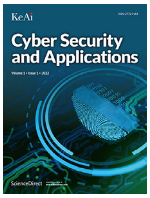
Cyber Security and Applications 1 (2023) 100018

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/



Sanitizable signature scheme with privacy protection for electronic medical data sharing☆

Zhiyan Xu[a], Min Luo[b,*], Cong Peng[b], Qi Feng[b]

[a] The Hubei Education Cloud Service Engineering Technology Research Center, College of Computer, Hubei University of Education, Wuhan, 430205, Hubei province, China

[b] The Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, Hubei province, China

ARTICLE INFO

Keywords:
Electronic medical records
Sanitizable signature
Privacy protection
Provable security

ABSTRACT

Electronic medicine has received more and more attention because of its ability to provide more efficient and better medical services. However, the characteristics of electronic medical networks make them more vulnerable to security threats such as data integrity and user privacy leakage. Traditional digital signatures cannot meet the diversity and privacy requirements of medical data applications. Sanitizable signatures incorporate sanitization capabilities into signatures to allow designated sanitizers to modify variable parts of a message in a controlled manner without the cooperation of the original signer. This paper uses the key-exposure free chameleon hash function to convert the data sanitization operation into using trapdoor keys to find collisions in the key-exposure free chameleon hash function, and builds a privacy-preserving sanitizable signature scheme. Security analysis and performance evaluation demonstrate that our new scheme achieves public verifiability, which greatly reduces computing costs while effectively ensuring data security and user privacy, and is especially suitable for electronic medical data sharing scenarios.

1. Introduction

Electronic medicine is a reshaping and adjustment of the traditional medical model by the Internet [1]. It can break through the limitations of objective factors such as time and space, and objects, establish a new medical system that adapts to the information age, and at the same time promote the information management of medical records [2]. Compared with the traditional medical model, the value of electronic medical records is no longer limited to the application scope of medical treatment, scientific research and teaching, and more involved in hospital management, insurance claims, judicial evidence collection and other fields [3,4].

In today's global economic and technological integration, electronic medicine and data sharing have become a new trend in the development of medical and health services, and have been highly valued by governments of many countries. Medical records contain a large amount of patient health data, and the authenticity and availability of information are critical to the correct use of medical data. A slight error will endanger the patient's life and property safety, resulting in irreparable losses. Data security and availability in electronic medical data sharing has become a huge challenge for the development of electronic medical health [5].

Many researchers have proposed many cryptographic schemes from the technical level to solve the above problems, among which digital signature is an important means to protect the authenticity and availability of data [6]. Doctors can use digital signatures to ensure the integrity, authenticity and availability of patient medical records. However, not all medical data sharing applications require access to patients' complete medical records. For example, when medical records are used for patient medical reimbursement, insurance company accountants only need real information about the patient's treatment and insurance numbers, but not the rest of the patient's medical record [7].

Unnecessary medical information leakage will lead to patients suffering from unpredictable hazards such as biometric information leakage, telephone scams, and spam sales [8]. To protect patient privacy, one solution is to require doctors to sign only information relevant to medical reimbursement. However, every time there is a new subset of the medical record that needs to be shared, the signing doctor needs to

☆ The work was supported by the National Natural Science Foundation of China (Nos. 61902115, 61972294, 61932016), the Natural Science Foundation of Hubei Province (No. 2021CFB584).

[*] Corresponding author.
E-mail address: mluo@whu.edu.cn (M. Luo).

https://doi.org/10.1016/j.csa.2023.100018
Received 7 August 2022; Received in revised form 1 January 2023; Accepted 19 April 2023
Available online 23 April 2023

repeat the signing process, which will result in excessive computation overhead, and may even make it impossible to re-sign the document due to reasons such as the departure of the doctor concerned [9].

Sanitizable signature [10] is a type of digital signature that supports controlled modification of signed messages. It can break through the limitations of traditional digital signatures and support designated sanitizers to modify the signed messages as required, and this process does not require any interaction with the signer. These characteristics enable sanitizable signature to not only ensure the integrity, authenticity, and availability of medical data, but also effectively hide sensitive patient information.

1.1. *Our research contributions*

In this paper, we put forward a sanitizable signature scheme with privacy protection for electronic medical data sharing (SSPM) which could better support the integrity, authenticity of medical data and effectively hide sensitive patient information. The main contributions are as belows.

- *Firstly*, we propose a typical system model of electronic medical data sharing scenarios that is closer to practical applications.
- *Secondly*, we present a SSPM scheme which can meet the actual security needs in the electronic medical data sharing.
- *Finally*, we provide detailed security analysis and performance evaluation for our SSPM scheme.

1.2. *Organization of the paper*

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 describes the problem statement related to our SSPM scheme, followed by details of the proposed SSPM scheme for electronic medical data sharing in Section 4. In Sections 5 and 6, the security analysis and performance evaluation of the scheme are followed. Finally, we present the conclusions of this paper in the last section.

2. Related work

Standard digital signature does not allow any form of modification to the signed message [11], so it cannot take into account the needs of data integrity and privacy protection in electronic medical data sharing scenarios. In order to solve the above problems, malleable digital signature technology has received extensive attention in recent years. The existing research work can be roughly divided into three directions: homomorphic signatures [12], redactable signatures [13] and sanitizable signatures [14].

Homomorphic signatures [15] take multiple signed messages as input and can be used to compute functions on authenticated datasets. In this signature mechanism, any entity can derive a valid signature about $f(m)$, but complex homomorphic operations make such signatures inefficient. The idea of redactable signature comes from the literature [16], and then literatures [17] and [18] respectively give its formal definition, the signature mechanism allows anyone who holds the signature verification public key to delete the message specified in the signed message block and derive a new signature, but it only supports delete operations on message blocks, and does not support the accountability property.

The concept of sanitizable signatures was first proposed in 2005 by Ateniese et al. [10] based on standard digital signature schemes and chameleon hash [19]. It enables a semi-trusted entities called sanitizer to modify signed message blocks in a controlled manner. Let σ be a valid signature of the message $M = \{m_1, m_2, \dots, m_n\}$, the sanitizer can modify the message block m_i that is allowed to be modified into a new message block m_i' without breaking the verifiability of the message signature. Sanitization is essentially that the sanitizer uses a trapdoor to find conflicts in the chameleon hash function without key exposure. Even if some message blocks are changed, the signature corresponding to the message will not be changed, so the original signature remains valid for the sanitized message [20]. Sanitizable digital signatures allow sanitizers to have their own keys and can derive new messages and corresponding signatures, providing more flexibility than redactable signatures.

Brzuska et al. [9] presented the first formalized security model for sanitizable signature schemes, which defined five basic security properties of sanitizable signature schemes. Gong et al. [21] analyzed the formal security model proposed in the literature [9], pointed out that the security model is vulnerable to rights forgery attack, and gave new definitions of security properties such as unforgeability, immutability, and accountability. Subsequently, Krenn et al. [22] conducted further research on the above model, and present more secure property variants of unforgeability, privacy, transparency, and accountability.

Unlinkability was introduced by Brzuska et al. [23] as a privacy-preserving property, which can ensure that anyone other than the signer and the sanitizer cannot obtain any information about the new signature after sanitization, even after the original signature has been knowing. Pohls et al. proposed the concept of the hidden property [18], which means that outsiders cannot know which parts of the signed message are allowed to be modified, and the literature [24,25] further gave a formal definition of this property. At present, sanitizable signature schemes that satisfy invisibility and unlinkability respectively already exist, but whether the two security properties can be integrated is still an open question.

3. Preliminaries

In this paper, we use k to represent the system security parameters, $[n]$ represents the set $\{1, 2, 3, \dots, n\}$, M represents the message space. Next, we first present key-exposure free chameleon hash function, and then the system model and the framework of our scheme are introduced. At last, security requirements of our scheme are followed.

3.1. *Key-Exposure free chameleon hash function*

The key-exposure free chameleon hash function (denoted as CH) introduced in our proposed scheme of this paper mainly includes the following algorithms [26,27].

- $KGen_z(1^\kappa) \rightarrow (sk_z, P_z)$ is a polynomial time algorithm, where κ is the security parameter, sk_z is the trapdoor key of the function CH, and P_z is the commitment key of the function CH.
- $Eval(m_i, i, \zeta_i, P_z) \rightarrow h_i$ is a polynomial-time evaluation algorithm, where m_i is a message block, i is the index of message block m_i, ζ_i is a random number, P_z is the commitment key of the function CH, and h_i is a hash value.
- $Inv(m_i', i, h_i, sk_z) \rightarrow \zeta_i'$ is a trapdoor collision finding algorithm, where m_i' is a different message, i is the data block index, h_i is the chameleon hash value on the input (m_i, ζ_i), sk_z is the trapdoor key, and ζ_i' is a random number such that $Eval(m_i', i, \zeta_i', P_z) = h_i$.
- $Vrfy(P_z, h_i, m_i, \zeta_i) \rightarrow 1/0$ is a verification algorithm, where P_z is the commitment key, h_i is a chameleon hash value, m_i is a message block, ζ_i is a random number. If $h_i = Eval(m_i, i, \zeta_i, P_z)$ holds, then outputs 1, otherwise outputs 0.

3.2. *System model*

As demonstrated in Fig. 1, we give the system model of our SSPM scheme. According to the practical application requirements of the electronic medical system, there are the following types of participants in the sanitizable signature scheme.

- *Signer.* The signer is responsible for generating the original signature for the patient's medical record, designating a sanitizer for the signed message to be shared, and for generating a proof of the message-signature pair so that judges use to decide who is responsible for the message-signature pair in the event of ambiguity.

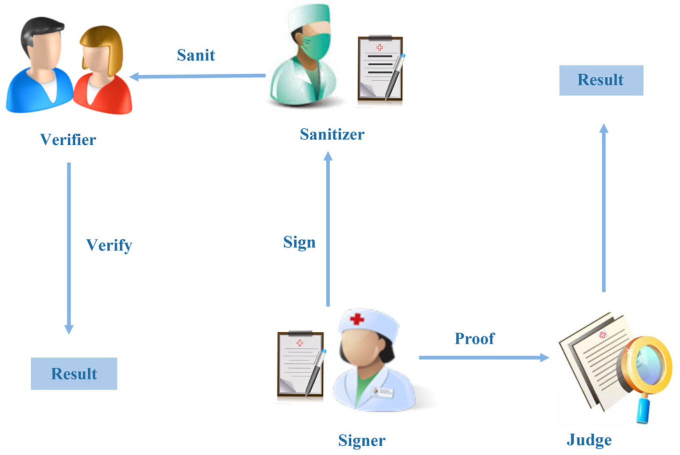


Fig. 1. The system model of our SSPM scheme for electronic medical scenario .

- *Sanitizer.* The sanitizer is responsible for performing the message update operation on the signed message that is allowed to be modified, and is also responsible for deriving a new message-signature pair for the sanitized message.
- *Verifier.* The verifier is responsible for verifying the validity of the sanitized message-signature pair, and the output result is $1/0$, where "1" indicates that the verification passed and the data is valid, and "0" indicates that the verification failed and the data is invalid.
- *Judge.* The judge decides which of the signer and the sanitizer should be responsible for the final message-signature pair based on the proof generated by the signer, and the output result is $\pi \in \{1, 0\}$, where "1" means that the signer is responsible for the message-signature pair, and "0" indicates that the sanitizer is responsible for the message-signature pair.

3.3. Framework of the SSPM scheme

Our proposed SSPM scheme is a collection of the following PPT algorithms as below.

- *Setup*$(1^\kappa) \rightarrow params$ is an initialization algorithm, where κ is the security parameter, $params$ is the system parameter list.
- $KGen_s$ $(params, ID_S) \rightarrow (sk_s, P_s)$ is a probabilistic algorithm executed by the signer, where $ID_S \in \{0,1\}^*$ is the identity and (sk_s, P_s) is the key pair of the signer S.
- $KGen_z$ $(params, ID_Z) \rightarrow (sk_z, P_z)$ is a probabilistic algorithm executed by the sanitizer, where $ID_Z \in \{0,1\}^*$ is the identity and (sk_z, P_z) is the key pair of the sanitizer Z.
- $Sign$ $(params, sk_s, P_z, m) \rightarrow \sigma_m$ is a signature algorithm executed by the signer, where m is the message, and σ_m is the signature on the message m.
- $Sanit$ $(params, sk_z, P_z, \sigma_m, Mut) \rightarrow \sigma_m^{'}$ is a sanitization algorithm executed by the sanitizer, where $m^{'}$ is the sanitized message, Mut is a description of information that needs to be sanitized on m, and $\sigma_m^{'}$ is the sanitized message-signature pair.
- *Verify*$(P_s, P_z, \sigma_m) \rightarrow \{0/1\}$ is a verification algorithm executed by any third party, where 1 or 0 as outputs to indicate whether the signature σ_m is validated.
- *Proof*$(sk_s, P_z, \sigma_m) \rightarrow \pi$ is a proof algorithm executed by the signer, where π as outputs to indicate whether the signature σ_m is generated by the signer or the sanitizer.

3.4. Security requirements

A SSPM scheme need to meet the following functions and security requirements:

- *Correctness*. To ensure that each signature generated by the signer or the sanitizer in our SSPM scheme can be correctly verified.
- *Unforgeability*. To ensure that an adversary cannot forge a legal signature without knowing the signer's private key or obtaining the trapdoor of chameleon hash.
- *Public verifiability*. To ensure that the message signature pair can be verified by any third party.
- *Accountability*. To determine which party is responsible for a given message-signature pair based on the $Proof$ generated by the signer.
- *Privacy*. To ensure that patient sensitive information is protected to the greatest extent possible during medical data sharing.
- *Immutability*. To ensure that the sanitizer can only modify the parts of the message that are allowed to be modified.

4. Our proposed SSPM scheme

To slove the security and privacy issues in electronic medical system while improving the efficiency, we propose a SSPM scheme which includes seven phases: $Setup$, $KGen_s$, $KGen_z$, $Sign$, $Sanit$, $Verify$ and $Proof$. The details are described as below.

4.1. Setup

A system parameter list is generated after obtaining the security parameter k by executing $Setup$.

1. Generate a cyclic group G with the prime order q, and g is a generator of G.
2. Select a key-exposure free chameleon hash function CH.
3. Randomly select h_1, h_2:$\{0,1\} \rightarrow Z_q$.
4. Publish the parameter list $Params = (G, q, g, CH, h_1, h_2)$.

4.2. $KGen_s$

The signer produces his/her key pair by executing $KGen_s$.

1. Randomly select $sk_s \in Z_q^*$ as the secret key and keep secret.
2. Calculates $P_s = g^{sk_s}$ as the public key and keep public.

4.3. $KGen_z$

The santizer produces his/her key pair by executing $KGen_z$.

1. Randomly select $sk_z \in Z_q^*$ as the secret key and keep secret.
2. Calculates $P_z = g^{sk_z}$ as the public key and keep public.

4.4. Sign

The signer S produces a message-signature pair σ_m on the message m by executing $Sign$.

1. Inputs parameter list $params$, signer's secret key sk_s, sanitizer's public key P_z, and message m.
2. Divide m into n blocks, that is, $m = \{m_i\}_{i\in[n]}$.
3. Set a variable block index Mut of size t and the mutable message blocks as $\{m_i\}_{i\in Mut}$.
4. For $i \in [n]\backslash Mut$, computes $h_i = h_1(m_i, P_s, P_z)$.
5. For $i \in Mut$, randomly select $\zeta_i \in Z_q$, set $\zeta = \{\zeta_i\}_{i\in Mut}$ and compute $h_i = Eval(m_i, i, \zeta_i, P_z)$.
6. Set $h_m = \{h_i\}_{i\in[n]}$ and randomly select $\omega \in Z_q$.
7. Compute $u = h_2(g^\omega, P_z, h_m)$ and $v = \omega/(u + sk_s)$.
8. Returns the message-signature pair $\sigma_m = (m, \zeta, h_m, u, v)$.

4.5. Sanit

The sanitizer Z produces a new message-signature pair $\sigma_m^{'}$ on the message $m^{'}$ by executing $Sanit$.

1. Inputs parameter list $params$, the sanitizer's key pair P_z, sk_z, the mutable block index Mut and the message-signature pair $\sigma_m = (m, \zeta, h_m, u, v)$.
2. For all $i \in Mut$, randomly select $m_i' \in M$ and compute $\zeta_i' = Inv(m_i', i, h_i, sk_z)$, make $Eval(m_i', i, \zeta_i', P_z) = h_i = Eval(m_i, i, \zeta_i, P_z)$
3. Set $m' = (\{m_i\}_{i\in[n]\backslash Mut}, \{m_i'\}_{i\in Mut})$ as the sanitzied message
4. Set $\zeta' = \{\zeta_i'\}_{i\in Mut}$.
5. Returns the sanitzied message-signature tuple $\sigma_m' = (m', \zeta', h_m, u, v)$.

4.6. *PVerify*

The any third party verifies the signature $\sigma_m = (m, \zeta, u, v)$ by executing $PVerify$.

1. Inputs system parameters $params$, public key pairs (P_s, P_z) and message-signarure pairs $\sigma_m = (m, \zeta, h_m, u, v)$.
2. If $\forall i \in Mut$, we have $Vrfy(P_z, i, m_i, h_i, \zeta_i) = 1$ and $\forall i \in [n]\backslash Mut$, we have $h_i = h_1(m_i, P_s, P_z)\ P_s, P_z)$, go to the next step, otherwise reject.
3. Verify

$$v = h_2((P_s g^u)^v, P_z, h_m) \tag{1}$$

3. If $Equ.(1)$ holds, emits "1" and accept σ_m; Otherwise emit "0" and reject.

4.7. *Proof*

The signer produces a proof indicating who is responsible for the signature by executing $Proof$.

1. Inputs system parameters $params$, key pairs (sk_s, P_s), the sanitizer's public key P_z, and the message m^*.
2. If $m^* = m$, return a proof $\pi = 1$, indicting that m^* originates from the signer.
3. If $m^* \neq m$ and has the same signature and hash values with m, return a proof $\pi = 0$, indicting that m^* is generated by the sanitizer.

5. Security analysis

In this section, we analyze the security of our presented SSPM scheme. We first give the correctness of our proposed scheme, and then demonstrate that our proposal can satisfy correctness, unforgeability, public verifiability, accountability, privacy, immutability.

5.1. *Correctness*

Our SSPM scheme satisfies correctness. From the construction of the SSPM scheme, we can find that the verification process may have the following two cases:

1. If an original signature $\sigma_m = (m, \zeta, h_m, u, v)$ generated by the signer via executing the $Sgin$ algorithm, then verify σ_m is valid as the belowing steps:
   a. If $\forall i \in Mut$, then we have $Vrfy(P_z, i, m_i, h_i, \zeta_i) = 1$.
   b. If $\forall i \in [n]\backslash Mut$, then we have $h_i = h_1(m_i, P_s, P_z)\ P_s, P_z)$, go to the next step, otherwise reject.
   c. Verify

$$v = h_2((P_s g^u)^v, P_z, h_m) \tag{2}$$

2. If a sanitized signature $\sigma_m' = (m', \zeta', h_m, u, v)$ generated by the the sanitizer via executing the *Sanit* algorithm, then verify σ_m' is valid as the belowing steps:
   a. If $\forall i \in Mut$, then we have $Vrfy(P_z, i, m_i', h_i, \zeta_i') = 1$.
   b. If $\forall i \in [n]\backslash Mut$, we have $h_i = h_1(m_i, P_s, P_z)$, go to the next step, otherwise reject.
   c. Verify

$$v = h_2((P_s g^u)^v, P_z, h_m) \tag{3}$$

Clearly, both σ_m and σ_m' pass the above verification and are valid. Therefore, the correctness of our scheme holds.

5.2. *Unforgeability*

Our SSPM scheme can meet the unforgeability. According to the characteristics of chameleon hash, with the exception of sanitizers authorized to forge signatures of sanitized message blocks in a controlled manner, other adversaries cannot successfully forge message signatures without obtaining the trapdoor of chameleon hash, and the trapdoor is kept secret, which means no one can forge the signature generated by the signer or sanitizer. Therefore, the unforgeability of our scheme holds.

5.3. *Public verifiability*

Our SSPM scheme can meet the public verifiability. From the PVerify algorithm we can find that the verification process only needs the participation of (P_s, P_z) and message-signarure pairs (m, ζ, h_m, u, v). That is, any third party can verify the authenticity of $\sigma_m = (m, \zeta, h_m, u, v)$ without the recipient's secret key. Therefore, the public verifiability of our scheme holds.

5.4. *Accountability*

Our SSPM scheme can meet accountability. From the $Proof$ algorithm, we can find that the original signer can follow the algorithm steps to produce convincing evidence to confirm which one of the signer and the sanitizer is responsible for the message-signature pair $\sigma_m = (m, \zeta, h_m, u, v)$. Therefore, the accountability of our scheme holds.

5.5. *Privacy*

Our SSPM scheme can meet the privacy. From the *Sanit* algorithm, The SSPM scheme proposed in this paper can effectively hide the sensitive information of patients and protect the privacy of patients to the greatest extent while ensuring that the data integrity can be verified through the document sanitization operation. Therefore, the accountability of our scheme holds.

5.6. *Immutability*

Our SSPM scheme satisfies immutability. From the $Sign$ algorithm, we can find that in the message signature pair $\sigma_m = (m, \zeta, h_m, u, v)$ generated by the signer, the unmodifiable message blocks are hashed using the collision resistant hash function h_1. In sanitized message signature pair $\sigma_m' = (m', \zeta', h_m, u, v)$, if m' produced by the sanitizer contains an unmodifiable block m_i', and $i \notin Mut$, the following two cases may exist:

1. $\sigma_m' = (m', \zeta', h_m, u, v)$, i.e. $m' \neq m$ and $(u', v') = (u, v)$, that means m' and m have the same hash value under the hash function h_1, which violates the collision resistance of h_1.
2. $(m', \zeta', h_m, u', v') \neq (m, \zeta, h_m, u, v)$, that means the sanitizer forged the signature (u', v') which violates the unforgeability of digital signatures used in the $Sign$ algorithm.

Therefore, the immutability of our scheme holds.

Table 1
Running time of different operations(ms).

Operations	T_{ma}	T_{mm}	T_{inv}	T_{hash}	$T_{\exp}$
Running time	0.0008	0.0011	0.189	0.003	0.913

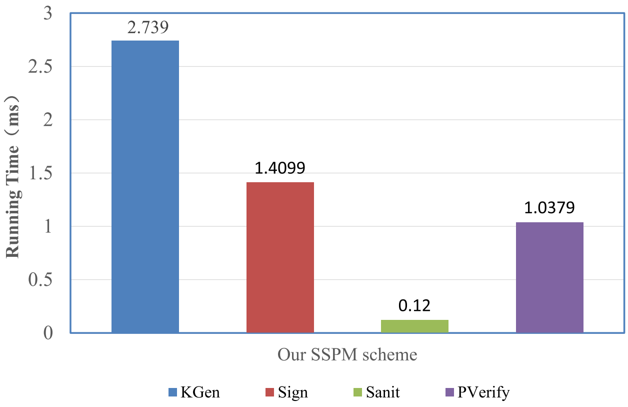


Fig. 2. Computation overhead of different phases in our SSPM scheme.

6. Performance evaluation

In this section, we analyze the performance of our proposed SSPM scheme and we mainly focus on the computation overhead of the scheme. For the convenience of description, let n denote the number of message blocks in the medical record and t denote the number of sanitized data blocks. where n and t are both small constants.

To achieve a credible security level, we set security parameter κ = 80 bits and select a non-singular elliptic curve $E : y^2 = x^3 + ax + b \bmod q$ for the schemes, where $a, b \in Z_q^*$, G is a group with order q on E, p and q are both prime numbers with a length of 160 bits. We use the MIRACL library [28] on a laptop computer (Intel core with I7-4770@3.4GHz CPU, 4GB random memory, and Windows7 operating system) to simulate our SSPM scheme.

Let T_{ma} represent a modular addition operation in Z_q, T_{mm} represent a modular multiplication operation in Z_q, T_{inv} represent a modular inverse operation in Z_q, $T_{\exp}$ represent an exponential operation in G, T_{mul} represent a scalar multiplication operation in G, T_{hash} represent a general hash operation in Z_q. The running time of different operations is shown in Table 1.

Since the computation overhead of $Setup$ and $Proof$ phases are almost negligible, we only consider the computation overhead of the $KGen_s$, $KGen_z$, $Sign$, $Sanit$ and $Verify$ phases in the scheme, where we use $KGen$ to denote $KGen_s$ and $KGen_z$ for convenience. The detailed analysis is as follows.

In $KGen_s$ and $KGen_z$ phases, we can find that our scheme needs to perform one exponentiation operation in G respectively. Therefore, the total cost of the $KGen$ phase is $2T_{\exp}$. In $Sign$ phase, we can find that our scheme needs to perform $(n+2)$ general hash operations, one modular addition operation, one modular multiplication operation, one modular inverse operation, and one exponentiation operation in G. Therefore, the total cost of the $Sgin$ phase is $(n+2)T_{hash} + T_{ma} + T_{mm} + T_{inv} + T_{\exp}$.

In *Sanit* phase, we can find that our scheme only needs to perform $2t$ general hash operations. Therefore, the total cost of the *Sanit* phase is $2tT_{hash}$. In $PVerify$ phase, we can find that our scheme needs to perform $(2t+1)$ general hash operations, one modular addition operation, one modular multiplication operation, and and one exponentiation operation in G. Therefore, the total cost of the $PVerify$ phase is $(2t+1)T_{hash} + T_{ma} + T_{mm} + T_{\exp}$.

As shown in Fig. 2, set $n = 100$ and $t = 20$, we can observe that the total computation cost of the $KGen$ phase is 1.826 ms, the computation cost of the $Sign$ phase in our SSPM scheme is 1.4099 ms, and the computation cost of the $Sanit$ phase is 0.12 ms, the computation cost of the $PVerify$ phase is 1.0379 ms. Obviously, the total computation cost of exponential operations, general hash operations, modular addition and modular multiplication operations involved in our scheme is also constant. Based on the above considerations, we consider our sterilizable signature scheme to be practical.

7. Conclusion

With the continuous development of electronic medicine, the application field of medical records is further expanded. Since medical records contain large amounts of health data and patient privacy, and security and privacy in medical data sharing are paramount. To solve these problems, we propose a sanitizable signature scheme, which is based on the key-exposure free chameleon hash, and converts the sanitization process of the message to use the trapdoor key to find collisions in the Chameleon hash function. Security analysis and performance evaluation demonstrate show that our SSPM scheme can not only guarantee the integrity of medical data, but also support patient privacy protection. Further, the scheme achieves public verification, which greatly reduces the computation cost of the receiver and is easier to deployment in electronic medical data sharing scenarios.

Declaration of Competing Interests

The authors declare that they have no conflict of interest.

CRediT authorship contribution statement

Zhiyan Xu: Validation, Data curation, Writing – original draft, Data curation. **Min Luo:** Conceptualization, Methodology, Writing – review & editing, Supervision.

References

[1] H. Wen, M. Wei, D. Du, X. Yin, A blockchain-based privacy preservation scheme in mobile medical, Secur. Commun. Netw. 2022 (2022).
[2] Z. Xu, M. Luo, N. Kumar, P. Vijayakumar, L. Li, Privacy-protection scheme based on sanitizable signature for smart mobile medical scenarios, Wirel. Commun. Mobile Comput. 2020 (2020).
[3] A.K. Jha, C.M. DesRoches, E.G. Campbell, K. Donelan, S.R. Rao, T.G. Ferris, A. Shields, S. Rosenbaum, D. Blumenthal, Use of electronic health records in us hospitals, N top N. Engl. J. Med. 360 (16) (2009) 1628–1638.
[4] A. Hoerbst, E. Ammenwerth, Electronic health records, Methods Inf. Med. 49 (04) (2010) 320–336.
[5] J.L. Fernández-Alemán, I.C. Señor, P.Á.O. Lozoya, A. Toval, Security and privacy in electronic health records: asystematic literature review, J. Biomed. Inform. 46 (3) (2013) 541–562.
[6] R. Guo, H. Shi, Q. Zhao, D. Zheng, Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems, IEEE Access 6 (2018) 11676–11686.
[7] R.B. Ness, J.P. Committee, et al., Influence of the HIPAA privacy rule on health research, JAMA 298 (18) (2007) 2164–2170.
[8] D. Wartenberg, W.D. Thompson, Privacy versus public health: the impact of current confidentiality rules, Am. J. Public Health 100 (3) (2010) 407–412.
[9] C. Brzuska, M. Fischlin, T. Freudenreich, A. Lehmann, M. Page, J. Schelbert, D. Schröder, F. Volk, Security of sanitizable signatures revisited, in: International Workshop on Public Key Cryptography, Springer, 2009, pp. 317–336.
[10] G. Ateniese, D.H. Chou, B.d. Medeiros, G. Tsudik, Sanitizable signatures, in: European Symposium on Research in Computer Security, Springer, 2005, pp. 159–177.
[11] Z. Xu, M. Luo, M.K. Khan, K.-K.R. Choo, D. He, Analysis and improvement of a certificateless signature scheme for resource-constrained scenarios, IEEE Commun. Lett. 25 (4) (2020) 1074–1078.
[12] C. Xie, J. Weng, D. Zhou, Revocable identity-based fully homomorphic signature scheme with signing key exposure resistance, Inf. Sci. (Ny) 594 (2022) 249–263.
[13] O. Sanders, Efficient redactable signature and application to anonymous credentials, in: IACR International Conference on Public-Key Cryptography, Springer, 2020, pp. 628–656.
[14] K. Samelin, D. Slamanig, Policy-based sanitizable signatures, in: Cryptographers Track at the RSA Conference, Springer, 2020, pp. 538–563.
[15] D. Catalano, D. Fiore, L. Nizzardo, On the security notions for homomorphic signatures, in: International Conference on Applied Cryptography and Network Security, Springer, 2018, pp. 183–201.
[16] R. Steinfeld, L. Bull, Y. Zheng, Content extraction signatures, in: International Conference on Information Security and Cryptology, Springer, 2001, pp. 285–304.

[17] C. Brzuska, H. Busch, O. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, et al., Redactable signatures for tree-structured data: definitions and constructions, in: International Conference on Applied Cryptography and Network Security, Springer, 2010, pp. 87–104.

[18] A. Kundu, E. Bertino, Privacy-preserving authentication of trees and graphs, Int. J. Inf. Secur. 12 (6) (2013) 467–494.

[19] S. Krenn, H.C. Pöhls, K. Samelin, D. Slamanig, Chameleon-hashes with dual long-term trapdoors and their applications, in: International Conference on Cryptology in Africa, Springer, 2018, pp. 11–32.

[20] L. Jiguo, Z. Liufu, L. Chengdong, L. Yang, H. Jinguang, W. Huaqun, Z. Yichen, Provably secure traceable attribute-based sanitizable signature scheme in the standard model, J. Comput. Res. Dev. 58 (10) (2021) 2253.

[21] J. Gong, H. Qian, Y. Zhou, Fully-secure and practical sanitizable signatures, in: International Conference on Information Security and Cryptology, Springer, 2010, pp. 300–317.

[22] S. Krenn, K. Samelin, D. Sommer, Stronger security for sanitizable signatures, in: Data Privacy Management, and Security Assurance, Springer, 2015, pp. 100–117.

[23] C. Brzuska, M. Fischlin, A. Lehmann, D. Schröder, Unlinkability of sanitizable signatures, in: International Workshop on Public Key Cryptography, Springer, 2010, pp. 444–461.

[24] J. Camenisch, D. Derler, S. Krenn, H.C. Pöhls, K. Samelin, D. Slamanig, Chameleon-hashes with ephemeral trapdoors, in: IACR International Workshop on Public Key Cryptography, Springer, 2017, pp. 152–182.

[25] M.T. Beck, J. Camenisch, D. Derler, S. Krenn, H.C. Pöhls, K. Samelin, D. Slamanig, Practical strongly invisible and strongly accountable sanitizable signatures, in: Australasian Conference on Information Security and Privacy, Springer, 2017, pp. 437–452.

[26] W. Gao, F. Li, X. Wang, Chameleon hash without key exposure based on Schnorr signature, Comput. Stand. Interfaces 31 (2) (2009) 282–285.

[27] X. Chen, F. Zhang, W. Susilo, H. Tian, J. Li, K. Kim, Identity-based chameleon hash scheme without key exposure, in: Australasian Conference on Information Security and Privacy, Springer, 2010, pp. 200–215.

[28] D.F. Pigatto, N.B.F. da Silva, K.R.L.J.C. Branco, Performance evaluation and comparison of algorithms for elliptic curve cryptography with El-Gamal based on MIRACL and RELIC libraries, J. Appl. Comput. Res. 1 (2) (2011) 95–103.