Security Analysis on a Secure Medical Data Sharing System in Digital Twin Environments

Mizuki Hayashi[1] and Keita Emura*[1,2]

[1.]Kanazawa University, Japan.
[2.]AIST, Japan.

August 26, 2026

Abstract

Gao et al. (IEEE Internet of Things Journal, 2025) proposed a medical data sharing system for digital twin environments using identity-based encryption (IBE), public-key encryption with keyword search (PEKS), and blockchain technologies. In this short note, we show that Gao et al.'s system allows unauthorized users to access other patients' medical data. We further show that the search server can obtain information about the queried keywords from the trapdoors (search queries). In addition, we analyze the procedure used to retrieve, from the blockchain, the IPFS (InterPlanetary File System) addresses storing encrypted medical data and encrypted keywords. Since these addresses are derived from labels that can be computed solely from public information and keywords, and because the keywords themselves are provided to the search server, we demonstrate that searchable encryption is unnecessary in the first place. Based on our security analysis, we argue that the proposed system requires a fundamental redesign.

1 Introduction

To investigate the extent to which medical data are considered in digital twin research, we searched the medical literature database PubMed[1] using the keyword "digital twin" and obtained 3,030 papers (as of August 26, 2026). Among these publications, we cite the survey paper by Laubenbacher et al. [3], which provides an overview of the current state of medical digital twins. The survey particularly focuses on oncology and cardiology, where development is most advanced, and discusses challenges related to data integration and privacy.

Our Motivation. Medical data contain sensitive information, such as patients' medical conditions, and therefore require appropriate protection when stored and processed electronically, regardless of whether they are used in digital twin environments. To the best of our knowledge, while conventional cryptographic primitives such as public/symmetric-key encryption, digital signatures, and hash functions have been deployed in medical settings, advanced cryptographic primitives, which provide functionality beyond that of conventional cryptographic primitives, have not yet seen widespread practical adoption. Although advanced cryptographic primitives offer various security and functionality advantages, their security definitions are often considerably more complex

*Corresponding Author: k-emura@se.kanazawa-u.ac.jp

[1] https://pubmed.ncbi.nlm.nih.gov/

than those of conventional cryptographic primitives, and a number of challenges must be addressed to ensure their proper use in practice.

In this short note, we conduct a security analysis of a medical data-sharing system for digital twin environments proposed by Gao et al. [2], which employs advanced cryptographic primitives including identity-based encryption (IBE) [4, 5] and public-key encryption with keyword search (PEKS) [1].

Our Contribution. In this short note, we show that Gao et al.'s system allows unauthorized users to access other patients' medical data. Our attack exploits the fact that the master secret key msk of the underlying IBE scheme (the Waters IBE scheme [5]) is shared among all users. From a cryptographic perspective, it is evident that msk, which enables the generation of secret keys for arbitrary identities, should never be shared. We believe that the issue stems from a misunderstanding of IBE, whose design and security model are considerably more complex than those of conventional cryptographic primitives.

Furthermore, we show that the cloud server responsible for keyword searches can obtain information about the queried keywords from the trapdoors. We also observe that encrypted medical data and encrypted keywords are stored in the InterPlanetary File System (IPFS). To retrieve the corresponding IPFS address, the system computes a keyword-dependent label, $\mathsf{Label}_w = H_1(\mathsf{PK}_{ID} \| w)$, where H_1 denotes a collision-resistant hash function. Consequently, the searchable encryption component provides no meaningful benefit in this setting. Based on these findings, we argue that the system requires a fundamental redesign.

Remark. We identified several ambiguous points in the paper by Gao et al. Since the interpretation of these points affects the attack scenario, we clarify our interpretation here.

In their system, for $\alpha, y, r \in \mathbb{Z}_p$, the search key $\mathsf{sk}_{\mathsf{search}} = (y, g_2^{\alpha})$ is defined. Gao et al. also set another value $\beta \in \mathbb{Z}_p$ which is not used within the system, and it is written as $\mathsf{sk}_{\mathsf{search}} = (y, g_2^{\beta})$ in only one place. From this, the possibility that different master keys are used depending on the purpose, with α used for encryption and β for searching, cannot be ruled out. However, considering that α is also used in the search part, and that g_2^{β} is included in the public parameters, and that $\mathsf{sk}_{\mathsf{search}}$ is the key that realizes the delegation function of search privileges, it is unlikely that β is used for search purposes. Therefore, in this short note, we interpreted $\mathsf{sk}_{\mathsf{search}} = (y, g_2^{\beta})$ as a typo for $\mathsf{sk}_{\mathsf{search}} = (y, g_2^{\alpha})$ and performed a security evaluation.

In Gao et al.'s paper, SK_S is inconsistently used as both the sender's secret key and the server's secret key. During encryption, SK_S serves as the sender's secret key since the computation of $H(w)^{\mathsf{SK}_S}$ requires it. However, during searching, SK_S is described as the server's secret key. Moreover, the Test algorithm requires the computation of PK_S^{δ}, where δ is generated during trapdoor generation and is unknown to the party executing Test. Since g_2^{δ} is included in the trapdoor, it is natural to compute PK_S^{δ} as $(g_2^{\delta})^{\mathsf{SK}_S}$. Furthermore, because $H(w)^{\mathsf{SK}_S}$ is also required during trapdoor generation, searching becomes impossible unless the role of SK_S is uniquely defined. To resolve this inconsistency, we focus on the phrase "Cloud Server (User's account)" in Gao et al.'s paper and interpret the sender as performing searches over ciphertexts stored on the cloud server. Under this interpretation, PEKS is effectively used in a manner similar to Symmetric Searchable Encryption (SSE). Although requiring the sender's secret key SK_S during searching appears unusual, this interpretation is more reasonable than requiring the cloud server's secret key during encryption.

Disclaimer. While we respect Gao et al.'s attempt to construct a medical data-sharing system using advanced cryptographic primitives and blockchain technologies, this short note demonstrates through concrete attacks that their system does not provide the intended level of security. Our

purpose is to identify and analyze these issues from the perspective of promoting the proper use of advanced cryptographic primitives, rather than to unfairly criticize the work of Gao et al. Given that the paper was published in the IEEE Internet of Things Journal, a highly influential JCR Q1 journal with Impact Factor 8.7, we believe it is worthwhile to point out that advanced cryptographic primitives such as IBE and PEKS can be misused even in papers published in top-tier venues. In particular, we highlight the sharing of the IBE master secret key and the blockchain-based IPFS access mechanism that undermines the rationale for employing PEKS.

2 Gao et al.'s Medical Data Sharing System

In this section, we introduce the medical data sharing system proposed by Gao et al. We omit the details of the distributed key generation process here. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order p, let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be generators and let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing. Let $H : \{0,1\}^* \to \{0,1\}^n$, $H' : \{0,1\}^* \to \mathbb{G}_1$, and $H_1 : \{0,1\}^* \to \{0,1\}^\lambda$ be collision resistant hash functions (where n is the ID length and λ is the security parameter). Also, let $u_0, \ldots, u_n \in \mathbb{G}_1$.

System Setup: Generate public parameters $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, p, g_1, g_2, g_1^\alpha, u_0, \ldots, u_n)$.

Patient Key Generation: For a patient's ID, compute $v = H(ID)$ and select $r, y \xleftarrow{\$} \mathbb{Z}_p$. Compute the encryption key $\mathsf{sk}_{\mathsf{enc}} = (g_2^\alpha (u_0 \prod u_i^{v_i})^r, g_2^r)$, the search key $\mathsf{sk}_{\mathsf{search}} = (y, g_2^\alpha)$, and the public key $\mathsf{pk}_{ID} = g_1^y$. Output the public key $\mathsf{PK}_{ID} = \mathsf{pk}_{ID}$ and the secret key $\mathsf{SK}_{ID} = (\mathsf{sk}_{\mathsf{enc}}, \mathsf{sk}_{\mathsf{search}})$.

Sender Key Generation: Although the paper by Gao et al. refers to this as $\mathsf{ServerKeyGen}$, we interpret it, as previously mentioned, as key generation for the sender (doctor), with SK_S being shared with the cloud server. Choose $x \xleftarrow{\$} \mathbb{Z}_p$ and output $(\mathsf{SK}_S, \mathsf{PK}_S) = (x, g_2^x)$. Note that the sender is assumed to possess g_2^α.

Doctor: Extract keywords from the patient's EHR. For each keyword w, generate a label $\mathsf{Label}_w = H_1(\mathsf{PK}_{ID} || w)$ and a set of IPFS addresses $\mathsf{PointerSet}_w = \{\mathsf{Addr}_1, \ldots, \mathsf{Addr}_\ell\}$ where the ciphertext of w is stored. Define the index structure as $\{(\mathsf{Label}_w, \mathsf{PointerSet}_w) \mid w \in \mathcal{W}\}$ and store it on the blockchain. Here, $\mathcal{W}$ is the keyword space. Using the patient's ID and PK_{ID}, as well as the sender's (doctor's) secret key SK_S, the patient's EHR information M and the keyword w related to the EHR content are encrypted as follows.

EHR Encryption: Choose $t \xleftarrow{\$} \mathbb{Z}_p$, and compute $C_1 = M \cdot e(g_1^\alpha, g_2)^t$, $C_2 = g_1^t$, and $C_3 = (u_0 \prod_{i=1}^n u_i^{v_i})^t$. Here, v_i is the i-th bit of $v = H(ID)$.

Keyword Encryption: Choose $k \xleftarrow{\$} \mathbb{Z}_p$, and compute $C_4 = w \cdot \mathsf{PK}_{ID}^k \cdot H(w)^{\mathsf{SK}_S}$ and $C_5 = g_1^k$.

Store $CT = (C_1, C_2, C_3, C_4, C_5)$ on IPFS. Compute the trapdoor T_w as follows.

Trapdoor Generation: Choose $\delta \xleftarrow{\$} \mathbb{Z}_p$. Using the sender's (doctor's) secret key SK_S and the patient's public key PK_{ID}, compute $t_1 = e(w \cdot H(w)^{\mathsf{SK}_S}, g_2^\alpha) e(\mathsf{PK}_{ID}, \mathsf{PK}_S^\delta)$ and $t_2 = g_2^\delta$, and output $T_w = (t_1, t_2)$.

When requesting access to a patient's medical data, the doctor is assumed to send T_w and C_4 to the patient (although not explicitly stated in Gao et al.'s paper, we adopt this interpretation because the cloud server uses (t_1, t_2) and C_4 to verify $\mathsf{T}_{\mathsf{auth}}$). After the request is approved,

the keyword w is sent to the cloud server (at this point, it becomes evident that searchable encryption is not actually being utilized). After obtaining the IBE ciphertext from the cloud server, the patient's EHR is retrieved by decrypting it.

Patient: Use $\mathsf{sk}_{\mathsf{search}}$ to generate a delegation trapdoor $\mathsf{T}_{\mathsf{auth}}$. Although it is named "trapdoor," it is not linked to a keyword. Therefore, it is interpreted as a value for authentication. In response to the request from the doctor, $\mathsf{T}_{\mathsf{auth}}$ is computed as follows.

Delegated Trapdoor Generation: Choose $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$. Using $\mathsf{sk}_{\mathsf{search}} = (y, g_2^{\alpha})$, the patient computes $t_3 = g_2^{\alpha}\mathsf{PK}_S^{r_1}$, $t_4 = g_2^{r_1}$, $t_5 = g_2^{y\alpha}\mathsf{PK}_S^{r_2}$, $t_6 = g_2^{r_2}$, and outputs $\mathsf{T}_{\mathsf{auth}} = (t_3, t_4, t_5, t_6)$.

Send $\mathsf{T}_{\mathsf{auth}}$, $T_w = (t_1, t_2)$ and C_4 to the cloud server.

Cloud Server: Verify the following using the shared doctor's secret key SK_S.

Verification of $\mathsf{T}_{\mathsf{auth}}$: Compute $t_7 = t_3/t_4^{\mathsf{SK}_S}, t_8 = t_5/t_6^{\mathsf{SK}_S}, \tilde{a} = e(\mathsf{PK}_{ID}, t_2^{\mathsf{SK}_S}) \cdot e(c_4, t_7)/t_1, \tilde{b} = e(c_5, t_8)$ and output 1 if $\tilde{a} = \tilde{b}$, and 0 otherwise.

Based on the above, after confirming that the patient has granted the doctor access rights to the medical data, the keyword w is obtained from the doctor. It generates a label from the keyword, and retrives the IPFS address using this label and the index structure stored on the blockchain (smart contract). The Test algorithm is then executed on the PEKS ciphertext stored in IPFS using the trapdoor and SK_S.

Test **Algorithm:** Compute $\tilde{c} = e(c_4, t_7), \tilde{d} = e(c_5, t_8)$ and output 1 if $\tilde{c} \cdot e(\mathsf{PK}_{ID}, \mathsf{PK}_S^{\delta}) = \tilde{d} \cdot e(t_1, g_2^{\alpha})$, and 0 otherwise. Although the computation of PK_S^{δ} can be interpreted as $t_2^{\mathsf{SK}_S}$, the expression is incorrect because $t_1 \in \mathbb{G}_T$ is used as an input to the pairing.

Blockchain: Store the index structures created by the doctor and the logs for each process.

3 Our Attacks

It should be noted that any entity can access the ciphertext stored on IPFS since a table containing labels and IPFS addresses is publicly available on the blockchain, and that the secret key is not used to compute the label ($\mathsf{Label}_w = H_1(\mathsf{PK}_{ID} || w)$). So, any entity possessing $\mathsf{sk}_{\mathsf{search}}$ containing the Waters IBE master secret key $\mathsf{msk} = g_2^{\alpha}$ can view the EHR M by $C_1/e(C_2, g_2^{\alpha}) = M$.

Furthermore, during the verification process for $\mathsf{T}_{\mathsf{auth}}$, the cloud server computes $t_3/t_4^{\mathsf{SK}_S} = g_2^{\alpha}$. This implies that the cloud server can also view the medical data. As previously mentioned, the use of searchable encryption is rendered meaningless. Nevertheless, it would be meaningful to evaluate the PEKS scheme proposed by Gao et al. For a trapdoor $T_w = (t_1, t_2)$, the following equation holds:

$$\begin{aligned} t_1/e(\mathsf{PK}_{ID}, t_2^{\mathsf{SK}_S}) &= e(w \cdot H(w)^{\mathsf{SK}_S}, g_2^{\alpha}) e(\mathsf{PK}_{ID}, \mathsf{PK}_S^{\delta}) / e(\mathsf{PK}_{ID}, t_2^{\mathsf{SK}_S}) \\ &= e(w \cdot H(w)^{\mathsf{SK}_S}, g_2^{\alpha}) \\ &= e(w \cdot H(w)^{\mathsf{SK}_S}, t_3/t_4^{\mathsf{SK}_S}) \end{aligned}$$

where $g_2^{\alpha} = t_3/t_4^{\mathsf{SK}_S}$. An adversary possessing SK_S declares two challenge keywords (w_0^*, w_1^*). The challenger randomly chooses $b \xleftarrow{\$} \{0, 1\}$, computes $T_w^* = (t_1, t_2)$, and returns it to the adversary.

If the cloud server obtains $\mathsf{T}_{\mathsf{auth}}$, it also becomes possible to acquire (t_3, t_4). If $t_1/e(\mathsf{PK}_{ID}, t_2^{\mathsf{SK}_S}) = e(w_0^* \cdot H(w_0^*)^{\mathsf{SK}_S}, t_3/t_4^{\mathsf{SK}_S})$ holds, then output $b = 0$, and $b = 1$, otherwise.

4 Conclusion

In this short note, we evaluated the security of the medical data sharing system proposed by Gao et al. We also observe that several entities in Gao et al.'s system are not clearly defined. For example, it is unclear whether Data Users and Data Researchers correspond to Doctors, Patients, or other entities. As a result, the system's security objectives and threat model are ambiguous, which we believe is one of the root causes of the vulnerabilities identified in this short note. These observations suggest that the system requires a fundamental redesign before providing the intended security guarantees.

Acknowledgment: This work was supported by JSPS KAKENHI Grant Number JP25H01106. This paper was edited for grammar using Copilot.

References

[1] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[2] Hongmin Gao, Yuhe Zhang, Libo Wang, Zhiquan Liu, and Zhaofeng Ma. A searchable encryption scheme for blockchain-based digital twins. *IEEE Internet Things Journal*, 12(20):42274–42289, 2025.

[3] Reinhard C. Laubenbacher, Borna Mehrad, Ilya Shmulevich, and Natalia A. Trayanova. Digital twins in medicine. *Nature Computational Science*, 4(3):184–191, 2024.

[4] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, volume 196, pages 47–53, 1984.

[5] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.