A Large-Scale Concurrent Data Anonymous Batch Verification Scheme for Mobile Healthcare Crowd Sensing

Jingwei Liu∗, Member, IEEE, Huijuan Cao, Qingqing Li†, Fanghui Cai, Xiaojiang Du∗, Senior Member, IEEE, and Mohsen Guizani†, Fellow, IEEE

Abstract—Recently, with the rapid development of big data, Internet of Things (IoT) brings more and more intelligent and convenient services to people’s daily lives. Mobile healthcare crowd sensing (MHCS), as a typical application of IoT, is becoming an effective approach to provide various medical and healthcare services to individual or organizations. However, MHCS still have to face to different security challenges in practice. For example, how to quickly and effectively authenticate masses of bio-information uploaded by IoT terminals without revealing the owners’ sensitive information. Therefore, we propose a large-scale concurrent data anonymous batch verification scheme for MHCS based on an improved certificateless aggregate signature. The proposed scheme can authenticate all sensing bio-information at once in a privacy preserving way. The individual data generated by different users can be verified in batch, while the actual identity of participants is hidden. Moreover, assuming the intractability of computational Diffie–Hellman problem, our scheme is proved to be secure. Finally, the performance evaluation shows that the proposed scheme is suitable for MHCS, due to its high efficiency.

Index Terms—Aggregate signature (AS), batch verification, mobile healthcare crowd sensing (MHCS), privacy preservation.

I. INTRODUCTION

INTERNET of Things (IoT), as a promising paradigm, can change the interactive way between networks and the physical world [1]. Meanwhile, with the popularization and development of wireless sensors, a new perceptual architecture—mobile crowd sensing (MCS) [2], [3], has emerged. It provides an important technical support for the integration of the physical world with higher layer applications in IoT. As an important application branch of MCS, mobile healthcare crowd sensing (MHCS) provides more convenient medical and healthcare services for organizations or individual.

Manuscript received January 18, 2018; revised February 20, 2018; accepted April 10, 2018. Date of publication April 19, 2018; date of current version May 8, 2019. This work was supported by the Joint Fund of the Ministry of Education of China under Grant 6141A022338, the 111 Project of China under Grant B08038, and in part by the Collaborative Innovation Center of Information Sensing and Understanding at Xidian University. (Corresponding author: Jingwei Liu.)

J. Liu, H. Cao, Q. Li, and F. Cai are with the State Key Lab of ISN, Xidian University, Xi’an 710071, China (e-mail: jliu@mail.xidian.edu.cn; caohuijuan345@163.com; 15229259171@163.com; caidoreen@163.com).

X. Du is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: dxj@ieee.org).

M. Guizani is with the Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID 83844 USA (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/JIOT.2018.2828463

MHCS, combining the merits of MCS with remote healthcare, is becoming a research hotspot. On one hand, participants in MCS upload health data collected by mobile terminals to cloud server and enjoy various services by healthcare organizations. On the other hand, remote healthcare system can provide health information and medical service anytime and anywhere, by analyzing the individual health data and patient vital signs submitted to remote health apps installed in mobile terminals or monitoring devices. Therefore, MHCS can not only provide real-time medical services to individual or community, but also improve the ability of healthcare organizations to monitor, track and control certain diseases in some regions.

However, there are still many security threats and privacy issues in MHCS: 1) the collected health data may deduce users’ sensitive information, such as identity, personal activities, and health status; 2) the data may be obtained or changed by an opponent, which will bring damage to people’s health and property, even people’s lives; and 3) these data collected by mobile devices should be processed safely in a real-time manner, otherwise the quality of medical service will be reduced. Therefore, the security and privacy preservation for MHCS is need to be considered emergently. So more and more security and privacy preservation schemes [4]–[18] for MCS and remote healthcare have been proposed recently. In [4] and [5], two essential key management schemes were introduced for the MHCS systems. Moreover, compared to the traditional healthcare systems, MHCS systems have to face more security threats, so the schemes [6]–[8] have studied relevant security issues. In addition, due to the high-risk of privacy leaking, participants in MHCS systems are often extremely concerned about privacy preservation issues. If their privacy cannot be protected effectively, they may not be willing to attend in MHCS. The schemes [9]–[18] provided different schemes for privacy preservation in MHCS systems. In this paper, we mainly focus on the security and privacy preservation for MHCS systems with large-scale concurrent data.

According to [19], a simple architecture of the MHCS system consists of MHCS participants, a cloud server, and healthcare organizations. In MHCS, as shown in Fig. 1, the cloud sever publishes sensing task for specific purpose. The participants receive a sensing task published from cloud sever, then they collect and upload the relevant health data to the sever. Meanwhile, the cloud sever will deliver the requested

2327-4662 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
information to specific organizations or healthcare institutes so as to make further analysis. However, millions of participants submit numerous biomedical data to the cloud sever, which will lead to data transmission obstacles and storage capacity burdens. An efficient approach named aggregate signature (AS) can improve the efficiency of the verification on numerous signatures and reduce the overhead of storage and bandwidth. The first AS scheme based on traditional public key cryptography was proposed by Boneh et al. [20] in 2003. It allowed multiple users to generate the signatures on different messages, respectively, and verify them in batch.

Following Boneh’s work, many AS schemes were proposed subsequently, but most of them were involved in the complicated certificate management problem. Thus, certificateless public key cryptosystem (CL-PKC) appeared to solve this issue. In 2007, Castro and Dahab [21] first introduced the concept of certificateless AS (CL-AS) that combined the merits of AS with CL-PKC. Then, Gong et al. [22] raised the formal security model for CL-AS in the same year. After the initial work, lots of CL-AS have been proposed [23]–[26].

In this paper, we put forward a large-scale concurrent data anonymous batch verification scheme for MHCS. The main work of this paper are summarized as follows.

1) The proposed scheme can provide bio-information batch verification and anonymous authentication for MHCS systems.

2) Based on the hardness of the computational Diffie–Hellman problem (CDHP), it is formally proved that our scheme is secure against the existential forgery attack on adaptively chosen message.

3) In the quantitative performance evaluation, our scheme achieves less computation overhead compared with the previous schemes. It is very suitable for the MHCS systems in practice.

The rest part of this paper is organized as follows. First, we introduce the reference model, security model, and design goals in Section II. In Section III, we improve a CL-AS scheme with the security proof. In Section IV, we describe the large-scale concurrent data anonymous batch verification scheme in detail. In Section V, we analyze the performance. Finally, we conclude this paper in Section VI.

II. MODELS AND DESIGN GOALS

For a better understanding, we first put forward the relevant models for MHCS, and then raise design goals.

A. Reference Model

The reference model for MHCS scenarios consists of four entities: 1) requestor; 2) data center (DC); 3) management server (MS); and 4) MHCS participants, as shown in Fig. 2.

1) Requestor: The requestors can submit healthcare sensing tasks to DC for some specific purposes. And they can further analyze the final report from DC to predict certain medical or health issues in some regions.

2) DC: It can publish and manage healthcare sensing tasks according to the demands of the requestors. Also, it is responsible for aggregating and verifying all collected health data from different participants.

3) MS: MS is a trusted third party who can manage the participants’ registration information in MHCS systems. It is in charge of issuing the a half private keys for legitimate participants and distributes the index of the participants to cover their actual identity. Here, DC can use the index to authenticate the uploaded health data from the participants.

4) MHCS Participants: MHCS participants refer to the mobile clients who collect and submit relevant health data using smart terminals for DC.

B. Security Model

To make better security analysis, we refer to the security model defined in [27], in which there are two types of opponents who are able (or unable) to replace certain participants’ public key without (or with) the MS’s private key. In this model, it can be proved that our scheme is secure against the above two kinds of opponents, if the following CDHP is
intractable. Here, we give the definition of the CDHP: in a large prime order $q$ cycle additive group $G_a$, $\forall x P, y P$ with a generator $P$ and unknown $x, y \in Z_q^*$, get $xyP$ finally.

C. Design Goals

Our design goals aim at designing a large-scale concurrent data anonymous batch verification scheme for MHCS, which achieves following properties.

1) **Batch Authentication**: The authentication information in the signed bio-data from large-scale MHCS participants could be aggregated and verified effectively by DC.

2) **Nonrepudiation**: MHCS participants cannot deny that they have submitted the related health data to DC.

3) **Anonymity**: Although DC can acquire and check the aggregated authentication message, it cannot obtain the real identity of the data provider.

III. IMPROVED CL-AS SCHEME

To provide a cryptographic essential for our design goals, we primarily propose an improved CL-AS scheme and then give the relevant security proof in this section. It can not only be used to realize batch verification, but also can deal with the key escrow problem of identity-based public key cryptosystem [28]. Due to these merits, it could be the key to solving the partial key problem of identity-based public key cryptosystem.

1) **Security**: We give the specification on the system $\langle l, q, P, G_a, G_m, e, H_1, H_2, Q_{KGC} \rangle$ as its private/public key pair. Then, KGC publishes the public parameters, while store $s_{KGC}$ as its private key secretly.

2) **Set-Partial-Key**: It consists of two part algorithms, one is to generate the partial key by a client or a signer, the other one is to compute the partial key by the KGC.

a) A client or a signer, marked as $C_i$, obtains his or her partial secret key by choosing $s_{1i} \in Z_q$ randomly and the partial public key by computing $Q_{1i} = s_{1i}P$.

b) $C_i$ sends his/her $id_i$ to KGC and request the partial key for the identity $id_i$. KGC calculates $Q_{2i} = H_1(id_i, Q_{1i})$, $S_{2i} = s_{KGC}Q_{2i}$ for it and distributes the half private key to $C_i$ through secure channels. Hence, $C_i$ can obtain the public key $\{Q_{1i}, Q_{2i}\}$ and the private key $(s_{1i}, S_{2i})$. Note that, each identity only can be used once.

3) **Signing**: The signer chooses $k_i \in Z_q^*$ randomly and then sign a message $m_i$, as follows:

$$V_i = k_iQ_{1i}$$

$$h_i = H_2(m_i, V_i)$$

$$U_i = S_{2i} + k_ih_is_{1i}Q_{KGC}.$$  (1)

Then, the signer view the pair $\sigma_i = \langle V_i, U_i \rangle$ as the signature on $m_i$.

4) **Verification**: To ensure the validity of the signature $\sigma_i$ signed by a $C_i$ on the message $m_i$, the verification procedure is as follows:

$$Q_{2i} = H_1(id_i, Q_{1i})$$

$$h_i = H_2(m_i, V_i)$$

$$e(U_i, P) = e(Q_{2i} + h_iV_i, Q_{KGC}).$$  (2)

Obviously, if the above equations hold, the signature is valid. Additionally, the proposed scheme also satisfies correctness

$$e(U_i, P) = e(S_{2i} + k_ih_is_{1i}Q_{KGC}, P)$$

$$= e(S_{2i}, P)e(k_ih_is_{1i}Q_{KGC}, P)$$

$$= e(s_{KGC}Q_{2i}, P)e(k_ih_is_{1i}P, Q_{KGC})$$

$$= e(Q_{2i}, s_{KGC}P)e(k_ih_iQ_{1i}, Q_{KGC})$$

$$= e(Q_{2i}, Q_{KGC})e(h_iV_i, Q_{KGC})$$

$$= e(Q_{2i} + h_iV_i, Q_{KGC}).$$
The σ = ⟨U, V⟩ is the final aggregated signature.

6) Aggregate Verification: On receiving an AS σ for aggregating Ci (from i = 1, 2, . . . , n) and the public key QKGC, the verifier will authenticate the AS. And the signature σ can be authenticated correctly, if the integrated formula holds: e(U, P) = e(Q2 + V, QKGC). Here, we give the proof of the equation on its correctness as follows:

\[
e(U, P) = e\left(\sum_{i=1}^{n} U_i, P\right) = \prod_{i=1}^{n} e(U_i, P)
= \prod_{i=1}^{n} e(S_{2i}, k_i h_i s_i 1, QKGC, P)
= \prod_{i=1}^{n} e(S_{2i}, P) e(k_i h_i s_i 1, QKGC, P)
= e(Q2, QKGC) e(V, QKGC)
= e(Q2 + V, QKGC).
\]

C. Security Proof

To make it convincing, it is proved that the proposed CL-AS scheme is existentially unforgeable against adaptively chosen message attacks in the random oracle model if the CDHP is intractable. As described in Section II, two types adversary, named A1 and A2 who attempt to forge a legal signature with different abilities (able/unable to use the PKC’s private key). We will prove the security of the proposed CL-AS under A1 and A2’s attacks, respectively. The detailed proofs are as follows.

**Theorem 1:** If the adversary A1 could break the proposed scheme by making q1/2 queries to H1/2, qk queries to extract-queries, ql queries to Secret-Key-Queries, qP queries to public-key-queries, qr queries to public-key-queries, and qsig to CLAS-sign-queries, so CDHP could be solved within

\[t + (q_1 + q_2 + q_k + q_l + q_P + q_r + q_{\text{sig}})m\]

with probability

\[\varepsilon' \geq \frac{1}{(q_k + 1)^e} \varepsilon.\]

**Proof:** Let C be a challenger trying to solve a CDHP instance (P, aP, bP) in Gq. For a, b ∈ Zq, we set X = aP and Y = bP. X, Y ∈ Gq is a CDHP instance in Gq. A1 interacts with C as the model in [24]. C sets QKGC = X.

Suppose A1 is a PPT Turing machine taking only open data as input, who has a advantage to break the proposed CLAS scheme with non-negligible probability. Given two random oracles which are H1 and H2, respectively, C gives the parameters (l, q, P, Ga, Gm, e, H1, H2, QKGC) to A1. C tries to simulate all above oracles to obtain the valid signatures of any message m; as the real signer. List L = (id1, s11, s2i, Q1i) is maintained by C. Throughout the proof process, ⊥ means the value of a variable is invalid. In particular, A1 can query as follows.

1) **H1-Queries:** On receiving a query I1 = ⟨id1, Q11⟩ on H1 from A1, with a list of tuple (I1, c1, α1, Q21), called Lc, C can simulate oracle H1 as follows.
   a) If I1 already exists in Lc, C outputs related Q2i.
   b) Otherwise, C sets c1 = 0 with probability λ and c1 = 1 with probability (1−λ). If c1 = 0, C chooses α1 ∈ Zq and outputs Q2i = baP ∈ Gq. If c1 = 1, then Q2i = α1P. In both cases, C inserts a tuple ⟨I1, c1, α1, Q2i⟩ to Lc.

2) **H2-Queries:** C simulates H2 by maintaining a list Lr with ⟨Ii, hi⟩. Here, Ii = (mi, Vi), mi ∈ {0, 1}n and Vi ∈ Gq. On inputting I1 to H2, C does as follows.
   a) If I1 already exists in Lr, C outputs the same answer.
   b) Otherwise, C chooses hi ∈ Zq and inserts a tuple ⟨I1, hi⟩ to Lr. Finally, it outputs hi as the answer.

3) **Extract-Queries:** A1 makes the query on ⟨id1, Q1i⟩.
   a) Firstly, C recovers the corresponding ⟨I1, c1, α1, Q2i⟩ from the list Lc. If c1 = 0, C returns failure. If c1 = 1 and L contains ⟨id1, s11, s2i, Q1i⟩, C checks if S2i = ⊥.
   b) If S2i = ⊥, C returns the current S2i to A1. Otherwise, H1(id1, Q1i) is set as α1P. C computes S2i = α1QKGC then C inserts a tuple ⟨id1, s11, s2i, Q1i⟩ to the list L and outputs S2i as the answer.
   c) Again, if c1 = 1, the list L does not contain ⟨id1, s11, s2i, Q1i⟩. Then, C sets S2i = ⊥ and computes S2i = α1QKGC. Finally, C inserts a tuple ⟨id1, s11, s2i, Q1i⟩ to L and replies S2i as output.

4) **Public-Key-Queries:** A1 makes the query on an identity id1.
   a) If ⟨id1, s11, s2i, Q1i⟩ is in L, C checks if Q1i = ⊥. If holds, C selects s11 ∈ Zq and Q1i = s11P. It updates ⟨Q1i, s11⟩ to L and replies Q1i to A1. Otherwise, C returns Q1i to A1.
   b) If ⟨id1, s11, s2i, Q1i⟩ is not in L, let S2i = ⊥, then selects a random s11 ∈ Zq and sets Q1i = s11P. C inserts a tuple ⟨id1, s11, s2i, Q1i⟩ to L and replies Q1i to A1.

5) **Secret-Key-Queries:** A1 makes the query on an identity id1.
   a) If ⟨id1, s11, s2i, Q1i⟩ is in L, C checks if s11 = ⊥. If holds, C selects a random s11 ∈ Zq. It also returns s11 and adds tuple ⟨id1, s11, s2i, Q1i⟩ to the list L. Otherwise, s11 = ⊥, C replies s11 to A1.
   b) If ⟨id1, s11, s2i, Q1i⟩ is not in L, C sets s11 = ⊥ and replies a random s11 ∈ Zq to A1.
6) Replace-Public-Key Queries: $A_1$ chooses new public key $Q’_{i1}$ for an identity $id_i$.
   a) If $\langle id_i, s_{i1}, Q_i, Q_{i1}\rangle$ is in $L$, $C$ sets $Q_{i1} = Q’_{i1}$ and $s_{i1} = L$. It updates a tuple $\langle id_i, s_{i1}, S_{2i}, Q_{i1}\rangle$ to the list $L$.
   b) If $\langle id_i, s_{i1}, S_{2i}, Q_{i1}\rangle$ is not in $L$, $C$ sets $Q_{i1} = Q’_{i1}$ and $s_{i1} = L$, then it inserts a tuple $\langle id_i, s_{i1}, S_{2i}, Q_{i1}\rangle$ to the list $L$.

7) CLAS-Sign-Queries: In this queries, $C$ provides valid signatures of any message $m_i$ of $id_i$ with list $\langle id_i, s_{i1}, S_{2i}, Q_{i1}\rangle$, $L_i = \langle (i, c_i, \alpha_i, Q_{2i})\rangle$, $L_v = \langle (j_i, h_{i})\rangle$, and answers the query as follows.
   a) If $L$ is not empty and $c_i = 1$, $C$ checks if $s_{i1} = L$. If $s_{i1} = L$, $C$ makes Public-Key-Queries to generate $s_{i1}$ and $Q_{i1} = s_{i1}P$.
   b) If $L$ is empty, $C$ makes Public-Key-Queries to generate $s_{i1}$ and $Q_{i1} = s_{i1}P$ and adds them to list $L$.
   c) $C$ tries to generate the signature. If $c_i = 1$, $C$ returns failure. Otherwise, $C$ picks a random $k_i \in \mathbb{Z}_{q^*}$, and computes

\[ V_i = k_iQ_{i1} \]
\[ h_i = H_2(m_i, V_i) \]
\[ U_i = S_{2i} + k_ih_is_{i1}Q_{KGC} \]  \hspace{1cm} (4)

Multiplying both side of (5) with $(h_i)^{-1}$ and both side of (6) with $(h_i')^{-1}$, we can obtain

\[ (h_i)^{-1}U_i = (h_i)^{-1}S_{2i} + (h_i)^{-1}h_is_{i1}Q_{KGC} \]
\[ (h_i')^{-1}U_i' = (h_i')^{-1}S_{2i} + (h_i')^{-1}h_is_{i1}Q_{KGC} \]  \hspace{1cm} (7)

Subtract (8) from (7)

\[ (h_i')^{-1}U_i' - (h_i)^{-1}U_i = [(h_i')^{-1} - (h_i)^{-1}]S_{2i}. \]  \hspace{1cm} (9)

Then, $C$ obtains $\langle id_i, s_{i1}, S_{2i}, Q_{i1}\rangle$ in $L$ and $\langle (i, c_i, \alpha_i, Q_{2i})\rangle$ in $L_v$, respectively. If $c_i = 1$, $C$ aborts. Otherwise, if $c_i = 0$, $Q_{2i} = \alpha_iP$, now $Q_{KGC} = aP = s_{KGC}P$. Because of $S_{2i} = aQ_{2i} = \alpha_iabP$, we can obtain

\[ (h_i')^{-1}U_i' - (h_i)^{-1}U_i = \left[(h_i')^{-1} - (h_i)^{-1}\right]abP \]  \hspace{1cm} (10)

\[ abP = \left[(h_i')^{-1}U_i' - (h_i)^{-1}U_i\right] \times \left[(h_i')^{-1} - (h_i)^{-1}\right]^{-1} \alpha_i^{-1}. \]  \hspace{1cm} (11)

Therefore, $C$ finds $abP$ as the solution to CDHP and solves CDHP with the probability

\[ \varepsilon' \geq \frac{1}{(q_k + 1)\varepsilon}. \]

There are three events needed by $C$ to succeed: $E_1$ is the result of any extract-queries raised by $A_1$ does not abort. $E_2$ represents $A_1$ generates a valid signature that can be verified. $E_3$ represents the probability that $A_1$ outputs a valid forgery and $C$ does not leave the game. The probability of $C$ success is that all the three events mentioned above happen

\[ P[E_1 \land E_2 \land E_3] = P[E_1]P[E_2|E_1]P[E_3|E_2 \land E_1]. \]

1) Claim 1: The probability of $E_1$ happening is at least $(1 - \lambda)^{qk}$, because $P[c_i = 1] = (1 - \lambda)$ and it takes at least $q_k$ queries. So, $P[E_1] \geq (1 - \lambda)^{q_k}$.  
2) Claim 2: The probability that $E_2|E_1$ happens is at least $\varepsilon$. So $P[E_2|E_1] \geq \varepsilon$.  
3) Claim 3: The probability that $E_3|E_2 \land E_1$ happens is at least $\lambda$, because $P[c_i = 0] = \lambda$, and $E_1|E_2$ both happen. So $P[E_3|E_2 \land E_1] \geq \lambda$.

Therefore, we can conclude that the probability of all three events happening is as follows:

\[ P[E_1 \land E_2 \land E_3] = P[E_1]P[E_2|E_1]P[E_3|E_2 \land E_1] = (1 - \lambda)^{q_k}\varepsilon. \]

We suppose $\lambda = [1/(q_k + 1)]$. Then

\[ \varepsilon' \geq (1 - \lambda)^{q_k}\varepsilon \]
\[ \varepsilon' \geq \frac{1}{(q_k + 1)}\left[1 - \frac{1}{\varepsilon}\right]^{q_k}\varepsilon. \]

If $q_k$ is sufficiently large, $(1 - (1/(q_k + 1)))^q$ tends to $(1/\varepsilon)$. So, the final probability is as follows:

\[ \varepsilon' \geq \frac{1}{(q_k + 1)}\varepsilon. \]

A forged AS could be generated in the following way by $C$:

\[ U = \sum_{i=1}^{n} U_i \]
\[ V = \sum_{i=1}^{n} h_iV_i. \]  \hspace{1cm} (12)

**Theorem 2:** The proposed CL-AS scheme is existentially unforgeable against the second kind of adversary $A_2$ assuming the CDHP is hard.

**Proof:** This security property also relies on the hardness of CDHP. Assuming the CDHP is intractable, we can prove that our scheme is secure in the similar way in Theorem 1. Thus, we omit the proof in detail.

**IV. LARGE-SCALE CONCURRENT DATA ANONYMOUS BATCH VERIFICATION SCHEME FOR MHCS**

Due to the unique security requirements of MHCS, we design an anonymous batch verification scheme for large-scale concurrent data. It can provide privacy-preserving batch verification of the uploaded health data in MHCS and achieve multiuser access authentication.
A. Scheme Description

The proposed scheme consists of five algorithms, such as: 1) initialization; 2) registration; 3) signing; 4) anonymous aggregation; and 5) batch verification. Here, we list some notations in Table I to facilitate our understanding. Then, we give the assumption of the time synchronization between the requested DC and MHCS participants. The proposed scheme is introduced as follows.

1) Initialization: MS establishes an enrollment system as follows.

   a) MS define $G_a$ as a additive group, $G_m$ as a multiplicative group, $q$ as the order, $P$ as the generator of $G_a$, $e: G_a \times G_a \rightarrow G_m$ as a bilinear map, $H_1 : \{0, 1\}^* \times G_a \rightarrow G_a$ and $H_2 : \{0, 1\}^* \times G_a \rightarrow \mathbb{Z}_q^*$ as two secure hash functions.

   b) Given $l$, MS selects its private key $s_{MS}$ randomly and calculates its public key $Q_{MS} = s_{MS}P$. Then, it opens the system parameters $\langle l, q, P, G_a, G_m, e, H_1, H_2, Q_{MS}\rangle$. We suppose that DC regards $\langle s_{DC}, Q_{DC}\rangle$ as its long-term key pair, where $Q_{DC} = s_{DC}P$.

2) Registration: A participant and the MS perform the following steps to access a DC as follows.

   a) The participant, marked as $C_i$, chooses a random number $s_{1i} \in \mathbb{Z}_q^*$ as the half private key, and it obtains $S_{2i}$ from MS who computes $S_{2i} = s_{MS}Q_{2i}$, where $Q_{2i} = H_1(id_i, Q_{1i})$ as the other half part private key. $C_i$ sets $\langle s_{1i}, S_{2i}\rangle$ as its private key. Then, it sends $\langle id_i, Q_{1i}\rangle$ to MS.

   b) Upon receiving $\langle id_i, Q_{1i}\rangle$, MS chooses a random number $a_i \in \mathbb{Z}_q^*$ and calculates

   $Q_{2i} = H_1(id_i, Q_{1i})$
   $S_{2i} = s_{MS}Q_{2i}$
   $\text{index}_{si} = a_iS_{2i}$
   $\text{index}_{vi} = a_iQ_{2i}$.

   Thus, MS stores serial number $sn_i = \langle id_i, Q_{1i}, Q_{2i}, \text{index}_{si}, \text{index}_{vi}\rangle$. Then, it sends $SN_i = \text{index}_{vi}$ and $\text{index}_{si}$ to the participant with $id_i$. All of the registration information should be transmitted via a secure channel.

3) Signing: $C_i$ chooses a random number $k_i \in \mathbb{Z}_q^*$ and a time stamp $t_i$, where $t_i$ is the system time to maintain the freshness of the message, and calculates

   $V_i = k_iQ_{1i}$
   $h_i = H_2(m_i||t_i, V_i)$
   $U_i = \text{index}_{si} + k_ih_i\{s_{1i}Q_{MS}\}$
   $SN'_i = E_{Q_{DC}}(SN_i||h_i||t_i)$.

Each required sensing data could be verified by

   $e(U_i, P) = e(\text{index}_{si} + h_iV_i, Q_{MS})$.

Table I

| Notations | Description |
|-----------|-------------|
| $G_a$     | a cyclic additive group of order $q$ |
| $G_m$     | a cyclic multiplicative group of order $q$ |
| $\sigma_i$| digital signature of the participant with $ID_i$ |
| $\sigma$  | aggregate signature |
| $Q_{DC}$  | DC’s public key |
| $Q_{MS}$  | MS’s public key |
| $\langle Q_{1i}, Q_{2i}\rangle$ | the public key of the participant with $ID_i$ |
| $l$       | system security parameter |
| $H_1(\cdot)$ | a hash function: $\{0, 1\}^* \times G_a \rightarrow G_a$ |
| $H_2(\cdot)$ | a hash function: $\{0, 1\}^* \times G_a \rightarrow \mathbb{Z}_q^*$ |
| $P$       | a generator of $G_a$ |
| $e(\cdot)$ | a bilinear map: $G_a \times G_a \rightarrow G_m$ |
| $m_i$     | healthcare data of the participant with $ID_i$ |
| $A_i$     | An adversary on type $i$ |
| $s_{DC}$  | DC’s private key |
| $s_{MS}$  | MS’s private key |
| $\langle s_{1i}, S_{2i}\rangle$ | the private key of the participant with $ID_i$ |
| $q$       | a large prime number |

4) Anonymous Aggregation: DC plays a role of the aggregator to merge all collected authentication information of different participants to a single verification message. Upon receiving $\langle U_i, V_i, m_i, SN'_i\rangle$, DC calculates $D_{Q_{DC}}(SN'_i) = SN_i||h_i||t_i$. For an aggregate set of $n$ participants $C_1, C_2, \ldots, C_n$ and a set of signatures $\langle U_i, h_iV_i\rangle$, when the time $T$ is up, DC aggregates all the received signatures as follows:

   $U = \sum_{i=1}^{n} U_i$
   $V = \sum_{i=1}^{n} h_iV_i$
   $\text{index}_v = \sum_{i=1}^{n} \text{index}_{vi}$.
Then, DC treats $\sigma = (U, V, \text{index}_v)$ on all health data $\langle m_1, m_2, \ldots, m_n \rangle$ as the aggregated authentication message.

5) **Batch Verification**: As illustrated in Fig. 3, DC verifies the validity of $e(U, P) = e(\text{index}_v + V, Q_{MS})$. If the equation holds, DC approves all health data uploaded by participants within the time slot $T$ as legal data. Otherwise, DC aborts this procedure. Here, DC can verify the validity of the equation as follows:

$$e(U, P) = e\left(\sum_{i=1}^{n} U_i, P\right)$$
$$= \prod_{i=1}^{n} e(U_i, P)$$
$$= \prod_{i=1}^{n} e(\text{index}_{si} + k_i h_i S_1 Q_{MS}, P)$$
$$= \prod_{i=1}^{n} e(\text{index}_{si}, P) e(k_i h_i S_1 Q_{MS}, P)$$
$$= e(\text{index}_v, Q_{MS}) e(V, Q_{MS})$$
$$= e(\text{index}_v + V, Q_{MS})$$.

**B. Security Analysis**

For convincing, we analyze the security of the large-scale concurrent data anonymous batch verification scheme in this part.

**Theorem 3**: The proposed scheme satisfies batch authentication, nonrepudiation, and anonymity.

**Proof**: We will give the proof as follows.

1) **Batch Authentication**: The proposed scheme is secure due to the intractability of the CDHP. So DC can authenticate the identities of MHCS participants by their signatures on health data. Meanwhile, it can aggregate all signatures from large-scale participants to a single verification message and verify the message by checking $e(U, P) = e(\text{index}_v + V, Q_{MS})$. Thus, our scheme can achieve anonymous batch verification.

2) **Nonrepudiation**: In our scheme, MHCS participants cannot deny that they have submitted the health data. DC can verify their signature via the corresponding public key. Then, MS can find serial number $s_n^i$ according to the public key and obtain the real identity of the participant.

3) **Anonymity**: In the phase of aggregate verification, due to the distribution of $S_{n}^i$ being random, DC cannot get the real identity of the MHCS participant from $S_{n}^i$. Therefore, even if the opponent has unlimited computing power, it is unable to guess the actual participant’s identity with the non-negligible advantage. Thus, the proposed scheme achieves anonymity.

**V. PERFORMANCE EVALUATION**

In this section, we evaluate the performance of the proposed scheme in two aspects, including computation overhead and storage overhead. Firstly, comparing our scheme with other three existing schemes, we assess the performance of the computation overhead in terms of the computation complexity and time overhead on signing, anonymous aggregation, and batch verification. Then, we analyze the storage overhead of the proposed scheme.

**A. Computation Overhead**

1) **Computation Complexity**: We select three existing schemes [23], [24], [29] to compare the computation complexity with our scheme. Due to the computation overhead is mostly caused by three basic cryptographic operations, so we mainly focus on the time consumption of these operations. Here, we only count on computation consumption, while the precomputation efforts are omitted. We define $P$ as a pairing operation, $S$ as a scalar multiplication in $G_a$ and $H$ as hash functions.

Table II shows the complexity comparison between different schemes. We find that, in the signing stage, our scheme only requires $nH + 2nS$ operations, while the schemes in [23], [24], and [29] require $4nH + 3nS$, $nH + 3nS$, and $nH + 3nS$, respectively. In the verification stage, our scheme
needs \( nH + 2nP + nS \) operations, rather than \( 5nH + 4nP + 2nS \) in [23], \( 2nH + 3nP + 3nS \) in [24], and \( 2nH + 3nP + 2nS \) in [29]. In addition, in aggregation stage, only our scheme needs \( 2nS \) scalar multiplications, but it only requires two pairing operations in aggregate verification stage. Hence, compared with the schemes in [23], [24], and [29], our scheme has the least total computation overhead in all four stages: 1) signing; 2) verification; 3) aggregation; and 5) aggregate verification.

Meanwhile, Fig. 4 shows the comparison of computation cost between different schemes. And we also find that our scheme has lowest computation complexity than the other schemes [23], [24], [29], with the increasing of the number of participants. As a whole, our scheme achieve the best performance of the computation complexity.

2) Time Overhead: In order to evaluate and test the performance of time overhead on our scheme, we compare our scheme with other three schemes [23], [24], [29]. For quantitative analysis, we first construct a simulation platform to measure the time overhead. The simulation environment is Ubuntu 16.04 LTS over an Intel Core i5 1.6 GHz \( \times \) 2 processor. We choose type A curve in the pairing-based cryptography library \( y^2 = x^3 + x \), to complete the simulation. Here, we assume that \( n \) participants try to upload their health data in a certain time slot \( T \).

Next, we view aggregation as the integration of aggregation and aggregate verification. Then, we record the start time from the beginning of the signing stage to simulate these schemes. Therefore, we can obtain the time overhead of different schemes as shown in Fig. 5. Compared with the schemes in [23], [24], and [29], the proposed scheme can save 61.33%, 42.16%, 41.99% running time, respectively, when the number of participants is 50.

B. Storage Overhead

In the proposed scheme, the DC needs to store all collected authentication information of different participants continuously until batch verification is done. Meanwhile, as the aggregator, DC can, in real time, merge the collected authentication information into a single verification message, due to the advantage of (16). When time \( T \) is up, DC can verify these data in batch. Therefore, the storage overhead of the proposed scheme can be reduced differently according to the number of MHCS participants. For quantitative analysis, we adopt the type A curve with base field size of 512 bits, the cyclic group order of 160 bits, and the embedding degree 2. So, \( U_i = 512 \) bits, \( V_i = 512 \) bits, and \( SN'_i = 160 \) bits. Here, we assume that the size of health data \( m_i \) is 160 bits as [30].

As mentioned before, the verification information of the participant \( i \) is \( \langle U_i, V_i, m_i, SN_i' \rangle \). Therefore, the corresponding storage overhead of the authentication data is \( SO_i = 512 + 512 + 160 + 160 = 1344 \) bits. Here, \( SO_i \) denotes the storage overhead of the participant \( i \). When the time \( T \) is up, the total storage overhead of the \( n \) participants in this time slot is \( SO = 512 + 512 + 160n + 160 = 160n + 1184 \) bits. Otherwise, if the verification stage does not utilize the scheme in the
batch mode, the total storage overhead of the $n$ participants is $SO' = 1344n$ bits. For better demonstration, we depict the storage overhead on the aforementioned two cases in Fig. 6. Then, we can conclude that the storage overhead is greatly reduced in the batch mode.

For all above, the proposed scheme achieves a better performance in terms of computation overhead and storage overhead. It is efficient and suitable for MHCS.

VI. CONCLUSION

In this paper, based on an improved CL-AS algorithm, we design an anonymous batch verification scheme for large-scale concurrent data in MHCS scenarios. It meets the EUF-CMA security in the random oracle model based on the intractability of the CDHP. And it can achieve three properties including batch authentication, nonrepudiation, and anonymity. Moreover, our scheme can be deployed in MHCS system to offer batch health data authentication and privacy preservation simultaneously. Through quantitative performance analysis, we find that the proposed scheme achieves lower computation overhead and provides better efficiency compared with the existing schemes, and its storage overhead is also reduced greatly. The proposed scheme is an efficient solution for the MHCS systems.

REFERENCES

[1] F. Xia, L. T. Yang, L. Wang, and A. Vinel, “Internet of Things,” Int. J. Commun. Syst., vol. 25, no. 9, pp. 1101–1102, 2012.
[2] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: Current state and future challenges,” IEEE Commun. Mag., vol. 49, no. 11, pp. 1–6, Nov. 2011.
[3] B. Guo, Z. Yu, X. Zhou, and D. Zhang, “From participatory sensing to mobile crowd sensing,” in Proc. IEEE Int. Conf. Pers. Ubiquit. Comput. (Pervasive Comput. Workshop) (PERCOM Workshops), Budapest, Hungary, 2014, pp. 593–598.
[4] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, “A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks,” IEEE Trans. Wireless Commun., vol. 8, no. 3, pp. 1223–1229, Mar. 2009.
[5] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, “An effective key management scheme for heterogeneous sensor networks,” Ad Hoc Netw., vol. 5, no. 1, pp. 24–34, 2007.
[6] L. Wu, X. Du, and J. Wu, “MobiFish: A lightweight anti-phishing scheme for mobile phones,” in Proc. IEEE 23rd Int. Conf. Comput. Commun. Netw. (ICCCN), Shanghai, China, 2014, pp. 1–8.
[7] Y. Cheng, X. Fu, X. Du, B. Luo, and M. Guizani, “A lightweight live memory forensic approach based on hardware virtualization,” Inf. Sci., vol. 379, pp. 23–41, Feb. 2017.
[8] L. Wu, X. Du, and J. Wu, “Effective defense schemes for phishing attacks on mobile computing platforms,” IEEE Trans. Veh. Technol., vol. 65, no. 8, pp. 6678–6691, Aug. 2016.
[9] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, “Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing,” in Proc. IEEE INFOCOM, San Francisco, CA, USA, 2016, pp. 1–9.
[10] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, “ARTSense: Anonymous reputation and trust in participatory sensing,” in Proc. IEEE INFOCOM, Turin, Italy, 2013, pp. 2517–2525.
[11] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, “TPF: A trajectory privacy-preserving framework for participatory sensing,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 6, pp. 874–887, Jun. 2013.
[12] I. Boutis and V. Kalogeraki, “Privacy preservation for participatory sensing data,” in Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom), San Diego, CA, USA, 2013, pp. 103–113.
[13] F. Qiu, F. Wu, and G. Chen, “SLICER: A slicing-based K-anonymous privacy preserving scheme for participatory sensing,” in Proc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst. (MASS), Hangzhou, China, 2013, pp. 113–121.
[14] B. Riedl, V. Grascher, S. Fenz, and T. Neubauer, “Pseudonymization for improving the privacy in E-health applications,” in Proc. IEEE 41st Ann. Hawaii Int. Conf. Syst. Sci., Waikoloa, HI, USA, 2008, p. 255.
[15] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, “Certificateless remote anonymous authentication schemes for wireless body area networks,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 332–342, Feb. 2014.
[16] H. Zhu, L. Gao, and H. Li, “Secure and privacy-preserving body sensor data collection and query service,” Sensors, vol. 16, no. 2, p. 179, 2016.
[17] X. Li, Q. Wen, W. Li, H. Zhang, and Z. Jin, “Secure privacy-preserving biometric authentication scheme for telecare medicine information systems,” J. Med. Syst., vol. 38, no. 11, p. 139, 2014.
[18] C.-L. Chen, T.-T. Yang, M.-L. Chiang, and T.-F. Shih, “A privacy authen- tication scheme based on cloud for medical environment,” J. Med. Syst., vol. 38, no. 11, p. 143, 2014.
[19] N. D. Lane et al., “A survey of mobile phone sensing,” IEEE Commun. Mag., vol. 48, no. 9, pp. 140–150, Sep. 2010.
[20] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and verifi- ably encrypted signatures from bilinear maps,” in Proc. EUROCRYPT, 2003, pp. 416–432.
[21] R. Castro and R. Dahab, “Efficient certificateless signatures suitable for aggregation,” IACR Cryptol. ePrint Archive, Rep. 2007/454, 2007, p. 454, [Online]. Available: https://eprint.iacr.org/2007/454
[22] Z. Gong, Y. Long, X. Hong, and K. Chen, “Two certificateless aggregate signatures from bilinear maps,” in Proc. IEEE 8th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel Distrib. Comput. (SNPD), vol. 3, Qingdao, China, 2007, pp. 188–193.
[23] H. Tu, D. He, and B. Huang, “Reattack of a certificateless aggregate signature scheme with constant pairing computations,” Sci. World J., vol. 2014, Mar. 2014, Art. no. 343715.
[24] A. K. Malhi and S. Batra, “An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks,” Discr. Math. Theor. Comput. Sci., vol. 17, no. 1, pp. 317–338, 2015.
[25] L. Zhang and F. Zhang, “A new certificateless aggregate signature scheme,” Comput. Commun., vol. 32, no. 6, pp. 1079–1085, 2009.
[26] H. Xiong, Q. Wu, and Z. Chen, “Strong security enabled certificateless aggregate signatures applicable to mobile computation,” in Proc. IEEE 3rd Int. Conf. Intell. Netw. Collaborat. Syst. (INCoS), Fukuoka, Japan, 2011, pp. 92–99.
[27] X. Huang, Y. Mu, D. Wang, and W. Wu, “Certificateless aggregate signature with constant pairing computations,” Inf. Forensics Security, vol. 8, no. 6, pp. 874–887, Apr. 2012.
[28] A. Shamir, “Identity-based cryptosystems and signature schemes,” in Proc. Workshop Theory Appl. Cryptograph. Techn., Santa Barbara, CA, USA, 1984, pp. 47–53.
[29] S. Gao, Z. Yuan, Z. Chen, and F. Li, “An efficient certificateless aggregate signature with constant pairing computations,” Inf. Sci., vol. 219, pp. 225–235, Jan. 2013.
[30] K. Ren, W. Lou, K. Zeng, and P. J. Moran, “On broadcast authentication in wireless sensor networks,” IEEE Trans. Wireless Commun., vol. 6, no. 11, pp. 4136–4144, Nov. 2007.

Fig. 6. Storage overhead of our scheme.
Jingwei Liu (M’11) received the B.S. degree in applied mathematics and M.S. and Ph.D. degrees in communication and information systems from Xidian University, Xi’an, China, in 2001, 2004, and 2007, respectively. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. He has authored or co-authored over 40 papers in journals and conference proceedings. His current research interests include information security, network security, and cryptography. Dr. Liu is a member of the Chinese Association for Cryptologic Research.

Huijuan Cao is currently pursuing the M.S. degree in communication and information systems at Xidian University, Xi’an, China. Her current research interests include network security and privacy preserving.

Qingqing Li is currently pursuing the M.S. degree in communication and information systems at Xidian University. Her current research interests include IoT security and privacy preserving.

Fanghui Cai is currently pursuing the M.S. degree in electronics and communication engineering at Xidian University, Xi’an, China. Her current research interests include authentication protocols and privacy preserving.

Xiaojiang Du (M’03–SM’09) received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2002 and 2003, respectively. He is currently a Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. He has authored or co-authored over 270 journal and conference papers. His current research interests include security, systems, wireless networks, and computer networks. Dr. Du serves on the Editorial Boards of three international journals.

Mohsen Guizani (S’85–M’89–SM’99–F’09) received the bachelor’s degree (with Distinction), master’s degree in electrical engineering, and master’s and doctorate degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor and the Electrical and Computer Engineering Department Chair with the University of Idaho, Moscow, ID, USA. His current research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. Dr. Guizani currently serves on the Editorial Boards of several international technical journals and is the founder and the Editor-in-Chief of Wireless Communications and Mobile Computing (Wiley). He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the Transmission, Access, and Optical Systems Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005.