An Authentic and Privacy-Preserving Scheme Towards E-Health Data Transmission Service

Qing Fan, Yumeng Xie, Chuan Zhang, *Member, IEEE*, Ximeng Liu, *Senior Member, IEEE*, and Liehuang Zhu, *Senior Member, IEEE*

***Abstract*—The e-health system enables online healthcare by supporting health data transmission services on medical platforms. Considering the frequent privacy breaches in e-health systems and the issuance of relevant regulations, it is important to ensure the authenticity and privacy of health data. Existing e-health systems either fail to provide data authenticity or neglect privacy protection after patients leave the system. In this article, we put forward a secure and efficient e-health system for data transmission, named PPED, to solve this dilemma. In PPED, we explore a regular signature and a forward-secure signature, which guarantee data authenticity and give the signature a valid period. Then, a specific epochal signature scheme is designed by combining two signature schemes with the time-lock puzzle. Since expired epochal signatures are forgeable, patients after leaving the e-health system can forge expired signatures to deny their relationship with the signed data, thus achieving privacy protection. Detailed security analysis demonstrates the PPED realizes data authenticity and user privacy. Extensive experiments evaluate our system and the results show it is practical in terms of running time.**

***Index Terms*—E-health system, epochal signature, data authenticity, privacy-preserving.**

I. Introduction

Electronic health (e-health) is an emerging field in the intersection of medical informatics, public health and business, including digital hospital, telemedicine services, portable communication platforms and so on [1], [2]. Such systems transmit patients' personal health information to healthcare providers who reply with suggestions. The effective data exchange between patients and healthcare providers enables more efficient and flexible healthcare services.

Despite the convenience of e-health systems, data breaches frequently happen. American NextGen Healthcare system [3] and HCA Healthcare system [4] both suffered severe personal data leakage, potentially leading to identity theft and illicit data trade. Similarly, a popular e-health app in China Dr Ding Xiang [5] faced deactivation due to privacy breaches, and the French medical software provider Dedalus Biologie [6] was fined 1.5 million Euros due to non-compliant use of data. These incidents result in increasing concerns about health data privacy issues. Hence, many official regulations of privacy preservation were published, like General Data Protection Regulation (GDPR) [7] and Health Insurance Portability and Accountability Act (HIPAA) [8]. The GDPR requires the use of appropriate technical measures to ensure the integrity and confidentiality of personal data[1] and HIPPA further clarifies that individuals' health information should be properly protected.[2] Furthermore, many international standards [9], [10], [11], [12] indicate that e-health systems should provide unified information security and privacy protection services through encryption, authentication and authorisation.

To protect health data and comply with legal requirements, it is necessary to implement protection mechanisms to defend against deliberate or unintentional leakage and misuse of health data. Ideally, a secure e-health system only utilizes transmitted personal health data for medical cures, as well as protects health data even if patients leave the system. However, realizing such a privacy-preserving e-health system is challenging considering the requirements of reliability, security, and practicality. Specifically, the reliability of healthcare systems usually depends on whether the authentic medical data can be transmitted to healthcare providers. Security means the patient's personal data should be protected by effective methods even after patients leave the system. Besides, the interaction of patients and doctors is an ideal function in a practical medical platform, which directly influences communication efficiency, medical suggestion accuracy and user experience perception.

Although plenty of research is devoted to privacy-preserving e-health system design, they cannot meet reliability, security, and practicality at the same time. In order to enhance the reliability of e-health systems, patients are usually required to transmit unique credentials along with their personal information to support

Received 31 December 2022; revised 25 October 2023; accepted 22 August 2024. Date of publication 28 August 2024; date of current version 9 October 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62302037, Grant 62202051, and Grant 62102027, in part by China Postdoctoral Science Foundation Grant 2021M700435, Grant 2021TQ0042, and Grant 2021TQ0041, in part by BIT Research and Innovation Promoting Project under Grant 2022YCXZ031, in part by Shandong Provincial Key Research and Development Program under Grant 2021CXGC010106, and in part by the Beijing Institute of Technology Research Fund Program for Young Scholars, and the Young Elite Scientists Sponsorship Program by CAST under Grant 2023QNRC001. *(Qing Fan and Yumeng Xie are contributed equally to this work.) (Corresponding author: Chuan Zhang.)*

Qing Fan, Chuan Zhang, and Liehuang Zhu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: qingf@bit.edu.cn; chuanz@bit.edu.cn; liehuangz@bit.edu.cn).

Yumeng Xie is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: yumengx@bit.edu.cn).

Ximeng Liu is with the College of Mathematics and Computer Science, Fujian Provincial Key Laboratory of Information Security of Network Systems, Fuzhou University, Fuzhou 350025, China (e-mail: snbnix@gmail.com).

Digital Object Identifier 10.1109/TSC.2024.3451145

[1] See GDPR Chapter 2. Principles Art.5, https://gdpr-info.eu/art-5-gdpr/

[2] See Summary of the HIPAA Privacy Rule, https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

TABLE I
COMPARISON WITH PRIOR WORKS

Scheme	[3]	[4]	[5]	[6]	[13]	[14]	[16]	[20]	[21]	[22]	[26]	[27]	Ours
Data authenticity	✔	✔	✔	✔	✔	✔	✔	✘	✘	✘	✔	✘	✔
User privacy	✘	✘	✘	✘	✘	✘	✘	✔	✔	✔	✘	✘	✔
Interactive session	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘	✔	✔
Key update	✘	✘	✘	✘	✔	✘	✘	✘	✘	✘	✘	✘	✔
Key extract	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔
Deniability	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔

✔: Property is supported.
✘: Property is not provided.

doctors in confirming data authenticity. Unfortunately, these credentials are unique, undeniable and often retained within the system, rendering patient privacy vulnerable once they exit the system. Despite previous efforts to improve reliability in ref. [13], [14], [15], [16], [17], [18], the security of patient information after leaving the system remains inadequately protected. In addition, ref. [19], [20], [21], [22], [23], [24] elaborated some security-improved e-health schemes by using cryptographic technologies (e.g., dynamic searchable encryption and attribute-based encryption). However, these schemes did not consider data authenticity and were not equipped with interaction functions. Another line of work related to e-health systems is to achieve scheme practicality. Deebak et al.'s [25] enables patients to directly interact with healthcare providers. Nevertheless, due to the usage of digital signatures, the undeniability may leak the privacy of medical data left in the system. Table I compares our work with real-world e-health systems and systems designed in academic research. Based on the comparison results, we can see a practical e-health system that provides both reliability and privacy preservation still deserves to be investigated.

In this paper, we answer this open problem by proposing a Privacy-Preserving scheme for E-health Data Transmission system, named PPED. The construction of PPED is based on the following idea: By leveraging two newly designed signature algorithms and the time-lock puzzle, we design a specific epochal signature scheme and utmost utilize its unforgeability and deniability to construct a practical e-health system with reliability and higher security. Our identity-based epochal signature scheme establishes a connection between a user's identity and his key pairs. When transmitting the health data, a patient applies the epochal signature scheme to sign the data. Since epochal signatures are time-limited and regularly updated, a valid signature with unforgeability guarantees data authenticity, while an expired signature can be easily forged, thus providing deniability. This deniability allows patients to deny the holding of old health data, thereby ensuring patient privacy protection. Overall, our scheme does not change the nature of the digital signature and provides a good guarantee of data authenticity and integrity. Furthermore, some auxiliary information about the user is published with every evolution of the key, which allows anyone to forge valid signatures of transmitted health data in expired sessions, thus the health information of patients is protected in the opposite view. In summary, the contributions of this paper are summarized in the following:

- We formulate the problem of a privacy-preserving scheme for e-health systems with data authenticity, identify a concrete system model, and establish a well-defined threat model based on adversaries' different attack objectives.
- We propose a new privacy-preserving scheme based on the epochal signature, realizing precise health data transmission and patients' privacy protection. Particularly, we first show a newly designed identity-based signature. Then, a forward-secure pseudorandom generator is innovatively used to transform the regular signature into a forward-secure one. Next, we apply these two signature schemes into the framework of epochal signatures to get the final scheme, which is the core technique of our PPED system.
- Detailed security analysis demonstrates that the PPED is equipped with completeness, unforgeability and deniability, preserving both data authenticity and user privacy. Additionally, the performance evaluation of our scheme is presented. We conduct experiments of six algorithms and analyze running efficiency with respect to the periods maximum E and the valid periods maximum V. Computation cost and communication cost analysis demonstrate our scheme is practical.

Organization: The remainder of this paper is organized as follows. In Section II, we introduce system model, threat model and design goals. In Section III, we give the preliminary including cryptographic primitives used to construct PPED system, a new identity-based signature scheme and a new forward-secure signature scheme. After that, we elaborate on details of our proposed scheme in Section IV and analyze its security in Section V. In Section VI, we conduct experiments to evaluate the performance of our scheme. Finally, we give a brief introduction of related work in Section VII and conclude our work in Section VIII.

II. PROBLEM STATEMENT

In this section, we give an overview of our PPED system including model delineation, threat model and design goals.

A. System Model

As shown in the Fig. 1, the architecture of PPED consists of four entities: the key generation center (KGC), the medical service platform (MSP), doctors and patients.

- *KGC* is a trusted third party, e.g., the government agency, responsible for the system initialization. KGC is also in

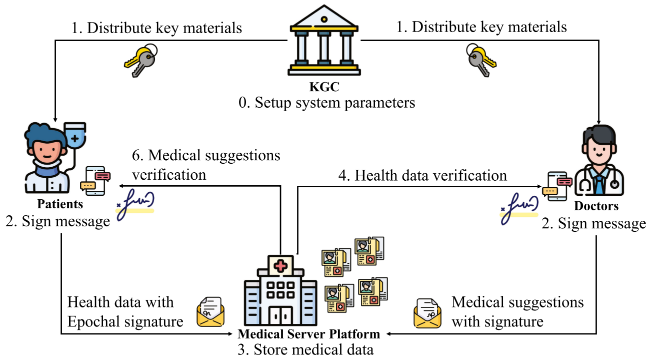

1. Distribute key materials
1. Distribute key materials
KGC
0. Setup system parameters
Patients
2. Sign message
6. Medical suggestions verification
4. Health data verification
Doctors
2. Sign message
Health data with Epochal signature
Medical suggestions with signature
Medical Server Platform
3. Store medical data

Fig. 1. PPED system model.

charge of generating key materials for patients and doctors. After each epochal signature, it updates the patients' keys.
- *MSP* is a medical service platform supporting the data exchange between patients and doctors. Patients transmit health data to MSP and doctors read patient health conditions from MSP. Then, doctors give treatment suggestions and patients read them through MSP. After the interaction period, the MSP preserves all transmitted medical data.
- *Patients* are users of MSP who sign their personal health data using the epochal signature, and submit them to MSP. Then patients get treatment suggestions with regular signatures by doctors from MSP and perform verification.
- *Doctors* are primary service providers, who read patient health data from the MSP and verify the data reliability. Then doctors give and sign treatment plans on the MSP.

In PPED, KGC first sets a series of parameters to initialize the system (step 0). Then, patients and doctors send their identities to KGC to register, and KGC generates personal key pairs for them (step 1). The patient who wants to connect with the doctor sends his health data along with a signature to MSP (step 2). When receiving the data from MSP, the doctor verifies data authenticity (step 4) and returns his medical treatment suggestions and the corresponding signature (step 5) if verification succeeds. MSP transmits information from the doctor to the patient, and then the patient checks the received data. During one data exchange, MSP stores the whole medical data transmitted between the doctor and patient (step 3).

B. Threat Model

In our model, KGC completes the secret key transmission by a secure channel. The MSP provides different channels in different periods. *During the data transmitting period*, the MSP opens a confidential channel for health data exchange, but it cannot avoid malicious data putting. *After interaction*, the MSP closes the channel and may share medical data with some commercial businesses. Doctors are honest and believable individuals, who honestly perform verification of transmitted data and will not leak patients' privacy. Based on the information that adversaries may derive, we consider two types of adversaries.

- *Type-I attack:* Adversaries can choose a number of messages and generate their corresponding signatures. This attack corresponds to the notion of unforgeability of epochal signature, which is *Epochal Existential UnForgeability under Chosen Message Attacks* or **EEUF-CMA**.
- *Type-II attack:* If the doctor or MSP is compromised, the message, signature and public information of the patient will probably be disclosed after the data exchange process. In this case, malicious parties can know the authenticity and validity of a signature. This attack corresponds to the *Deniability* of the epochal signature.

It is obvious that these two forms of attack are proposed for two different needs in e-health systems.

C. Design Goals

According to system model and threat models, our system should fulfil the following goals:
- *Completeness:* PPED is supposed to be complete, which means the honestly generated signature should pass the verification successfully. That is to say, when receiving the real health data and its corresponding honest signature, a doctor will not mistakenly reject the data.
- *Unforgeability:* Every signature created in our system must be unforgeable. In other words, nobody can forge signatures of doctors and patients to spread false information about patients, or upload forged signatures to the platform to save false medical content.
- *Deniability:* If an expired signature of the patient is gotten by a malicious party, a simulated signature can be created by utilizing the public key and auxiliary information published with every key evolution. Deniability guarantees that a simulated signature is indistinguishable from the real one.

III. Preliminary

In this section, we first introduce some basic cryptographic knowledge. Then, we propose an extensive identity-based signature and a new forward-secure signature, which are solid foundations of the epochal signature. For simplicity, Table II lists the notations used in this work.

A. Basic Knowledge

Bilinear Pairing: $\mathbb{G}_1$ is an additive cyclic group generated by P, whose order is a prime number q, and $\mathbb{G}_2$ is a multiplicative cyclic group with the same order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties:
- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in Z_q$.
- Non-degeneracy: $e(P, Q) \neq 1$ for all $P, Q \in \mathbb{G}_1$.
- Computablility: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

k-CAA problem: In the additive group $(\mathbb{G}_1, +)$, for an integer k, $x \in Z_q$, $P \in \mathbb{G}_1$, given $\{P, Q = xP, h_1, \ldots, h_k \in Z_q, \frac{1}{h_1+x}P, \ldots, \frac{1}{h_k+x}P\}$, to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \ldots, h_k\}$. We say that k-CAA is (t, ϵ)-hard for all t-time

TABLE II
LIST OF NOTATIONS

Notation	Description
KGC	Key Generation Center
$\overline{\Sigma}$	Regular signature
$\hat{\Sigma}$	Forward-secure signature
Σ	Epochal signature
Z_r^*	non-zero element in finite filed Z_r
H_1'	$\{0,1\}^* \rightarrow Z_r^*$
H_2'	$\mathbb{G} \times \{0,1\}^* \rightarrow Z_r^*$
H_1	$\{0,1\}^\lambda \times \{0,1\}^* \rightarrow \{0,1\}^\lambda$
(pk, sk)	System secret key pairs for epochal signature
$\hat{pk}$	Public key of $\hat{\Sigma}$
$\hat{sk}_i$	Private key for period i of $\hat{\Sigma}$
$\overline{pk}_i$	Public key for period i of $\overline{\Sigma}$
$\overline{sk}_i$	Private key for period i of $\overline{\Sigma}$
r_i	Pseudo-random string for period i
$\boldsymbol{\hat{sk}}$	All private keys for $\hat{\Sigma}$
$\boldsymbol{sk_r}$	All pseudo-random strings
$kinfo_i$	Key information for period i
$pinfo_i$	Public information published in period i
E	The maximum number of periods
V	The valid period length of epochal signatures
Δt	The time length of one period

adversaries $\mathcal{A}$, we have

$$\mathbf{Adv}_{\mathcal{A}}^{k-CAA} = \Pr\left[\mathcal{A}\left(P, Q = xP, \frac{1}{h_1 + x}P, \ldots, \frac{1}{h_k + x}P\right) = \frac{1}{h + x}P \middle| x \in Z_q, P \in \mathbb{G}_1, h_1, \ldots, h_k \in Z_q, h \notin \{h_1, \ldots, h_k\}\right].$$

Forward-secure pseudorandom generator: A FWPRG [28] is a generator which on input k_{t-1} at each time period t produces a pair of pseudorandom values k_t and r_t. It suffices that each of k_t and r_t are individually pseudorandom and r_t is indistinguishable from a truly random sequence.

The Time-Lock (TL) Puzzle: Time-lock puzzles [29] enable anyone to recover the plaintext after performing a certain amount of sequential decryption computation. A time-puzzle TL consists of two algorithms:

- $TL.lock(1^\lambda, t, m) \rightarrow c$: take the security parameter 1^λ, the time period of t and message m, and return a ciphertext c;
- $TL.unlock(n, T, c) \rightarrow m$: take the ciphertext c, the time T and number n as input, return message m.

Digital Signature: Digital signature schemes provide a cryptographic mechanism that allows individuals to digitally sign documents, messages, or data, thereby providing a means to validate the origin and integrity of the information.

B. The Proposed Identity-Based Signature Scheme

We design a new identity-based signature that utilizes a user's ID to generate key pairs. Hence, in an e-health system, identity-based signature forms can better verify the authenticity of health data and its relationship to its owner. Our identity-based signature scheme ($\overline{\Sigma}$) inspired by **MB-IBS** in [30], where the secret master key is transformed from a uni-variate polynomial to a binary polynomial by the simple addition operation. Our scheme is parameterized by a cyclic group $\mathbb{G} = \langle P \rangle$ of prime order r, another group generator Q, a finite field Z_r and two hash functions H_1' and H_2'. This scheme consists of four algorithms:

- *Setup:* The KGC generates a random secret binary polynomial $s(x, y) = \sum_{i=0}^{d} s_i(x^i + y^i) \in Z_r[x, y]$ which is its master secret key. Then the KGC publishes the points P, Q, $g = e(P, Q)$ and s_iP for $i = 0, \ldots, d$.
- *KeyGen:* Given a user's identity, KGC computes $u = H_1'(ID)$ and sets $y = 1$ to get user's private key $Q_{su} = s(u, 1)^{-1}Q$. The corresponding public key of the user can be computed from u and s_iP as $P_u = \sum_{i=0}^{d}(u^i s_i P + s_i P) = s(u, 1)P$.
- *Sign:* Given a message $m \in \{0, 1\}^*$, the algorithm generates a random integer $\omega \in Z_r^*$ and computes: $R = \omega P_u$, $h = H_2'(R, m)$, $S = (\omega + h)^{-1}Q_{su}$. The signature attached to m is the pair (R, S).
- *Verify:* Given m and (R, S), the algorithm computes $h = H_2'(R, m)$ and checks whether $e(R + hP_u, S) = g$.

C. The Proposed Forward-Secure Signature Scheme

In this part, we propose an ID-based signature scheme with forward security ($\hat{\Sigma}$). A forward-secure signature scheme updates the key over time, which can resist the attack towards previously signed messages, thereby the previous health data in the e-health system can be protected.

This scheme comes from the regular identity-based signature scheme in Section III-B through adopting the transformation technique proposed in [28] and utilizing a forward-secure pseudorandom generator (FWPRG) mentioned above. The scheme $\hat{\Sigma}$ is a tuple of five algorithms:

- *Setup:* The KGC generates a random secret binary polynomial $s(x, y) = \sum_{i=0}^{d} s_i(x^i + y^i) \in Z_r[x, y]$ which is its master secret key. Then the KGC publishes the points P, Q, $g = e(P, Q)$ and s_iP for $i = 0, \ldots, d$.
- *KeyGen:* The process is performed by KGC as follows.
  1) Compute the $u = H_1'(ID)$, where ID is the user's identity.
  2) Compute the signer's initial private key $sk_0 = s(u, 1)^{-1}Q$ and the base public key $pk_0 = \sum_{i=0}^{d}(u^i + 1)s_iP = s(u, 1)P$.
  3) Choose a random seed k_0 for FWPRG.
     For $t = 1$ to T do

$$(k_t, l_t) \leftarrow \text{FWPRG}(k_{t-1}),$$

$$\hat{sk}_t \leftarrow s(u, l_t)^{-1}Q, \hat{pk}_t \leftarrow s(u, l_t)P,$$

$$m_t \leftarrow (pk_0, t, \hat{pk}_t),$$

$$\sigma_t \leftarrow \overline{\Sigma}.\text{Sign}(sk_0, m_t):$$

$$R_t = \omega_t pk_0, \text{where } \omega_t \xleftarrow{\$} Z_r^*$$

$$h_t = H_2'(R_t, m_t)$$

$$S_t = (\omega_t + h_t)^{-1} sk_0$$

$$\sigma_t = (R_t, S_t)$$

$$C_t \leftarrow (pk_0, t, \hat{pk}_t, \sigma_t)$$

   4) Erase sk_0 and $k_t, l_t, \hat{sk}_t$ for $t = 1, \ldots, T$ and secretly store k_0.
   5) Store $C_t, t = 1, \ldots, T$ and publish the pk_0.
- *Update:* To update the secret key, the KGC:
   1) $(k_t, l_t) \leftarrow \text{FWPRG}(k_{t-1})$.
   2) $\hat{sk}_t = s(u, l_t)^{-1}Q, \hat{pk}_t = s(u, l_t)P$.
   3) Retrieve C_t and verify that values pk_0 and t in it are correct and also check that the public key $\hat{pk}_t$ in it equals the public key generated in the previous step. If any of these checks fail, abort.
   4) Store k_t and $\hat{sk}_t$ secretly and erase k_{t-1}.
- *Sign:* To sign message $M \in \{0, 1\}^*$, the user:
   1) Retrieve current values of C_t and $\hat{sk}_t$.
   2) $\sigma \leftarrow \overline{\Sigma}.\text{Sign}(\hat{sk}_t, M)$:
      Generate a random integer $\omega \in Z_r^*$.
      Compute $R = \omega\hat{pk}_t$, $h = H_2'(R, M)$, $S = (\omega + h)^{-1}\hat{sk}_t$.
      The signature attached to M is the pair $\sigma = (R, S)$.
   3) Outputs the signature pair $s = (C_t, \sigma)$.
- *Verify:* Given the base public key pk_0, a message M, a time period t, and a signature string s, the verifier:
   1) Parse s into values C_t' and σ'.
   2) Parse C_t' to get the values $(pk_0', t', \hat{pk}_t', \sigma_t')$.
   3) Check that $pk_0' = pk_0$ and $t' = t$.
   4) Verify that $\overline{\Sigma}.Verify((pk_0, t, \hat{pk}_t'), \sigma_t')$, namely checks that $e(R_t' + h_t' pk_0, \sigma_t') = g$.
   5) Verify that $\overline{\Sigma}.Verify(\hat{pk}_t', \sigma')$, namely checks that $e(R' + h'\hat{pk}_t', S') = g$.
      If *all* checks succeed output VALID, otherwise output FALL.

IV. Proposed PPED Scheme

This section includes two main parts. The first part gives the succinct profile of the PPED scheme and outlines its main idea. The second part presents detailed five phases of PPED.

A. Scheme Overview

In our construction, KGC first initializes the system to set a series of parameters. Before activating the system, all users including patients and doctors register in KGC to get their own key pairs. In specific, patients get keys from the key generation of epochal signature, and doctors' keys from that of a regular identity-based signature. Additionally, doctors' keys remain unchanged while patients' keys are evolved with every epoch. After getting keys and epochal public information, a patient can sign the health data and produce epochal signatures. A doctor can receive the patient's condition from the medical service platform and verify message authenticity. Then the doctor signs his treatment advice and generates regular signatures, which are transmitted to the patient to be verified. After the data exchange, the MSP stores whole medical data, and anyone can simulate valid signatures of health data in expired sessions.

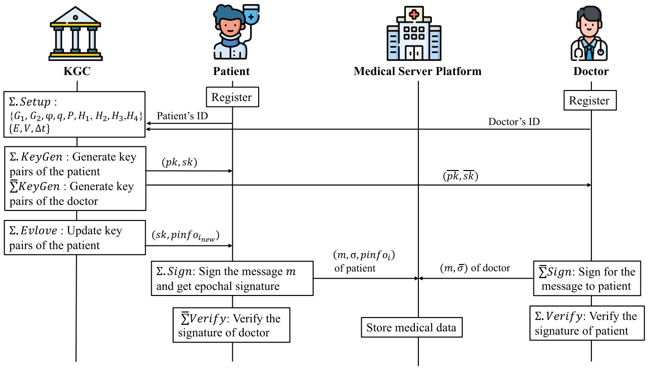

KGC
Patient
Medical Server Platform
Doctor
Register
Register
$\Sigma.Setup$:
$\{G_1, G_2, \varphi, q, P, H_1, H_2, H_3, H_4\}$
$\{E, V, \Delta t\}$
Patient's ID
Doctor's ID
$\Sigma.KeyGen$: Generate key pairs of the patient
$\overline{\Sigma}KeyGen$: Generate key pairs of the doctor
(pk, sk)
$(\overline{pk}, \overline{sk})$
$\Sigma.Evlove$: Update key pairs of the patient
$(sk, pinfo_{i_{new}})$
$\Sigma.Sign$: Sign the message m and get epochal signature
$(m, \sigma, pinfo_i)$ of patient
$(m, \overline{\sigma})$ of doctor
$\overline{\Sigma}Sign$: Sign for the message to patient
$\overline{\Sigma}Verify$: Verify the signature of doctor
Store medical data
$\Sigma.Verify$: Verify the signature of patient

Fig. 2. Interactive process among entities in PPED.

Algorithm 1: Σ.Setup.

```
Input: $1^\lambda$
Output: $s(x, y), P, Q, g, s_iP$
  1: Generate $s(x, y) = \sum_{i=0}^{d} s_i(x^i + y^i) \in Z_r[x, y]$
  2: Compute $g = e(P, Q)$ and $s_iP$ for $i = 0, \ldots, d$
```

At a high level, the data transmission process utilizes an identity-based epochal signature, which is a tuple of seven algorithms: Σ.Setup, Σ.KeyGen, Σ.Evolve, Σ.Sign, Σ.Verify, Σ.KExtract, and Σ.Simulate. In the specific construction, we use a regular signature scheme $\overline{\Sigma}$ (proposed in Section III-B) and a forward-secure one $\hat{\Sigma}$ (proposed in Section III-C) to establish the whole epochal signature scheme. Moreover, we innovate the structure of key pairs and introduce the concept of public information to satisfy the goals set in Section II, rather than change the nature of signatures. Since a digital signature is usually generated by the signer's private key, to simulate an expired signature, some auxiliary information relating to keys is published. Hence, the key idea is publishing some information with every key-evolving period and using this public epoch information to forge expired signatures. An attacker who obtains a real signature cannot distinguish simulated signatures from real ones so that the victim can deny ownership of the real signature.

B. PPED Scheme

The construction is based on the IBS scheme in Section III and the forward-secure signature designed in Section III-C. Let e, H_1, H_1' and H_2' be the same as the notation table, namely Table II. Fig. 2 presents the information transfer process of our system. Concretely, we describe five main phases of PPED scheme as follows.

Phase 1: System Initialization

KGC begins with running setup Algorithm 1, where it chooses coefficients s_i and generates a binary polynomial as the master secret key. Then KGC calculates s_iP and publishes $\{P, Q, g, s_iP\}$ as public parameters.

Phase 2: Users Registration

- For patients registering, they send their identities to KGC to generate personal public and private keys. The detailed process is shown in Algorithm 2 (Σ.KeyGen). The key generation algorithm of $\hat{\Sigma}$ is run first to create the initial

Algorithm 2: Σ.KeyGen.

```
Input: $ID, \Delta t, E, V$
Output: $(pk, sk)$
$\hat{\Sigma}.KeyGen:$
    $u \leftarrow H_1'(ID)$
compute $\hat{sk}_E \leftarrow s(u,1)^{-1}Q$
and $\hat{pk}_E \leftarrow \sum_{i=1}^{d}(u^i+1)s_iP = s(u,1)P$
choose a random seed $k_E$ for FWPRG
for $i \in \{E-1, \ldots, -V\}$ do
        $(k_i, l_i) \leftarrow$ FWPRG$(k_{i+1})$
        $\hat{sk}_i \leftarrow s(u,l_i)^{-1}Q, \hat{pk}_i \leftarrow s(u,l_i)P$
        $m_i \leftarrow (\hat{pk}, i, \hat{pk}_i)$
        $\alpha_i = (\beta_i, \gamma_i) \leftarrow \overline{\Sigma}$.Sign$(\hat{sk}_E, m_i)$
        $C_i \leftarrow (\hat{pk}_E, i, \hat{pk}_i, \alpha_i)$
erase $\hat{sk}_E$ and $k_i, l_i, \hat{sk}_i$ for $i = -V, \ldots, E-1$
store $C_i$ for $i = -V, \ldots, E-1$, publish $\hat{pk}_E$
$r_E \xleftarrow{\$} \{0,1\}^\lambda$
for $i \in \{E-1, \ldots, -V\}$ do
$\hat{\Sigma}.Update:$
    $(k_i, l_i) \leftarrow$ FWPRG$(k_{i+1})$
    $\hat{sk}_i \leftarrow s(u,l_i)^{-1}Q, \hat{pk}_i \leftarrow s(u,l_i)P$
retrieve $C_i$, check $\hat{pk}_E$ is correct, $i$ is the current
    epoch, and $\hat{pk}_i$ is equal to the public key generated in
    previous step; abort if the check fails
store $k_i$ and $\hat{sk}_i$, erase $k_{i+1}$
$r_i \leftarrow H_1(r_{i+1}, \hat{pk}||i+1||0)$
end for
$\widehat{sk} = [\widehat{sk}_{-V}, \ldots, \widehat{sk}_0, \ldots, \widehat{sk}_E]$
$sk_r = [r_{-V}, \ldots, r_0, \ldots, r_E]$
$t_0 = now()$
$pk = (\hat{pk}_E, t_0, \Delta t, E, V)$, $sk = (sk_r, pk, \widehat{sk}, 0, \perp, \perp)$
```

key pairs. Then KGC updates and hashes $E+V$ times to get $\hat{sk}$ and sk_r which contain all intermediate results respectively. The public key of a patient pk consists of the long-term public key of $\hat{\Sigma}$, current time t_0, time interval Δt and other parameters. The corresponding secret key sk is a six-tuple that contains the public key pk, all pseudorandom strings sk_r, all secret keys of $\hat{\Sigma}$, the number i of evolution being executed.

- For doctors registering, they send identities to KGC for private key generation. First, KGC computes $u = H_1'(ID)$ and $s(u,1)^{-1}Q$ as the doctor's private key, where $s(u,1) = \sum_{i=1}^{d}(u^i+1)$. Then KGC sends the private key $s(u,1)^{-1}Q$ to the doctor through a secure channel, and the doctor can get his public key by calculating $\sum_{i=0}^{d}(u^is_iP + s_iP) = s(u,1)P$.

In our scheme, patients and doctors are two important roles who perform registration through a trusted KGC. Patients registration is achieved by key generation of the epochal signature scheme, and doctors registration is accomplished by key generation of the regular signature.

Phase 3. Key evolution and public information generation

Algorithm 3: Σ.Evolve.

```
Input: $sk$
Output: $sk', pinfo_{i_{new}}$
$i_{new} \leftarrow i+1, i_{exp} \leftarrow i-V$
if $t = t_0 + i_{new} \cdot \Delta t$ then
$r_{new} \leftarrow sk[i_{new}], r_{exp} \leftarrow sk_r\ [i_{exp}]$
$\hat{sk}_{new} \leftarrow \widehat{sk}[i_{new}], \hat{sk}_{exp} \leftarrow \widehat{sk}\ [i_{exp}]$
end if
$\overline{\Sigma}.KeyGen:$
    $r_{\overline{pk}} \leftarrow H_1(r_{new}, \hat{pk}_E||i_{new}||1)$
    $\overline{sk}_{new} \leftarrow s(u, r_{\overline{pk}})^{-1}Q$
    $\overline{pk}_{new} \leftarrow s(u, r_{\overline{pk}})P$
$m_{tl} \leftarrow (r_{new}||\hat{sk}_{new}||H_1(r_{new}, \hat{pk}||i_{new}||2))$
$TL.lock(1^k, V \cdot \Delta t, m_{tl}) \rightarrow tl:$
  $\varphi(N) \leftarrow (p-1)(q-1),$
  $T \leftarrow V \cdot \Delta t,$
  $b \leftarrow 2^T \pmod{\varphi(N)}, a_T \leftarrow 2^b \pmod{N}$
  $tl \leftarrow m_{tl} \oplus a_T$
$kinfo_i \leftarrow (\overline{pk}_{new}||i_{new}||r_{exp}||\hat{sk}_{exp}||tl)$
$\hat{\Sigma}$.Sign$(\hat{sk}_i, kinfo_i) \rightarrow \hat{\sigma}_i:$
retrieve $C_i = (\hat{pk}_E, i, \hat{pk}_i, \alpha_i)$ and $\hat{sk}_i$
choose a nonce $\omega_i \in Z_r^*$
compute $R_i \leftarrow \omega_i \cdot \hat{pk}_i, h_i \leftarrow H_2'(R_i, kinfo_i),$
  $S_i \leftarrow (\omega_i + h_i)^{-1}\hat{sk}_i$
    $\hat{\sigma}_i \leftarrow (C_i, R_i, S_i)$
$pinfo_{i_{new}} = (\overline{pk}_{new}, i_{new}, r_{exp}, \hat{sk}_{exp}, tl, \hat{\sigma}_i)$
$sk_{i_{new}} = (sk_r, \hat{pk}, \hat{sk}', i_{new}, \overline{sk}_{new}, pinfo_{i_{new}})$
```

In this phase, the KGC completes key evolution $sk_{i_{new}}$ and public information $pinfo_{i_{new}}$ generation for the new period i_{new}. As shown in Algorithm 3, the KGC first generates a nonce r_{new} and a key $\hat{sk}_{new}$ for the new period, and generates a nonce r_{exp} and a key $\hat{sk}_{exp}$ for the expired period. Then, upon inputting the random number r_{new}, public key $\hat{pk}$, the new period tag i_{new}, and identity information u, the KGC subsequently executes the key generation of regular signature, time lock puzzle and signing of forward-secure signature. Finally, the KGC sends the public information $pinfo_{i_{new}}$ and evolved key $sk_{i_{new}}$ of new period to the patient.

Phase 4. Patient-Doctor data exchange

When the patient needs some medical services, he will transmit health data with the epochal signature to the MSP, and the doctor verifies the data integrity and authenticity. The doctor aborts if the verification fails; otherwise, the doctor gives feedback along with the signature to the MSP. Finally, the patient verifies the authenticity of data from the doctor and believes it if the verification succeeds. Details are as follows.

- First, the patient signs his health data m according to Algorithm 4, that is he runs the sign algorithm of regular signature $\overline{\Sigma}$ with inputting current keys sk and data m. The signing output $\overline{\sigma}$ and the public information of current epoch $pinfo_i$ constitute the epochal signature σ. Then, the patient sends data m, signature σ, the public key pk and current epoch number i to the MSP.

Algorithm 4: Σ.{\textrm{Sign}}.

```
Input: $sk, m$
Output: $\sigma$
$\overline{\Sigma}$.Sign :
    $R \leftarrow \omega \cdot \overline{pk}$, where $\omega \in Z_r^*$ is random
    $h \leftarrow H_2'(R, pinfo||m)$
    $S \leftarrow (\omega + h)^{-1}\overline{sk}$
    $\overline{\sigma} \leftarrow (R, S)$
$\sigma = (\overline{\sigma}, pinfo)$
```

Algorithm 5: Σ.Verify.

```
Input: $pk, m, i, \sigma$
Output: $0/1$
$pk = \hat{pk}_E, t_0, \Delta t, E, V$
$\sigma = \overline{\sigma}, pinfo_{i'}$
$\overline{\sigma} = (R', S')$
$pinfo_{i'} = \overline{pk}_{i'}, i', r_{i'-V}, \hat{sk}_{i'-V}, tl_{i'}, \hat{\sigma}$
if $i \leq 0 \vee i' \leq 0 \vee i' + V \leq i \vee i' > i \vee i > E$ then
return 0
end if
$\hat{\Sigma}.Verify$ :
parse $\hat{\sigma}$ into values $C_{i'}, R_{i'}, S_{i'}$
parse $C$ to get the values $(\hat{pk}_E', i', \hat{pk}_{i'}, \alpha_{i'})$
check $\hat{pk}_E' = \hat{pk}_E$, $i' = i \rightarrow \mathbf{0/1}$
parse $\alpha_{i'} := (\beta_{i'}, \gamma_{i'})$, $m_{i'} := (\hat{pk}_E, i', \hat{pk}_{i'})$
compute $h_1 \leftarrow H_2'(\beta_{i'}, m_{i'})$
check whether $e(\beta_{i'} + h_1\hat{pk}_E, \gamma_{i'}) = g \rightarrow \mathbf{0/1}$
compute $h_2 \leftarrow H_2'(R_{i'}, pinfo_{i'})$
check whether $e(R_{i'} + h_2\hat{pk}_{i'}, S_{i'}) = g \rightarrow \mathbf{0/1}$
If all checks succeed output $\boldsymbol{b_0 = 1}$, otherwise
    output $\boldsymbol{b_0 = 0}$.
$\overline{\Sigma}.Verify$ :
    $h_3 \leftarrow H_2'(R_i', pinfo_{i'}||m)$
Check whether $e(R' + h_3\overline{pk}_{i'}, S') = g \rightarrow \boldsymbol{b_1 = 0/1}$
$\boldsymbol{b \leftarrow b_0 \wedge b_1}$
```

- Second, the doctor verifies validity of (m, σ, pk, i) by Algorithm 5 through the MSP. This algorithm checks whether the signature is during the valid epoch. Aborts if the condition is not satisfied; otherwise, $\hat{\Sigma}.Verify$ and $\overline{\Sigma}.Verify$ are run. If and only if all verification processes succeed, the doctor trusts the health data has authenticity and integrity. After that, the doctor gives his diagnosis results or treatment plans based on health data, and signs corresponding information by running sign algorithm of regular signature ($\overline{\Sigma}$.Sign). In final, the doctor replies with his treatment and signature to the patient.
- Eventually, the patient obtains the diagnosis information and verifies them by $\overline{\Sigma}.Verify$. Thus, the patient realizes reliable medical treatment. Particularly, patients can question further problems by repeating above two steps.

Algorithm 6: Σ.KExtract.

```
Input: $pk, pinfo_i$
Output: $sk^*$
$\hat{pk} = pk$, $\hat{sk}_{e-1} = \hat{sk}_{i-V}$
if $0 \leqslant l \leqslant e - 1$ then
$\hat{sk}_l \leftarrow \hat{\Sigma}.Update(\hat{sk}_{l+1})$
$r_l \leftarrow H_1(r_{l+1}, \hat{pk}||l + 1||0)$
end if
$\hat{sk}^* = [\hat{sk}_0, \hat{sk}_1, \ldots, \hat{sk}_e]$
$sk_r^* = [r_0, r_1, \ldots, r_e]$
$sk^* = (sk_r^*, \hat{pk}, \hat{sk}^*, 0, \perp, \perp)$
```

Algorithm 7: Σ.Simulate.

```
Input: $pk, m, pinfo_i$
Output: $\sigma_i^*$
$\Sigma.KExtract(pk, pinfo_i) \rightarrow sk_i^*$
$\Sigma$.Sign$(sk_i^*, m) \rightarrow \sigma_i^*$
```

Phase 5. Deniability after information leakage

In order to protect patients' privacy after leakage, this phase explains the reason that patients can deny their connection with the disclosed information. The answer is that anyone can simulate an expired signature distinguishing from the real one by Algorithm 7. The following are details.

- Upon the public pk and $pinfo_i$, anyone can run Algorithm 6 to exact a simulated key sk^* by the update algorithm $\hat{\Sigma}.Update$ of forward-secure signature;
- Taking sk^* and data m as input, the signing algorithm Σ.Sign can be executed by anyone and outputs a simulated signature of m distinguishing from the real one.

Above two steps form the $\Sigma.Simulate$ algorithm of epochal signature. Since the distinguishability of the simulated signature from the real signature, the relationship between the patient and health data m is broken.

V. Security Analysis

In this section, we analyze the security of our proposed schemes from "Completeness", "Unforgeability" and "Deniability" on the basis of system goals set in Section II.

A. Completeness Analysis

The completeness of our system requires the honestly generated signature can pass the verifying algorithm to ensure that real health data will not be rejected by doctors, which depends on the scheme's correctness. Hence, we first give the definition of correctness of our epochal signature scheme Σ and prove its correctness under this definition.

Informally, correctness means all the unexpired signatures produced by Σ.Sign must be accepted by Σ.Verify. The formal definition is shown as follows:

Definition 1: For any message m, the epochal signature scheme is correct if

$$\forall k \in \mathbb{N}, E \in ploy(k), V \in \{1, \ldots, E-1\},$$

$$(pk, sk) \leftarrow \Sigma.\text{KeyGen}(ID, \Delta t, E, V, \Sigma.\text{Setup}(1^k)),$$

$$i \in \{1, \ldots, E-1\}, i' \in \{i, \ldots, \min(e+V, E)\}:$$

$$\Sigma.\text{Verify}\left(pk, i', \Sigma.\text{Sign}(sk_i, m), m\right) = 1,$$

where sk_i is the resulting secret key from i'th execution of key evolution Σ.Evolve.

Based on the above definition, we have Theorem 1.

Theorem 1: Σ is correct in the sense of Definition 1.

Proof: According to the conditions shown in the definition of correctness, the challenged signature is less than V epochs old and the current epoch i is not past the whole lifetime of the key, namely $i < E$. Moreover, the epoch of the challenged signature should not be a future one. Reviewing the verification algorithm, it is acknowledged that the verification can only succeed if either b_0 or b_1 is set to false.

b_0 indicates the result of verifying $\hat{\sigma}$ under $\hat{pk}$, where $\hat{\sigma}$ was created for the exact message honestly. The correctness of $\hat{\Sigma}$ implies $b_0 = 1$ can be proved as follows. Given a signature $(s, \sigma) \leftarrow \mathbf{Sign}(M, \hat{sk}_t)$, $\mathbf{Verify}(pk_0, M', t', s)$ must be "1" if $M' = M, t' = t$.

1) For one aspect, σ' is correct. For $\sigma' \leftarrow (R'_t, S'_t), m'_t \leftarrow (pk_0, t', \hat{pk}'_t)$, compute $h'_t = H'_2(R'_t, m'_t)$ and we have

$$\begin{aligned} & e(R'_t + h'_t pk_0, S'_t) \\ & = e(x'_t pk_0 + h'_t pk_0, (x_t + h'_t)^{-1} sk_0) \\ & = e((x'_t + h'_t) pk_0, (x'_t + h'_t)^{-1} sk_0) \\ & = e(pk_0, sk_0) = e(s(u,1)P, s(u,1)^{-1}Q) \\ & = e(P, Q) = g. \end{aligned}$$

2) For another aspect, σ is correct. For $\sigma \leftarrow (M, R', S')$, compute $h' = H'_2(R', M)$ and we have

$$\begin{aligned} & e(R' + h' \hat{pk}'_t, S') \\ & = e(x' \hat{pk}'_t + h' \hat{pk}'_t, (x' + h')^{-1} \hat{sk}'_t) \\ & = e(\hat{pk}'_t, \hat{sk}'_t) = e(s(u, r_{t'})P, s(u, r_{t'})^{-1}Q) \\ & = e(P, Q) = g. \end{aligned}$$

b_1 indicates the result of verifying $\overline{\sigma}$ under $\overline{pk}$, where $\overline{\sigma}$ was created for the exact message honestly. The correctness of $\overline{\Sigma}$ implies $b_1 = 1$, which is proved as follows. Given a signature $\overline{\sigma} = (R, S) \leftarrow \mathbf{Sign}(m, P_u, Q_{su})$, $\mathbf{Verify}(m', P_u, (R, S))$ must be "1" if $m' = m$.

$$\begin{aligned} & e(R + hP_u, S) \\ & = e(xP_u + hP_u, (x+h)^{-1} Q_{su}) = e(P_u, Q_{su}) \\ & = e(s(u,l)P, s(u,l)^{-1}Q) = e(P, Q) = g. \end{aligned}$$

Combined this means that $urcorner(b_0 \wedge b_1)$ is always false, therefore the Σ is correct.

```
Exp$^{EEUF-CMA}_{\Sigma,\mathcal{F}}(1^\lambda, \Delta t, E, V)$
1. $(pk, sk) \leftarrow \Sigma.KeyGen(1^\lambda, \Delta t, E, V)$;
   $t_0 = now()$;
   $i \leftarrow 0$;
   $queries = [\emptyset, ..., \emptyset]$.
2. $\sigma, m \leftarrow \mathcal{F}^{\Sigma.Evolve', \Sigma.Sign'}(pk)$.
3. $ret \leftarrow \Sigma.Verify(pk, i, \sigma, m)$
4. for $i' \in \{max(0, i-V), ..., e\}$:
   abort_if($(m, i') \in queries[i']$).

$\Sigma.evolve'()$:
   abort_if($t = E$);
   abort_if($now() \leq t_0 + i \cdot \Delta t$);
   $i \leftarrow i + 1$;
   $pinfo_i, sk \leftarrow \Sigma.evlove(sk)$;
   return $pinfo_i$

$\Sigma.sign'()$ :
   abort_if($now() \leq t_0 + i \cdot \Delta t$);
   $\sigma \leftarrow \Sigma.sign(sk, m)$;
   $queries[i] \cup = m$;
   return $\sigma$
```

Fig. 3. The unforgeability game for epochal signatures.

Theorem 2: PPED achieves completeness if the signature scheme Σ is correct

Proof: System completeness refers to the successful execution of health data transmission in the e-health system. Recall Theorem 1, we prove correctness, which guarantees that a real signature can pass the verification successfully. Consequently, when a patient generates a real and valid signature on their health data, the doctor can receive it successfully, thus facilitating the execution of the data transfer process. This observation confirms the completeness of our system.

B. Unforgeability Analysis

Here we prove our signature scheme is unforgeable. Unforgeability means that no one can forge the signatures of doctors and patients, guaranteeing the authenticity of the data being transmitted and stored by the platform. We begin with showing the formal definition of unforgeability. Then we prove our new identity-based signature scheme is unforgeable, which is significant for the scheme design. After that, we explain the unforgeability of our forward-secure signature scheme. At last, we use a series of game hops to show our specific epochal signature scheme is secure.

The formal notion of unforgeability for an epochal signature is defined by [31], and we introduce this definition into our signature scheme since it is also epoch-based. In the following, we first give an experiment $Exp^{EEUF-CMA}_{\Sigma,\mathcal{F}}$, as shown in Fig. 3, to simulate the unforgeability game played between an Adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Then we define the security of PPED based on this experiment.

Definition 2: Our epochal signature construction Σ is unforgeable under the definition of Epochal Existential

Unforgeability under Chosen Message Attacks (EEUF-CMA) if there is no polynomial time forger $\mathcal{F}$ that has a non-negligible advantage of wining the $Exp^{EEUF-CMA}_{\Sigma,\mathcal{F}}$

$$\forall \mathcal{F} \in PPT, \lambda \in \mathrm{N}, E \in ploy(\lambda), V \in \{1, \ldots, E-1\}:$$

$$Pr[Exp^{EEUF-CMA}_{\Sigma,\mathcal{F}}(1^{\lambda}, \Delta t, E, V) = 1] =$$

$$Adv^{EEUF-CMA}_{\Sigma,E,V,\mathcal{F}}(1^{\lambda}, \Delta) \leq negl(\lambda).$$

Before proving the security of Σ, we first explain the unforgeability of our proposed ID-based signature scheme under the hardness of **k-CAA** problem introduced in Section III. Here we use the definition of [32], which gives a standard security model of a regular identity-based signature scheme.

Definition 3: We claim that an ID-based signature scheme, which is composed of four algorithms **Setup**, **KeyGen**, **Sign**, **Verify** is secure against existential forgery on adaptively chosen message and identities attacks (ID-EUF-CMA) if there is no efficient adversary $\mathcal{A}$ that has a non-negligible chance winning a challenger $\mathcal{C}$ in the following game:

1. $\mathcal{C}$ performs **Setup** algorithm to obtain system parameters and sends to $\mathcal{A}$.

2. $\mathcal{A}$ performs the following series of queries:

- Hash function query. Given the queried input, $\mathcal{C}$ computes the hash function value and transmits the results to $\mathcal{A}$.
- **KeyGen** query. $\mathcal{A}$ sends the requested identity ID to $\mathcal{C}$. Take ID as input, $\mathcal{C}$ honestly runs **KeyGen** algorithm and returns the private key corresponding to ID.
- **Sign** query. Take an identity ID and a message m as input, $\mathcal{C}$ honestly runs **Sign** algorithm and outputs the corresponding signature.

3. $\mathcal{A}$ outputs a three tuple (ID^*, m^*, σ), where ID^* and (ID^*, m^*) cannot be the inputs of any query to **KeyGen** and **Sign** respectively. $\mathcal{A}$ wins the game means that σ is a valid signature of m^* generated by a party with ID^*.

Here we divide our proof into two steps according to the core idea of [32]. The first step of our proof is reducing the problem to the ID-fixed situation. For this case, we introduce the following lemma.

Lemma 1: If there is an algorithm $\mathcal{A}_0$ for an adaptively chosen message and identity attack to our scheme with running time t_0 and advantage ϵ_0, then there is an algorithm $\mathcal{A}_1$ for an adaptively chosen message and given identity attack which has running time $t_1 \leq t_0$ and advantage $\epsilon_1 \leq \epsilon_0(1 - 1/r)/q_{H_1}$, where q_{H_1} is the maximum number of queries to H_1 made by $\mathcal{A}_0$. Additionally, $\mathcal{A}_1$ execute same times of queries as $\mathcal{A}_0$ to hash functions, Extract, and Sign.

In this lemma, a non-identity-based scheme can be obtained by fixing an ID in identity-based scheme. Then the algorithm $\mathcal{A}_1$ can be seen as an adversary against the non-identity-based scheme. Additionally, $\mathcal{A}_1$ is accessible to the key-generation oracle to get secret keys of identities which are not equal to the fixed one, as well as the signing oracle and hash functions.

Here, we are prepared to design an algorithm that solves k-CAA problem with the assumption of $\mathcal{A}_1$'s existence.

Lemma 2: If there is an algorithm $\mathcal{A}_1$ for an adaptively chosen message and given ID attack to our scheme with running time t_1 and advantage ϵ_1, and queries H_1, H_2, **Sign** and **KeyGen** at most $q_{H_1}, q_{H_2}, q_S, q_E$ times respectively, then there is an algorithm $\mathcal{A}$ that can solve q_S-CCA problem with advantage $\epsilon' \geq (\frac{q_S}{q_H})^{q_S} \cdot \epsilon$ within the same running time $t' = t$.

Proof: To proof this lemma, we first make some ratioanl assumptions. In specific, for any ID, $\mathcal{A}_1$ queries $H_1(\text{ID})$ and **KeyGen**(ID) at most once, and $\mathcal{A}_1$ queries $H_1(\text{ID})$ before ID becomes input of any query to H_1, **KeyGen** and **Sign**. We will utilize the algorithm $\mathcal{A}_1$ to construct an algorithm $\mathcal{A}$ to solve q_S-CAA problem. We fix an identity ID and put $Q_{su} = \frac{1}{q}Q, P_u = aP$ and suppose $\mathcal{A}$ is given a challenge: Given Q_{su}, $P_u = aP$, $R = xP_u$ $h_1, h_2, \ldots, h_{q_S} \in Z_r$ and $\frac{1}{h_1+x}P$, $\frac{1}{h_2+x}P, \ldots, \frac{1}{h_{q_S}+x}P$, to compute $\frac{1}{h+x}P$.

Now $\mathcal{A}$ acts as the role of the signer and will response to hash oracle queries and sign queries. It is assumed that $\mathcal{A}$ never repeats a hash query or a signature query.

1) $\mathcal{A}$ prepares q_H answers $\{w_1, w_2, \ldots, w_{q_{H_2}}\}$ of hash oracle queries of H_2, where h_1 to h_{q_S} are distributed in this response set at random.

2) $\mathcal{A}_1$ asks a hash oracle query on message m_i for $1 \geq i \geq q_H$. $\mathcal{A}$ returns w_i to $\mathcal{A}_1$ to response the hash oracle query of $\mathcal{A}_1$ on m_i.

3) $\mathcal{A}_1$ executes a signature oracle query for w_i. If $w_i = h_j$, $\mathcal{A}$ responses $1/(h_j + x)P$ to $\mathcal{A}_1$ as result. Otherwise, the process aborts, which means $\mathcal{A}$ has failed.

4) Lastly, $\mathcal{A}_1$ stops and outputs $(m, (R, S))$, such that the hash value of (R, m) is some w_l and $w_l \notin \{h_1, \ldots, h_{q_S}\}$. As $(m, (R, S))$ is a successful forgery and $H_2(R, m) = w_l$, it satisfies: $e(R + w_l P_u, S) = e(P, Q)$.

In conclusion, our identity-based signature scheme is secure. Combined the property shown above, the unforgeability of our forward-secure scheme can be deduced as follows.

Theorem 3: The forward-secure signature $\hat{\Sigma}$ proposed in Section III-C satisfies existential unforgeability under adaptive chosen messages and identities attacks (ID-FS-EUF-CMA).

Proof: As mentioned in Section III, there is a universal method in [28], which can change a regular signature into a forward-secure one. This work also declares that if the regular underlying scheme is unforgeable and a forward-secure pseudorandom generator is used in the transforming process, the newly constructed signature is forward-secure signature scheme with unforgeability. We have already explained our identity-based signature scheme is secure against existential forgery under ID-EUF-CMA attack above, therefore our FSIBS scheme is naturally unforgeable. Thus, our forward-secure identity-based signature is ID-FS-EUF-CMA.

Now we formally prove the unforgeability of our epochal signature scheme.

Theorem 4: Our specific epochal signature scheme based on identity Σ is secure against an existential forgery for adaptive chosen messages attacks(ID-EUF-CMA).

Proof: The key idea of our proof is reducing the security of the scheme to the security of used technologies. E is set as the maximum number of epochs. We will present our signature scheme Σ is secure by a series of game-hopping as follows:

Game 0: This game is set as identical to the original **EEUF-CMA**-game and $Pr[break_0]$ denotes the advantage that a forger $\mathcal{F}$ is successful in showing a forgery.

Game 1: We assume the forger will succeed in creating a forgery in the epoch i and stop if this is not the case. This abort will happen by the end of epoch $i+V$ because the validity of signatures is V epochs. Our guess has probability E^{-1} to be right, having

$$Pr[break_0] \leq E \cdot Pr[break_1].$$

In this game, all evaluations of $H_1(r_i, \cdot)$ are replaced with random values. We make use of PRF-assumption for H_1 to alter all later values of r as well in reserve order to realize the replacement.

Game 2: In this game, we substitute outputs of all $H(r_e, \cdot)$ into random values. Let *Game 2.E := Game.1* and $Game2.k$ means *Game 2* is in the period k ($k \in \{E-1, \ldots, i\}$). Then we choose a series of random values to replace the results of all evaluations of $H(r_{k+1}, \cdot)$. In order to make this operation reasonable, we set up a challenger of PRF and use the oracle provided by it, rather than computing the value of H directly. This change is reasonable because r_{k+1} is a truly random value by $Game2.(k+1)$. If the challenger's internal bit is 0 then we are in $Game2.(k+1)$, otherwise we are in $Game2.k$. Thus, we enable any adversary who can detect this change to be transformed into an adversary $\mathcal{A}_{2.k}$ who can break the PRF-security of H

$$Pr[break_{2.k+1}] \leq Pr[break_{2.k}] + Adv^{PRF}_{H,\mathbf{A}_{2.k}}(1^\lambda).$$

Because there are no more than E hops like this, we have

$$Pr[break_1] \leq Pr[break_2] + E \cdot \mathcal{A}_2^P RF(1^\lambda).$$

Game 3: In this game, we encapsulate random strings in $tl_i, \ldots, tl_{i+V-1}$. The sub-hops are needed to perform this. Let *Game 3.V := Game.2* and $Game3.k$ means *Game 3* is in the period $k (k \in \{V-1, \ldots, 0\})$. Then we use a random string to replace the value encapsulated in tl_{i+k}. This operation is rational because the time lock puzzle has the temporal hiding property. In order to achieve this, a hiding-challenger is initialized and queried with two inputs m_0 and m_1, where $r_t||\hat{sk}_t$ is set as m_0 and a randomness with the same length is m_1. Since the string used to run the time lock puzzle is random by $Game3.(k+1)$ and a time lock is less than $V \cdot \Delta t$, our replacement is reasonable. When the challenger's internal bit is 0, we are in $Game3.(k+1)$. Otherwise, the encapsulation of a random string will be output, then we are in $Game3.k$. Therefore any adversary who is able to distinguish these two games can be transformed into $\mathcal{A}_{3.k}$ against the hiding property of the time lock puzzle. Then we can conclude

$$Pr[break_{3.(k+1)}] \leq Pr[break_{3.k}] + Adv^{IND-NMA}_{TL,\mathcal{A}_{3.k}}(1^\lambda, V \cdot \Delta t).$$

Through defining $Game3 := Game3.0$ and combining these sub-game-hops, we have

$$Pr[break_2] \leq Pr[break_3] + V \cdot Adv^{IND-NMA}_{TL,\mathbf{A}_3}(1^\lambda, V \cdot \Delta t).$$

Game 4: We change the key generation process as follows: After setting up r and sk_r we directly calculate $r_{i'}, \overline{pk}_{i'}, \overline{sk}_{i'}, tl_{i'}$ for all epochs $i' \in \{i-V+1, \ldots, i+V\}$. Then we set up an ID-FS-EUF-CMA-challenger for the forward secure signature scheme $\hat{\Sigma}$ and get its public key $\hat{pk}$ that will be used by us rather than generating our own. After that, we query the signatures for public keys of the regular scheme $\overline{\Sigma}$. First we require the challenger to implement key updates $E-i-V+1$ times. For each $k \in \{V-1, \ldots, 0\}$ we do that:

(1) query a signature $\hat{\sigma}_{i+k}$ for key information of epoch i, namely $kinfo_i = (\overline{pk}_{i+k}||i+k||r_{i+k-V}||tl_{e+i-V})$.

(2) query a key evolution.

After that, we query the evolved secret key $\hat{sk}_{i-1}$. By utilizing this key we can set up $\hat{sk}$ for the first $i-1$ epochs and finish the key generation honestly. In first $i-1$ epochs, the algorithms $\Sigma.evolve$ and $\Sigma.sign$ do not change. However from the i'th epoch $\Sigma.evolve$ is modified to use the corresponding $\hat{\sigma}_i$ and $tl_{i'}$ that we prepared in the key generation process rather than calculating them as normal, and sk' is set as $\perp$. All values given to the adversary are still sampled from the same distribution in *Game 4* which makes our change in this part hard to be undetected.

If a valid forgery for epoch i is shown by the adversary successfully, we check whether the forward-secure signature is equal to $\hat{\sigma}_i$. If they are not equal, we can transmit it to ID-FS-EUF-CMA-challenger and win the ID-FS-EUF-CMA-game and stop. Otherwise, we repeat the process as before. The games are perfectly indistinguishable unless the forger manages to forge $\hat{\sigma}_i$. Therefore any adversary capable of finding this change can be transformed into a new adversary $\mathcal{A}_5$, who can break the ID-FS-EUF-CMA-security of $\hat{\Sigma}$, we get that

$$Pr[break_3] \leq Pr[break_4] + Adv^{ID-FS-EUF-CMA}_{\hat{\Sigma},E,\mathcal{A}_4}(1^\lambda).$$

Game 5: We finally consider the situation where the adversary attacks the regular scheme $\overline{\Sigma}$. When it comes to computing $\overline{pk}_i$ and $\overline{sk}_i$, we initialize an ID-EUF-CMA challenger for $\overline{\Sigma}$ and utilize the challenged public key as $\overline{pk}_i$. This replacement is rational since the randomness used for the key generation process is truely random in *Game 2*. We query the signature from the challenger whenever we need to sign a message using $\overline{sk}_i$. Any adversary who can win *Game 5* can be transformed into an new adversary $\mathcal{A}_6$ with the same probability of breaking the regular signature scheme $\overline{\Sigma}$, namely

$$Pr[break_5] = Adv^{EUF-CMA}_{\overline{\Sigma},\mathcal{A}_5}(1^k).$$

```
$\mathbf{Exp}^{Deniability}_{\Sigma,\mathcal{S},\mathcal{J}}(1^\lambda,\Delta t,E,V)$
$(pk,sk)\leftarrow\Sigma.KeyGen(1^\lambda,\Delta t,E,V)$;
$b\leftarrow\{0,1\}^*$;
  $m,i_0,i_1=\mathcal{J}(pk,sk)$;
  $\sigma=\perp$.
abort_if$(\vee i_0+i_1\geq E\vee i_0<0\vee i_1<V)$
$(pinfo_i,sk)\leftarrow\Sigma.evlove(sk)$,
    for $i\in\{1,...,i_0\}$.
if $b=0$, $\sigma\leftarrow\Sigma.sign(sk,m)$.
$(pinfo_i,sk)\leftarrow\Sigma.evlove(sk)$,
    for $i\in\{i_0+1,...,i_0+i_1\}$.
if $b=1$, $\sigma\leftarrow\mathcal{S}(m,i,pinfo_{i_0+i_1})$.
$b'=\mathcal{J}(\sigma,sk)$.
return $b'=b$
```

Fig. 4. The deniability game for epochal signatures.

In conclusion, we get that for all forgers $\mathcal{F}$

$$Adv^{EUF-CMA}_{\Sigma,E,V,\mathcal{F}}(1^k,\Delta t)\leq$$

$$E\cdot(E\cdot Adv^{PRF}_{H,\mathcal{A}_{PRF}}+V\cdot Adv_{TL}+$$

$$Adv^{ID-FS-EUF-CMA}{}_{\hat{\Sigma},\mathcal{A}_4}+Adv^{ID-EUF-CMA}{}_{\hat{\Sigma},\mathcal{A}_5}).$$

In fact, if $H_1,TL,\hat{\Sigma}$ and $\overline{\Sigma}$ provide security guarantees respectively, our epochal signature scheme Σ is unforgeable in the sense of Definition 2.

C. Deniability Analysis

Deniability means that when faced with the accidental disclosure of old data and corresponding signatures, the patient can deny his relationship with them in order to protect personal privacy. Intuitively if there is a simulator capable of generating any expired signatures that are distinguishable from real ones, an epoch-based signature scheme is said to be deniable. In order to prove our scheme with deniability, we first introduce an experiment $\mathbf{Exp}^{Deniability}_{\Sigma,\mathcal{S},\mathcal{J}}$, as shown in Fig. 4, to simulate the game between a simulator $\mathcal{S}$ and a judge $\mathcal{J}$.

Definition 4: Our PPED scheme Σ is deniable if there is a with polynomial time simulator $\mathcal{S}$, then there is no judge $\mathcal{J}$ that can win $\mathbf{Exp}^{Deniability}_{\Sigma,\mathcal{S},\mathcal{J}}$ with the probability $>\frac{1}{2}$

$$\forall\lambda\in\mathrm{N},E\in poly(\lambda),V\in\{1,\ldots,E-1\}:\exists\mathcal{S}\in PPT:$$

$$\forall\mathcal{J}\in TM:Pr[\mathbf{Exp}^{Deniability}_{\Sigma,\mathcal{S},\mathcal{J}}(1^\lambda,\Delta t,E,V)=1]\leq\frac{1}{2}.$$

Theorem 5: Σ is deniable in sense of Definition 4.

Proof: The simulator $\mathcal{S}$ first executes the key-Extractor Algorithm, which uses the information $pinfo_e$ to generate a secret key that is identical to the real key for any expired epochs. Next, $\mathcal{S}$ performs the signing algorithm as same as an honest party. In this case, if the sk generated by the simulator is certainly identical to the real secret key, the deniability of our scheme can be inferred from the whole definition of the simulator and the fact that key evolution algorithm is deterministic. In order to make the secret key equivalent in fact, we should focus on the only difference between it and the real one. It is that the data structure does not return back as keys or randomness evaluations preventing its use in future periods. Given the process of the game, this information is not necessary when the simulator runs. Thus, the only difference is one that does not affect the generated signatures as the signing process does not need that information. Hence, the keys are equivalent for any past epochs and the simulated signatures are exact as if they were honestly generated. As $pinfo_i$ and the related parts of sk_i are identical to the ones that would be used by an honest party, the resulting signature is indistinguishable from the real one from the perspective in information theoretical. Even if $\mathcal{S}$ is called more than once, $\mathcal{J}$ is impossible to learn b due to the perfect indistinguishability which limits him to guess only one random bit with a probability of success of $\frac{1}{2}$.

In conclusion, the proposed scheme Σ is deniable.

VI. Performance Analysis

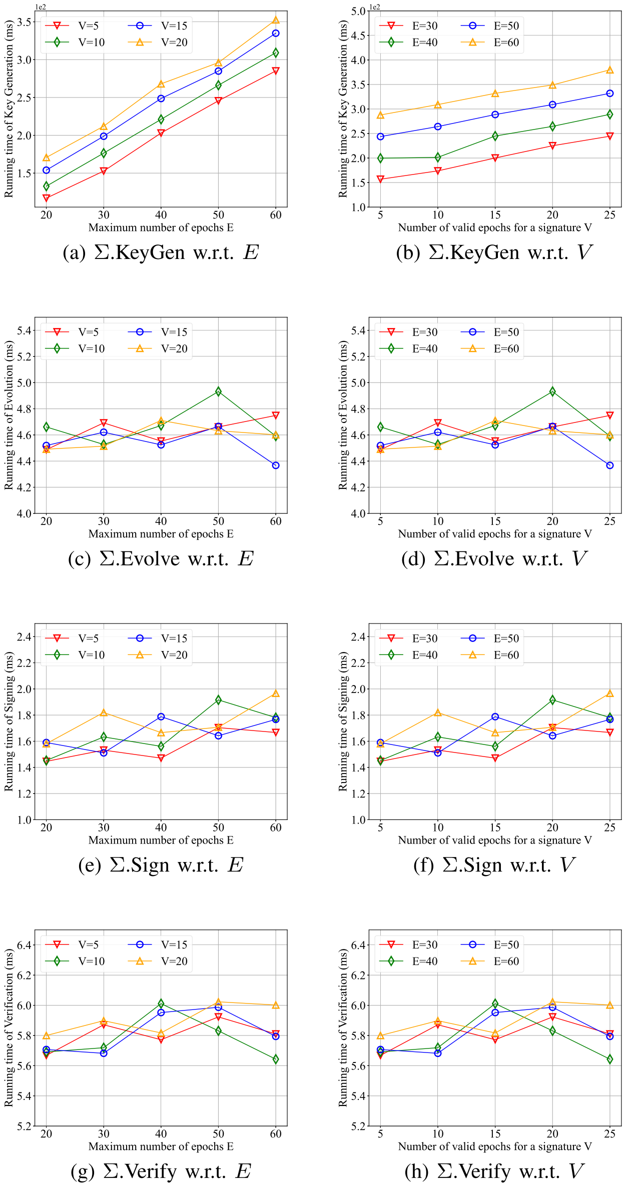
In this section, we analyze the performance of the proposed scheme in terms of both computational and communication overhead. Specifically, we implement the Σ to measure the execution time of key generation, key evolution, signing, verification, key extraction, and simulation algorithms; we also evaluate the communication cost of Σ in the view of patients.

A. Computation Cost Evaluation

Setup and Implementation: In order to analyze the factors that have an impact on the computation overhead of our scheme, we implement our scheme in C++. Specifically, we sample the parameters of Type A curve in PBC library, and use the pseudorandom function in AES algorithm to realize the forward-secure pseudorandom generator FWPRG. The degree of the master secret polynomial d is set to 10. The experiments are conducted on a virtual machine with 4 GB RAM and 3.3 GHz AMD R9 5900 HX CPU.

Experimental Results: In our evaluation, each experiment result is the average running time of 30 tests. The maximum periods number E and valid periods number V are two most important parameters in Σ, which will directly affect the running time of our scheme. Therefore, we first show the running time of four algorithms of Σ: key generation, key evolution, signing, and verification, under different settings of E and V. For the principle of controlling variables, we consider E and V separately in experiments. In detail, we test the running time of above four algorithms when V is set to be constant and E varies from 20 to 60. Then, we repeatedly run algorithms when fixing E and changing V from 5 to 25. The comparison results are shown in Fig. 5.

The computational cost evaluation of key generation is illustrated in Fig. 5(a) and (b). We can find that the running time of the key generation algorithm is approximately linear with the increase of E and V. This is because a larger E or V leads to more executions of $\hat{\Sigma}.KeyGen$ and $\hat{\Sigma}.Update$ to generate pseudorandom-seed tuple sk_r and secret key tuple $\hat{sk}$. In Fig. 5(c) and (d), the running time of Σ. Evolve fluctuates around 4.6 ms in the different setting of E and V. This is because each evolution is to generate the key and public information for the current phase, and is irrelevant to numbers of period E and V. Additionally, we can observe from Fig. 5(e) and (f) that



(a) Σ.KeyGen w.r.t. E

(b) Σ.KeyGen w.r.t. V

(c) Σ.Evolve w.r.t. E

(d) Σ.Evolve w.r.t. V

(e) Σ.Sign w.r.t. E

(f) Σ.Sign w.r.t. V

(g) Σ.Verify w.r.t. E

(h) Σ.Verify w.r.t. V

Fig. 5. Runing time of Σ.KeyGen, Σ.Evolve, Σ.Sign, Σ.Verify.

the running time of Σ.Sign is stable at approximately 1.6ms since its computation cost comes from the once execution of $\overline{\Sigma}$.Sign which is only related to information of current phase. The same as two former algorithms, the computation cost of Σ.Verify is also independent of E and V. Fig. 5(g) and (h) show the running time is about 5.8 ms regardless of E and V, as the time consumption of Σ.Verify is from once $\hat{\Sigma}.Verify$ and once $\overline{\Sigma}.Verify$.

Σ.KExtract and Σ.Simulate are two crucial algorithms for realizing deniability. The simulator obtains public key and public information of the current epoch i to simulate expired signatures. Thus, except for E and V, we also consider the effect of i on Σ.KExtract and Σ.Simulate. We respectively set E=40 and V=5, 10, 15, 20, as well as V=10 and E=20, 30, 40, 50 for experiments. Tables III and IV show that the running time of Σ.KExtract and Σ.Simulate increase with the growth of i and unrelated to E and V. This is because the number of key updates and hashing operations in algorithms is determined by i, which can be inferred from details in Algorithms 6 and 7.

B. Communication Cost Evaluation

In this part, we evaluate the communication overhead of a patient transferring a message within an epoch in the system. In the experiment, we use a 128-bit elliptic curve and set the security parameter in the time lock to 1,024 bits, choose a 3rd-degree polynomial as the master key, and sign a 512-bit message. The communication cost of patients mainly consists of two parts. One part comes from the transmission of public key pk, message m, signature σ, and the number of the current epoch i to the doctor, so that the doctor can confirm the authenticity of the message. This process consumes 717.5 KB. The other part comes from the process of uploading the public key pk and public information $pinfo$ to the platform for performing key extraction and signature forgery. The communication cost of this process is 592 KB. Thereby, the total communication cost of our scheme is 1309.5 KB.

VII. Related Work

Nowadays, an increasing number of people hope to obtain more quality and comprehensive healthcare services. To realize this, electronic health has become indispensable due to its timeliness and convenience. However, the development of e-health is still in its early stages and data authenticity and privacy issues still remain to be solved for practical applications.

The authenticity of an e-health system is critical and any incorrect data could lead to significant consequences, thereby many works deployed verification mechanisms into systems. Ding et al. [16] proposed a lightweight IoT-based health system, which used the identity-based cryptosystem to verify patients' health data authenticity and improve the system reliability. To prevent malicious users from tampering with real health data, Lu et al. [26] focused on the trusted platform module and utilized the digital signature to implement data authentication. Additionally, Deebak et al. [25] proposed an authentication protocol verifying the authenticity of data transmitted between any two entities in their e-health system, among medical WBAN sensors, personal patient devices and the service authority center. Another authentication scheme based on WBAN was recently proposed by Yang et al. [13]. They proposed an efficient and anonymous authentication mechanism for healthcare servers, which achieved comprehensive authentication and significantly decreased the computation overhead. Furthermore, Yan et al. [14] applied digital watermarking technology in an e-health system to propose a multi-watermarking scheme to guarantee the authenticity of medical images. Liu et al. [33] designed a new redactable signature scheme to establish a lightweight e-health system that supports data redaction and data authentication during the dissemination process.

Patient privacy is another key element in e-health systems. There are many works on how to protect patient data privacy, and attribute-based encryption is one of the most commonly used techniques. Specifically, Zhang et al. [20] proposed a large universe ciphertext-policy attributed-based encryption with partially hidden access policies, which can hide specific and

TABLE III
THE TIME COST OF Σ.KEXTRACT

Time(ms)	E=40				V=10			
	V=5	V=10	V=15	V=20	E=20	E=30	E=40	E=50
$i = 5$	9.990	10.234	9.539	9.751	9.990	10.234	9.539	9.751
$i = 10$	20.567	19.327	21.112	21.696	20.567	19.327	21.115	21.696
$i = 15$	30.159	29.056	29.977	31.867	30.159	29.056	29.997	31.867
$i = 20$	39.163	38.613	39.001	40.236	39.163	38.613	39.001	40.236
$i = 25$	50.342	50.836	51.435	52.007	50.342	50.863	51.435	52.007

TABLE IV
THE TIME COST OF Σ.SIMULATE

Time(ms)	E=40				V=10			
	V=5	V=10	V=15	V=20	E=20	E=30	E=40	E=50
$i = 5$	11.203	11.776	11.178	12.630	12.027	11.668	12.475	13.861
$i = 10$	23.021	22.001	22.964	24.763	21.897	22.964	23.217	24.886
$i = 15$	33.510	33.641	32.974	34.143	33.001	32.936	32.809	35.127
$i = 20$	44.321	44.013	43.647	44.612	44.236	43.647	44.478	43.855
$i = 25$	53.177	53.498	52.696	54.707	53.666	52.696	53.330	55.741

sensitive attribute values of patients. Recently, Zhang et al. [34] utilized attributed-based encryption to design a lightweight fine-grained e-health system. This system achieved a fully hidden access policy and verifiable decryption mechanism, realizing secure personal health record sharing. Additionally, searchable encryption is a typical method to protect patient privacy in some data sharing or retrieval scenarios. In [35], the authors employed dynamic searchable encryption and bloom filter to present an e-health system, where patients' health information is generated and stored in the cloud securely and patients can control search capability for health service providers. The work of Xu et al. [13] presented an attribute set based keyword search scheme to ensure that only some specific data is searchable for an authorized user. Besides, the study of Yang et al. [27] combined attribute-based encryption and searchable encryption together to propose a lightweight sharable and traceable secure e-health system. This scheme achieves access control to encrypted health data and supports traitor tracing when patients' identities are disclosed. Apart from that, there are other schemes [36], [37], [38], [39], [40] that target patient privacy protection in E-health systems.

While the efforts mentioned above focus on e-health systems, most of them do not guarantee the authenticity and privacy of health data simultaneously. Furthermore, these systems only concentrate on the patient-doctor interaction stage without adequately considering the case after patients leave e-health systems. Those two processes are simultaneously considered in our scheme. The authenticity of transmitted data and the privacy of patients leaving systems are ensured in our scheme well.

VIII. CONCLUSION

In this paper, we study challenging problems in the e-health scenario and propose a reliable and privacy-preserving e-health diagnostic system PPED. By designing a regular signature and a forward-secure signature, we construct a specific epochal signature scheme. PPED exploits the unforgeability and deniability of our epochal signature, and helps patients to be securely and privately diagnosed by doctors. PPED is practical and feasible for real-world e-health systems, as it not only provides strong protection for patients but also introduces acceptable running time and communication cost. In the future, we will further reduce the time and communication overhead and improve the performance during the whole process of e-health system. In addition, we assume that messages and signatures will be stored securely after they are transmitted to the platform, but the exact method is not considered. Later, we will consider using other cryptographic technologies to provide further privacy protection.

REFERENCES

[1] N. A. Azeez and C. Van der Vyver, "Security and privacy issues in e-health cloud-based system: A comprehensive content analysis," *Egyptian Inform. J.*, vol. 20, no. 2, pp. 97–108, 2019.

[2] P. Kishore, S. K. Barisal, K. V. Kumar, and D. P. Mohapatra, "Security improvement and privacy preservation in e-health," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[3] Nextgen healthcare: Award-winning EHR/EMR software, 1996. [Online]. Available: https://www.nextgen.com/

[4] HCA healthcare: Giving people a healthier tomorrow, 2000. [Online]. Available: https://hcahealthcare.com/

[5] Dr ding xiang - professional healthy lifestyle platform, 2002. [Online]. Available: https://dxy.com/

[6] Dedalus global, 2021. [Online]. Available: https://www.dedalus.com/global/en/

[7] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," in *A Practical Guide*, 1st ed., Cham, Switzerland: Springer International Publishing, 2017.

[8] A. Act, "Health insurance portability and accountability act of 1996," U.S. Public Law 104.191, U.S. House of Rep., Washington, DC, USA, 1996. [Online]. Available: https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996

[9] ISO, "Requirements for an electronic health record architecture," 2019. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:18308:ed-1:v1:en

[10] ISO, "Security requirements for archiving of electronic health records – principles," 2012. [Online]. Available: https://www.iso.org/obp/ui#iso:std:iso:ts:21547:ed-1:v1:en

[11] ISO, "Information technology-security techniques-information security management systems," 2019. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso-iec:27000:ed-5:v1:en

[12] ISO, “Health informatics – electronic health record communication,” 2019. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:13606:-1:ed-2:v1:en

[13] X. Yang, X. Yi, S. Nepal, I. Khalil, X. Huang, and J. Shen, “Efficient and anonymous authentication for healthcare service with cloud based WBANs,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2728–2741, Sep./Oct. 2022.

[14] F. Yan, H. Huang, and X. Yu, “A multiwatermarking scheme for verifying medical image integrity and authenticity in the internet of medical things,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8885–8894, Dec. 2022.

[15] H. Wang and Y. Song, “Secure cloud-based EHR system using attribute-based cryptosystem and blockchain,” *J. Med. Syst.*, vol. 42, no. 8, pp. 1–9, 2018.

[16] R. Ding, H. Zhong, J. Ma, X. Liu, and J. Ning, “Lightweight privacy-preserving identity-based verifiable IoT-based health storage system,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8393–8405, Oct. 2019.

[17] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, “Blockchain-based data preservation system for medical data,” *J. Med. Syst.*, vol. 42, no. 8, pp. 1–13, 2018.

[18] C. Xu, N. Wang, L. Zhu, C. Zhang, K. Sharif, and H. Wu, “Reliable and privacy-preserving top-k disease matching schemes for e-healthcare systems,” *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5537–5547, Apr. 2022.

[19] Y. Liu, J. Yu, J. Fan, P. Vijayakumar, and V. Chang, “Achieving privacy-preserving DSSE for intelligent iot healthcare system,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 2010–2020, Mar. 2022.

[20] Y. Zhang, D. Zheng, and R. H. Deng, “Security and privacy in smart health: Efficient policy-hiding attribute-based access control,” *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[21] Y. Zhang, D. He, M. S. Obaidat, P. Vijayakumar, and K.-F. Hsiao, “Efficient identity-based distributed decryption scheme for electronic personal health record sharing system,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 384–395, Feb. 2021.

[22] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh, and X. Liu, “Secure fine-grained encrypted keyword search for e-healthcare cloud,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1307–1319, May/Jun. 2021.

[23] W. Yang, S. Wang, and Y. Mu, “An enhanced certificateless aggregate signature without pairings for e-healthcare system,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 5000–5008, Mar. 2021.

[24] H. Attaullah et al., “Fuzzy-logic-based privacy-aware dynamic release of IoT-enabled healthcare data,” *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4411–4420, Mar. 2022.

[25] B. D. Deebak and F. Al-Turjman, “Smart mutual authentication protocol for cloud based medical healthcare systems using internet of medical things,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 346–360, Feb. 2021.

[26] D. Lu et al., “xTSeH: A trusted platform module sharing scheme towards smart IoT-eHealth devices,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 370–383, Feb. 2021.

[27] Y. Yang, X. Liu, R. H. Deng, and Y. Li, “Lightweight sharable and traceable secure mobile health system,” *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 78–91, Jan./Feb. 2020.

[28] H. Krawczyk, “Simple forward-secure signatures from any signature scheme,” in *Proc. 7th ACM Conf. Comput. Commun. Secur.*, 2000, pp. 108–115.

[29] R. L. Rivest, A. Shamir, and D. A. Wagner, “Time-lock puzzles and timed-release crypto,” 1996. [Online]. Available: https://hdl.handle.net/1721.1/149822

[30] N. McCullagh and P. S. L. M. Barreto, “Efficient and forward-secure identity-based signcryption,” Cryptology ePrint Archive, Report 2004/117, 2004. [Online]. Available: https://eprint.iacr.org/2004/117

[31] A. Hülsing and F. Weber, “Epochal signatures for deniable group chats,” in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 1677–1695.

[32] J. C. Choon and J. Hee Cheon, “An identity-based signature from gap Diffie-Hellman groups,” in *Proc. Int. Workshop Public Key Cryptography*, Springer, 2003, pp. 18–30.

[33] J. Liu, J. Yang, W. Wu, X. Huang, and Y. Xiang, “Lightweight authentication scheme for data dissemination in cloud-assisted healthcare IoT,” *IEEE Trans. Comput.*, vol. 72, no. 5, pp. 1384–1395, 2023.

[34] L. Zhang, W. You, and Y. Mu, “Secure outsourced attribute-based sharing framework for lightweight devices in smart health systems,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 3019–3030, Sep./Oct. 2022.

[35] L. Yang, Q. Zheng, and X. Fan, “RSPP: A reliable, searchable and privacy-preserving e-healthcare system for cloud-assisted body area networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[36] X. Ge, J. Yu, R. Hao, and H. Lv, “Verifiable keyword search supporting sensitive information hiding for the cloud-based healthcare sharing system,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 8, pp. 5573–5583, Aug. 2022.

[37] A. S. Rajput and B. Raman, “Privacy-preserving distribution and access control of personalized healthcare data,” *IEEE Trans. Ind. Inform.*, vol. 18, no. 8, pp. 5584–5591, Aug. 2022.

[38] X. Liu, R. H. Deng, K.-K. R. Choo, and Y. Yang, “Privacy-preserving outsourced clinical decision support system in the cloud,” *IEEE Trans. Serv. Comput.*, vol. 14, no. 1, pp. 222–234, Jan./Feb. 2021.

[39] Z. Ma et al., “Lightweight privacy-preserving medical diagnosis in edge computing,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1606–1618, May/Jun. 2022.

[40] W. Zhang, Y. Lin, J. Wu, and T. Zhou, “Inference attack-resistant e-healthcare cloud system with fine-grained access control,” *IEEE Trans. Serv. Comput.*, vol. 14, no. 1, pp. 167–178, Jan./Feb. 2021.

Qing Fan received the PhD degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2022. She is currently a postdoctoral research fellow with the School of Cyberspace Science and Technology, Beijing Institute of Technology. Her research interests include applied cryptography and information security.

Yumeng Xie received the BS degree in applied mathematics from the Beijing University of Technology, in 2022. She is currently working toward the master's degree with the School of Computer Science and Technology, Beijing Institute of Technology. Her research interests include applied cryptography and secure data services in cloud computing.

Chuan Zhang (Member, IEEE) received the PhD degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2021. From 2019 to 2020, he worked as a visiting PhD student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently an assistant professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include secure data services in cloud computing, applied cryptography, machine learning, and blockchain.

Ximeng Liu (Senior Member, IEEE) received the BSc degree in electronic engineering from Xidian University, Xi'an, China, in 2010, and the PhD degree in cryptography from Xidian University, China, in 2015. Currently, he is a full professor with the College of Mathematics and Computer Science, Fuzhou University, China. Also, he is a research fellow with the School of Information System, Singapore Management University, Singapore. His research interests include cloud security, applied cryptography and Big Data security.

Liehuang Zhu (Senior Member, IEEE) received the PhD degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004. He is currently a professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.