_k|R_{ik})$ in the MHT, where R_{ik} is a random value to prevent signature forgery. The inner node is the hash of the two children, e.g., $L_9 = H(L_1|L_2)$. The root of the MHT is also computed by hashing the concatenation of its two children, i.e., $Root_i = H(L_{13}|L_{14})$. Then, the sender obtains $Root_i$, which is the predication outcome of the message m_i based on the prediction table PT_i.

4.5 Signature Generation

After generating the commitment K_0, constructing the prediction table with a local coordinate, and producing the MHT's root $Root_1$ for the next beacon B_1, the sender broadcasts the first beacon in a time frame. It contains public keys, time stamp T_0, and other important parameters (such as, its local coordinate system). We format the first beacon as $B_0 = \{m_0, S_0, Cert\}$, where $m_0 = \{T_0, I_0, \vec{P_0}, K_0, \vec{x}, \vec{y}, Root_1\}$ is signed by ECDSA, and a $Cert$ is issued by a CA.

For I_i, being at the position $\vec{P_i}$ and time T_i, the vehicle will locate the leaf node corresponding to $\vec{P_i}$ in the MHT, and broadcast the necessary values and off-path nodes of this leaf in m_i. We define off-path nodes are the siblings of the nodes on the path from one leaf to the root of MHT. For example, in Fig. 4, the car shows the leaf associated with the current location and time. At T_1, the sender moves to $\vec{P_1} = \vec{P_0} + \vec{M_2}$, associated with L_2. Hence, m_1 includes the random value and off-path nodes: $\{R_{12}, L_1, L_{10}, L_{14}\}$.

Similarly, m_2 also includes the random value and off-path nodes for I_2.

To construct the signature of m_i, the sender first picks the TESLA key K_i for the interval I_i. Then, by performing the steps of position prediction and MHT construction, it obtains the root value $Root_{i+1}$ for I_{i+1}. Finally, the sender signs m_i and $Root_{i+1}$ with K'_i. As shown in Fig. 4, the signature of m_1 includes the TESLA signature $MAC_{K'_1}(m_1)$ and $MAC_{K'_1}(Root_2)$.

Thus, except the first beacon, the broadcast B_i includes the message m_i, the signature S_i, and the TESLA key K_{i-1} which is disclosed for receivers to verify previous beacons.

Reducing the communication overhead. As the random value and off-path nodes are contained in the message, the size of beacon is larger than before. To reduce the communication overhead, we could decrease the number of off-path nodes with Huffman hash tree instead of Merkle hash tree. Note that, if Huffman hash tree is used to reduce the communication overhead, it will take effect only when an OBU predicts its movement accurately [7].

4.6 Self-Generated MAC Storage

In a time frame, as the first beacon B_0 is signed by ECDSA, a receiver will directly store K_0, $Root_1$ and other local parameters if it passes the verification. Except B_0, when the receiver gets the signature of a beacon B_i, it will store a self-generated MAC to reduce memory cost. Algorithm 1 depicts the operations of the receiver.

Algorithm 1. Self-Generated MAC.

```
Require: Beacon $B_i$, Local secret key $SK_{loc}$
 1: Check the security condition;
 2: if not satisfied then
 3:   Drop the beacon
 4: else
 5:   Compute
      $MAC_{RS_{i+1}} = MAC_{SK_{loc}}(MAC_{K'_i}(Root_{i+1}))$
 6:   Store $MAC_{RS_{i+1}}$
 7:   if $K_{i-1}$ is valid then
 8:     Reconstruct the MHT's root node $Root'_i$
 9:     Recompute
        $MAC'_{RS_i} = MAC_{SK_{loc}}(MAC_{K'_{i-1}}(Root'_i))$
10:     if Search $(MAC'_{RS_i}) == 1$ then
11:       Accept $m_i$
12:       Free memory for $MAC_{RS_i}$
13:     else
14:       Compute
          $MAC_{MS_i} = MAC_{SK_{loc}}(MAC_{K'_i}(m_i))$
15:       Store $m_i$ and $MAC_{MS_i}$
16:     end if
17:     Verify previously received messages
        Free memory for $m_g$ and $MAC_{MS_g}(g < i)$
18:   end if
19: end if
```

The security of the basic TESLA scheme depends on the TESLA keys that remain secret until a pre-determined time period [23]. PBA builds on the basic TESLA scheme, so the receiver must verify the key K_i, which is used to generate the signature of the beacon, has not yet been disclosed by the sender (Line 1). If this security condition does not hold, the receiver must drop the beacon, because it cannot assure the authenticity any more (Line 2 and 3). Otherwise, it recomputes the MAC of the signed prediction outcome with a local secret key SK_{loc}: $MAC_{RS_{i+1}} = MAC_{SK_{loc}}(MAC_{K'_i}(Root_{i+1}))$ (Line 5). Note that, SK_{loc} is only known by the receiver. The receiver stores this shortened MAC (i.e., $MAC_{RS_{i+1}}$) until the next interval I_{i+1} (Line 6). The lifetime of $MAC_{RS_{i+1}}$ is one interval in memory since it is only useful to achieve instant verification of B_{i+1}.

The incoming B_i also contains the TESLA key K_{i-1}. The receiver will check whether it can use K_{i-1} to verify B_i and some previous unverified beacons (Line 7). To verify B_i, the receiver first reconstructs the MHT's root node $Root'_i$ (Line 8, we present the reconstruction process in Sec. 4.7). It then calculates the shortened MAC (i.e., $MAC'_{RS_i} = MAC_{SK_{loc}}(MAC_{K'_{i-1}}(Root'_i))$) (Line 9), and compares it with the one stored in memory. If a matching MAC is found (Line 10), m_i is authenticated (Line 11) and the receiver can free the memory (Line 12). If none of the stored MACs match MAC'_{RS_i}, the receiver considers that the prediction outcome of the message lost. Thus, it will compute the shortened MAC of the message (i.e., $MAC_{MS_i} = MAC_{SK_{loc}}(MAC_{K'_i}(m_i))$) (Line 14), store m_i and MAC_{MS_i} (Line 15), and wait for the later key for authentication. Moreover, the disclosed TESLA key K_{i-1} might allow the receiver to verify previously received messages and then free the memory (Line 17).

Here, we set the size of original MACs to be 160 bits and the size of short MACs 32 bits. Given the interval of 100 ms as suggested by the IEEE standard, we will prove that receivers could use shorter MACs to store signatures without decreasing security. We also find that the receiver's memory consumption is related to the packet loss rate in VANETs. Assuming the lifetime of beacons to be N, we will discuss the upper-limit of memory consumption for PBA in Section 5.3.

4.7 Signature Verification

For the first beacon B_0, ECDSA signature can provide the property of non-repudiation. It helps the receiver ensure that the sender is accountable for the parameters such as the initial position $\vec{P}_0$ and the commitment of hash chains K_0, and thus prevents drivers from broadcasting malicious information.

To verify the following signed B_i, the receiver verifies the validity of K_{i-1} by following the one-way key chain back to K_0 signed with ECDSA. It recomputes the root value $Root'_i$ of MHT given relevant values in the m_i, and checks whether it matches $Root_i$ stored in the memory. If not, the receiver will verify m_i with the later TESLA key.

In the example of Fig. 4, the receiver gets the tree root $Root_1$ from the first beacon. In I_1, it reconstructs L_2 from the values (e.g., R_{12}) in the message, and calculates the hash tree root based on L_2 and the off-path hashes $\{L_1, L_{10}, L_{14}\}$. If the calculated root $H(H(H(L_1|L_2)|L_{10})|L_{14})$ matches $Root_1$, the receiver is convinced that the sender moves $\vec{M}_2$ distance from I_0 to I_1, being located at $\vec{P}_1 = \vec{P}_0 + \vec{M}_2$. In I_2, the receiver of B_2 reconstructs the hash tree root as before, and then does MAC operations towards the root with the keys K'_1 and SK_{loc}. If the value matches MAC_{RS_2} stored in the memory, the receiver is convinced that the sender moves $\vec{M}_7$ distance from I_1 to I_2, being located at $\vec{P}_2 = \vec{P}_1 + \vec{M}_7$.

Public key rebroadcasting. As K_0 is only sent at the beginning of a time frame, if a vehicle A encounters a vehicle C after C broadcasts its current K_0, A cannot verify C's beacons until the next time frame. To overcome this issue, we may consider that vehicle C signs K_0 by ECDSA with the certificate every second (10 beacons) on demand. Hence, after waiting several beacon intervals, the receiver A is able to authenticate beacons.

Here, we do not specialize how often vehicle C signs K_0 by ECDSA as we only give a general solution of broadcast authentication in VANETs. It is absolutely possible to consider the length of time frame and the frequency of ECDSA signature when we have a specific application. The system designer can easily modify our scheme according to the applications' needs. For example, in an application where time demand is tight, vehicle A may send a request packet to vehicle C for K_0, and C will return the ECDSA signature immediately. After getting it, vehicle A can initiate authentication with this trust commitment.

5 Analysis

In this section, we first prove that PBA is secure. Then, we discuss the performance of PBA in wireless lossy environments. Finally, we analyze the storage requirements of PBA. We assume the packet loss rate is p, and a beacon's lifetime is N $(N \geq 1)$ intervals from the time that a sender generates the beacon.

5.1 Security Proof

PBA relies on the symmetric cryptographic functions (hashes and MACs) and the basic TESLA scheme. We begin by assuming these cryptographic functions are secure. The security of the TESLA scheme has been proved in previous work [24]. Besides the basic TESLA scheme, new mechanisms are proposed in PBA to provide more properties. On one hand, a sender broadcasts a MAC before it sends the beacon to support instant authentication. On the other hand, by using a secret key on the received MAC, the receiver generates a shortened MAC to reduce the possibility of memory-based DoS attacks. However, these new mechanisms will become useless if they enable adversaries to spoof other vehicles. Here, we show a detailed security proof of PBA.

Theorem 1. *If the underlying MAC algorithms and hash chains are secure, given a receiver vehicle's key is securely kept, PBA provides a negligible probability that an attacker could forge a legitimately authenticated message in the context of VANETs, independent of the attacker's computational capability.*

To prove this theorem, we need to prove the following two lemmas.

Lemma 1. *Assuming that the underlying MAC algorithms and hash chains are secure, broadcasting the MAC of a message's prediction outcome is secure.*

Proof. Based on the known MAC, the aim of the attackers is to generate false messages and pretend to be the original sender. To achieve this purpose, they will try all kinds of methods to be successful.

First, an attacker may try to find a different prediction outcome $\hat{Root}_{i+1}$, which results in the same MAC as the original $Root_{i+1}$: $MAC_{K'_i}(Root_{i+1}) = MAC_{K'_i}(\hat{Root}_{i+1})$. However, producing such an outcome means the underlying MAC was not secure under an adaptive chosen-message attack.

Second, an attacker may want to get the undisclosed TESLA key K_i before the sender broadcasts it so that it can produce any valid MAC and message pair. However, to successfully find such an undisclosed key, the attacker should defeat the one-way property of hash chains, which is not feasible on computation.

Finally, an attacker may intend to create a message $\tilde{m}$, where there is some new $\tilde{L}_z$ such that $\tilde{L}_z \neq L_z$ but $\hat{Root}_{i+1} = Root_{i+1}$ in the MHT. Provided that it succeeds, there must exist a collision of hash function. Without loss of generality, we will show that with $z = 1$. The attacker constructs the root of MHT, i.e., $\hat{Root}_{i+1}$, with a structure like Fig. 3b.

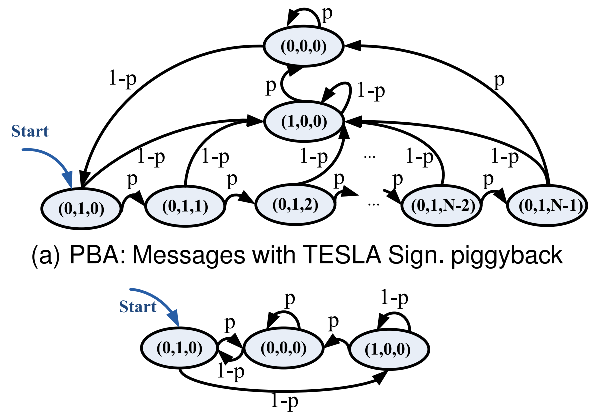
Let $\tilde{L}_9 = H(\tilde{L}_1|L_2)$. If $L_9 = \tilde{L}_9$, there exists a collision of hash function. Else, we have $L_9 \neq \tilde{L}_9$. Let $\tilde{L}_{13} = H(\tilde{L}_9|L_{10})$. If $L_{13} = \tilde{L}_{13}$, then $L_9|L_{10}$ and $\tilde{L}_9|L_{10}$ form another collision of H. Else, we have $L_{13} \neq \tilde{L}_{13}$. With $\hat{Root}_{i+1} = Root_{i+1}$, it produces a collision: $L_{13}|L_{14}$ and $\tilde{L}_{13}|L_{14}$. Therefore, a collision of hash function must exist at certain step. □

Lemma 2. *Provided a receiver's key SK_{loc} is securely kept, PBA provides security guarantees based on the parameters (t_I, W_B), where t_I is the size of the beacon interval and W_B is the network bandwidth.*

Proof. Without a receiver's key SK_{loc}, an attacker has no method to calculate the shortened MAC for one prediction outcome. Therefore, as the best strategy, the attacker will broadcast MACs as many times as possible for a given beacon interval to make the receiver record a new shortened MAC. Then, the attacker tries to spoof a message D' with one sender's valid TESLA key K_i to correspond to a MAC value $\tilde{MAC}_{RS}$. The receiver will mistakenly trust the attacker if he previously stored the shortened MAC: $\tilde{MAC}_{RS} = MAC_{SK_{loc}}(MAC_{K'_i}(D'))$.

Provided that the size of the shortened MAC is X_s bits, there are 2^{X_s} MACs in all. Hence, to successfully forge an arbitrary message, the attacker should send $2^{X_s} \log 2^{X_s}$ MACs on average in a beacon interval. The probability of the attacker successfully spoofing a message is $\frac{W_B \cdot t_I}{G_m \cdot 2^{X_s} \cdot \log 2^{X_s}}$, where W_B is the DSRC bandwidth, t_I is the size of beacon interval, and G_m shows the average length of beacons.

We exaggerate the bandwidth of VANETs to be 100 Mbps. According to the IEEE standard, we pick the value of t_I from 100 to 300 ms. If we choose the size of original MAC 160 bits and shortened MAC 32 bits, the probability of success is reduced to 10^{-6}. Therefore, although a receiver only stores a shortened MAC, the attacker is



(a) PBA: Messages with TESLA Sign. piggyback

(b) Messages without TESLA Sign. piggyback

Fig. 5. Markov chains for authentication in lossy situations.

different to make the receiver trust a forged message in VANETs. □

5.2 Authentication in Wireless Lossy Situations

In this part, we consider how long our PBA scheme takes for an authentication of one beacon, when packet loss occurs in VANETs.

First, we will show how many intervals are needed to authenticate a beacon on average. As shown in Fig. 5a, we use a Markov Chain to model the interaction between packet losses and our authentication scheme, where the current state includes three elements. The first element shows whether the authentication has occurred (i.e., Authenticated State with 1 or Unauthenticated State with 0). The second element presents the loss or reception of the later packet that produces the MAC value (i.e., 0 or 1 MAC). The last element shows how many beacon intervals or delays from a receiver have taken place (i.e., $0, 1, 2, \ldots, N-1$). Fig. 5b shows the Markov Chain which represents the authentication process without TESLA signatures piggyback. In this case, the receiver will not store the messages. The main difference between Fig. 5a and 5b is that the receiver in (b) only verifies the beacon based on the previous prediction outcome.

We analyze the average time taken in the transition states of the Markov Chains, and get average beacon intervals for a receiver to verify a beacon broadcast by a sender.[1] For PBA, we find that on average T_{f_a} intervals are required to successfully authenticate one beacon, which is related with the average number of beacons required to reach {(1, 0, 0)} state from{(0, 1, 0)} state.[2]

$$T_{f_a} = \frac{1}{(1-p)(1-p^N)} - 1. \tag{2}$$

Similarly, by analyzing the chain in Fig. 5b, we find that on average T_{f_b} intervals are required to authenticate one beacon without TESLA signatures piggyback:

$$T_{f_b} = \frac{1}{(1-p)^2} - 1. \tag{3}$$

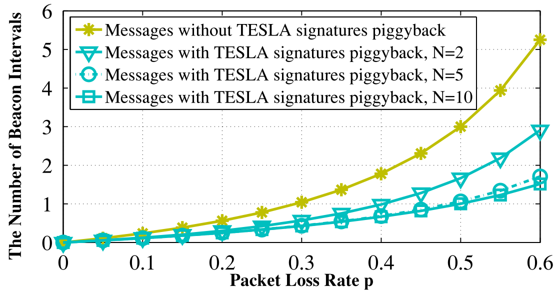


Fig. 6. Average intervals for a receiver to successfully authenticate a beacon as the packet loss rate p grows.

As N increases, the impact level in Equation (2) decreases faster than in Equation (3), which is also shown in Fig. 6. When p increases, more intervals are taken on authentication as expected. Through all these curves, we observe that messages with TESLA signatures piggyback are authenticated in less intervals with a larger N. Nevertheless, the improved gain becomes small when N continues increasing.

In all, we find that PBA resists packet losses effectively due to TESLA signatures piggyback. In lossy environments, the performance of authentication delay can be further improved with a large value of N. However, more storage overhead will be introduced with a larger N. We will discuss this issue in next section.

5.3 Storage Requirements

In our scheme, we only store smaller MACs to prevent memory-based DoS attacks. Here, we discuss the upper-limit of memory consumption for our PBA scheme, which is a function of how much data are broadcast by senders in an interval and how long these data are stored by receivers.

We first get the expression of average memory consumption for one beacon. With the Markov Chain, we model the interaction between the packet loss and memory cost for PBA. The states in the chain encode that how many intervals from a receiver have occurred (i.e., $0, 1, 2, \ldots, N$), and the receiver stores either the shortened MAC or both the message and the shortened MAC. We use Q to indicate the matrix of one-step transition probability Q_{ij}. For long process, the probability of each state j is expressed by Π_j, which is the unique solution of $\Pi_j = \sum_{i=0}^{N} \Pi_i Q_{ij}$ and $\sum_{j=0}^{N} \Pi_j = 1$. Then, on average E_s storage is needed for one beacon, where X_s is the size of the shortened MAC of the prediction outcome, $|m_c|$ is the average length of the message, and X_m is the size of the shortened MAC of the message.

$$E_s = X_s \cdot \Pi_0 + (X_m + |m_c|) \cdot \left(\sum_{i=1}^{N-1} i \cdot \Pi_i + (N-1) \cdot \Pi_N \right). \tag{4}$$

To compute the upper-limit of memory consumption F_s, we also need to consider how much data are broadcast by senders in one interval. Therefore, given that the bandwidth of VANET channel is W_B, a receiver will at

[1] Here, we neglect the verification time produced by hash operations, and only calculate it in unit of beacon interval for simplicity.

[2] Note that, this analysis excludes the authentication of the first beacon.

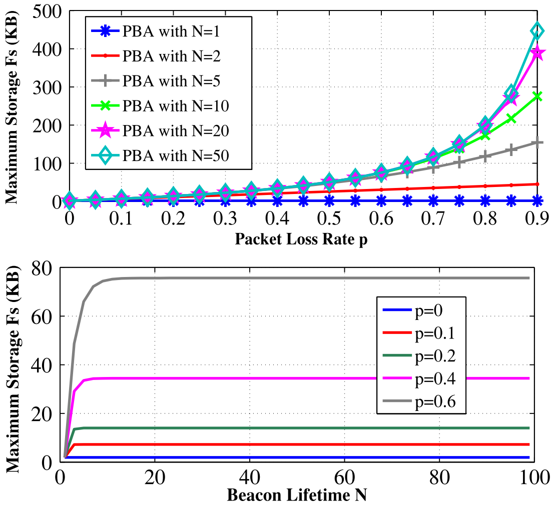


Fig. 7. The estimated maximum storage overhead F_s as the function of beacons' lifetime N and the packet loss rate p, with $W_B = 6$ Mbps, $t_I = 100$ ms, $G_m = 160$ Bytes, $|m_c| = 100$ Bytes, and $X_s = X_m = 4$ Bytes.

most store the maximum number of beacons sent in one beacon interval (t_I) times average memory saving for one beacon :

$$F_s = \frac{t_I \cdot W_B}{G_m} \cdot E_s, \tag{5}$$

where G_m represents the average length of beacons.

Fig. 7 shows the maximum storage overhead of PBA with different combinations of N and p, given $W_B = 6$ Mbps and $t_I = 100$ ms. When $N \geq 2$, the curve of F_s increases rapidly as the packet loss rate p grows, since more beacons are verified by the TESLA mechanism leading to higher storage overhead.

Given a value of $p = 0.2$, there are about 80 percent of messages that could be instantly authenticated by the receiver. For the rest of messages, the receiver is able to handle them before their lifetime when N is not small (e.g., $N \geq 10$). Therefore, for a value of p, we can see that F_s would keep a maximum value when N continues increasing. According to the analysis, it is worth noting that an OBU can process the maximum number of data required to be stored even with a limited memory space of 1 MBytes.

6 Simulation Results

To evaluate the performance of PBA, we use NS-3 to simulate the algorithm in a variety of VANET topologies. First, we consider a sender vehicle sends a beacon every 100 ms, and moves along the trajectory pre-defined for the simulation. The receiver vehicle receives the beacons with the probability $1 - p$. Then, we simulate PBA together with ECDSA, TESLA and VAST [31] in more real road situations, with more sources sending beacons.

The parameters commonly used in VANETs are listed in Table 1. Moreover, a prediction table is required to model the vehicle's future positions. Actually, some car suppliers or application providers of VANETs could offer advanced traffic statistics model to build the accurate prediction table. For simulation, however, we construct a large prediction table to cover most of a vehicle's movements in a beacon interval, with 129 km/h of maximum speed limit. We set the block unit to be 2 meters with commodity GPS's positioning accuracy. For each beacon interval, we make use of six layers of MHT in our simulation.

TABLE 1
Parameters

Parameter	Value
ECDSA's Generation Time	7 ms
ECDSA's Verification Time	22 ms
Hash or MAC Operation Time	1 μs
ECDSA Signature Size	512 bits
MAC, MAC Key Size	160 bits
Vehicle's Radio Range	300 meters
Bandwidth of DSRC Channel	6 Mbps
Beacon's Lifetime N	5 or 10 (0.5 or 1 sec)
Time Frame n	$10 - 500$ $(1 - 50$ sec)
Packet Loss Rate p	$0 - 0.6$
Traffic Density	$2 - 100$ vehicles

6.1 Single-Neighbor Case

We first discuss the impact of the time frame n, the packet loss rate p, and the lifetime of beacons N on our PBA scheme. We will evaluate PBA based on these four metrics:

- Sender's computational overhead, which is expressed by the average time for a beacon's signature generation;
- Receiver's computational overhead, which is expressed by the average time for a beacon's signature verification;
- Packet processing rate of a receiver, which is defined as the ratio of beacons successfully verified to beacons received;
- Storage cost for a beacon's verification, which is defined as the total amount of Bytes stored by vehicles.

Fig. 8 shows the performance of PBA with various p and N under different time frames. Both the sender and receiver's computational cost reduce with the increasing of time frame. This is because hash and MAC operations, which are done much faster than the operations of ECDSA verification, have a high proportion in the overall computation, especially when the time frame is set to be a large value. From the results, we can see that PBA only requires about 0.1 ms to sigh a beacon and less than 1 ms to verify the beacon, which significantly outperforms the standard ECDSA scheme.

Fig. 9 shows that the packet processing rate is affected by both p and N. When p begins to increase due to wireless losses or highly dynamic environments, some beacons are lost so that the incoming beacons will be not verified instantly and buffered in the queue. If N is large enough, the receiver can verify the beacons even under high packet loss rate (e.g., $p = 0.6$). In this case, PBA can still maintain almost 100 percent packet processing rate. Otherwise, the curve of packet processing rate declines when beacons are out of date and then dropped. On the storage overhead, we compare the simulation results with theoretical analysis obtained by Equation (4) in Section 5.3. We find the theoretical analysis predicts the performance very accurately.

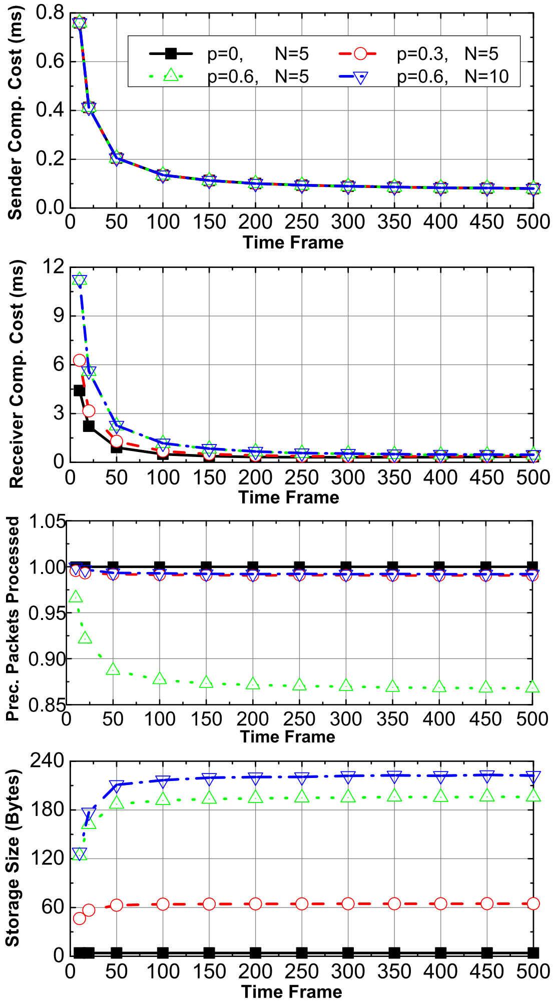


Fig. 8. PBA: simulation for different time frames.

As a summary, our simulation results confirm that PBA reduces the computational cost of sender and receiver drastically. It can resist packet losses, and maintain high packet processing rate with low storage overhead even at highly dynamic environments.

6.2 Multi-Neighbors Case

In this part, we will simulate the performance of PBA in a road topology under different traffic density and packet loss rate p. We also compare it with other three authentication schemes for V2V communications. We set the time frame to be 20 seconds, and the lifetime of beacons to be one second. Other parameters are set as default shown in Table 1.

Except the two metrics of packet processing rate and storage cost, we will use another new metric for evaluation: overall delay. It is defined as the total authentication time of a valid beacon from the time that it is produced by a sender to the time that it is accepted by a receiver.

6.2.1 Traffic Density Scenarios

We first analyze PBA's performance with different traffic density, and evaluate it in lossless scenarios of VANETs where there is no packet loss.

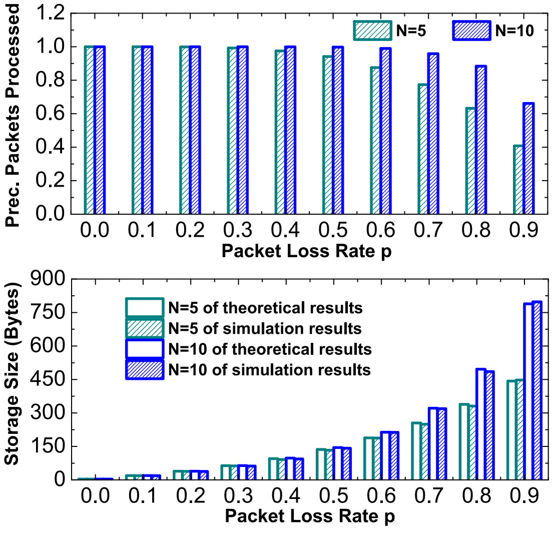


Fig. 9. PBA: simulation for different packet loss rate, and comparison with theoretical analysis.

From Fig. 10, it can be observed that PBA's overall delay is lower than other three protocols, especially as the number of OBUs within the radio range increases. It can achieve instant authentication by a few hash and MAC operations based on the previously broadcast prediction outcomes. For ECDSA, even in a low-density traffic scenario, the overall delay reaches the maximum as most of beacons cannot be verified before the deadline. The schemes of TESLA and VAST do not authenticate a message with a MAC until the MAC's key disclosure. Here, the disclosure delay is set to be one beacon interval, so receivers need to store the message for 0.1 second and then verify it.

We investigate the impact of traffic density on the packet processing rate. The TESLA-based authentication schemes (TESLA, VAST and our PBA scheme) work pretty well even in high-density traffic scenarios. In terms of overhead, we can see that PBA's excellent performance will not be affected by the traffic density.

Through all the scenarios, PBA performs best with a little authentication delay and storage overhead, and almost 100 percent of received packets authenticated.

6.2.2 Wireless Lossy Scenarios

In lossy scenarios, when one vehicle sends beacons, other neighbor vehicles receive these beacons with probability $1 - p$. We test our scheme under the traffic density of average 20 cars within the radio range.

As shown in Fig. 11, the increase of p slightly extends the overall delay and storage overhead of PBA. In particular, when a number of prediction outcomes are lost in highly dynamic networks, receivers should buffer a mass of beacons in the queue and wait for future TESLA keys to verify them.

The simulation results also highlight the effectiveness of the TESLA signatures piggyback mechanism when packet losses happen in VANETs. Even when p grows to 0.6, our PBA scheme could maintain excellent packet processing rate. With more wireless errors, ECDSA decreases the computational load due to less requests of signature verification. It can be observed that the performance of VAST changes rapidly with different p. It performs well when p is

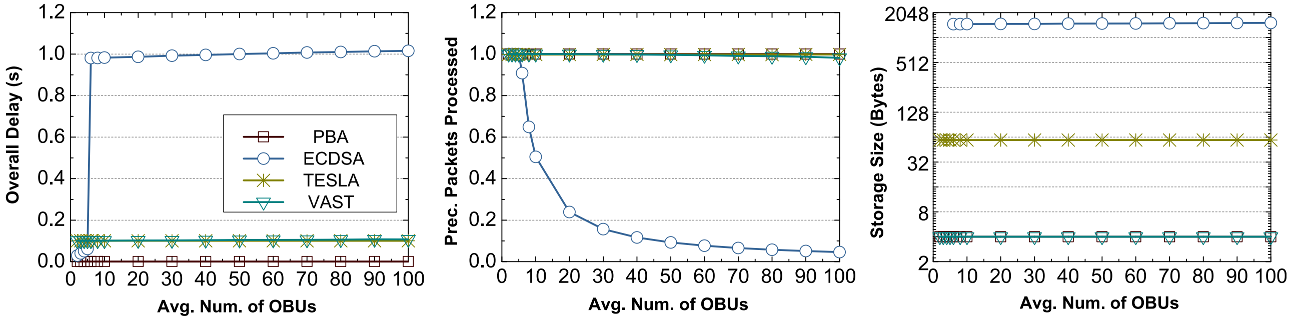


Fig. 10. Performance comparison results with different traffic density: overall delay (left), packet processing rate (middle), and storage cost (right).

no larger than 0.3. However, when p keeps increasing, the performance becomes bad since the operations of ECDSA verification become significant in the overall computation.

In face of wireless losses, we conclude that PBA is not only effective but also efficient. It verifies nearly 99 percent of beacons at extremely low delay with small storage overhead.

7 Related Work

Many related studies have been reported on authentication issues for VANETs [7], [8], [9], [10], [15], [16], [17], [18], [19], [26], [27], [20], [31], [32], [33], [34], [35]. These works mainly try to solve one of these three problems: key or certificate management, privacy-preservation and efficient broadcast authentication.

In [32], Studer et al. propose a key management scheme to satisfy the security and privacy requirements in VANETs. They use short-lived keys to sign messages to preserve the OBU's privacy, and revoke the certificate timely if the OBU's misbehavior is detected. In [33], Hass et al. make use of Certificate Revocation Lists (CRLs) to distribute the revocation information in VANETs, which could help a receiver OBU check the revocation status of a sender. As the size of CRL is expected to be large, they use a Bloom filter [36] to store the certificate identifiers, which would take less memory and computational overhead to determine whether a certificate is on the CRL or not. To reduce the authentication delay caused by checking the long CRL, Wasef et al. [34] employ a keyed MAC function to do fast checking process for the OBU's certificate.

There are also some works concentrating on the problem of privacy issues for VANETs. To hide the identity of the signer, group signature-based schemes [37], [38] are made use of in [20], [27], [32]. However, these schemes would fail if a group manager who possesses the group master key arbitrarily reveals the group member's identity. In addition, for V2V communications, the selection of group leader will sometimes become a bottleneck as OBUs could not find a trusted entity among vehicles. In [35], the authors introduce a random key-set based authentication protocol to preserve the vehicles' privacy. To achieve the conflicting goals of privacy and traceability, Sun. et al. [17] propose a privacy-preserving defense scheme by combining the mechanism of pseudonyms and the technology of identity-based threshold signature [39].

For efficient broadcast authentication, there are some works [8], [9], [10] using batch signature verification [11] or aggregate signature schemes[12] for V2I communications. An RSU will verify multiple received signatures at the same time such that the total verification time could be reduced. In their schemes, the computational cost is mainly dominated by a few operations of pairing and a number of operations of point multiplication over the elliptic curve [13]. It is affordable for RSUs, but expensive for OBUs to verify the messages [14]. Furthermore, if attackers inject false beacons, it is so hard for the receiver to locate them that these schemes are also vulnerable to computation-based DoS attacks [15]. In addition, there are some works [16], [19] that rely on RSUs or other vehicles to achieve the authentication for vehicular communications. However, these schemes must assume the RSUs or vehicles as cooperators are trusted (or at least semi-trusted) in VANETs. Moreover, the performance of authentication delay cannot be guaranteed for multiply transmissions, especially when the packet loss rate is high.

For resource-limited environments, researchers have explored lightweight broadcast authentication schemes,

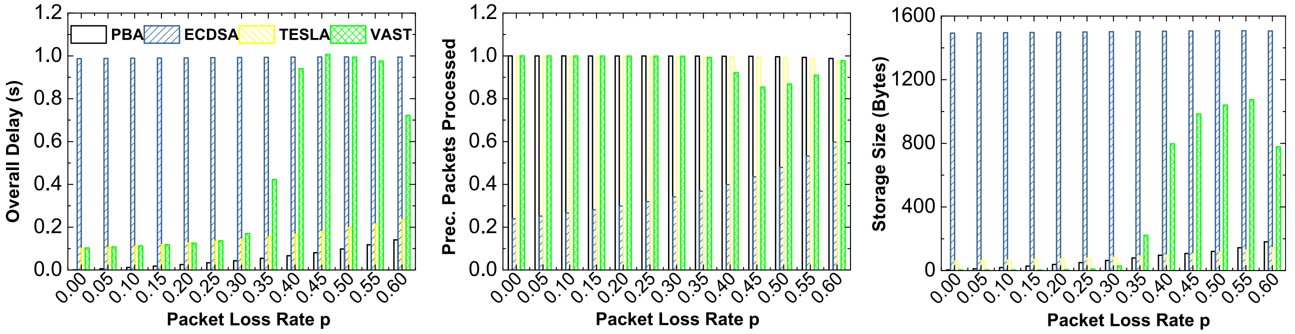


Fig. 11. Performance comparison results with different wireless losses: overall delay (left), packet processing rate (middle), and storage cost (right).

such as TESLA-based authentication schemes [22], [23], [24], [26], [31]. Stude et al. [31] propose VAST to provide a wide range of possible authentication properties. Unfortunately, similar to the basic TESLA, VAST does not enable instant authentication. In safety-related applications, delayed verification is not favorable when the receiver wants to instantly verify the time-sensitive messages. Hsiao et al. [7] propose a one-time signature scheme named FastAuth to provide lightweight, timely and non-repudiation authentication for vehicle-to-vehicle communications. In FastAuth, they use chained Huffman hash trees to generate a common public key and minimize the signature size for beacons sent during one prediction interval. As far as we know, FastAuth first exploits the predictability of future beacons to achieve the instant authentication in VANETs. However, there is one drawback in FastAuth: once the receiver misses a beacon, it cannot work in the rest of the current prediction interval. To deal with packet losses, they add the schemes of Reed-Solomon (RS) Coding [40] and Public Key Rebinding. However, more communication overhead is required in wireless lossy environments, as well as the computational overhead. Our PBA scheme is motivated by FastAuth, but it belongs to TESLA-based authentication schemes. With TESLA signatures piggyback, our PBA could resist packet losses naturally.

8 Conclusion

For V2V communications, we propose an effective, efficient and scalable broadcast authentication scheme to provide both computation-based DoS attacks resilient and packet losses resilient in VANETs. Moreover, PBA has the advantage of fast verification by leveraging the predictability of beacons for single-hop relevant applications. To defend against memory-based DoS attacks, PBA only keeps shortened MACs of signatures to reduce the storage overhead.

By theoretical analysis, we show PBA is secure and robust in the context of VANETs. Through a range of evaluations, PBA has been demonstrated to perform well even under high-density traffic scenarios and lossy wireless scenarios. In the future, we will try to study how our scheme could be improved given accurate prediction models. For some vehicular applications, it is also important to consider the privacy issues. We will address how to satisfy both security and privacy requirements in the future work.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61472250), National Key Technology R&D Program (No. 2012BAH46B02), Doctoral Fund of Ministry of Education of China (No. 20120073110094), Innovation Programs by Shanghai Municipal Science and Technology Commission (No. 13511504000, No. 14511100300), China Scholarship Council (No. 201306230026). Finally, the authors would like to thank the anonymous reviewers for their helpful comments. Dawu Gu is the corresponding author of the article.

References

[1] ASTM E2213-03-Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems-5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Sep. 2003.

[2] F. Bai, H. Krishnan, V. Sadekar, G. Holland, and T. Elbatt, "Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective," in *Proc. IEEE Workshop Automotive Netw. Appl.*, pp. 1–25, 2006.

[3] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proc. Fourth Workshop Hot Topics Netw.*, Nov. 2005.

[4] S. B. Lee, G. Pan, J. S. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. ACM Mobihoc*, pp. 150–159, 2007.

[5] M. Raya and J. P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, 2007.

[6] IEEE Std 1609.2-2013 - IEEE standard for wireless access in vehicular environments—Security services for applications and management messages, Apr. 2013.

[7] H. C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Flooding-resilient broadcast authentication for vanets," in *Proc. ACM Mobicom*, pp. 193–204, Sep. 2011.

[8] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, pp. 816–824, 2008.

[9] 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 Trans. Vehicular Technol.*, vol. 60, no. 1, pp. 248–262, Jan. 2011.

[10] K. Shim, "Reconstruction of a secure authentication scheme for vehicular ad hoc networks using a binary authentication tree," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5586–5393, Nov. 2013.

[11] M. Bellare, J. A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures, " in *Proc. EUROCRYPT*, pp. 236–250, 1998.

[12] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. EUROCRYPT*, pp. 416–432, 2003.

[13] D. Hankerson, J. L. Hernandez, and A. Menezes, "Software implementation of elliptic curve cryptography over binary fields, " in *Proc. Cryptographic Hardware Embedded Syst.*, pp. 1–24, 2000.

[14] T. Unterluggauer and E. Wenger, "Efficient pairings and ecc for embedded systems," in *Proc. Cryptographic Hardware Embedded Syst.*, pp. 298–315, 2014.

[15] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1974–1983, Apr. 2009.

[16] X. Lin and X. Li, "Achieving efficient cooperative message authentication in vehicular ad hoc networks," *IEEE Trans. Vehicular Technol.*, vol. 62, no. 7, pp. 3339–3348, Sep. 2013.

[17] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, Sep. 2010.

[18] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in vanets," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 61–629, Mar. 2011.

[19] C. Zhang, X. Lun, R. Lu, P. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Trans. Vehicular Technol.*, vol. 57, no. 6, pp. 3357–3368, Nov. 2008.

[20] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM*, pp. 1903–1911, 2008.

[21] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications from a perspective of vehicular network engineers," in *Proc. ACM Mobicom*, pp. 329–340, 2010.

[22] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA CryptoBytes*, 2002.

[23] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. Symp. Netw. Distributed Syst. Secur.*, pp. 35–46, 2001.

[24] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Secur. Privacy*, pp. 56–73, 2000.

[25] R. C. Merkle, "Secrecy, authentication, and public key systems," *PhD Dissertation*, Stanford Univ., Stanford, CA, USA, 1979.

[26] C. Lyu, D. Gu, X. Zhang, S. Sun, and Y. Tang, "Efficient, fast and scalable authentication for vanets," in *Proc. WCNC*, pp. 1768–1773, 2013.

[27] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *Proc. VANET*, pp. 19–28, 2007.

[28] J. T. Chiang and Y. C. Hu, "Cross-layer jamming detection and mitigation in wireless broadcast networks," in *Proc. ACM Mobicom*, pp. 346–349, 2007.

[29] W. Shen, P. Ning, X. He, and H. Dai, "Ally friendly jamming: How to jam your enemy and maintain your own wireless connectivity at the same time," in *Proc. IEEE Symp. Secur. Privacy*, pp. 174–188, 2013.

[30] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo, "Real-time detection of denial-of-service attacks in ieee 802.11p vehicular networks," *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 110–113, Jan. 2014.

[31] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient vanet authentication," *J. Commun. Netw.*, vol. 11, no. 6, pp. 574–588, Dec. 2009.

[32] A. Studer, E. Shi, F. Bai, and A. Perrig, "Tacking together efficient authentication, revocation, and privacy in vanets," in *Proc. SECON*, pp. 1–9, 2009.

[33] J. J. Haas, Y. Hu, and K. P. Laverteaux, "Design and analysis of a lightweight certificate revocation mechanism for vanet," in *Proc. VANET*, pp. 89–8, 2009.

[34] A. Wasef and X. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 78–89, Jan. 2013.

[35] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing privacy using symmetric random key-set in vehicular networks," in *Proc. Int. Symp. Autonomous Decentralized Syst.*, pp. 344–351, 2007.

[36] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[37] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. CRYPTO*, pp. 41–55, 2004.

[38] D. Chaum and E. Van Heyst, "Group signatures," in *Proc. EUROCRYPT*, pp. 257–265, 1991.

[39] J. Baek and Y. Zheng, "Identity-based threshold signature scheme from the bilinear pairings," in *Proc. ITCC*, pp. 124–128, 2004.

[40] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

Chen Lyu received the BS and MS degrees in Telecommunications Engineering from Xidian University of China, X'ian, China, in 2007 and 2010, respectively. She is currently working toward the PhD degree in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, Shanghai, China. She was a visiting student at the University of California, Davis, CA, from 2013 to 2014. Her research interests include wireless security and privacy.

Dawu Gu received the BS degree in applied mathematics in 1992, and the MS and PhD degrees in cryptography in 1998, both from Xidian University of China, X'ian, China. He is currently a professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University (SJTU), Shanghai, China. His current research interests include cryptography, side channel attack, and software security. He leads the Laboratory of Cryptology and Computer Security (LoCCS) at SJTU. He has published over 150 scientific papers in academic journals and conferences, and has owned 15 innovation patents. He was the winner of Yangtze River Scholar Distinguished Professors Program in 2014 and New Century Excellent Talent Program in 2005, both made by the Ministry of Education of China. He serves as board members of China Association of Cryptologic Research and Shanghai Computer Society. He also serves as several technical editors for *China Communications*, *J. Cryptologic Research*, and *Information Network Security*, etc. He has been invited as Chairs and TPC members for many conferences and workshops.

Yunze Zeng received the BS degree in electrical engineering from Beijing Jiaotong University, Beijing, China, in 2012. He is currently working toward the PhD degree in the Department of Computer Science at the University of California, Davis, CA. He was an exchange student at the University of California, San Diego, CA, in 2011. He is the recipient of the Best Paper Awards from ACM BodyNets 2013 and IFIP Networking 2014. His current research interests include wireless networking and mobile computing.

Prasant Mohapatra received the doctoral degree from Pennsylvania State University, Philadelphia, PA, in 1993. He is currently a professor in the Department of Computer Science at the University of California (UC), Davis, CA. He is currently serving as the associate chancellor of UC Davis. He was the Department Chair of Computer Science from 2007 to 2013, and held the Tim Bucher Family Endowed Chair Professorship during that period. In the past, he has been on the faculty at Iowa State University and Michigan State University. He is the Editor-in-Chief of the *IEEE Transactions on Mobile Computing*. He has served on the editorial boards of the *IEEE Transactions on Computers*, *IEEE Transactions on Mobile Computing*, *IEEE Transaction on Parallel and Distributed Systems*, *ACM WINET*, and *Ad Hoc Networks*. He has served as the Program Chair and the General Chair and has been on the program/organizational committees of several international conferences. He received the Outstanding Engineering Alumni Award in 2008. His research interests include wireless networks, mobile communications, cybersecurity, and Internet protocols.