IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2019, accepted December 30, 2019, date of publication January 6, 2020, date of current version February 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964040

Security Analysis of an Identity-Based Signature From Factorization Problem

GANGLIN ZHANG, YONGJIAN LIAO, YU FAN, AND YIKUAN LIANG
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Yongjian Liao (liaoyj@uestc.edu.cn)

This work was supported in part by the Sichuan Science and Technology Program under Grant 2017GZDZX0002 and Grant 2017GZDZX0001.

ABSTRACT Many sensitive data are generated by resource-limitation devices in the Vehicular ad hoc network (VANET). When these data are divulged, people 's life and property will be threatened. To solve these problems, Wei *et al.* proposed a lightweight privacy-preserving protocol based on RSA assumption for VANET and they claimed that their protocol was secure and low overhead. In this paper, first of all, we show that the basic signature scheme to be used in Wei et al.'s protocol is not secure, i.e., the user's private key will be revealed from the pairs of message-signatures, which causes the protocol to be insecure. We also show that our security analysis is feasible and effective in practice from the theory and experiments. Then we construct a new identity-based signature scheme based RSA assumption and prove it is existentially unforgeable under the chosen message attack without random oracle. Finally, we update the Wei et al.'s protocol and do some experiments to evaluate the efficiency of our scheme in the updated protocol.

INDEX TERMS Common modulus attack, security analysis, VANETs privacy-preserving, IBS.

I. INTRODUCTION

The internet of things(IoT) is a future network that connects everything [2]. Unlike traditional internet made up of computers, various hardware devices, sensors and computers are connected to form the network in the IoT with the development of communication technology and wireless technology. This enables the application of intelligent computing in people's lives to be realized [3]–[6].

Although IoT brings a lot of convenience to people, many security problems also emerge. First of all, people's sensitive data such as location privacy, identity privacy, and personal preferences are exposed to IoT. Then, if hackers control some components in IoT, it will not only damage people's property, but even threaten people's lives [7]–[12].

Vehicular ad hoc network (VANET) is an important branch of IoT. It can improve vehicle and road safety, traffic efficiency, and convenience as well as comfort to both drivers and passengers. At the early stages, Road Side Unit(RSU) and On Board Unit(OBU) make up VANET. OBU represents the moving car. It can be the vehicle-mounted system, mobiles of driver etc. RSU is a roadside infrastructure. It can be traffic lights, streetlights, etc. One OBU node communicates with another distant OBU node by sending the message to nearby RSU nodes. RSU nodes can collect information about nearby road conditions, communicate with OBU nodes and communicate with other RSU nodes [13]. With the development and popularization of cloud computing, VANET uses the cloud to do complex computing. Because of the cloud, people can use VANET to plan driving paths, regulate traffic, monitor traffic accidents, etc. Obviously, the current VANET consists of RSU, OBU and cloud [14].

However, there are many attacks in the VANET. S.S. Tangade et al. divided these attacks into 17 categories: Passive Attacks, Insider Attacks, Insider Attacks, Outsider Attacks, Malicious Attacks, Rational attacks, Local attacks, Bogus Information, Alteration Attacks, Sybil Attack, Denial of Service (DoS), Identity Revealing, Black Hole, Message suppression, Timing Attack, Tunneling and Social attack [15].

To protect the security of communication information and hide OBU identity privacy under the limited computing resource, Wei *et al.* [1] proposed a lightweight privacy-preserving protocol based on the RSA assumption. Furthermore, the protocol uses an identity-based signature(IBS) scheme from RSA. IBS is a type of signature scheme which allows users to verify others signatures without exchanging private or public keys, without keeping key directories, and without using the services of a third party [16]. In IBS,

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

users' public key is generated from users' identity information like name, address, e-mail, etc. As far as we know, Guillou et al. proposed the first IBS scheme based on RSA [17]. Xing *et al.* extended Guillou et al.'s scheme and proposed an IBS scheme with message recovery from RSA [18]. Herranz *et al.* used Guillou et al.'s scheme to design an identity-based ring signatures from RSA [19].

A. CONTRIBUTION

We analyze the security of Wei et al.'s protocol and find a security defect in the IBS scheme [1] on which the protocol depends. Then, we analyze the impact of the security defect in the IBS on the protocol [1] and find that an RSU node can get the private keys of the OBU nodes with which it communicates through the common modulus attack. Moreover, we prove that the common modulus attack is feasible and effective in practice from the theory and experiments. Next, we propose an improved IBS scheme to replace Wei et al.'s [1] to resolve the security defect. We give the updated details of the protocol after applying the improved scheme and prove our improved IBS scheme is secure against chosen-message attack without random oracle. At last, we analyze the efficiency of our identity-based signature scheme in the updated protocol.

B. ORGANIZATION

The rest part of the paper is organized as follows. In section II, we introduce some necessary basic knowledge. In section III, we review the protocol proposed by Wei *et al.*, analyze its security and measure the difficulty of implementing the common modulus attack in this protocol by doing theoretical derivation and experiments. In section IV, we propose an improved IBS scheme to resolve the defect. Besides, we prove our IBS scheme is secure against chosen-message attack without random oracle. We apply our IBS scheme to the VANET protocol [1] in Section V. Finally, we conclude our work in Section VI.

II. PRELIMINARY

Here, we recall some basic knowledge that will be used.

A. EXTENDED EUCLIDEAN ALGORITHM

The Euclidean algorithm is an algorithm for solving the greatest common divisor of two given integers.

Let a and b be two integers. Without loss of generality, let $a \geq b$. For $a = bt + r$, where the integers t and $b > r \geq 0$, we have $gcd(a, b) = gcd(b, r)$.

More details about the Euclidean algorithm can be found in [20]. Here, we extend them to lots of integers.

Definition 1: We define $gcd(a_1, a_2, \ldots, a_n)$ as the greatest common divisor of integers $a_1, a_2, \ldots, a_n$.

The extended Euclidean algorithm is an extension to the Euclidean algorithm. It has the following result [21]:

Given integers $a_1, a_2, \ldots, a_n$ and let d be the greatest common divisor of them. There exists an efficient algorithm to compute integers $s_1, s_2, \ldots, s_n$ that satisfy $a_1s_1 + a_2s_2 + \ldots + a_ns_n = d$.

B. THE ASSUMPTION OF RSA PROBLEM

We give the following definition of RSA problem with reference to [18].

Assume $n = pq$, where p and q are two large primes. For an element $y \in Z_n^*$ and a prime number e such that $gcd(e, \phi(n)) = 1$, it is difficult to compute an element x in Z_n^* such that $x^e = y \ mod \ n$, where $\phi(n) = (p-1)(q-1)$.

RSA Assumption: We define that the RSA assumption is a $(t_R, \epsilon_R)-$RSA assumption if the probability of any adversary solving the RSA problem is at least ϵ_R in time t_R.

C. COMMON MODULUS ATTACK

Here, we describe the common modulus attack [22] in brief. Suppose an adversary gets two different ciphertexts $c_1 = m^{e_1} \ mod \ n$ and $c_2 = m^{e_2} \ mod \ n$ of for the same m, where (e_1, n) and (e_2, n) are two public keys of two users and $gcd(e_1, e_2) = 1$. Then it can use the extended Euclidean algorithm to compute the plaintext m as follows.

It first computes r and t such that

$$e_1r + e_2t = gcd(e_1, e_2) = 1.$$

Then, it can compute:

$$c_1^r c_2^t = m^{e_1r+e_2t} \ mod \ n = m^{gcd(e_1,e_2)} \ mod \ n = m.$$

We will use the attack to analyze the IBS scheme [1] in section III.

D. IBS MODEL AND ITS SECURITY MODEL

The model of IBS [1] is described bellow.

- **Setup:** The key generation center (KGC) generates the system public parameters PP and the system master secret key sk.
- **Ext:** For any user with its identity id, the KGC generates its private key sk_{id} and sends it to the user secretly.
- **Sign:** For any message m, a user uses its private key to produce a signature σ_m.
- **Ver:** Any user can a signature σ_m of a message m if if the algorithm **Ver** outputs 1; Otherwise, the σ_m of a message m is invalid.

Next, the security model of IBS defined by Wei *et al.* [1] is as follows.

In the security model, the adversary can issue limited queries of private key generation for identity set U and signatures for challenge identity $id^* \notin U$.

- Setup: The challenger generates the public key and gives it to the adversary.
- Query: The adversary adaptively does the following queries and the number of queries is limited.
  - The adversary randomly chooses u_0 identities $\{id_i : i = 1, \ldots, u_0\}$. The challenger answers each query-identity id_i by running Ext algorithm.
  - The adversary selects a challenge identity $id^* \neq id_i (i = 1, .., u_0)$ and randomly chooses l messages $m_1, \ldots, m_l$ with respect to id^*. The challenger gets the private key of id^* by running Ext algorithm.

Then, the challenger runs Sign algorithm to compute the signature for each message.

- Forgery: The adversary generates a valid signature for id^* and $m^*(m^* \neq m_i)$.

Definition 2: We call the IBS scheme is $(t, \epsilon)-$secure if the probability of any adversary breaking our scheme is at least ϵ in time t.

III. ANALYSIS OF THE PROTOCOL

Wei et al.'s protocol is constructed by using one identity-based signature (IBS) scheme and two outsourcing algorithms [1]. At first, we review the IBS scheme and find a security defect in security analysis. Next, we show the impact of the security defect in the IBS scheme on the protocol [1]. At last, we estimate the difficulty of carrying out the common modulus attack to the protocol in practice by doing theoretical derivation and experiments.

A. RECALL OF THE IBS SCHEME

The IBS scheme [1] is defined bellow.

- **Setup:** Assume $n = pq$, where p and q are two large primes. Select a random element $g \in Z_n$, hash functions $H : Z_n^2 \to Z_n$ and $H_0 : U \times Z_n \to Z_n$, where U is the identity set. $pk = (g, n, U, H, H_0)$ is the public key and $sk = (p, q)$ is the master key.
- **Ext:** Select a random element $v_{id} \in Z_n$ for identity $id \in U$. Then, compute $w_{id} = H_0(id, v_{id})$ and $g_{id} = g^{\frac{1}{w_{id}}} (mod\ n)$. The private key of identity $id \in U$ is (g_{id}, v_{id}).
- **Sign:** Given a message $m \in Z_n$, the signer id randomly chooses r and computes

$$\sigma = g_{id}^{H(m,r)} mod\ n.$$

  The signature of message m is (v_{id}, r, σ).
- **Ver:** A message receiver accepts the message m with a signature(m, v_{id}, r, σ) if

$$g^{H(m,r)} = \sigma^{w_{id}} mod\ n$$

  holds, where $w_{id} = H_0(id, v_{id})$. Otherwise, the message receiver rejects it.

B. SECURITY ANALYSIS OF THE IBS SCHEME

In the subsection, we show how an adversary $\mathcal{A}$ uses the common modulus attack to forge a valid signature for the IBS scheme constructed by Wei *et al.* [1].

According to the security model of IBS scheme, after the adversary $\mathcal{A}$ receives the challenge identity id^*, it can continue to query signatures of some messages.

It randomly chooses some messages $m_1, \ldots$ with respect to id^*. The challenger $\mathcal{C}$ computes these signatures $((m_1, v_{id^*}, r_1, \sigma_1), (m_2, v_{id^*}, r_2, \sigma_2), \ldots)$ by running Sign algorithm, and sends them to the adversary $\mathcal{A}$. $\mathcal{A}$ computes all $H(m_i, r_i)$ and stops querying when $gcd(H(m_1, r_1), \ldots, H(m_j, r_j)) = 1$. Since there exists an efficient algorithm to compute $s_1, \ldots, s_j$ such that $H(m_1, r_1)s_1 + \ldots + H(m_j, r_j)s_j = 1$, which is in section II, $\mathcal{A}$ can get the following value, private key of the user.

$$\sigma_1^{s_1} \ldots \sigma_j^{s_j} = g_{id^*}^{H(m_1,r_1)s_1+\ldots+H(m_j,r_j)s_j} mod\ n = g_{id^*}.$$

Thus, $\mathcal{A}$ can generates a signature of any message after it computes the private key g_{id^*} of the user. The privacy preserving protocol in VANETs [1] based on the outsourcing computations and the IBS scheme, so the protocol also is insecure.

C. SITUATION OF IMPLEMENTING THE COMMON MODULUS ATTACK IN PRACTICE

From the previous description, it can be concluded that the condition for performing the common modulus attack in the protocol is to obtain a sequence of message/signature pairs $(m_1, (r_1, \sigma_1)), (m_2, (r_2, \sigma_2)), \ldots, (m_j, (r_j, \sigma_j))$, where

$$gcd(H(m_1, r_1), H(m_2, r_2), \ldots, H(m_j, r_j)) = 1.$$

We view the hash function as random function, therefore, the output of the hash function is random value. If $gcd(\dot{,} \ldots, \dot{)} = 1$ holds, it requires to a large number of signatures which exceeds the limitation of the number of the queries, the common modulus attack for the IBS scheme [1] is difficult to implement in practice. To estimate the difficulty of carrying out the common modulus attack for the IBS scheme in practice, we do the theoretical derivation and experiments.

1) THEORETICAL ESTIMATION

Here, we first recall a Theorem 1 which refers to [23].

Theorem 1 [23]: The probability that two integers should be prime to one another is $\frac{6}{\pi^2}$, where π is the circular constant.

To estimate the possibility of the common modulus attack implemented to this protocol in practice, we give the Lemma 1.

Lemma 1: Let $a_1, \ldots, a_i$ $(i \geq 2)$ be random integers, p is the possibility of $gcd(a_1, a_2, \ldots, a_i) = 1$. Then p must satisfy:

$$p \geq 1 - (1 - \frac{6}{\pi^2})^{i(i-1)/2}$$

Proof: Assume three envents:

- ϵ_1: For any random integers a and b, $gcd(a, b) = 1$.
- ϵ_2: There is at least one pair (a_x, a_y) with $gcd(a_x, a_y) = 1$, where a_x, a_y in $\{a_1, a_2, \ldots, a_i\}$ $(i \geq 2)$.
- ϵ_3: $gcd(a_1, a_2, \ldots, a_i) = 1$.

The greatest common divisor is 1, which is equivalent to relatively prime. So we can get $Pr[\epsilon_1] = \frac{6}{\pi^2}$ according to Theorem 1. Thus

$$Pr[\epsilon_2] = 1 - (1 - Pr[\epsilon_1])^{\binom{2}{i}},$$

we can compute

$$Pr[\epsilon_2] = 1 - (1 - \frac{6}{\pi^2})^{i(i-1)/2}.$$

TABLE 1. Time spent attacking.

Length	Attack Times	Total Time Cost	Average Time Cost
1024 bits	1000	11144 ms	11 ms
2048 bits	1000	57639 ms	57 ms

If there is at least one pair (a_x, a_y) with $gcd(a_x, a_y) = 1$ in $a_1, a_2, \ldots, a_i$ $(i \geq 2)$, we can get $gcd(a_1, a_2, \ldots, a_i) = 1$. But there may be no one pair (a_x, a_y) with $gcd(a_x, a_y) = 1$ in $a_1, a_2, \ldots, a_i$ $(i \geq 2)$, if $gcd(a_1, a_2, \ldots, a_i) = 1$. So we can know $\epsilon_2 \subset \epsilon_3$ and $Pr[\epsilon_2] \leq Pr[\epsilon_3]$. At last, we can get

$$Pr[\epsilon_3] \geq 1 - (1 - \frac{6}{\pi^2})^{i(i-1)/2},$$

Since $Pr[\epsilon_3] = p$,

$$p \geq 1 - (1 - \frac{6}{\pi^2})^{i(i-1)/2}.$$

When $i = 3$, $p \geq 94\%$. This means that the adversary with a probability 94% at least can succeed to compute the user's private key, when it get 3 signature/message pairs. So the common modulus attack is easily implemented in the protocol.

2) EXPERIMENT ESTIMATION

We use experiments to measure the difficulty of implementing the common modulus attack for this protocol in practice. We randomly choose 10 1024-bit integers and 10 2048-bit integers as the modulus. For each modulus, we randomly choose 100 sets integers and the greatest common divisor of each set is 1. For each set of integers, we calculate the coefficients that make their sum be 1 and count the running time. We do our experiments by using Java on the Win10 operation system over a computer with Intel I7 8550U CPU and 16 GB memory. TABLE 1 shows the results of our experiments. The results show that it is very easy to implement the common modulus attack for the protocol in practice.

IV. IMPROVED SCHEME

To resolve the defect of the IBS scheme [1], we improve and propose a new IBS scheme for the protocol [1]. Moreover, we prove our improved IBS scheme is secure against chosen-message attack without random oracle by using the security model proposed by Wei *et al.* [1], which has been described in section II-D.

A. IMPROVED IBS SCHEME

The details of our improved IBS scheme is as follows:

- **Setup:** Assume $n = pq$, where p and q are two large primes. Select a random element $g \in Z_n^*$, hash functions $H : Z_n^2 \to Z_n$ and $H_0 : U \times Z_n \to Z_n$, where U is the identity set. $pk = (g, n, U, H, H_0)$ is the public key and $sk = (p, q)$ is the master secret key.
- **Ext:** Select a random element $v_{id} \in Z_n$ for identity $id \in U$. Then compute $w_{id} = H_0(id, v_{id})$ and $g_{id} = g^{\frac{1}{w_{id}}} (mod\ n)$. The private key of id is (g_{id}, v_{id}).
- **Sign:** Given a message $m \in Z_n$, the signer id randomly chooses $r \in Z_n$, $a \in Z_n$ and computes

$$\sigma_1 = (g_{id})^{aH(m,r)} mod\ n,$$
$$\sigma_2 = g^a\ mod\ n.$$

  The signature of message m is $(v_{id}, r, \sigma_1, \sigma_2)$.
- **Ver:** A message receiver accepts the message m with a signature $(v_{id}, r, \sigma_1, \sigma_2)$ if

$$\sigma_2{}^{H(m,r)} = \sigma_1{}^{w_{id}}\ mod\ n$$

  holds. Otherwise, the message receiver rejects it.

B. SECURITY PROOF

Before proving the security of our IBS scheme, we first recall the following lemma [24], which will be used.

Lemma 2 [24]: Given $x, y \in Z_n^*$ and $a, b \in Z$ such that $x^a = y^b$, one can efficiently compute $z \in Z_n^*$ such that $z = y^{\frac{gcd(a,b)}{a}}$.

Theorem 2: If $(t_R, \epsilon_R)-$ RSA assumption holds, the signature scheme is $(t, \epsilon)-$secure and

$$\epsilon \approx (\frac{e-1}{e})^2 \epsilon_R,\ t \approx t_R - c_Z(u_0 + l),$$

where (e, y, n) is the given RSA challenge,e is a large prime, c_Z is a constant that depends on Z_n, u_0 is the number of private key queries and l is the number of signature queries.

Proof: Let $\mathcal{A}$ be an adversary and $\mathcal{C}$ be a challenger. Then security game between $\mathcal{A}$ and $\mathcal{C}$ is constructed as follows. $\mathcal{C}$ is given RSA problem (n, e, y) and needs to find a z that satisfies $y = z^e\ mod\ n$. Let H_1, H_2 be chameleon hash functions, U be the identity set and u_0 be the number of identities chosen by $\mathcal{A}$.

- **Setup.** The challenger $\mathcal{C}$ computes

$$g = y^{\prod_{j=1}^{u_0} w_j}\ (mod\ n)$$

  for w_j, where w_j is randomly chosen such that $gcd(w_j, e) = 1$ and let $w = \prod_{j=1}^{u_0} w_j$. Then, $\mathcal{C}$ sends public key $pk = (g, n, U, H, H_0)$ to the adversary $\mathcal{A}$.
- **Query.** The adversary $\mathcal{A}$ can make a polynomial number of the following queries:
  - **Query of private key.** The adversary $\mathcal{A}$ randomly chooses u_0 identities (denoted by $U_0 = \{id_i : i = 1, \ldots, u_0\}$) to query. To answer the query, the challenger $\mathcal{C}$ makes $w_{id_j} = w_j$ and uses the trapdoor to driver v_{id_j} from $w_{id_j} = H_0(id_j, v_{id_j})$. Next, $\mathcal{C}$ computes

$$g_{id_j} = y^{e_{id_j}}\ (mod\ n),$$

    where $e_{id_j} = w/w_{id_j}$. At last, $\mathcal{C}$ returns the private key (g_{id_j}, v_{id_j}) for the identity $id_j (j = 1, \ldots, u_0)$.
  - **Query of signature.** The adversary $\mathcal{A}$ selects a challenge identity $id^* \neq id_i$ $(i = 1, \ldots, u_0)$

and randomly chooses l message $m_1, \ldots, m_l$ with respect to id^*. The challenger $\mathcal{C}$ randomly chooses number $b_1, \ldots, b_l$ and set $w_{id^*} = e$. Then, $\mathcal{C}$ drivers v_{id^*} from $w_{id^*} = H_0(id^*, v_{id^*})$ and r_i from $H(m_i, r_i) = b_i w_{id^*}$. Next, $\mathcal{C}$ randomly chooses number $a_1, \ldots, a_l$ from Z_n and computes

$$\sigma_{1_i} = (g_{id^*})^{a_i H(m_i, r_i)} \ mod \ n,$$
$$\sigma_{2_i} = g^{a_i} \ mod \ n.$$

At last, $\mathcal{C}$ returns $(v_{id^*}, r_i, \sigma_{1_i}, \sigma_{2_i})$ as the signature of message m_i.

- **Forgery.** For the change identity id^*, the adversary $\mathcal{A}$ forges a valid signature $(v_{id^*}, r, \sigma_1, \sigma_2)$ of a message m_0, where $\sigma_2 = g^a \ mod \ n$ and ${\sigma_1}^{w_{id}} = {\sigma_2}^{H(m_0, r)} \ mod \ n$. In other words, ${\sigma_1}^e = y^{waH(m_0, r)} \ mod \ n$.

Firstly, $\mathcal{C}$ can get $gcd(w, e) = 1$, because $w = \prod_{j=1}^{u_0} w_j$ and $gcd(w_j, e) = 1$. If $gcd(aH(m_0, r), e) \neq 1$, $\mathcal{C}$ aborts. Otherwise, the challenger $\mathcal{C}$ can use Lemma 2 to compute

$$x = \sigma_1^{\frac{gcd(waH(m_0,r),e)}{waH(m_0,r)}} \ mod \ n,$$

then $\mathcal{C}$ can get

$$\begin{aligned} x^e &= (\sigma_1^e)^{\frac{gcd(waH(m_0,r),e)}{waH(m_0,r)}} \ mod \ n \\ &= y^{waH(m_0,r)\frac{1}{waH(m_0,r)}} \ mod \ n \\ &= y \ mod \ n. \end{aligned}$$

So x is the answers of the RSA problem (n, e, y) and the challenger $\mathcal{C}$ solves the RSA problem. In the case of $gcd(aH(m_0, r), e) = 1$, the challenger $\mathcal{C}$ can construct a solution to solve the RSA problem. Because the probability of $gcd(a, e) = 1$ is $\frac{e-1}{e}$ and the probability of $gcd(H(m_0, r), e) = 1$ is $\frac{e-1}{e}$, the probability of $gcd(aH(m_0, r), e) = 1$ is $(\frac{e-1}{e})^2$. Thus, we can get that the probability of breaking our scheme is

$$\epsilon \approx (\frac{e-1}{e})^2 \epsilon_R.$$

Assume the time of exponentiation on Z_n is c_Z, so we can compute that the time of breaking our time is $t \approx t_R - c_Z(u_0 + l)$. ■

Thus, our IBS scheme is secure against the chosen identity attack and the chosen message attack.

V. OUR IMPROVED PROTOCOL

We first recall the outsourcing algorithm in [1], and then construct our improved protocol.

A. OUTSOURCING ALGORITHMS

Wei *et al.* proposed two outsourcing algorithms for exponential operation $u^a (mod \ n)$ to a cloud server [1].

Algorithm 1 (*A*1)

Assume u is public and a_i is secret, where $i = 1, \ldots, n_0$. The algorithm uses a untrusted cloud server to compute u^{a_i}.

- **Setup.** At first, the client randomly choose a number a_0 and computes and saves u^{a_0}. Then, the client sends $a_i - a_0$ and u to the cloud server.
- **Outsourcing computation.** The cloud computes $u^{a_i - a_0}$ and returns the result to the client.
- **Output.** the client computes $u^{a_i} = u^{a_0} \cdot u^{a_i - a_0}$.

Algorithm 2 (*A*2)

Assume u and a_i are secret, where $i = 1, \ldots, n_0$. The algorithm uses a untrusted cloud server to compute u^{a_i} without revealing u and a_i.

- **Setup.** At first, the client randomly choose a number a_0 and computes and saves u^{a_0}. Then, it randomly chooses a 2×2 invertible matrix H. Next, it sends $a_i - a_0$ and

$$A_i = H \cdot \begin{pmatrix} u & r_i \\ 0 & u^l \end{pmatrix} \cdot H^{-1}$$

to the cloud server, where r_i is randomly chosen and l is any small integer, such as 2.
- **Outsourcing computation.** The cloud server computes $B_i = A_i^{a_i - a_0}$ and returns the result to the client.
- **Verification and output.** The client computes $C_i = H^{-1} B_i H$ and gets $(C_i)_{11}, (C_i)_{22}$. It first checks whether $(C_i)_{11}^2 = (C_i)_{22}$ or not. If it holds, then this means $(C_i)_{11} = u^{a_i - a_0}$. The client outputs $u^{a_1} = u^{a_0} \cdot u^{a_i - a_0}$.

1) COMPATIBLE WITH OUTSOURCING ALGORITHMS

Our scheme can also use the Wei et al.'s outsourcing algorithms [1] to reduce computational overhead. We have

$$\sigma_1 = (g_{id})^{aH(m,r)} mod \ n = A_2(g_{id}, aH(m, r)),$$
$$\sigma_2 = g^a \ mod \ n = A_1(g, a).$$

B. OUR CONSTRUCTION

We replace the original IBS scheme [1] with our improved IBS scheme in the protocol [1]. FIGURE1 shows process of signature and verification in our updated protocol. The detail of our updated protocol is as follows:

- **Setup:** TA chooses two large primes p and q, a random element $g \in Z_n^*$ and collision resistant hash functions $H : Z_n^2 \to Z_n, H_0 : U \times Z_n \to Z_n$, where U is the identity set. Then, TA computes $n = pq$. At last, TA sets the system master secret key $sk = (p, q)$ and public key $pk = (g, n, U, H, H_0)$.
- **Key Generation:** This step is divided into three substeps.
  - TA chooses e, d such that $e \cdot d = 1 (mod \ \phi(n))$ and let e public. Then, TA make d be the secret key.

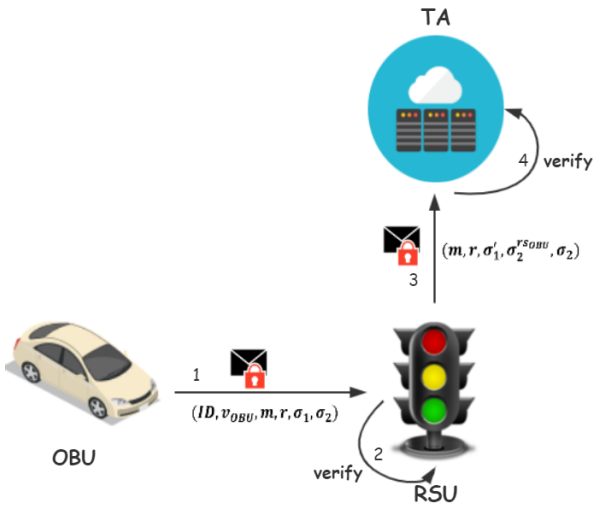

FIGURE 1. Model of our improved protocol.

- OBU randomly selects x_{OBU} and computes $v_{OBU} = g^{x_{OBU}}$. Then OBU selects k, computes $w_1 = H_1(g^k \parallel ID)$, $w_2 = k + w_1 \cdot v_{OBU}$ and sends (ID, w_1, w_2, v_{OBU}) to TA, where $H_1 : 0, 1^\lambda \rightarrow Z_n$. If $w_1 = H_1(g_2^{w_2} \cdot v_{OBU}^{-w_1} \parallel ID)$, then TA can make sure that v_{OBU} and ID are real identification of OBU. Ultimately, TA computes $w_{OBU} = H_0(ID, v_{OBU})$ and sends private key $g_{OBU} = g^{w_{OBU}^{-1}}$ to OBU. Here, TA can use the above outsourcing algorithm, then $g^{w_{OBU}^{-1}} = A1(g, w_{OBU}^{-1})$, $g_{OBU} = A1(g, w_2)$ and $v_{OBU}^{-w_1} = A2(v_{OBU}, -w_1)$.
- RSU builds its own public encryption algorithm Enc_{RSU} with public key and secret key (pk_{RSU}, sk_{RSU}).

- **Key Generation for Re-signature:** TA randomly chooses s_{OBU} and computes $A1(g, s_{OBU}) = g^{s_{OBU}}$. RSU's re-signature key for OBU is $(ID, g^{s_{OBU}}, y_{OBU})$, where $y_{OBU} = d \cdot w_{OBU} \cdot s_{OBU}$. Meanwhile, TA appends ID, g^{soBU} to a list T which can be used to trace the OBU's real identity.
- **OBU Signature:** The message m signed by OBU is $m = ID_{type} \parallel PL \parallel Time$, where ID_{type} is message type, PL is message load payload and $Time$ is the accurate time of the message generation. OBU executes the following algorithms:
  - For message $m = ID_{type} \parallel PL \parallel Time$, OBU randomly selects a, r then executes outsourcing algorithms Algorithm 1 and Algorithm 2 to obtain
    $$\sigma_1 = (g_{OBU})^{aH(m,r)} mod\ n = A_2(g_{OBU}, aH(m, r)),$$
    $$\sigma_2 = g^a\ mod\ n = A_1(g, a).$$
  - OBU encrypts $M = (ID, v_{OBU}, m, r, \sigma_1, \sigma_2)$ by using the public key of RSU and sends $Enc_{RSU}(M)$ to RSU.
- **Re-signature:** RSU gets $M = (ID, v_{OBU}, m, r, \sigma_1, \sigma_2)$ from $Enc_{RSU}(M)$, and checks whether $\sigma_2^{H(m,r)} = \sigma_1^{H_0(ID, v_{OBU})}\ mod\ n$ or not. If the equation holds, then RSU can use re-signature key to compute $\sigma_1' = \sigma_1^{ry_{OBU}}$ and broadcasts $(m, r, \sigma_1', \sigma_2^{rs_{OBU}}, \sigma_2)$, where $\sigma_1' = A1(\sigma_1, ry_{OBU}), \sigma_2^{rs_{OBU}} = A1(\sigma_2, (rs_{OBU}))$.
- **Verification:** $(m, r, \sigma_1', \sigma_2^{rs_{OBU}}, \sigma_2)$ can be verified by any party. If $(\sigma_1')^e = (\sigma_2^{r \cdot s_{OBU}})^{H(m,r)}$ holds, the verifier returns 1, Otherwise, the verifier returns 0.
- **Tracing and revocation:** TA does the tracing process. TA and the RSU execute the revocation process together.
  - **Tracing.** If $(\sigma_1')^e = (\sigma_2^{r \cdot s_{OBU}})^{H(m,r)}$, TA can trace the real identity of the corresponding OBU. TA uses its secret key and outsourcing algorithm $A1$ to compute $r^{-1}(mod\ \phi(n))$, $a^{-1}(mod\ \phi(n))$ and
    $$A1(A1(\sigma_2^{r \cdot s_{OBU}}, r^{-1}), a^{-1}) = (g^{ar \cdot s_{OBU}})^{r^{-1}a^{-1}} = g^{s_{OBU}},$$
    then TA gets the corresponding $ID, g^{s_{OBU}}$ from local list T.
  - **Revocation.** Once TA discovers a malicious vehicle OBU,TA sends $g^{s_{OBU}}$ to the RSU to cancel this OBU. What's more, TA and RSU delete $ID, g^{s_{OBU}}$ from list T.

TABLE 2. Signing time cost and Verifying time cost.

Scheme	Length	Signing Average Time Cost	Verifying Average Time Cost
Wei et al.'s scheme [1]	1024 bits	1.3 ms	2.4 ms
	2048 bits	8.4 ms	16.2 ms
our scheme	1024 bits	3.8 ms	2.7 ms
	2048 bits	24.4 ms	16.2 ms

C. COMPARATIVE ANALYSIS OF EFFICIENCY

We analyze the effect of our improved IBS scheme on the efficiency of Wei et al.'s protocol [1].

- **Storage Cost.** Because of the size of the public key, the master secret key and user's private key is not changed, storage cost is not changed after using our improved IBS scheme.
- **Communication Cost.** In our improved IBS scheme, there is one more σ_2 for each signature. So the signature of our scheme is 1024 bits longer than the original IBS scheme [1] and the data sent by the OBU to the RSU increases to 3360 bits.
- **Computation Cost.** The Setup step and Ext step of our improved IBS scheme are the same as the original IBS scheme [1], so we only need to analyze the computation cost of Sign step and Ver step. We choose different length of modulus n and count the time that running Sign step and Ver step 1000 times need in the two IBS scheme. We do this experiment by using Java on Win10 operation system over a computer with

Intel I7 8550U CPU and 16 GB memory. According to TABLE 2, although our improved IBS scheme consumes three times as much time on the signing as the Wei et al.'s IBS scheme [1], it is still acceptable. Furthermore, TABLE 2 also shows that our improved IBS scheme and Wei et al.'s IBS scheme [1] take the same amount of time to verify the signature.

VI. CONCLUSION

In this paper, we first showed that there was a security defect in the IBS scheme [1] that the protocol depended and attackers could use the common modulus attack to get the user's private key. Then, we analyzed the impact of the security defect in the IBS scheme on the protocol [1]. In addition, we theoretically and experimentally proved that the protocol was vulnerable to the common modulus attack in practice. Next, we proposed an improved IBS scheme, updated the protocol and proved our scheme was secure against chosen-message attack without random oracle. At last, we analyzed the efficiency of our scheme in the updated protocol.

REFERENCES

[1] Z. Wei, J. Li, X. Wang, and C.-Z. Gao, "A lightweight privacy-preserving protocol for VANETs based on secure outsourcing computing," *IEEE Access*, vol. 7, pp. 62785–62793, 2019.

[2] P. Wang, R. Valerdi, S. Zhou, and L. Li, "Introduction: Advances in IoT research and applications," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 239–241, Apr. 2015.

[3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[4] R. K. Kodali, G. Swamy, and B. Lakshmi, "An implementation of IoT for healthcare," in *Proc. Intell. Comput. Syst.*, 2016.

[5] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 2, pp. 112–121, Jan. 2014.

[6] A. Das, P. Dash, and B. K. Mishra, "An innovation model for smart traffic management system using Internet of Things (IoT)," in *Cognitive Computing for Big Data Systems Over IoT*. Cham, Switzerland: Springer, 2018, pp. 355–370.

[7] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017.

[8] J. Sathishkumar and D. R. Patel, "A survey on Internet of Things: Security and privacy issues," *Int. J. Comput. Appl.*, vol. 90, no. 11, pp. 20–26, Mar. 2014.

[9] C. C. Niu, K. C. Zou, Y. L. Ou Yang, G. J. Tang, and Y. Zou, "Security and privacy issues of the Internet of Things," *Appl. Mech. Mater.*, vols. 416–417, pp. 1429–1433, Sep. 2013.

[10] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2014.

[11] Y. Liao, Y. Liu, Y. Liang, Y. Wu, and X. Nie, "Revisit of certificateless signature scheme used to remote authentication schemes for wireless body area networks," *IEEE Internet Things J.*, to be published.

[12] M. Ramadan, Y. Liao, F. Li, S. Zhou, and H. Abdalla, "IBEET-RSA: Identity-based encryption with equality test over RSA for wireless body area networks," *Mobile Netw. Appl.*, pp. 1–11, Apr. 2019.

[13] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, Aug. 2012, doi: 10.1007/s11235-010-9400-5.

[14] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: A generic cloud computing model for vehicular Ad Hoc networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 96–102, Feb. 2015.

[15] S. S. Tangade and S. S. Manvi, "A survey on attacks, security and trust management solutions in VANETS," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–6.

[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1985, pp. 47–53.

[17] L. C. Guillou and J. J. Quisquater, "A 'paradoxical' identity-based signature scheme resulting from zero-knowledge," in *Proc. Int. Cryptol. Conf. Adv. Cryptol.-Crypto*, Santa Barbara, CA, USA, Aug. 1988.

[18] X. Wang and H. Qian, "A novel identity-based signature with message recovery from RSA," in *Proc. Int. Conf. Electron. Mech. Eng. Inf. Technol.*, vol. 9, Aug. 2011, pp. 4827–4831.

[19] J. Herranz, "Identity-based ring signatures from RSA," *Theor. Comput. Sci.*, vol. 389, nos. 1–2, pp. 100–117, Dec. 2007.

[20] B. Sunar, *Euclidean Algorithm*. Boston, MA, USA: Springer, 2005, pp. 204–206.

[21] Wikipedia. (Jul. 2019). *Extended Euclidean Algorithm*. [Online]. Available: https://doi.org/10.1007/s11235-010-9400-5

[22] A. J. Menezes, S. A. Vanstone, and P. C. van Oorschot, *Handbook of Applied Cryptography*. 1997.

[23] G. Hardy and E. Wright, *An Introduction to the Theory of Numbers*, vol. 355. 2009.

[24] D. Cash, R. Dowsley, and E. Kiltz, "Digital signatures from strong RSA without prime generation," in *Public-Key Cryptography—PKC*, J. Katz, Ed. Berlin, Germany: Springer, 2015, pp. 217–235.