FABSS: Attribute-Based Sanitizable Signature for Flexible Access Structure

Ruo Mo[1(✉)], Jianfeng Ma[1], Ximeng Liu[2], and Qi Li[3]

[1] Xidian University, Xi'an, China
593430655@qq.com
[2] Singapore Management University, Singapore, Singapore
[3] Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract. In the Electronic Health Record (EHR) system, digital signature is utilized to prevent the medical data from being tampered. However, users update their medical data frequently and have to sign these medical data from scratch after updating. Besides, traditional signature attests the identity of the individual signing the records, which leads to vast computation cost and the privacy leakage. In this paper, we obfuscate users identity information with attribute sets and introduce a semi-trusted participant–sanitizer to propose the Flexible Attribute-Based Sanitizable Signature (FABSS) scheme. We prove that our scheme is unforgeable under generic group model. Through comparison, the FABSS scheme not only reduces the users computation overhead, but also supports flexible access structures to implement expressively fine-grained access control.

Keywords: Flexible attribute-based access control
Sanitizable signature · Unforgeability · Anonymity
Information privacy

1 Instruction

The EHR system is considered as a sustainable solution for improving the quality of medical care, referring to the systematized collection of patient and population electronically-stored health information in a digital format. In the EHR system, it is important to guarantee the authentication and integrity of medical records, and thus digital signature is utilized in the EHR system. However, the secret signing key of the signature attests to the identity information of patients, such as names, ages which are not supposed to be shown to the public. With attribute based signature (ABS), patients sign the records with attributes signing key issued by attribute authorities according to the patients' attributes, such as $age : >45, profession : teacher, workunits : Xidian\ University$, etc. The signature attests not to the identities of patients but some of their attributes, which protects the identity privacy of patients and achieves anonymous authentication. Hence, ABS adapts to the EHR system.

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 39–50, 2018.
https://doi.org/10.1007/978-3-319-89500-0_3

In the system, the health data of patients are updated frequently. However, traditional digital signature including ABS prohibits any alteration of the original medical data once it is signed and has to be regenerated from scratch once parts of the original records are changed, which increases the computation overhead of users, leading to inefficiency. Sanitizable signature allows a semi-trusted party sanitizer to modify certain portions of the health records in the original signature. Thus, the signer needs to sign the records only once, which reduces the computation cost of the signer. In the sanitizing phase of the original signature, the sanitizer can generate the sanitized signature without interacting with the signer for signing keys, thus the sanitizer cannot forge the signature of the original signer. In addition, the process of sanitizing does not impact the verification.

In this paper, to address the efficiency and identity privacy problems in the EHR system, we propose a novel Flexible Attribute-Based Sanitizable Signature (FABSS) scheme. Specifically, major contributions of this paper are twofold.

- We introduce the sanitizable signature mechanism into present ABS with which patients sign the record with their attribute signing keys. When the records need to be updated, the patients deliver the original signature and the modifiable parts of the original message to the sanitizer, then the sanitizer can modify the message on the signature directly.
- Comparing with existing schemes, the FABSS scheme not only reduces the users computation overhead, but also preserves the anonymity and information privacy of users. Besides, the proposed scheme supports flexible access structure consisting of any AND, OR and threshold gates, which can provide expressively fine-grained access control.

1.1 Related Work

The notion of sanitizable signature was first proposed by Ateniese et al. [1], which allows a semi-trusted party sanitizer to modify certain portions of a signed message and produce a valid sanitized signature without interaction with the original signer for secret keys. Besides, They also define two necessary security requirements: (1) unforgeability, which means only legitimate signers can generate valid signatures. (2) information privacy, which means the sanitized message and corresponding signature should not reveal the original message. Nevertheless, they did not provide formal specifications for these properties. Brzuska et al. [2,3] introduced another security requirements unlinkability which prevents that one can link the sanitized message-signature pair of the same document and deduce the original message. Obviously, unlinkability is a variant of information privacy. Canard et al. [4] gave a generic construction of trapdoor sanitizable signatures which the candidate sanitizer cannot produce the sanitized signature until receiving the trapdoor from the signer. [5] utilized accountable chameleon hash and presented an accountable trapdoor sanitizable signature based on [4]. However, these schemes did not present a concrete construction of sanitizable signatures. Hence, several concrete sanitizable signature scheme [6–8] were proposed with

thorough security proofs. Although these works make a significant contribution to the development of sanitizable signature, none of them considers the identity privacy of users and thus cannot be applied to EHR system.

ABS is converted from attribute-based encryption (ABE) [9,10]. Maji et al. [11] presented an ABS scheme which supported flexible access structure. However, the unforgeability proof of [11] was given under generic group model. In [12], Li et al. gave the construction of two efficient ABS schemes supporting threshold predicate in random oracle model and standard model, respectively. Okamoto and Takashima [13] proposed an ABS scheme for non-monotone access structure which introduced NOT gate into threshold access structure and was provably secure in the standard model. To achieve flexible access control as well as more secure level, [14,15] presented flexible ABS schemes in random oracle model and standard model, respectively. Concerning on the authority management of attributes, Li et al. [16] presented a formalized construction of multi-authority ABS scheme supporting threshold gates.

In this paper, we refer to two present ABSS schemes [17,18]. However, only [17] gave the concrete construction of ABSS. Thus, we compare our FABSS scheme with [17] and [8,12] to illustrate the advantage of our scheme in function and efficiency.

The remainder of this paper is organized as follows. The preliminary knowledge of our scheme is in Sect. 2. In Sect. 3 we define the algorithm model and security model. We propose the specific construction of FABSS scheme and corresponding security proof in Sects. 4 and 5, respectively. In Sect. 6 we compare our scheme with existing works and the paper is concluded in Sect. 7.

2 Preliminaries

2.1 Flexible Access Structure

The threshold access structure in ABS is composed of one threshold and several attributes. A user can generate a valid signature only if the size of the intersection of his attribute sets and the access structure attribute sets exceeds the threshold value. Simple access control can be achieved with threshold access structure, such as {'A' AND 'B' AND 'C'}, {'A' OR 'B' OR 'C'}.

The flexible access structure consists of a number of thresholds and attributes, in which each interior node is a threshold gate. Besides aforementioned structures, we can define expressive access control in large-scale attribute sets through changing the breadth and depth of the structure, such as {{'A' AND 'B'} OR 'C'}, {{'A' OR 'B'} AND 'C'}, etc.

2.2 Monotone Span Program

Suppose f is a monotone boolean function. A monotone span program over a field $\mathbb{F}$ for f is a $l \times t$ matrix $\mathbf{M}$, and f takes input as the mapping of each row

of the matrix $\mathbf{M}$ with a labeling function $z(\cdot)$. The monotone span satisfies the following equation:

$$f(x_1, \ldots, x_n) = 1 \Leftrightarrow \exists \boldsymbol{v} \in \mathbb{F}^{1\times l} \; s.t. \; \boldsymbol{v}\mathbf{M} = [1, 0, \ldots, 0] \text{ and } (\forall i : x_{z(i)} = 0 \Rightarrow v_i = 0).$$

Every monotone boolean function can be presented by some monotone span programs. The flexible attribute access structure is composed of several (t_i, l_i) threshold gates, meaning that we can obtain the secret from at least t_i of l_i attributes. The size of the matrix $\mathbf{M}$ depends on the specifications of these threshold gates. With i (t_i, l_i) threshold gates, we can construct a matrix $\mathbf{M}$ with length $l = \sum_i (l_i - 1) + 1$ and width $t = \sum_i (t_i - 1) + 1$.

2.3 Designated Instruction

The Designated Instruction (DI) refers to designated parts of the original message for updating. Let $m = m_1 m_2 \ldots m_n$, where m_k is defined as the bit at index k of message m. Let $DI \subseteq \{1, \ldots, n\}$ denote the set of the indexes that is going to be updated. Obviously, DI is classified into two groups, where $DI_1 = \{k \in DI : \; m_k = 0, \; m'_k = 1\}$, $DI_2 = \{k \in DI : \; m_k = 1, \; m'_k = 0\}$.

3 Algorithm Model and Security Model

3.1 Algorithm Model

The FABSS scheme is parametrized by five algorithms below.

$(Params, MSK) \leftarrow \mathbf{Setup}(1^\lambda)$. Algorithm **Setup** takes a security parameter λ as input and outputs the public parameters $Params$ and the master secret key MSK.

$SK_\mathcal{A} \leftarrow \mathbf{KeyGen}(MSK, \mathcal{A})$. With the MSK, the **KeyGen** algorithm outputs the attribute private key $SK_\mathcal{A}$ based on the signing request of each patient on his attribute set $\mathcal{A} \subseteq \mathbb{A}$.

$\sigma \leftarrow \mathbf{Sign}(Params, SK_\mathcal{A}, m, f)$. The patient endorses a message m for the access structure f with his signing key $SK_\mathcal{A}$, resulting in the signature σ.

accept*/*reject $\leftarrow \mathbf{Verify}(Params, m, f, \sigma)$. The algorithm **Verify** allows the health professionals to verify whether the message is signed by a legitimate patient. It outputs ***accept*** if the signature is valid, otherwise ***reject***.

$(m', \sigma') \leftarrow \mathbf{Sanitize}(Params, m, \sigma, DI)$. According to the DI provided by the patient, the algorithm **Sanitize** outputs the sanitized message m' and corresponding signature σ'.

3.2 Security Model

Definition 1 (Correctness). *The FABSS scheme is correct if for* $(Params, MSK) \leftarrow$ ***Setup****, the message* m*, attribute sets* $\mathcal{A}$ *that satisfy the access structure* f *and the signing key* $SK_\mathcal{A} \leftarrow$ ***KeyGen***$(MSK, \mathcal{A})$*,*

$$\boldsymbol{Verify}(Params, m, f, \boldsymbol{Sign}(params, SK_\mathcal{A}, m, f)) = \mathbf{accept}.$$

Definition 2 (Unforgeability). *We prove the unforgeability under selective-predicate attack which is weaker than the adaptive predicate attack. The FABSS scheme is unforgeable under selective-predicate attack provided that the advantage of any polynomial-time adversary in the following experiment is negligible:*

- *The adversary chooses the challenge access structure* f^*.
- *The challenger runs* $(Params, MSK) \leftarrow$ ***Setup*** *and gives* $Params$ *to the adversary.*
- *The adversary can make a polynomial bounded number of queries to oracles* ***KeyGen***$(MSK, \mathcal{A})$ *and* ***Sign***$(Params, SK_{\mathcal{A}}, m, f)$.
- *The adversary outputs the purported message-signature pair forgery* (m^*, σ^*).

The adversary wins if m^* *is never queried to the* ***Sign*** *oracle, and none of* $\mathcal{A}$ *queried to the* ***KeyGen*** *oracle satisfy* f, *and* ***Verify***$(Params, m^*, f^*, \sigma^*) =$ **accept**.

Definition 3 (Anonymity). *Anonymity means that the signature would not reveal anything about the identity or attributes of the signer except the attributes in the access structure. The FABSS scheme is anonymous if for all access structure* $f, (Params, MSK) \leftarrow$ ***Setup***, *attributes* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *that satisfy* f, *attribute signing key* $SK_{\mathcal{A}_1} \leftarrow$ ***KeyGen***$(MSK, \mathcal{A}_1)$ *and* $SK_{\mathcal{A}_2} \leftarrow$ ***KeyGen***$(MSK, \mathcal{A}_2)$, *the distribution of* $\sigma_{\mathcal{A}_1} \leftarrow$ ***Sign***$(Params, SK_{\mathcal{A}_1}, m, f)$ *is identical to that of* $\sigma_{\mathcal{A}_2} \leftarrow$ ***Sign***$(Params, SK_{\mathcal{A}_2}, m, f)$.

Definition 4 (Information Privacy). *The FABSS scheme achieves information privacy if for all access structure* $f, (Params, MSK) \leftarrow$ ***Setup***(1^λ), *attribute* $\mathcal{A}$ *satisfying* $f, SK_{\mathcal{A}} \leftarrow$ ***KeyGen***$(MSK, \mathcal{A})$, *message* m_1, m_2 *and a sanitized message* m', *where* m' *differs from* m_1 *and* m_2 *only at bits that are allowed to be sanitized, the distribution of* $\sigma_1' \leftarrow$ ***Sanitize***(***Sign***$(Params, SK_{\mathcal{A}}, m_1, f), Params, UI)$ *and* $\sigma_2' \leftarrow$ ***Sanitize***(***Sign***$(Params, SK_{\mathcal{A}}, m_2, f), Params, UI)$ *are identical.*

4 Our FABSS Scheme

Our scheme supports the access structure whose monotone span program $\mathbf{M}$ has width at most t_{max} which is an arbitrary number. $\mathbb{A} \subseteq \mathbb{Z}_p^*$ is the universe of all possible attributes in the access structure f. The original message $m = m_1 m_2 \dots m_n \in \{0,1\}^n$.

Setup: Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic groups of prime order p. Choose a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and random generators: $g_1 \leftarrow \mathbb{G}_1, g_{2_0}, \dots, g_{2_{t_{max}}} \leftarrow \mathbb{G}_2$. Choose random $a_0, a, b \leftarrow \mathbb{Z}_p^*, u', u_1, \dots, u_n \leftarrow \mathbb{Z}_p$ and set $A_0 = g_{2_0}^{a_0}, A_j = g_{2_j}^a$ and $B_j = g_{2_j}^b (\forall j \in [t_{max}]), U' = g_1^{u'}, U_k = g_1^{u_k} (\forall k \in [n])$. Public parameters are $g_{2_0}, \dots, g_{2_{t_{max}}}, A_0, \dots, A_{t_{max}}, B_0, \dots, B_{t_{max}}, g_1, U', U_1, \dots, U_n$. Master secret keys are a_0, a, b.

KeyGen: On inputting master secret keys and the attribute set $\mathcal{A} \subseteq \mathbb{A}$, choose random $K \leftarrow \mathbb{G}_1$ and set $K_0 = K^{1/a_0}, K_z = K^{1/(a+bz)} (\forall z \in \mathcal{A})$. The signing key is $SK_{\mathcal{A}} = (K, K_0, \{K_z \mid z \in \mathcal{A}\})$.

Sign: First convert the claim-predicate f into its corresponding monotone span program matrix $\mathbf{M} \in (\mathbb{Z}_p)^{l\times t}$, mapping each row of $\mathbf{M}$ with the row labeling function $z : [l] \to \mathbb{A}$. Then compute the vector $\boldsymbol{v}$ that corresponds to the desirable assignment for $\mathcal{A}$. Pick random $r \leftarrow \mathbb{Z}_p^*, r_1, \ldots, r_l \leftarrow \mathbb{Z}_p$ and compute $Y = K^r, S_i = (K_{z(i)}^{v_i})^r \cdot (U' \prod_{k=1}^n U_k^{m_k})^{r_i} (\forall i \in [l]), W = K_0^r, P_j = \prod_{i=1}^l (A_j B_j)^{\mathbf{M}_{i,j}\cdot r_i} (\forall j \in [t])$. The signature is $\sigma = (Y, W, S_1, \ldots, S_l, P_1, \ldots, P_t)$.

Verify: With $(Params, \sigma = (Y, W, S_1, \ldots, S_l, P_1, \ldots, P_t), m, f)$, the verifier first converts f into its corresponding monotone span program $\mathbf{M} \in (\mathbb{Z}_p)^{l\times t}$, with row labeling $z : [l] \to \mathbb{A}$. If $Y = 1$, the verifier outputs ***reject***, otherwise checks $e(W, A_0) \stackrel{?}{=} e(Y, g_{2_0})$,

$$\prod_{i=1}^{l} e(S_i, (A_j B_j^{z(i)})^{\mathbf{M}_{i,j}}) \stackrel{?}{=} \begin{cases} e(Y, g_{2_1}) e(U' \prod_{k=1}^n U_k^{m_k}, P_1), & j = 1, \\ e(U' \prod_{k=1}^n U_k^{m_k}, P_j), & j > 1, \end{cases}$$

for each $j \in [t]$. The verifier returns ***accept*** if all the equations above hold, otherwise ***reject***.

Sanitize: The sanitizer obtains σ and the DI from the signer. Pick random $\tilde{r}_1, \ldots, \tilde{r}_l \leftarrow \mathbb{Z}_p$ then compute $Y' = Y, S_i' = S_i \frac{\prod_{k\in I_1} U_k^{r_i}}{\prod_{k\in I_2} U_k^{r_i}} (U' \prod_{k=1}^n U_k^{m_k'})^{\tilde{r}_i} (\forall i \in [l])$, $W' = W, P_j' = P_j \prod_{i=1}^l (A_j B_j)^{\mathbf{M}_{i,j}\cdot \tilde{r}_i} (\forall j \in [t])$.

5 Security Analysis

Proof (Correctness). When the signature of either the original message or the sanitized message is signed by the signer whose attributes fit the access structure f, it can be successfully checked by the verification.

Verification:

$$e(W, A_0) = e(K_0^r, g_{2_0}^{a_0}) = e(K^r, g_{2_0}) = e(Y, g_{2_0}),$$

$$\begin{aligned} \prod_{i=1}^{l} e(S_i, (A_j B_j^{z(i)})^{\mathbf{M}_{i,j}}) &= \prod_{i=1}^{l} e((K_{z(i)}^{v_i})^r \cdot (U' \prod_{k=1}^{n} U_k^{m_k})^{r_i}, g_{2_j}^{a+bz(i)\cdot \mathbf{M}_{i,j}}) \\ &= e((K^{\sum_{i=1}^l v_i \cdot \mathbf{M}_{i,j}})^r, g_{2_j}) \cdot e((U' \prod_{k=1}^{n} U_k^{m_k})^{r_i}, g_{2_j}^{(a+bz(i))\cdot \mathbf{M}_{i,j}}) \\ &= \begin{cases} e(Y, g_{2_1}) e((U' \prod_{k=1}^n U_k^{m_k}), P_1), & j = 1, \\ e((U' \prod_{k=1})^n U_k^{m_k}), P_j), & j > 1. \end{cases} \end{aligned}$$

Sanitization: From the definition of DI we note that $m_k' - m_k$ is 1 when $k \in I_1, -1$ when $k \in I_2$, and 0 otherwise. Thus we can conclude that

$$
\begin{aligned}
S_i' &= S_i \frac{\prod_{k\in I_1} U_k^{r_i}}{\prod_{k\in I_2} U_k^{r_i}} (U' \prod_{k=1}^{n} U_k^{m_k'})^{\tilde{r}_i} \\
&= (K_{z(i)}^{v_i})^r U'^{(r_i+\tilde{r}_i)} (\prod_{k=1}^{n} U_k^{m_k})^{r_i} (\prod_{k=1}^{n} U_k^{(m_k'-m_k)})^{\tilde{r}_i} (\prod_{k=1}^{n} U_k^{m_k'})^{\tilde{r}_i} \\
&= (K_{z(i)}^{v_i})^r U'^{(r_i+\tilde{r}_i)} (\prod_{k=1}^{n} U_k^{m_k'})^{(r_i+\tilde{r}_i)}, \\
P_j' &= P_j \prod_{i=1}^{l} (A_j B_j)^{\mathbf{M}_{i,j}\cdot\tilde{r}_i} = \prod_{i=1}^{l} (A_j B_j)^{\mathbf{M}_{i,j}\cdot(\tilde{r}_i+r_i)}.
\end{aligned}
$$

From the sanitization we can see that the distribution of the santized signature is identical to that of the original signature, so the verification fits both of them. □

Proof (Unforgeability). We can prove that our FABSS scheme is unforgeable under selective-predicate attack in the generic group model. Here we present the full proof of unforgeability.

For $Y = K^r \leftarrow \mathbb{G}_1$ and $W = K_0^r = K^{r/a_0} \leftarrow \mathbb{G}_1$, we suppose $Y = g_1^y, W = g_1^{y/a_0}$. Similarly, suppose $S_i = g_1^{s_i}, P_j = g_{2_j}^{p_j}$. We can derive that $S_i = g_1^{\frac{yv_i}{a+bz(i)} + u'r_i + \sum_{k=1}^{n} u_k m_k r_i}, P_j = g_{2_j}^{\sum_{i=1}^{l}(a+bz(i))\mathbf{M}_{i,j}\cdot r_i}$. So $s_i = \frac{yv_i}{a+bz(i)} + u'r_i + \sum_{k=1}^{n} u_k m_k r_i, p_j = \sum_{i=1}^{l}(a+bz(i))\mathbf{M}_{i,j}\cdot r_i$. Then we get $s_i(a+bz(i))\mathbf{M}_{i,j} = yv_i\mathbf{M}_{i,j} + r_i(u' + \sum_{k=1}^{n} u_k m_k)(a+bz(i))\mathbf{M}_{i,j}$. Then $\sum_{i=1}^{l}(s_i(a+bz(i))\mathbf{M}_{i,j}) = \sum_{i=1}^{l}(yv_i\mathbf{M}_{i,j} + r_i(u' + \sum_{k=1}^{n} u_k m_k)(a+bz(i))\mathbf{M}_{i,j})$. We assume that $\sum_{i=1}^{l} v_i \cdot \mathbf{M}_{i,j} = \boldsymbol{d} = [1, 0, \ldots, 0]$, then we conclude that $p_j = \frac{1}{u' + \sum_{k=1}^{n} u_k m_k} \cdot \big[\sum_{i=1}^{l}(s_i(a+bz(i))\mathbf{M}_{i,j}) - yd_j\big]$.

Therefore, we can define that the oracle $\mathbf{Sign}(Params, SK_{\mathcal{A}}, m, f)$ generates signatures in the following way: Let $\mathbf{M} \in (\mathbb{Z}_p)^{l\times t}$ be the monotone span program for f, with row labeling function $z : [l] \to \mathbb{A}$.

- Pick random $s_1, \ldots, s_l \leftarrow \mathbb{Z}_p^*$.
- For all $j \in [t]$, compute $p_j = \frac{1}{u' + \sum_{k=1}^{n} u_k m_k} \cdot \big[\sum_{i=1}^{l}(s_i(a+bz(i))\mathbf{M}_{i,j}) - yd_j\big]$, where $\boldsymbol{d} = [1, 0, \ldots, 0]$.
- Output $\sigma = (g_1^y, g_1^{y/a_0}, g_1^{s_1}, \ldots, g_1^{s_l}, g_{2_1}^{p_1}, \ldots, g_{2_t}^{p_t})$.

We assume that there is an efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. For any generic-group adversary, the simulator registers each group element's discrete logarithm in the following formal variables: $\Sigma = \{a_0, a, b, u', \lambda_0\} \cup \{\lambda_j \mid j \in [t_{max}]\} \cup \{x_\mu \mid \mu \in [\Lambda]\} \cup \{s_i^{(q)}, y^{(q)} \mid q \in [\nu], i \in [l^{(q)}]\}$, where Λ is the number of queries made to the **KeyGen** oracle, ν is the number of **Sign** queries made by the adversary, and $l^{(q)}$ is the length of the monotone span program corresponding to the qth signature query.

The simulation associates each group element with aforementioned formal variables. For each group element in its collection, the simulator keeps track of

its discrete logarithm and gives it to the adversary as the encoding of the group element. In the simulation, the group elements are expressed as follows:

Public key components are generated by **Setup**: 1, representing the generator g_1. λ_0, representing $g_{2_0} = g_1^{\lambda_0}$. $\lambda_0 a_0$, denoting $A_0 = g_1^{\lambda_0 a_0}$. $\{\lambda_j \mid j \in [t_{max}]\}$, indicating $g_{2_j} = g_1^{\lambda_j}$. $\{\lambda_j a \mid j \in [t_{max}]\}$, standing for $A_j = g_1^{\lambda_j a}$. $\{\lambda_j b \mid j \in [t_{max}]\}$, representing $B_j = g_1^{\lambda_j b}$. u', denoting $U' = g_1^{u'}$. $\{u_k \mid k \in [n]\}$, indicating $U_k = g_1^{u_k}$.

Signing key components are given by **KeyGen**. Let $\mathcal{A}_\mu$ be the μth set of attributes queried to **KeyGen**: x_μ, representing $K^{(\mu)} = g_1^{x_\mu}$. x_μ/a_0, denoting $K_0^{(\mu)} = g_1^{x_\mu/a_0}$. $\{x_\mu/(a+bz) \mid z \in \mathcal{A}_\mu\}$, indicating $K_z^{(\mu)} = g_1^{x_\mu/(a+bz)}$.

Sign queries. For the qth signature query on message $m^{(q)}$ under the predicate $f^{(q)}$ made by the adversary, let $\mathbf{M}^{(q)} \in (\mathbb{Z}_p)^{l^{(q)} \times t^{(q)}}$ be the monotone span program corresponding to $f^{(q)}$, with row labeling $z^{(q)} : [l^{(q)}] \to \mathbb{A}$: $\{s_i^{(q)} \mid i \in [l^{(q)}]\}$, representing $S_i^{(q)} = g_1^{s_i^{(q)}}$. $y^{(q)}$, denoting $Y^{(q)} = g_1^{y^{(q)}}$. $y^{(q)}/a_0$, standing for $W^{(q)} = g_1^{y^{(q)}/a_0}$. $\{p_j^{(q)} \mid j \in [t^{(q)}]\}$, where $p_j^{(q)} = \frac{\lambda_j}{u' + \sum_{k=1}^{n^{(q)}} u_k^{(q)} m_k^{(q)}} \cdot \left[\sum_{i=1}^{l^{(q)}} (s_i^{(q)}(a + bz^{(q)}(i))\mathbf{M}_{i,j}) - y^{(q)} d_j\right]$, representing $P_j^{(q)} = g_1^{p_j^{(q)}}$.

Now the adversary outputs a forgery signature $\sigma^* = (g_1^{y^*}, g_1^{w^*}, g_1^{s_1^*}, \ldots, g_1^{s_{l^*}^*}, g_1^{p_1^*}, \ldots, g_1^{p_{t^*}^*})$ on a predicate f^* and message m^* such that $(m^*, f^*) \neq (m^{(q)}, f^{(q)})$ for all q. $\mathbf{M}^* \in (\mathbb{Z}_p)^{l^* \times t^*}$ is the corresponding monotone span program with row labeling $z^*(\cdot)$. The discrete logarithm of the forgery has to satisfy $y^* \neq 0, w^* \neq 0$, for $Y^* \neq 1, W^* \neq 1$ and $\sum_{i=1}^{l^*} s_i^* \mathbf{M}_{i,j}^*(a + bz^*(i))\lambda_j = y^* d_j \lambda_j + (u' + \sum_{k=1}^{n^*} u_k^* m_k^*) p_j^*$, these constraints can hold with non-negligible probability only if two sides of the equation are functionally equivalent.

Then we will prove if the two sides of the equation are functionally equivalent, there has to be a contradiction: there exists a $\mu_0 \in [\Lambda]$ such that $f^*(\mathcal{A}_{\mu_0}) = 1$. Namely, the adversary may generate a signature using the signing key $SK_{\mathcal{A}_{\mu_0}}$ that has been queried before but meets the new claim-predicate f^*, and thus the output is not a forgery.

Assume $\mathbb{L}(\Gamma)$ is the set of all multilinear polynomials over the set of terms Γ with coefficients in $\mathbb{Z}_p$. Let $\mathbb{H}(\Gamma) \subset \mathbb{L}(\Gamma)$ be the subset of homogeneous polynomial.

We know that $y^*, w^*, s_1^*, \ldots, s_{l^*}^*, p_1^*, \ldots, p_{t^*}^* \in \mathbb{L}(\Gamma)$, where $\Gamma = \{1, a_0, \lambda_0, u', u_k\} \cup \{\lambda_j, a\lambda_j, b\lambda_j \mid j \in [t_{max}]\} \cup \{x_\mu, x_\mu/a_0, x_\mu/(a+bz) \mid \mu \in [\Lambda], z \in \mathcal{A}_\mu\} \cup \{s_i^{(q)}, y^{(q)}, w^{(q)}, p_j^{(q)} \mid q \in [\nu], i \in [l^{(q)}], j \in [t^{(q)}]\}$. We can exclude certain terms by comparing terms between the equation, then for y^* we can get $y^* \in \mathbb{H}(\{x_\mu \mid \mu \in [\Lambda]\} \cup \{y^{(q)} \mid q \in [\nu]\})$. It is obvious that $\lambda_j \mid (u' + \sum_{k=1}^{n^*} u_k^* m_k^*) p_j^*$ and thus $\lambda_j \mid p_j^*$. So, $p_j^* \in \mathbb{H}(\{\lambda_j, a\lambda_j, b\lambda_j\} \cup \{p_j^{(q)} \mid q \in [\nu]\})$.

Suppose p_j^* has a λ_j term. Then the right side has monomials λ_j and $b\lambda_j$. Because y^* has no a or b term, $y^* d_j \lambda_j$ cannot contribute a λ_j monomial. Therefore $\sum_{i=1}^{l^*} s_i^* \mathbf{M}_{i,j}^*(a + bz^*(i))\lambda_j$ cannot contribute a monomial with λ_j alone, so $p_j^* \in \mathbb{H}(\{a\lambda_j, b\lambda_j\} \cup \{p_j^{(q)} \mid q \in [\nu]\})$.

Suppose p_j^* has a $p_j^{(q)}$ term. Then $(u' + \sum_{k=1}^{n^*} u_k^* m_k^*) p_j^*$ will contribute the term of $(\frac{u' + \sum_{k=1}^{n^*} u_k^* m_k^*}{u' + \sum_{k=1}^{n^{(q)}} u_k^{(q)} m_k^{(q)}}) \cdot p_j^{(q)}$. Since $\sum_{k=1}^{n^*} u_k^* m_k^* \neq \sum_{k=1}^{n^{(q)}} u_k^{(q)} m_k^{(q)}$ for any q, this is a proper rational. Neither y^* nor $\{s_i^*\}_{i \in l^*}$ can yield terms in the final equation with a factor of $\frac{u' + \sum_{k=1}^{n^*} u_k^* m_k^*}{u' + \sum_{k=1}^{n^{(q)}} u_k^{(q)} m_k^{(q)}}$. Hence, $p_j^* \in \mathbb{H}(\{a\lambda_j, b\lambda_j\})$.

Consider j_0 such that $d_{j_0} \neq 0$. As neither $(u' + \sum_{k=1}^{n^*} u_k^* m_k^*) p_{j_0}^*$ nor $\sum_{i=1}^{l^*} s_i^* \mathbf{M}_{i,j_0}^* (a + bz^*(i)) \lambda_{j_0}$ can contribute a monomial of this form, y^* cannot have a $y^{(q)}$ term. Therefore, $y^* \in \mathbb{H}(\{x_\mu \mid \mu \in [\Lambda]\})$. Finally we conclude that $p_j^* \in \mathbb{H}(\{a\lambda_j, b\lambda_j\}), y^* \in \mathbb{H}(\{x_\mu \mid \mu \in [\Lambda]\})$.

To make the expression equal, some parts of the left side have x_μ to fit y^* and the other parts do not have x_μ to satisfy p_j^*. So, we can break s_i^* up into two parts: one whose terms involve x_μ variables, and one whose terms do not. Suppose $s_i^* = t_i^*(X_i) + \delta^*(\Gamma \setminus X_i)$, where $X_i = \{\frac{x_\mu}{a + bz^*(i)} \mid z(i) \in \mathcal{A}_\mu, \mu \in [\Lambda]\}$ is to cancel out the term $(a + bz^*(i))$ from the left side. For $t_i^* \in \mathbb{H}(X_i)$, it is apparent for all $j \in [t]$ that $\sum_{i=1}^{l^*} t_i^* \mathbf{M}_{i,j}^* (a + bz^*(i)) = y^* d_j = y^* \sum_{i=1}^{l^*} v_i^* \mathbf{M}_{i,j}^*$, because of the equality of two sides of the equation, we get for all $i \in [l]$ that $t_i^* \mathbf{M}_{i,j}^* (a + bz^*(i)) = y^* v_i^* \mathbf{M}_{i,j}^*$.

Take account of any x_{μ_0} that has a non-zero coefficient in y^*. Construct v_i^*, for $i \in [l]$, by defining $v_i^* = \frac{1}{[x_{\mu_0}] y^*} \left[\frac{x_{\mu_0}}{a + bz^*(i)}\right] t_i^*$, where the $[x_{\mu_0}] y^*$ denotes the coefficient of the term x_{μ_0} in y^*, $\left[\frac{x_{\mu_0}}{a + bz^*(i)}\right] t_i^*$ denotes the coefficient of the term $\frac{x_{\mu_0}}{a + bz^*(i)}$ in t_i^*. $\boldsymbol{v}^*$ is a vector composed of constants, which satisfies the equation $\boldsymbol{v}^* \mathbf{M}^* = [d_1, \ldots, d_t] = [1, 0, \ldots, 0]$. Further, when $v_i^* \neq 0$, the set $\mathcal{A}_{\mu_0}$ surely contains the attribute $z^*(i)$, which means $x_{z^*(i)} \neq 0$. By the properties of the monotone span program, it must be the case that $f^*(\mathcal{A}_{\mu_0}) = 1$, thus the signature is not a forgery. □

Proof (Anonymity). In our construction, the signature will not reveal which attributes of the signer's attributes $\mathcal{A}$ are used to sign the message, because any attribute subset satisfying the access structure f can generate a valid signature. Thus, we only need to prove that the signer's identity among all users is kept anonymous even when $\mathcal{A} = \mathbb{A}$, where $\mathbb{A}$ is the attributes in f.

First, the challenger runs **Setup** to get the public parameters $Params$ and master secret keys MSK. The adversary outputs two attributes $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfying f, and conducts **KeyGen** to get signing keys $SK_{\mathcal{A}_1} = (K_1, K_{0_1}, \{K_{z_1} \mid z \in \mathcal{A}_1\}$ and $SK_{\mathcal{A}_2} = (K_2, K_{0_2}, \{K_{z_2} \mid z \in \mathcal{A}_2\}$, respectively. Let $K_\theta, K_{0_\theta} = K_\theta^{1/a_0}, K_{z_\theta} = K_\theta^{1/(a+bz)}$ for each $z \in \mathcal{A}_\theta$, where $\theta \in \{1, 2\}$.

Then the adversary asks the challenger to generate a signature for message m^* with the signing key from either $SK_{\mathcal{A}_1}$ or $SK_{\mathcal{A}_2}$. The challenger chooses a random bit $b \in \{1, 2\}$ and outputs a signature $Y = K^r, W = K_0^r, S_i = K_{z(i)}^{v_i} \cdot (U' \prod_{k=1}^{n} U_k^{m_k})^{r_i}, P_j = \prod_{i=1}^{l} (A_j B_j)^{\mathbf{M}_{i,j} \cdot r_i}$ by the algorithm **Sign** with the signing key $SK_{\mathcal{A}_b} = (K_b, K_{0_b}, \{K_{z_b} \mid z \in \mathcal{A}_b\}$. On the basis of Monotone Span Program, it is obvious that it could be generated from either $SK_{\mathcal{A}_1}$ or

$SK_{\mathcal{A}_2}$. Hence, if the signature is generated from $SK_{\mathcal{A}_1}$ for $\mathcal{A}_1$, it could also be generated from $SK_{\mathcal{A}_2}$ for $\mathcal{A}_2$. Thus, our FABSS scheme satisfies anonymity. □

Proof (Information Privacy). From the construction of our scheme, the signature of message $m' = m'_1 m'_2 \dots m'_n$ is $\sigma = (Y = K^r, S_i = (K_{z(i)}^{v_i})^r \cdot (U' \prod_{k=1}^{n} U_k^{m_k})^{r_i} (\forall i \in [l]), W = K_0^r, P_j = \prod_{i=1}^{l} (A_j B_j^{z(i)})^{\mathbf{M}_{i,j} \cdot r_i} (\forall j \in [t]))$. The sanitized signature of message m_1 resulting in m' is $\sigma'_1 = (Y = K^r, \{S_i = (K_{z(i)}^{v_i})^r \cdot (U' \prod_{k=1}^{n} U_k^{m'_k})^{r_i + \tilde{r}_i} : (\forall i \in [l])\}, W = K_0^r, \{P_j = \prod_{i=1}^{l} (A_j B_j^{z(i)})^{\mathbf{M}_{i,j} \cdot (r_i + \tilde{r}_i)} : (\forall j \in [t])\})$, where $r, r_i, \tilde{r}_i$ are random numbers. So the distribution of σ is identical to that of σ'_1. Similarly, the distribution is identical to σ'_2 of message m_2 resulting in m'. Hence, the distribution of σ'_1 and σ'_2 are identical and our scheme preserves the information privacy. □

6 Performance Analysis

Through comparing with existing scheme functionally in Table 1, our FABSS scheme not only reduces the patients computation cost, but also preserves the privacy of patients. Meanwhile, the FABSS scheme achieves flexible access structure and fine-grained access control. Thus, our scheme applies to the EHR system.

Table 1. Functional analysis

	FABSS	ABSS [17]	ABS [12]	SS [8]
Reduce patients' computation cost	✓	✓	×	✓
Flexible access structure	✓	×	×	×
Anonymity	✓	✓	✓	×
Fine-grained access control	✓	✓	✓	×

In Table 2 we specify the efficiency of our scheme. For the ease of exposition we assume $\mathbb{G}_1, \mathbb{G}_2$ are symmetric, treating $\mathbb{G}_1$ as the base group and $\mathbb{G}_2$ as the bilinear group $\mathbb{G}_T$ in our scheme. In scheme [8,12,17], n denotes the sum of the attributes in the system, m is the length of the message, ω is the signers attributes, the threshold value is expressed by k and $d \geq k$. I is the order of UI. In our scheme, we first convert f into the matrix $\mathbf{M}^{l \times t}$, then denote the length and width of the matrix by l and t, respectively, where $l = n, t = k$. EX is the number of the exponent arithmetic and P is the number of the pairing arithmetic.

From Table 2 we find that our scheme exceeds in $Key.Size, Key.Gen$, $Sig.Size, Sansig.Size$ than that of [12,17] and is inferior to that of [8], because [8] does not consider the privacy of users and thus does not include the attribute sets. The size of $Params, MSK$ is similar to that of [12,17]. Furthermore, the computation cost of $Sig.Gen, Sansig.Gen$ and $Verify$ is longer than that of [8,17], which is due to the flexible access structure with matrix $\mathbf{M}$ and admissible.

Table 2. Efficiency analysis

	FABSS	ABSS [17]	ABS [12]	SS [8]
$Params$	$(3t+m+5)\mathbb{G}_1$	$(m+n+4)\mathbb{G}_1+\mathbb{G}_2$	$(m+3)\mathbb{G}_1+\mathbb{G}_2$	$(m+4)\mathbb{G}_1+\mathbb{G}_2$
MSK	$3\mathbb{Z}_p$	$\mathbb{Z}_p$	$\mathbb{Z}_p$	$\mathbb{Z}_p$
$Key.Size$	$(\omega+2)\mathbb{G}_1$	$2(\omega+d-1)\mathbb{G}_1$	$2(\omega+d-1)\mathbb{G}_1$	-
$Key.Gen$	$(\omega+1)EX$	$3(\omega+d-1)EX$	$3(\omega+d-1)EX$	-
$Sig.Size$	$(l+t+2)\mathbb{G}_1$	$2(n+d-k+1)\mathbb{G}_1$	$(n+d-k+2)\mathbb{G}_1$	$2\mathbb{G}_1$
$Sansig.Size$	$(l+t+2)\mathbb{G}_1$	$2(n+d-k+1)\mathbb{G}_1$	-	$2\mathbb{G}_1$
$Sig.Gen$	$(lt+2l+m+2)EX$	$(4n+6d-4k+m+2)EX$	$(2(n+2d-k)+m+2)EX$	$(m+3)EX$
$Sansig.Gen$	$(l(t+1)+m+I)EX$	$(4(n+d-k)+m+I+2)EX$	-	$(m+I+2)EX$
$Verify$	$(tl+t+3)P+(2tl+m)EX$	$2(n+d-k+1)P+mEX$	$(n+d-k+2)P+mEX$	$3P+mEX$

7 Conclusion

In order to reduce the computation cost and keep the identity privacy of users in the EHR system, we propose the Flexible Attribute-Based Sanitizable Signature (FABSS) scheme. Security demonstration shows that our scheme is unforgeable and preserves the anonymity and information privacy of the users. Compared with existing scheme, our scheme not only reduce the users' computation cost when data updating, but also supports flexible access structure defining expressive access control in large-scale users. Further efforts can be made on enhancing the security model of our FABSS scheme. In addition, we will exploit multi-authority FABSS scheme in which the attributes are assigned by different attribute authorities.

Acknowledgement. This work is supported by the National High Technology Research and Development Program (863 Program) (No. 2015AA016007, No. 2015AA017203), the Key Program of NSFC Grant (No. U1405255, No. U1135002), the National Natural Science Foundation of China (No. 61502248) and the NUPTSF (No. 215008). The authors would like to thank the editors and the anonymous reviewers for their constructive comments that would help us to improve this paper.

References

1. Ateniese, G., Chou, D.H., de Medeiros, B., Tsudik, G.: Sanitizable signatures. In: di Vimercati, S.C., Syverson, P., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 159–177. Springer, Heidelberg (2005). https://doi.org/10.1007/11555827_10
2. Brzuska, C., Fischlin, M., Freudenreich, T., Lehmann, A., Page, M., Schelbert, J., Schröder, D., Volk, F.: Security of sanitizable signatures revisited. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 317–336. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00468-1_18
3. Brzuska, C., Fischlin, M., Lehmann, A., Schröder, D.: Unlinkability of sanitizable signatures. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 444–461. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_26
4. Canard, S., Laguillaumie, F., Milhau, M.: Trapdoor sanitizable signatures and their application to content protection. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 258–276. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_16

5. Lai, J., Ding, X., Wu, Y.: Accountable trapdoor sanitizable signatures. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 117–131. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38033-4_9
6. Miyazaki, K., Hanaoka, G., Imai, H.: Digitally signed document sanitizing scheme based on bilinear maps. In: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, pp. 343–354 (2006)
7. Yuen, T.H., Susilo, W., Liu, J.K., Mu, Y.: Sanitizable signatures revisited. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 80–97. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89641-8_6
8. Agrawal, S., Kumar, S., Shareef, A., Rangan, C.P.: Sanitizable signatures with strong transparency in the standard model. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Inscrypt 2009. LNCS, vol. 6151, pp. 93–107. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16342-5_7
9. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of 13th ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
10. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
11. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_24
12. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Proceedings of 5th ACM Symposium on Information, Computer and Communications Security, pp. 60–69 (2010)
13. Okamoto, T., Takashima, K.: Efficient attribute-based signatures for non-monotone predicates in the standard model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 35–52. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_3
14. Su, J., Cao, D., Zhao, B., Wang, X., You, I.: ePASS: an expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things. Future Gener. Comput. Syst. **33**, 11–18 (2014)
15. Rao, Y.S., Dutta, R.: Efficient attribute-based signature and signcryption realizing expressive access structures. Int. J. Inf. Secur. **15**, 81–109 (2016)
16. Li, J., Chen, X., Huang, X.: New attribute-based authentication and its application in anonymous cloud access service. Int. J. Web Grid Serv. **11**, 125–141 (2015)
17. Liu, X., Ma, J., Xiong, J., Ma, J., Li, Q.: Attribute based sanitizable signature scheme. J. Commun. **34**, 148–155 (2013)
18. Xu, L., Zhang, X., Wu, X., Shi, W.: ABSS: an attribute-based sanitizable signature for integrity of outsourced database with public cloud. In: Proceedings of 5th ACM Conference on Data and Application Security and Privacy, pp. 167–169 (2015)