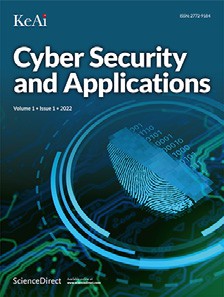
[Cyber Security and Applications 1 (2023) 100018](https://doi.org/10.1016/j.csa.2023.100018)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

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[☆](#_bookmark2)

Zhiyan Xu[a](#_bookmark0), Min Luo [b](#_bookmark1),[∗](#_bookmark3), Cong Peng [b](#_bookmark1), Qi Feng [b](#_bookmark1)

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*

a r t i c l e i n f o a b s t r a c t

*Keywords:*

Electronic medical records Sanitizable signature Privacy protection Provable security

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.

# Introduction

Electronic medicine is a reshaping and adjustment of the traditional medical model by the Internet [[1]](#_bookmark11). It can break through the limita- tions of objective factors such as time and space, and objects, estab- lish a new medical system that adapts to the information age, and at the same time promote the information management of medical records [[2]](#_bookmark12). Compared with the traditional medical model, the value of elec- tronic 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]](#_bookmark13).

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 govern- ments 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 endan- ger the patient’s life and property safety, resulting in irreparable losses.

Data security and availability in electronic medical data sharing has be- come a huge challenge for the development of electronic medical health [[5]](#_bookmark14).

Many researchers have proposed many cryptographic schemes from the technical level to solve the above problems, among which digital sig- nature is an important means to protect the authenticity and availability of data [[6]](#_bookmark15). Doctors can use digital signatures to ensure the integrity, au- thenticity 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]](#_bookmark16).

Unnecessary medical information leakage will lead to patients suf- fering from unpredictable hazards such as biometric information leak- age, telephone scams, and spam sales [[8]](#_bookmark17). 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.](https://doi.org/10.13039/501100003819) [61902115](#_bookmark2)[,](https://doi.org/10.13039/501100003819) [61972294](#_bookmark2)[,](https://doi.org/10.13039/501100003819) [61932016](#_bookmark2)[), the Natural Science Foundation of Hubei Province (No.](https://doi.org/10.13039/501100003819) [2021CFB584](#_bookmark2)[).](https://doi.org/10.13039/501100003819)

∗ Corresponding author.

*E-mail address:* [mluo@whu.edu.cn](mailto: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

2772-9184/© 2023 The Authors. Publishing Services by Elsevier B.V. on behalf of KeAi Communications Co., Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

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]](#_bookmark18).

Sanitizable signature [[10]](#_bookmark19) 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 san- itizers 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. *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 ef- fectively 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 se- curity needs in the electronic medical data sharing.
    - *Finally*, we provide detailed security analysis and performance eval- uation for our SSPM scheme.
  1. *Organization of the paper*

The rest of this paper is organized as follows. [Section 2](#_bookmark5) presents re- lated work. [Section 3](#_bookmark4) 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](#_bookmark6). In [Sections 5](#_bookmark8) and 6, the se- curity analysis and performance evaluation of the scheme are followed. Finally, we present the conclusions of this paper in the last section.

# Related work

Standard digital signature does not allow any form of modification to the signed message [[11]](#_bookmark20), 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 sig- nature technology has received extensive attention in recent years. The existing research work can be roughly divided into three directions: ho- momorphic signatures [[12]](#_bookmark21), redactable signatures [[13]](#_bookmark22) and sanitizable signatures [[14]](#_bookmark23).

sanitized message [[20]](#_bookmark30). 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]](#_bookmark18) presented the first formalized security model for sanitizable signature schemes, which defined five basic security prop- erties of sanitizable signature schemes. Gong et al. [[21]](#_bookmark32) analyzed the formal security model proposed in the literature [[9]](#_bookmark18), 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]](#_bookmark33) conducted further research on the above model, and present more secure property vari- ants of unforgeability, privacy, transparency, and accountability.

Unlinkability was introduced by Brzuska et al. [[23]](#_bookmark34) 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]](#_bookmark28) further gave a for- mal 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.

# Preliminaries

In this paper, we use *𝑘* to represent the system security parameters,

[*𝑛*] represents the set {1*,* 2*,* 3*,* … *, 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.

* 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]](#_bookmark31).

* + - *𝐾𝐺𝑒𝑛𝑧* (1*𝜅* ) → (*𝑠𝑘𝑧 , 𝑃𝑧* ) is a polynomial time algorithm, where *𝜅* is the security parameter, *𝑠𝑘𝑧* is the trapdoor key of the function *𝐶𝐻*, and

*𝑃𝑧* is the commitment key of the function *𝐶𝐻*.

* + - *𝐸𝑣𝑎𝑙*(*𝑚𝑖 , 𝑖, 𝜁𝑖 , 𝑃𝑧* ) → *ℎ𝑖* is a polynomial-time evaluation algorithm, where *𝑚𝑖* is a message block, *𝑖* is the index of message block *𝑚𝑖* ,

*𝜁𝑖* is a random number, *𝑃𝑧* is the commitment key of the function

*𝐶𝐻*, and *ℎ𝑖* is a hash value.

′ ′

Homomorphic signatures [[15]](#_bookmark24) take multiple signed messages as in- put 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 inef-

ficient. The idea of redactable signature comes from the literature [[16]](#_bookmark25),

and then literatures [[17]](#_bookmark26) and [[18]](#_bookmark27) respectively give its formal definition, the signature mechanism allows anyone who holds the signature verifi- cation 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]](#_bookmark19) based on standard digital signature schemes and chameleon hash [[19]](#_bookmark29). 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 *,* … *, 𝑚𝑛* }, the sanitizer can modify the message block *𝑚𝑖* that is allowed to be modified into a new message block *𝑚*′ without breaking the verifiability of the message signature.

*𝑖*

Sanitization is essentially that the sanitizer uses a trapdoor to find con- flicts in the chameleon hash function without key exposure. Even if some message blocks are changed, the signature corresponding to the mes- sage will not be changed, so the original signature remains valid for the

* + - *𝐼𝑛𝑣*(*𝑚𝑖 , 𝑖, ℎ𝑖, 𝑠𝑘𝑧* ) → *𝜁𝑖* is a trapdoor collision finding algorithm,

where *𝑚*′ is a different message, *𝑖* is the data block index, *ℎ* is the chameleon hash value on the input (*𝑚𝑖 , 𝜁𝑖* ), *𝑠𝑘𝑧* is the trapdoor key, and *𝜁* ′ is a random number such that *𝐸𝑣𝑎𝑙*(*𝑚*′ *, 𝑖, 𝜁* ′ *, 𝑃* ) = *ℎ* .

*𝑖*

*𝑖*

*𝑖 𝑖 𝑖 𝑧 𝑖*

* + - *𝑉 𝑟𝑓 𝑦*(*𝑃𝑧 , ℎ𝑖 , 𝑚𝑖 , 𝜁𝑖* ) → 1∕0 is a verification algorithm, where *𝑃𝑧* is the

commitment key, *ℎ𝑖* is a chameleon hash value, *𝑚𝑖* is a message block,

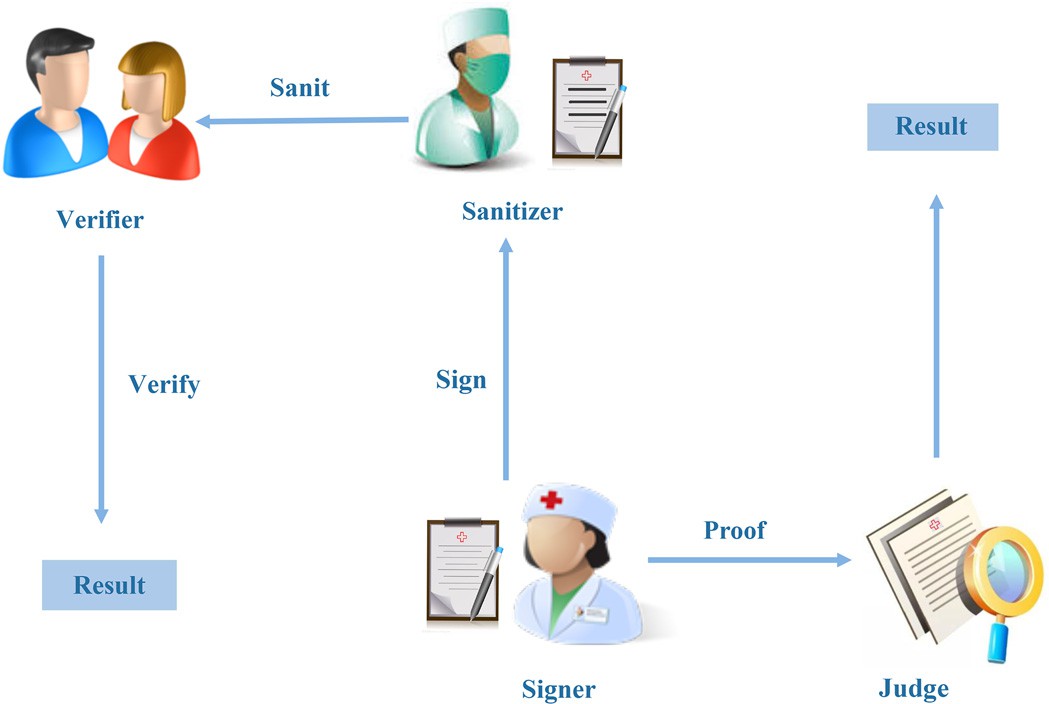
*𝜁𝑖* is a random number. If *ℎ𝑖* = *𝐸𝑣𝑎𝑙*(*𝑚𝑖 , 𝑖, 𝜁𝑖 , 𝑃𝑧* ) holds, then outputs

1, otherwise outputs 0.

* 1. *System model*

As demonstrated in [Fig. 1](#_bookmark7), we give the system model of our SSPM scheme. According to the practical application requirements of the elec- tronic 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 up- date 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.

sanitized message-signature pair, and the output result is 1∕0, where *• Verifier.* The verifier is responsible for verifying the validity of the

”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 *𝜋* ∈ {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.

* 1. *Framework of the SSPM scheme*

Our proposed SSPM scheme is a collection of the following PPT al- gorithms as below.

* + - *Setup*(1*𝜅* ) → *𝑝𝑎𝑟𝑎𝑚𝑠* is an initialization algorithm, where *𝜅* is the se- curity parameter, *𝑝𝑎𝑟𝑎𝑚𝑠* is the system parameter list.
    - *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 sig- nature 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.

message-signature pair based on the *𝑃 𝑟𝑜𝑜𝑓* generated by the signer. *• Accountability*. To determine which party is responsible for a given

* + - *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.

# Our proposed SSPM scheme

To slove the security and privacy issues in electronic medical sys- tem while improving the eﬃciency, we propose a SSPM scheme which

includes seven phases: *𝑆𝑒𝑡𝑢𝑝*, *𝐾𝐺𝑒𝑛𝑠* , *𝐾𝐺𝑒𝑛𝑧* , *𝑆 𝑖𝑔𝑛*, *𝑆 𝑎𝑛𝑖𝑡*, *𝑉 𝑒𝑟𝑖𝑓 𝑦* and

*𝑃 𝑟𝑜𝑜𝑓* . The details are described as below.

* 1. *Setup*

rameter *𝑘* by executing *𝑆𝑒𝑡𝑢𝑝*. A system parameter list is generated after obtaining the security pa-

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} → *𝑍𝑞* .
4. Publish the parameter list *𝑃 𝑎𝑟𝑎𝑚𝑠* = (*𝐺, 𝑞, 𝑔, 𝐶𝐻, ℎ*1*, ℎ*2).
   1. *𝐾𝐺𝑒𝑛𝑠*

The signer produces his/her key pair by executing *𝐾𝐺𝑒𝑛𝑠* .

1. Randomly select *𝑠𝑘𝑠* ∈ *𝑍* ∗ as the secret key and keep secret.

*𝑞*

1. Calculates *𝑃𝑠* = *𝑔𝑠𝑘𝑠* as the public key and keep public.
   1. *𝐾𝐺𝑒𝑛𝑧*

The santizer produces his/her key pair by executing *𝐾𝐺𝑒𝑛𝑧* .

* + - *𝐾𝐺𝑒𝑛𝑠* (*𝑝𝑎𝑟𝑎𝑚𝑠, 𝐼𝐷𝑆* ) → (*𝑠𝑘𝑠 , 𝑃𝑠* ) is a probabilistic algorithm exe-

cuted by the signer, where *𝐼𝐷* ∈ {0*,* 1}∗ is the identity and (*𝑠𝑘 , 𝑃* )

1. Randomly select *𝑠𝑘*

*𝑧* ∈ *𝑍* ∗ as the secret key and keep secret.

is the key pair of the signer

*𝑞*

*𝑆*

*𝑆*.

*𝑠 𝑠*

1. Calculates *𝑃𝑧* = *𝑔𝑠𝑘𝑧* as the public key and keep public.
   * + *𝐾𝐺𝑒𝑛𝑧* (*𝑝𝑎𝑟𝑎𝑚𝑠, 𝐼𝐷𝑍* ) → (*𝑠𝑘𝑧 , 𝑃𝑧* ) is a probabilistic algorithm exe-

cuted by the sanitizer, where *𝐼𝐷𝑍* ∈ {0*,* 1}∗ is the identity and

(*𝑠𝑘𝑧 , 𝑃𝑧* ) is the key pair of the sanitizer *𝑍*.

* + - *𝑆𝑖𝑔𝑛* (*𝑝𝑎𝑟𝑎𝑚𝑠, 𝑠𝑘𝑠 , 𝑃𝑧 , 𝑚*) → *𝜎𝑚* is a signature algorithm executed by the signer, where *𝑚* is the message, and *𝜎𝑚* is the signature on the
  1. *Sign*

The signer *𝑆* produces a message-signature pair *𝜎𝑚*

*𝑚* by executing *𝑆𝑖𝑔𝑛*.

on the message

message *𝑚*.

* + - *𝑆𝑎𝑛𝑖𝑡* (*𝑝𝑎𝑟𝑎𝑚𝑠, 𝑠𝑘𝑧 , 𝑃𝑧 , 𝜎𝑚 , 𝑀𝑢𝑡*) → *𝜎𝑚* is a sanitization algorithm ex-

′

ecuted by the sanitizer, where *𝑚*′ is the sanitized message, *𝑀𝑢𝑡* is a

1. Inputs parameter list *𝑝𝑎𝑟𝑎𝑚𝑠*, signer’s secret key *𝑠𝑘𝑠* , sanitizer’s pub- lic key *𝑃𝑧* , and message *𝑚*.
2. Divide *𝑚* into *𝑛* blocks, that is, *𝑚* = {*𝑚* } .

description of information that needs to be sanitized on *𝑚*, and *𝜎*′ is

*𝑚*

the sanitized message-signature pair.

* + *Verify*(*𝑃𝑠, 𝑃𝑧 , 𝜎𝑚* ) → {0 ∕1} is a verification algorithm executed by any

third party, where 1 or 0 as outputs to indicate whether the signature

*𝜎𝑚* is validated.

* + *Proof*(*𝑠𝑘𝑠, 𝑃𝑧 , 𝜎𝑚* ) → *𝜋* is a proof algorithm executed by the signer, where *𝜋* as outputs to indicate whether the signature *𝜎𝑚* is generated

by the signer or the sanitizer.

*𝑖 𝑖*∈[*𝑛*]

1. Set a variable block index *𝑀𝑢𝑡* of size *𝑡* and the mutable message blocks as {*𝑚𝑖* }*𝑖*∈*𝑀𝑢𝑡*.
2. For *𝑖* ∈ [*𝑛*]∖*Mut*, computes *ℎ𝑖* = *ℎ*1 (*𝑚𝑖 , 𝑃𝑠 , 𝑃𝑧* ).
3. For *𝑖* ∈ *𝑀𝑢𝑡*, randomly select *𝜁𝑖* ∈ *𝑍𝑞* , set *𝜁* = {*𝜁𝑖* }*𝑖*∈*𝑀𝑢𝑡* and compute

*ℎ𝑖* = *𝐸𝑣𝑎𝑙*(*𝑚𝑖 , 𝑖, 𝜁𝑖 , 𝑃𝑧* ).

1. Set *ℎ𝑚* = {*ℎ𝑖* }*𝑖*∈[*𝑛*] and randomly select *𝜔* ∈ *𝑍𝑞* .
2. Compute *𝑢* = *ℎ*2 (*𝑔𝜔, 𝑃𝑧 , ℎ𝑚* ) and *𝑣* = *𝜔*∕(*𝑢* + *𝑠𝑘𝑠* ).
3. Returns the message-signature pair *𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*).

*3.4. Security requirements*

A SSPM scheme need to meet the following functions and security

* 1. *Sanit*

The sanitizer *𝑍* produces a new message-signature pair *𝜎*′

on the

requirements:

message *𝑚*′

*𝑚*

by executing *𝑆𝑎𝑛𝑖𝑡*.

1. Inputs parameter list *𝑝𝑎𝑟𝑎𝑚𝑠*, the sanitizer’s key pair *𝑃𝑧 , 𝑠𝑘𝑧* , the mutable block index *𝑀𝑢𝑡* and the message-signature pair *𝜎𝑚* =

(*𝑚, 𝜁, ℎ , 𝑢, 𝑣*).

c. Verify

*𝑣* = *ℎ*2 ((*𝑃𝑠 𝑔𝑢*)*𝑣 , 𝑃𝑧 , ℎ𝑚* ) (3)

*𝑚*

1. For all *𝑖* ∈ *𝑀𝑢𝑡*

, randomly select *𝑚*′ ∈ *𝑀*

and compute *𝜁* ′ = *𝐼𝑛𝑣*(*𝑚*′ *,*

*𝑖, ℎ , 𝑠𝑘* ), make *𝐸𝑣𝑎𝑙*( ′ ′ ) = *ℎ* = *𝐸𝑣𝑎𝑙*(*𝑚 , 𝑖, 𝜁 , 𝑃* )

*𝑖*

*𝑖 𝑖*

*𝑖 𝑧*

*𝑚*

*𝑚𝑖 , 𝑖, 𝜁𝑖 , 𝑃𝑧 𝑖*

*𝑖 𝑖 𝑧*

’

*𝑖 𝑖*∈[*𝑛*]∖*Mut*

3. Set *𝑚* = ({*𝑚* }

’

*,* {*𝑚𝑖* }*𝑖*∈*Mut*

) as the sanitzied message

Clearly, both *𝜎𝑚*

and *𝜎*′ pass the above verification and are valid. There-

1. Set *𝜁* ′ = {*𝜁* ′ }

*𝑖*

*𝑖*∈*𝑀𝑢𝑡* .

fore, the correctness of our scheme holds.

1. Returns the sanitzied message-signature tuple *𝜎*′ = ( ′ ′ ).
   1. *PVerify*

*𝑚 𝑚 , 𝜁 , ℎ𝑚, 𝑢, 𝑣*

* 1. *Unforgeability*

ing

The any third party verifies the signature *𝜎𝑚*

*𝑃 𝑉 𝑒𝑟𝑖𝑓 𝑦*.

= (*𝑚, 𝜁, 𝑢, 𝑣*) by execut-

Our SSPM scheme can meet the unforgeability. According to the characteristics of chameleon hash, with the exception of sanitizers au- thorized to forge signatures of sanitized message blocks in a controlled

1. Inputs system parameters *𝑝𝑎𝑟𝑎𝑚𝑠*, public key pairs (*𝑃𝑠 , 𝑃𝑧* ) and message-signarure pairs *𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*).
2. If ∀*𝑖* ∈ *𝑀𝑢𝑡*, we have *𝑉 𝑟𝑓 𝑦*(*𝑃𝑧 , 𝑖, 𝑚𝑖 , ℎ𝑖 , 𝜁𝑖* ) = 1 and ∀*𝑖* ∈ [*𝑛*]∖*Mut*, we

have *ℎ𝑖* = *ℎ*1 (*𝑚𝑖 , 𝑃𝑠 , 𝑃𝑧* ) *𝑃𝑠 , 𝑃𝑧* ), go to the next step, otherwise reject.

1. Verify

*𝑣* = *ℎ*2 ((*𝑃𝑠 𝑔𝑢*)*𝑣 , 𝑃𝑧 , ℎ𝑚* ) (1)

3. If *𝐸𝑞𝑢.*(1) holds, emits ”1” and accept *𝜎𝑚* ; Otherwise emit ”0” and reject.

* 1. *Proof*

signature by executing *𝑃 𝑟𝑜𝑜𝑓* . The signer produces a proof indicating who is responsible for the

1. Inputs system parameters *𝑝𝑎𝑟𝑎𝑚𝑠*, key pairs (*𝑠𝑘𝑠 , 𝑃𝑠* ), the sanitizer’s public key *𝑃𝑧* , and the message *𝑚*∗.
2. If *𝑚*∗ = *𝑚*, return a proof *𝜋* = 1, indicting that *𝑚*∗ originates from the

signer.

1. If *𝑚*∗ ≠ *𝑚* and has the same signature and hash values with *𝑚*, return a proof *𝜋* = 0, indicting that *𝑚*∗ is generated by the sanitizer.

# 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.

* 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 *𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*) generated by the signer via executing the *𝑆𝑔𝑖𝑛* algorithm, then verify *𝜎𝑚* is valid as the be-

lowing steps:

* 1. If ∀*𝑖* ∈ *𝑀𝑢𝑡*, then we have *Vrfy*(*𝑃𝑧, 𝑖, 𝑚𝑖 , ℎ𝑖 , 𝜁𝑖* ) = 1.
  2. If ∀*𝑖* ∈ [*𝑛*]∖*Mut*, then we have *ℎ𝑖* = *ℎ*1 (*𝑚𝑖 , 𝑃𝑠 , 𝑃𝑧* ) *𝑃𝑠 , 𝑃𝑧* ), go to the

next step, otherwise reject.

* 1. Verify

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.

* 1. *Public verifiability*

Our SSPM scheme can meet the public verifiability. From the PVer- ify algorithm we can find that the verification process only needs the

participation of (*𝑃𝑠 , 𝑃𝑧* ) and message-signarure pairs (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*). That

is, any third party can verify the authenticity of *𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*) with-

out the recipient’s secret key. Therefore, the public verifiability of our

scheme holds.

* 1. *Accountability*

Our SSPM scheme can meet accountability. From the *𝑃 𝑟𝑜𝑜𝑓* algo- rithm, 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

*𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*). Therefore, the accountability of our scheme holds.

* 1. *Privacy*

Our SSPM scheme can meet the privacy. From the *Sanit* algorithm, The SSPM scheme proposed in this paper can effectively hide the sen- sitive 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 account- ability of our scheme holds.

* 1. *Immutability*

Our SSPM scheme satisfies immutability. From the *𝑆𝑖𝑔𝑛* algorithm, we can find that in the message signature pair *𝜎𝑚* = (*𝑚, 𝜁, ℎ𝑚, 𝑢, 𝑣*) gen-

the collision resistant hash function *ℎ*1. In sanitized message signature erated by the signer, the unmodifiable message blocks are hashed using pair *𝜎*′ = (*𝑚*′ *, 𝜁* ′ *, ℎ , 𝑢, 𝑣*), if *𝑚*′ produced by the sanitizer contains an unmodifiable block *𝑚*′ , and *𝑖* ∉ *𝑀𝑢𝑡*, the following two cases may exist:

*𝑖*

*𝑚 𝑚*

*𝑣* = *ℎ*2 ((*𝑃𝑠 𝑔𝑢*)*𝑣 , 𝑃𝑧 , ℎ𝑚* ) (2)

′ ′ ′

′ ′ ′ ′

1. If a sanitized signature *𝜎*’ = (*𝑚*’ *, 𝜁* ’ *, ℎ , 𝑢, 𝑣*) generated by the the san-

*𝑚 𝑚*

1. *𝜎𝑚* = (*𝑚 , 𝜁 , ℎ𝑚, 𝑢, 𝑣*), i.e. *𝑚* ≠ *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.

*𝑚 𝑚*

itizer via executing the *Sanit* algorithm, then verify *𝜎*′ is valid as the

*𝑚*

1. (*𝑚*′ *, 𝜁* ′ *, ℎ , 𝑢*′ *, 𝑣*′ ) ≠ (*𝑚, 𝜁, ℎ , 𝑢, 𝑣*), that means the sanitizer forged the

belowing steps:

* 1. If ∀*𝑖* ∈ *𝑀𝑢𝑡*, then we have *Vrfy*( ’ ’ ) = 1.

*𝑃 , 𝑖, 𝑚 , ℎ , 𝜁*

signature (*𝑢*′ *, 𝑣*′ ) which violates the unforgeability of digital signa-

tures used in the *𝑆𝑖𝑔𝑛* algorithm.

*𝑧 𝑖 𝑖 𝑖*

* 1. If ∀*𝑖* ∈ [*𝑛*]∖*Mut*, we have *ℎ𝑖* = *ℎ*1 (*𝑚𝑖 , 𝑃𝑠 , 𝑃𝑧* ), go to the next step,

otherwise reject.

Therefore, the immutability of our scheme holds.

**Table 1**

Running time of different operations(ms).



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operations | *𝑇𝑚𝑎* | *𝑇𝑚𝑚* | *𝑇𝑖𝑛𝑣* | *𝑇ℎ𝑎𝑠ℎ* | *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.

# 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* mod *𝑞* for the schemes, where *𝑎*, *𝑏* ∈ *𝑍*∗, *𝐺* is a group with order *𝑞* on *𝐸*, *𝑝* and *𝑞*

*𝑞*

are both prime numbers with a length of 160 bits. We use the MIRACL li- brary [[28]](#_bookmark33) on a laptop computer (Intel core with [I7-4770@3.4GHz](mailto:I7-4770@3.4GHz) CPU, 4GB random memory, and Windows7 operating system) to simulate our SSPM scheme.

Let *𝑇𝑚𝑎* represent a modular addition operation in *𝑍𝑞* , *𝑇𝑚𝑚* represent

a modular multiplication operation in *𝑍𝑞* , *𝑇𝑖𝑛𝑣* represent a modular in- verse operation in *𝑍𝑞* , *𝑇*exp represent an exponential operation in *𝐺*, *𝑇𝑚𝑢𝑙* represent a scalar multiplication operation in *𝐺*, *𝑇ℎ𝑎𝑠ℎ* represent a gen- eral hash operation in *𝑍𝑞* . The running time of different operations is

shown in [Table 1](#_bookmark9).

Since the computation overhead of *𝑆𝑒𝑡𝑢𝑝* and *𝑃 𝑟𝑜𝑜𝑓* phases are al-

most negligible, we only consider the computation overhead of the

*𝐾𝐺𝑒𝑛**𝑠*, *𝐾𝐺𝑒𝑛𝑧*, *𝑆𝑖𝑔𝑛*, *𝑆𝑎𝑛𝑖𝑡* and *𝑉 𝑒𝑟𝑖𝑓 𝑦* phases in the scheme, where we use *𝐾𝐺𝑒𝑛* to denote *𝐾𝐺𝑒𝑛𝑠* and *𝐾𝐺𝑒𝑛𝑧* for convenience. The detailed

analysis is as follows.

In *𝐾𝐺𝑒𝑛𝑠* and *𝐾𝐺𝑒𝑛𝑧* 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 2*𝑇*exp. In *𝑆𝑖𝑔𝑛* phase, we can find that our scheme needs to perform (*𝑛* + 2) general hash operations, one modular

inverse operation, and one exponentiation operation in *𝐺*. Therefore, addition operation, one modular multiplication operation, one modular the total cost of the *𝑆𝑔𝑖𝑛* phase is (*𝑛* + 2)*𝑇ℎ𝑎𝑠ℎ* + *𝑇𝑚𝑎* + *𝑇𝑚𝑚* + *𝑇𝑖𝑛𝑣* + *𝑇*exp.

2*𝑡* general hash operations. Therefore, the total cost of the *Sanit* phase In *Sanit* phase, we can find that our scheme only needs to perform is 2*𝑡𝑇ℎ𝑎𝑠ℎ*. In *𝑃 𝑉 𝑒𝑟𝑖𝑓 𝑦* phase, we can find that our scheme needs to perform (2*𝑡* + 1) general hash operations, one modular addition oper-

tion operation in *𝐺*. Therefore, the total cost of the *𝑃 𝑉 𝑒𝑟𝑖𝑓 𝑦* phase is ation, one modular multiplication operation, and and one exponentia- (2*𝑡* + 1)*𝑇ℎ𝑎𝑠ℎ* + *𝑇𝑚𝑎* + *𝑇𝑚𝑚* + *𝑇*exp.

As shown in [Fig. 2](#_bookmark10), 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 com- putation cost of the *𝑆𝑎𝑛𝑖𝑡* phase is 0.12 ms, the computation cost of the

*𝑃 𝑉 𝑒𝑟𝑖𝑓 𝑦* 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 con- stant. Based on the above considerations, we consider our sterilizable signature scheme to be practical.

# Conclusion

With the continuous development of electronic medicine, the ap- plication 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 sani- tization process of the message to use the trapdoor key to find collisions in the Chameleon hash function. Security analysis and performance eval- uation demonstrate show that our SSPM scheme can not only guarantee the integrity of medical data, but also support patient privacy protec- tion. Further, the scheme achieves public verification, which greatly re- duces 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).](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0001)
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).](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0002)
3. [A.K. Jha, C.M. DesRoches, E.G. Campbell, K. Donelan, S.R. Rao, T.G. Ferris,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0003)

[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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0003)

1. [A. Hoerbst, E. Ammenwerth, Electronic health records, Methods Inf. Med. 49 (04) (2010) 320–336.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0004)
2. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0005)
3. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0006)
4. [R.B. Ness, J.P. Committee, et al., Influence of the HIPAA privacy rule on health research, JAMA 298 (18) (2007) 2164–2170.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0007)
5. [D. Wartenberg, W.D. Thompson, Privacy versus public health: the impact of current confidentiality rules, Am. J. Public Health 100 (3) (2010) 407–412.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0008)
6. [C. Brzuska, M. Fischlin, T. Freudenreich, A. Lehmann, M. Page, J. Schelbert,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0009)

[D. Schröder, F. Volk, Security of sanitizable signatures revisited, in: International Workshop on Public Key Cryptography, Springer, 2009, pp. 317–336.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0009)

1. [G. Ateniese, D.H. Chou, B.d. Medeiros, G. Tsudik, Sanitizable signatures, in: Euro- pean Symposium on Research in Computer Security, Springer, 2005, pp. 159–177.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0010)
2. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0011)
3. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0012)
4. [O. Sanders, Eﬃcient redactable signature and application to anonymous creden- tials, in: IACR International Conference on Public-Key Cryptography, Springer, 2020,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0013)

[pp. 628–656.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0013)

1. [K. Samelin, D. Slamanig, Policy-based sanitizable signatures, in: Cryptographers Track at the RSA Conference, Springer, 2020, pp. 538–563.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0014)
2. [D. Catalano, D. Fiore, L. Nizzardo, On the security notions for homomorphic signa- tures, in: International Conference on Applied Cryptography and Network Security, Springer, 2018, pp. 183–201.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0015)
3. [R. Steinfeld, L. Bull, Y. Zheng, Content extraction signatures, in: International Con- ference on Information Security and Cryptology, Springer, 2001, pp. 285–304.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0016)
4. [C. Brzuska, H. Busch, O. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Man- ulis, C. Onete, A. Peter, B. Poettering, et al., Redactable signatures for tree-structured data: definitions and constructions, in: International Conference on Applied Cryp- tography and Network Security, Springer, 2010, pp. 87–104.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0017)
5. [A. Kundu, E. Bertino, Privacy-preserving authentication of trees and graphs, Int. J. Inf. Secur. 12 (6) (2013) 467–494.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0018)
6. [S. Krenn, H.C. Pöhls, K. Samelin, D. Slamanig, Chameleon-hashes with dual long-term trapdoors and their applications, in: International Conference on Cryp- tology in Africa, Springer, 2018, pp. 11–32.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0019)
7. [L. Jiguo, Z. Liufu, L. Chengdong, L. Yang, H. Jinguang, W. Huaqun, Z. Yichen, Prov- ably secure traceable attribute-based sanitizable signature scheme in the standard model, J. Comput. Res. Dev. 58 (10) (2021) 2253.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0020)
8. [J. Gong, H. Qian, Y. Zhou, Fully-secure and practical sanitizable signatures, in: In- ternational Conference on Information Security and Cryptology, Springer, 2010,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0021)

[pp. 300–317.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0021)

1. [S. Krenn, K. Samelin, D. Sommer, Stronger security for sanitizable signatures, in: Data Privacy Management, and Security Assurance, Springer, 2015, pp. 100– 117.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0022)
2. [C. Brzuska, M. Fischlin, A. Lehmann, D. Schröder, Unlinkability of sanitizable sig- natures, in: International Workshop on Public Key Cryptography, Springer, 2010,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0023)

[pp. 444–461.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0023)

1. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0024)
2. [M.T. Beck, J. Camenisch, D. Derler, S. Krenn, H.C. Pöhls, K. Samelin, D. Sla- manig, Practical strongly invisible and strongly accountable sanitizable signatures, in: Australasian Conference on Information Security and Privacy, Springer, 2017,](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0025)

[pp. 437–452.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0025)

1. [W. Gao, F. Li, X. Wang, Chameleon hash without key exposure based on Schnorr signature, Comput. Stand. Interfaces 31 (2) (2009) 282–285.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0026)
2. [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.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0027)
3. [D.F. Pigatto, N.B.F. da Silva, K.R.L.J.C. Branco, Performance evaluation and compar- ison of algorithms for elliptic curve cryptography with El-Gamal based on MIRACL and RELIC libraries, J. Appl. Comput. Res. 1 (2) (2011) 95–103.](http://refhub.elsevier.com/S2772-9184(23)00006-1/sbref0028)