Secure and Efficient Handover Authentication Based on Bilinear Pairing Functions

Daojing He, *Student Member, IEEE*, Chun Chen, *Member, IEEE*, Sammy Chan, *Member, IEEE*, and Jiajun Bu, *Member, IEEE*

***Abstract*—Seamless handover over multiple access points is highly desirable to mobile nodes, but ensuring security and efficiency of this process is challenging. This paper shows that prior handover authentication schemes incur high communication and computation costs, and are subject to a few security attacks. Further, a novel handover authentication protocol named *PairHand* is proposed. PairHand uses pairing-based cryptography to secure handover process and to achieve high efficiency. Also, an efficient batch signature verification scheme is incorporated into PairHand. Experiments using our implementation on laptop PCs show that PairHand is feasible in real applications.**

***Index Terms*—Wireless networks, security, privacy, efficiency, handover authentication.**

I. Introduction

NOWADAYS, wireless internet access services are offered through interconnected mobile telecommunication networks, WLANs, vehicular ad hoc networks (VANETs). To overcome the geographical coverage limit of each access point and provide seamless access service for mobile nodes (e.g., PDA, laptop PC, smart phone and vehicle), it is important to have an efficient handover protocol. One important module in the handover protocol is authentication. Regardless of the technology implemented, as shown in Fig. 1, a typical handover authentication scenario involves three entities: mobile nodes (MNs), access points (APs) and the authentication server (AS). Before entering the network, an MN registers to AS, then subscribes services and connects to an AP for accessing the network. When the MN moves from the current AP (e.g., $AP1$) into a new AP (e.g., $AP2$), handover authentication should be performed at $AP2$. Through handover authentication, AP2 authenticates the MN to identify and reject any access request by an unauthorized user. At the same time, a session key should be established between the MN and $AP2$ to provide confidentiality and integrity of the communication session. We further illustrate the above procedure by considering a VANET scenario. A typical VANET consists of a trusted authority (TA) (i.e., AS in this paper), a large number of vehicles equipped with wireless On-Board Units (OBUs) (i.e., MNs in this paper), and some stationary Roadside Units (RSUs) (i.e., APs in this paper). TA deploys RSUs and registers vehicles by granting the corresponding authentication keys. Each RSU receives and then verifies the traffic safety messages from the OBUs.

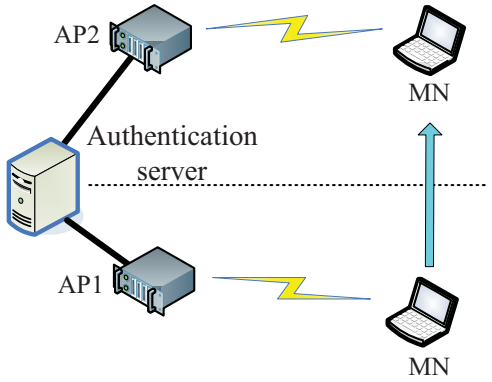

Fig. 1. Handover authentication overview.

Designing a handover authentication protocol is not an easy task. Generally, there are two major practical issues challenging the design. First, efficiency needs to be considered. An MN is generally constrained in terms of power and processing capability. Therefore, a handover authentication process should be computationally efficient. Further, such a process should be fast enough to maintain persistent connectivity for MNs. For example, to reduce the impact of bursty packet loss caused by handover, the IEEE is discussing a 50-ms limit on handover time, of which the authentication module should ideally take no more than 20 ms. However, most of the existing handover authentication protocols [1]–[12] incur high communication and computation costs in five aspects. (1) The conventional way of performing handover authentication [1]–[6] is to let $AP2$ contact AS who acts as a guarantor for vouching that an MN is its legitimate subscriber. This will incur more computation and communication delay, especially AS is often located in a remote location. (2) For mutual authentication and key establishment, all protocols without communicating with AS [7]–[12] require at least three handshakes between the MN and $AP2$ while other protocols [1]–[6] require at least four handshakes among the three entities. Data transmission is a costly operation in wireless networks: sending 1-bit over a wireless medium requires over 1000 times more energy than a single 32-bit computation [13]. (3) To provide robust security, employing a digital signature scheme is widely recognized as the most effective approach for handover authentication [6]–

Manuscript received June 29, 2011; revised August 21, 2011; accepted October 13, 2011. The associate editor coordinating the review of this paper and approving it for publication was V. K. N. Lau.

This work was supported by National Science Foundation of China (Grant No. 61070155), Program for New Century Excellent Talents in University (NCET-09-0685), and a grant from the Research Grants Council of the Hong Kong SAR, China [Project No. City U 111208].

D. He, C. Chen, and J. Bu are with the College of Computer Science, Zhejiang University, P.R. China (e-mail: hedaojinghit@gmail.com).

S. Chan is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eeschan@cityu.edu.hk).

Digital Object Identifier 10.1109/TWC.2011.110811.111240

[12]. Unfortunately, it is not efficient in communication, because the certificate has to be transmitted along with the digital signature as the message propagates in the network. This leads to more energy consumption on MNs. Moreover, to authenticate each digital signature, the corresponding receiver always takes two expensive signature verification operations. This is because the certificate needs to be authenticated as well. (4) To provide user anonymity, group signature-based protocols have been proposed in [11], [12]. However, the user revocation list needs to be distributed across the entire network in a timely manner. Further, the verification delay incurred in these protocols for each access request is linearly proportional to the number of revoked users. Therefore, the performance of these protocols may deteriorate when the number of revoked users is large. (5) Generally, an AP verifies each signature individually. When the arrival rate of signatures is high, a scalability problem emerges immediately, where the AP has much less time to verify each received signature. For example, in VANETs, each RSU could possibly communicate with hundreds of OBUs, each sending a safety related message to the RSU every 100-300 ms [14].

Second, security and privacy are serious concerns for the handover authentication service. However, all existing handover authentication protocols [1]–[12] are subject to a few security attacks in two aspects. On the one hand, users are deeply concerned about their privacy-related information such as the identity, position, and roaming route. Unfortunately, in most of the current handover authentication schemes (e.g., [1]–[5], [7]–[10]), it is commonly assumed that the APs are trustworthy and would keep users' privacy-related information confidential. However, since such information is extremely sensitive and coveted by many companies, which may use it to improve their business, such an assumption may not be valid. Therefore, a user should be protected from the prying eyes of APs. On the other hand, by Denial-of-Service (DoS) attacks, adversaries can exhaust the resources of AP and AS and render them less capable of serving legitimate MNs. Such attacks can be classified into two categories. (1) The conventional way of performing handover authentication [1]–[6] requires an AP to unconditionally forward any access request, valid or invalid, to AS, an adversary can easily launch DoS attacks on AS through an AP. (2) To avoid such a DoS attack, some recent handover authentication techniques (e.g., [7]–[12]) only require an MN and an AP to be involved in each protocol run. However, the AP needs to perform expensive cryptographic operations (e.g., pairing computation in [11], [12]) to check the validity of the sender. This checking is exploited by the adversary to make another type of DoS attack. That is, it can inject bogus access requests into the networks, force the APs that receive such messages to perform expensive verifications, and eventually exhaust their resources. Despite the necessity and importance, no research has been conducted to address this attack in handover authentication.

According to the above analysis, all existing handover authentication protocols fail to provide appropriate security and efficiency guarantees. In this case, users are reluctant to accept such mobile service. Thus, it is utterly important to provide an efficient handover authentication protocol for practical wireless networks. In this paper, we propose a novel handover authentication protocol called *PairHand*, which uses pairing-based cryptography to secure handover process and to reduce the communication and computation overheads of the involved entities. Also, it only requires two handshakes between an MN and an AP, and does not need to transmit or verify any certificate as in traditional public key cryptosystems. Further, we introduce an efficient batch signature verification scheme, in which each AP can simultaneously verify multiple received signatures.

The remainder of this paper is organized as follows. Section II discusses the security requirements and bilinear maps. Section III presents our protocol and Section IV analyzes and evaluates the protocol. Section V concludes the paper.

II. Security Requirements and Preliminaries

A. Security Requirements

As reported in [9]–[12], a strong handover authentication should satisfy the following seven security properties: (1) Subscription validation: An AP must authenticate MNs to ensure their legitimacy. (2) Server authentication: MNs should be allowed to authenticate the AP they visit to avoid potential deception and other malicious attacks. (3) Key establishment: A session key should be established between an MN and an AP to protect subsequent exchanged data between them. (4) User anonymity and untraceability: Except to AS, the user is anonymous and his activities are unlinkable to anyone including the visited AP. (5) Conditional privacy preservation: In some application scenarios, it is the liability for AS to reveal the related private information (e.g., identity, position) of an MN to law enforcement in case of emergency (e.g., enhanced 911 location service mandated by U.S. Federal Communications Commission). (6) Provision of user revocation: Service to an MN should be terminated once his subscription period ends. (7) Attack-resistance: The protocol should have the ability to resist the attacks in wireless networks (i.e., replay and DoS attacks) such that it can be applied in the real world.

B. Bilinear Maps

Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order q. Let P be an arbitrary generator of $\mathbb{G}$. Note that aP denotes P added to itself a times. Let $\hat{e} : \mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ be a bilinear map, which satisfies the following properties: (1) Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, where $P, Q\in\mathbb{G}$ and $a, b\in\mathbb{Z}_q^*$. Here $\mathbb{Z}_q^* = \{\rho \,|\, 1\leq\rho\leq q-1\}$. (2) Non-degenerate: $\hat{e}(P, P)\neq 1$. (3) Computable: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q\in\mathbb{G}$. The group that possesses such a map $\hat{e}$ is called a bilinear group, on which the Decisional Diffie-Hellman problem is easy to solve while the Computational Diffie-Hellman problem is believed hard [15]. For example, given $P, aP, bP, cP\in\mathbb{G}$ and any $a, b, c\in\mathbb{Z}_q^*$, there exists an efficient algorithm to determine whether $ab = c \mod q$ by checking $\hat{e}(aP, bP) \stackrel{?}{=} \hat{e}(P, cP)$, while there exists no algorithm that can compute $abP\in\mathbb{G}$ with non-negligible probability within polynomial time.

III. PairHand: The Protocol

When designing PairHand, we find that none of the existing privacy-aware cryptographic primitives, such as blind

signature, ring signature, and group signature techniques, suits our purpose given the security and efficiency requirements discussed above. Blind signature and ring signature can only provide unconditional privacy, while PairHand demands conditional privacy, and hence, revocable anonymity. Existing group signature schemes do provide revocable anonymity, but cannot meet high efficiency as described in Section I. Here we adopt the privacy preserving technique based on pseudonyms. Since MNs generally have large storage capacity, rendering the preloading of a large pool of pseudonyms from AS feasible. A recent work [16] quantitatively studied the storage space requirement for preloading anonymous keys (i.e., pseudonyms) and associated certificates for long term use (i.e., one year). Their results are obtained based on quantifying the upper and lower bounds on the pseudonym change interval for maintaining a satisfactory degree of privacy. Since the preloading method in our handover authentication protocol involves a pool of shorter-lived pseudonyms, the memory consumption is bounded by the results given by [16]. The preload-and-replenish mechanism has been proposed by many researchers and works efficiently. For example, it can be realized through some existing wireless infrastructure, such as Wi-Fi networks.

A. System Initialization

In system initialization phase, AS first initializes the whole system by running the following steps. Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order q. Let P be an arbitrary generator of $\mathbb{G}$. Let $\hat{e} : \mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$ be a bilinear map. Then, AS chooses a random number $s\in\mathbb{Z}_q^*$ as the *master key* and computes the corresponding public key $P_{pub} = sP$. Also, AS chooses two secure hash functions H_1 and H_2, where $H_1 : \{0,1\}^*\rightarrow\mathbb{G}$ and $H_2 : \{0,1\}^*\rightarrow\mathbb{Z}_q^*$.

In the end, AS publishes the public system parameters $params$ as $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub}, H_1, H_2\}$ and keeps the *master key* secretly. For each AP, AS computes $H_1(ID_{AP})$ as the public key, $sH_1(ID_{AP})$ as the private key, and sends them to the AP using a secure transmission protocol (e.g., wired transport layer security protocol), where ID_{AP} is the identity of the AP.

When an MN, say i, registers to AS with his real identity ID_i, AS first checks his validity. If MN i is valid, AS chooses a family of unlinkable pseudo-IDs $PID = \{pid_1, pid_2, \ldots\}$. For each pseudo-ID $pid_j\in PID$, AS computes the public key $H_1(pid_j)$ and the corresponding private key $sH_1(pid_j)$, and then securely sends all tuples $(pid_j, sH_1(pid_j))$ back to MN i. By this, MN i can constantly change its pseudo-ID to achieve identity privacy and location privacy in handover authentication process.

In the above procedure, AS adds $ExpiryDate$ into ID_{AP} and pid_j so that the public keys are only valid before the specified expiry date. After that date, the corresponding private key is revoked automatically if a new private key for a new expiry date is not provided by AS. This type of key can be delegated to an AP (rsp. mobile device) so that it is only usable before the expiry date even if the AP (rsp. the device) is compromised (rsp. lost or stolen). Moreover, there is no need to store public and private keys database in AS. Thus, PairHand can support user revocation.

B. Handover Authentication

Each AP broadcasts its identity as part of *beacon messages* that are periodically broadcasted to declare service existence.

To access the network, as illustrated in Fig. 2, an MN, say i, follows the handover authentication protocol as specified below, when an AP ($AP2$) is within his direct communication range.

1) MN i picks an unused pseudo-ID pid_i and the corresponding private key $sH_1(pid_i)$.

2) With the private key $sH_1(pid_i)$ and message $\mathcal{M}_i = (pid_i\|ID_{AP2}\|ts)$, MN i can compute the signature $\sigma_i = H_2(\mathcal{M}_i)\cdot sH_1(pid_i)$, where a timestamp ts is added by MN i to counter replay attacks and $\|$ indicates message concatenation operation. In this case, we assume that all network entities keep loose time synchronization via some existing time synchronization mechanisms such as GPS-system. Alternatively, instead of timestamp, a random number can be used to prevent replay attacks.

3) Subsequently, MN i unicasts the access request message $\{\mathcal{M}_i, \sigma_i\}$ to $AP2$.

4) Then, MN i computes the shared symmetric key with $AP2$: $K_{i-2} = \hat{e}(sH_1(pid_i), H_1(ID_{AP2}))$.

Upon receipt of $\{\mathcal{M}_i, \sigma_i\}$, $AP2$ proceeds as follows.

1) Check the time stamp ts to prevent replay attack. Examine $ExpiryDate$ included in pid_i to verify the service expiration time.

2) With $params$ assigned by AS, $AP2$ checks whether signature σ_i is valid if $\hat{e}(\sigma_i, P) = \hat{e}(H_2(\mathcal{M}_i)\cdot H_1(pid_i), P_{pub})$, as verified below.

$$\hat{e}(\sigma_i, P) = \hat{e}(H_2(\mathcal{M}_i)\cdot sH_1(pid_i), P)$$
$$= \hat{e}(H_2(\mathcal{M}_i)\cdot H_1(pid_i), sP) = \hat{e}(H_2(\mathcal{M}_i)\cdot H_1(pid_i), P_{pub})$$

3) $AP2$ further computes

$$K_{2-i} = \hat{e}(H_1(pid_i), sH_1(ID_{AP2})).$$

Note that K_{i-2} is equal to K_{2-i} since

$$K_{i-2} = \hat{e}(sH_1(pid_i), H_1(ID_{AP2})) = \hat{e}(H_1(pid_i), H_1(ID_{AP2}))^s$$
$$= \hat{e}(H_1(pid_i), sH_1(ID_{AP2})) = K_{2-i}$$

4) $AP2$ generates an authentication code $Aut = H_2(K_{2-i}\|pid_i\|ID_{AP2})$. Finally, $AP2$ sends $\{pid_i, ID_{AP2}\}$ and Aut to MN i.

Upon receiving $\{pid_i, ID_{AP2}, Aut\}$, MN i generates a verification code $Ver = H_2(K_{i-2}\|pid_i\|ID_{AP2})$ and compares it with Aut. If Ver matches Aut, then MN i believes $AP2$ is legitimate and has established the shared key K_{2-i}; otherwise, MN i rejects the connection.

The above protocol enables explicit mutual authentication between an AP and a legitimate MN; it also enables unilateral anonymous authentication for the MN. Upon successful completion of the protocol, the AP and the MN also establish a shared symmetric key used for the subsequent communication session. And this session is uniquely identified through (pid_i, ID_{AP2}).

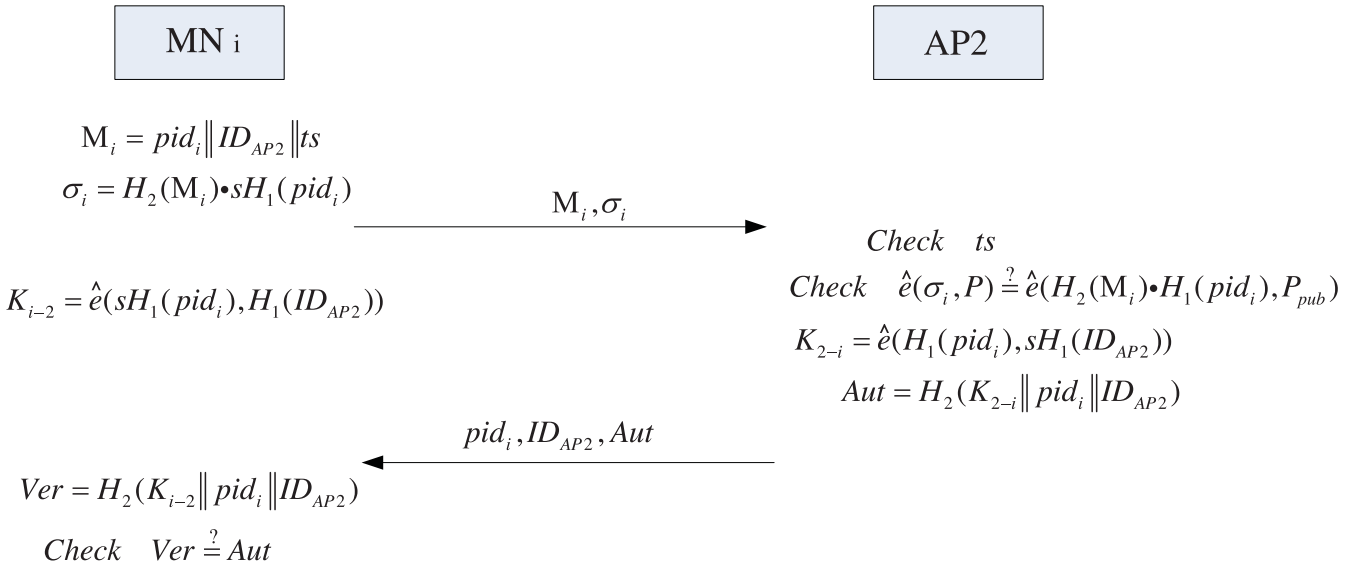

Fig. 2. The protocol run of PairHand.

Therefore, the computation cost by the AP for verifying a single signature is dominantly comprised of 1 point multiplication and 2 pairing operations. Note that the computation cost of a pairing operation is higher than the cost of a point multiplication operation.

5) Finally, AP2 securely transmits $\{\mathcal{M}_i, \sigma_i\}$ to AS. Upon receiving this message, AS can find the real identity of MN i according to the pseudo-ID included in $\mathcal{M}_i$. Thus, PairHand can provide conditional privacy. Since APs only notify AS of the authentication result after performing the handover authentication, this step does not affect the authentication time, and incurs much less overhead than other existing schemes [1]–[6], [11], [12].

C. Batch Authentication

Based on the network architecture as described above, once an AP receives an access request from an MN, the AP has to verify the signature of the message to ensure the validation of the corresponding MN.

Given n distinct access request messages denoted as $< \mathcal{M}_1, \sigma_1 >, < \mathcal{M}_2, \sigma_2 >, \ldots, < \mathcal{M}_n, \sigma_n >$, respectively, which are sent by n distinct MNs denoted as $MN_1, MN_2, \ldots, MN_n$, all signatures, denoted as $\sigma_1, \sigma_2, \ldots, \sigma_n$, are valid if $\hat{e}(\sum_{i=1}^{n} \sigma_i, P) = \hat{e}(\sum_{i=1}^{n} H_2(\mathcal{M}_i) \cdot H_1(pid_i), P_{pub})$. This batch verification equation follows since

$$\hat{e}(\sum_{i=1}^{n} \sigma_i, P) = \hat{e}(\sum_{i=1}^{n} H_2(\mathcal{M}_i) \cdot sH_1(pid_i), P)$$

$$= \hat{e}(\sum_{i=1}^{n} H_2(\mathcal{M}_i) \cdot H_1(pid_i), sP)$$

$$= \hat{e}(\sum_{i=1}^{n} H_2(\mathcal{M}_i) \cdot H_1(pid_i), P_{pub})$$

From the above batch verification equation, the computation cost that the AP spends on verifying n signatures is dominantly comprised of n point multiplication and 2 pairing operations. Therefore, the time for an AP to verify a large number of signatures sent by multiple users can be dramatically reduced, which can apparently reduce the connection loss ratio due to the potential bottleneck of signature verification at the AP.

D. DoS Attack Resistance

To prevent the DoS attack described in Section I, we propose a polynomial-based lightweight verification scheme which is inspired by [17]. We require that in the system initialization phase, AS randomly generates a bivariate t-degree polynomial $f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$ over a finite field F_p, where p is a large prime number, such that it has the property of $f(x, y) = f(y, x)$. When an MN, say i, registers to AS, for each pseudo-ID pid_j, AS computes a polynomial share of $f(x, y)$, that is, $f(pid_j, y)$, and then securely delivers them to MN i. Also, AS securely transmits $f(ID_{AP}, y)$ to each AP, where ID_{AP} is the identity of the AP. When MN i wants to access the network via $AP2$, it computes the common key $f(pid_j, ID_{AP2})$ by evaluating $f(pid_j, y)$ at point ID_{AP2}, and $AP2$ can compute the same key $f(pid_j, ID_{AP2}) = f(ID_{AP2}, pid_j)$ by evaluating $f(ID_{AP2}, y)$ at point pid_j. Then $AP2$ can use key $f(ID_{AP2}, pid_j)$ to verify the access request of MN i. As will be shown in Section IV.B, evaluation of the polynomial is very fast, and hence $AP2$ can efficiently verify the access request before performing expensive verification to mitigate the DoS attack. Here lightweight verification can effectively mitigate DoS attacks since an authorized user has a clear advantage over the adversary due to its prior knowledge of the communication key with each AP. On the other hand, the adversary has to guess the communication key first, before generating a valid access request. Thus, our method would not bring much burden to the low-power mobile devices. At the

TABLE I
FUNCTIONALITY AND PERFORMANCE COMPARISON BETWEEN PAIRHAND AND RELATED WORK

ME: modular exponentiation, RV : RSA verification

Protocols	Number of Parties	User Anonymity and Untraceability	Conditional Privacy Preservation	DoS Attack Resistance	Commun. Overhead	User Cryptographic Operations	E_{MN}
HMZCB [6]	3	Yes	Yes	No	2β+2δ	-	-
Method of [10]	2	No	No	No	3δ	4ME+1RV	-
YHWD [11]	2	Yes	No	No	3δ	6.25ECSM	124mJ
Priauth [12]	2	Yes	No	No	3δ	15.75ECSM+4Pairing	563mJ
PairHand	2	Yes	Yes	Yes	2δ	1ECSM+1Pairing	82mJ

TABLE II
TIMINGS FOR ECSM AND PAIRING OPERATIONS

	800MHz Processor		1.2GHz Processor		1.6GHz Processor		2GHz Processor	
	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing
Time(ms)	1.83	5.734	1.547	3.841	0.916	2.872	0.672	2.134

same time, it can resist DoS attacks from a powerful adversary. The proposed scheme is unconditionally secure and t-collusion resistant, which means that only when $t+1$ network identities are compromised, the secret polynomial f is disclosed. Based on this scheme, PairHand is modified as follows. If $AP2$ is under attack (e.g., $AP2$ finds the rate of received bogus access requests is more than the pre-defined threshold), it adds "Yes" and the identity ID_{AP2} into the *beacon messages*. An MN, say i, picks an unused pseudo-ID pid_i and then generates an authentication code $Aut2 = H_2(K\|\mathcal{M}_i\|\sigma_i)$, where $K = f(pid_i, ID_{AP2})$. Finally, MN i sends $Aut2$ and $(\mathcal{M}_i\|\sigma_i)$ to $AP2$. Upon receiving such a message, $AP2$ generates a verification code $Ver2 = H_2(K^*\|\mathcal{M}_i\|\sigma_i)$ and compares it with $Aut2$, where $K^* = f(ID_{AP2}, pid_i)$. Only if this verification is successful, $AP2$ performs expensive verification on the access request. Our implementations show that an authentication code with a 128-bit number as input just takes 1.1 μs on 1.2 GHz laptop PC.

IV. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

A. Security Analysis

We analyze the security of PairHand to verify whether the requirements mentioned in Section II have been satisfied. Note that requirements (3), (5), (6) and (7) have been analyzed in Section III.

Subscription validation: The signature $\sigma_i = H_2(\mathcal{M}_i)\cdot sH_1(pid_i)$ is actually a pseudo-ID-based signature. Without knowing the private key $sH_1(pid_i)$, it is infeasible to forge a valid signature. Because of the NP-hard computation complexity of Diffie-Hellman problem in $\mathbb{G}$, it is difficult to derive the private key $sH_1(pid_i)$ by way of pid_i, P and P_{pub}. Therefore, the pseudo-ID-based signature is unforgeable, and the property of subscription validation is achieved.

Server authentication: Similar to the proof of subscription validation, an adversary who does not know $AP2$'s private key $sH_1(ID_{AP2})$ cannot make legitimate authentication code Aut.

User anonymity and untraceability: In PairHand, each MN receives a family of pseudo-IDs and the corresponding private keys from AS at the time of registration. These pseudo-IDs, instead of the MN's real identity, are used in handover authentication phase for the purpose of privacy protection. Thus, only AS knows the relationship between a pseudo-ID and the real identity. Since there is no linkage between pseudo-IDs, except the MN and AS, anyone including the APs is unable to identify the MN or link two transactions initiated by the same MN.

B. Functionality Comparison and Performance Evaluation

Table I shows the functionality and performance comparison of PairHand and related work ([6], [10]–[12]). Note that the complexity of highly efficient operations such as hash function is omitted. The Elliptic Curve Scalar Multiplication (ECSM) operation of MIRACL [18] library and the most efficient Pairing operation of PBC library have been introduced into the implementation of PairHand. Our implementation results in Laptop PCs with different computational power (a single CPU) are summarized in Table II. Here we use MNT curve with order of 160 bits and embedding degree $k = 6$. Under this curve, elements in $\mathbb{Z}_q^*, \mathbb{G}$ and $\mathbb{G}_T$ are represented by 160, 161 and 960 bits. Considering the transmission overhead, the length of each access request $\{\mathcal{M}_i, \sigma_i\}$ is 30 bytes while the length of the response $\{pid_i, ID_{AP2}, Aut\}$ is 28 bytes, where the lengths of pid_i, ts and ID_{AP2} are 4, 2, 4 bytes, respectively. Here we assume that an MN runs on a 800 MHz processor while an AP runs on a 1.6 GHz processor, a successful handover authentication just takes 11.36 ms. Currently, the clock frequency of most Laptop PCs, PDAs and smart phones is greater than 800 MHz. Therefore, PairHand is efficient when employed on most mobile devices. Table I also shows the energy consumption at the MN, where it is assumed that an MN runs on a 800 MHz processor. It can be calculated as $E_{MN} = T_{MN}\times C$, where E_{MN} is the energy consumption, T_{MN} is the total computation time for handover authentication, and C is the CPU maximum power (10.88W). For communication overhead, we assume that the expected authentication message delivery cost between an AP and AS is β unit and that between an MN and an AP is δ unit, respectively. As shown in Table I, PairHand achieves all security requirements and is more efficient than the well-known protocols ([6], [10]–[12]).

Table III gives the execution time of evaluating a t-degree polynomial in Laptop PCs (a single CPU) when t varies. This

TABLE III
TIMINGS FOR EVALUATING A t-DEGREE POLYNOMIAL

	800MHz Processor				1.6GHz Processor				2.4GHz Processor			
	t=100	t=500	t=1000	t=1500	t=100	t=500	t=1000	t=1500	t=100	t=500	t=1000	t=1500
Time(ms)	0.333	1.645	3.265	4.397	0.148	0.799	1.623	2.359	0.102	0.517	1.075	1.661

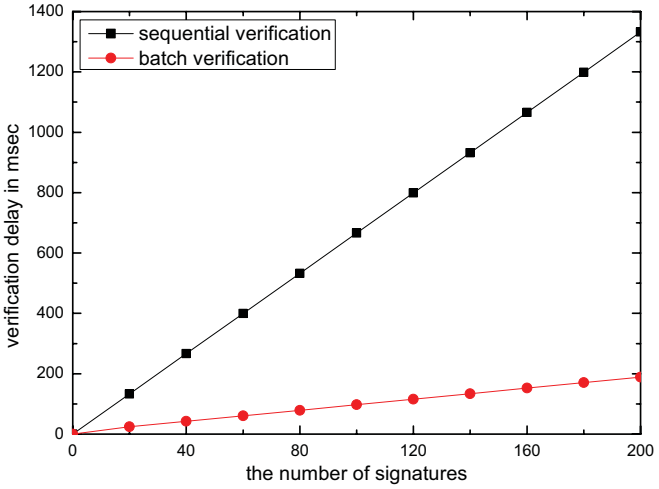

Fig. 3. Verification delay comparison.

requires $2t$ modular multiplications and t modular additions in a finite field F_p. In our implementation, p is set to 64 bit long for typical cryptosystems such as RC5. For example, the execution time on a 800 MHz laptop PC is 1.645 ms when $t = 500$. Thus, evaluation of the polynomial is very fast.

The verification delay of an AP against the number of the received message is plotted in Fig. 3. In this experiment, AP side programs have been implemented in C and executed in a 1.6 GHz laptop PC. The maximum number of signatures that can be verified simultaneously in 200 ms are 30 and 2177 messages for sequential verification and batch verification, respectively. In the context of the secure VANET application discussed in Section I, it means that with PairHand, an RSU can verify 2177 safety related messages every 200 ms.

V. CONCLUSION

In this paper, we have proposed a novel protocol to achieve secure and efficient handover authentication. The security analysis and experimental results show that the proposed approach is feasible for real applications.

REFERENCES

[1] European Telecommunications Standards Institute (ETSI), GSM 02.09: Security Aspects, 1993.
[2] 3rd Generation Partnership Project, 3GPP Specification: 3GPP TS 33.102, 3G Security, Security Architecture, Dec. 2002.
[3] C.-C. Chang, C.-Y. Lee, and Y.-C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Computer Commun.*, vol. 32, no. 4, pp. 611–618, 2009.
[4] Y.-P. Liao and S.-S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.
[5] H.-C. Hsiang and W.-K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.
[6] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Commun.*, vol. 34, no. 3, pp. 367–374, 2011.
[7] J. Choi and S. Jung, "A secure and efficient handover authentication based on light-weight Diffie-Hellman on mobile node in FMIPv6," *IEICE Trans. Commun.*, vol. E-91B, no. 2, pp. 605–608, 2008.
[8] Y. Kim, W. Ren, J. Jo, M. Yang, Y. Jiang, and J. Zheng, "SFRIC: a secure fast roaming scheme in wireless LAN using ID-based cryptography," in *Proc. ICC 2007*.
[9] J. Choi, S. Jung, Y. Kim, and M. Yoo, "A fast and efficient handover authentication achieving conditional privacy in V2I networks," LNCS 5764. Springer, pp. 291–300, 2009.
[10] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 54–56, 2010.
[11] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168–174, 2010.
[12] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 431–436, 2011.
[13] K. C. Barr and K. Asanovi, "Energy aware lossless data compression," *ACM Trans. Comput. Syst.*, vol. 24, no. 3, pp. 250–291, 2006.
[14] U.S. Department of Transportation, National Highway Traffic Safety Adminstration, Vehicle Safety Communications Project, Final Repot, 2006.
[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. Asiacrypt 2001*, vol. 2248, pp. 514–532.
[16] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.
[17] C. Blundo, *et al.*, "Perfectly-secure key distribution for dynamic conferences," in *Advances in Cryptology-Crypto 1992*, LNCS 740, PP. 471–486.
[18] M. Scott, Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL). Published by Shamus Software Ltd., http://www.shamus.ie/.