A PRIVACY- AWARE TRACKING AND TRACING SYSTEM

Ali M. Allam

Associate Professor, Communications, and Electronics Engineering Department, Faculty of Engineering, Helwan University, Egypt

ABSTRACT

The ability to track and trace assets in the supply chain is becoming increasingly important. In addition to asset tracking, the technologies used provide new opportunities for collecting and analyzing employee position and biometric data. As a result, these technologies can be used to monitor performance or track worker behavior, resulting in additional risks and stress for employees. Furthermore, contact tracing systems used to contain the COVID-19 outbreak have made positive patients' privacy public, resulting in violations of users' rights and even endangering their lives. To resolve this situation, a verifiable attribute-based encryption (ABE) scheme based on homomorphic encryption and zero-knowledge identification (ZKI) is proposed, with ZKI providing anonymity for data owners to resist tracking attacks and homomorphic encryption used to solve the problem of privacy leakage from location inquiries returned from a semi-honest server. Finally, theoretical security analysis and formal security verification show that our scheme is secure against the chosen plaintext attack (CPA) and other attacks. Besides that, our novel scheme is efficient enough in terms of user-side computation overhead for practical applications.

KEYWORDS

Privacy preservation; asset tracking; monitoring; contact tracing; COVID-19

1. INTRODUCTION

In recent years, object tracking and tracing have become widely used in many fields, including monitoring patients in the healthcare system and tracking assets in the supply chain. Detailed information about an entity's position and status allows for better logistics planning and scheduling, and it can also be used to combat pandemic spread in the healthcare field. Real-time load tracking services, for example, have been successfully implemented and provide a clear benefit to logistics companies [1], and contact tracing provided numerous benefits during the COVID-19 pandemic [2]. With the rise of tracking and tracing systems, aspects other than economic and health benefits must be considered. One of the issues that arise in industrial and health contexts is the exposure of personal data to a technical system via a potentially insecure communication channel.

As a result, these technologies can be used to monitor performance or track worker behavior, resulting in additional burdens and stress for employees. Also, these systems can track our movements and activities over time, which can help with crime detection and fraud investigation. However, without the necessary authority to obtain it, criminals or government agencies may abuse this information. To avoid this, we must ensure that the tracking or tracing system we use has sufficient privacy safeguards.

DOI: 10.5121/ijcnc.2023.15507

Most mobile devices use GPS to determine their position, produce a lot of tendency data, and perform a lot of operations, which is not suitable for devices with limited resources. For processing and storage, the user's device typically transmits the data it generates to the cloud [3]. As a result, confidential information about data owners is compromised, leaving them vulnerable to tracking attacks.

In cloud computing services, the cloud server is assumed to follow the honest-but-curious security model; therefore, the stored data must be encrypted to provide privacy for the data owner. Therefore, user authorization and access control are issues. The data owner must be aware of all possible authorized users in the system in advance and acquire their symmetric encryption key or public key, according to the research literature currently available [4]. Multiple users of big data applications will find this extremely difficult. Through more adaptable attribute management, attribute-based encryption (ABE) implements access control. The attribute set is used as a public key for data encryption, and users who fit that attribute set can decrypt the data [5]. Because of this, the issue of user authorization can be resolved without the need for data owners to know the identities of potential authorized users and associated key sets in advance.

This paper suggests a privacy protection tracking scheme for the industrial or health sectors. To illustrate the usage of our suggested scheme, we will apply it to COVID-19 patient tracing as an example to illustrate the execution of our proposed scheme. Our suggested approach will be based on the following idea, "The user controls access to his identity and location information stored in a cloud server." The following is a list of our main contributions:

- We preserve the user's identity by using an interactive zero-knowledge proof between the data owner and the data user with the cloud server. Authentication between them depends on the zero-knowledge proof method to avoid revealing the user's identity by the user.
- The user's location information is stored in a central database in encrypted form, and the user controls access to the data using the ABE encryption scheme.
- The cloud service provider performs a distance comparison between the data owner and data user without decrypting the information using a homomorphic encryption scheme.

The remainder of the paper is structured as follows. The related work is fully addressed in Section 2. Our models of the system are explained in Section 3. Some preliminaries are mentioned in section 4. Section 5 details the complete structure of our scheme. Section 6 looks at the security analysis of our proposed scheme. Section 7 demonstrates a performance comparison with the candidate scheme. This paper is finally concluded in Section 8.

2. RELATED WORK

Currently, privacy research is being applied in significant application situations like healthcare [6], traffic monitoring [7], and contact tracing [8]. We show in this section the relevant works in depth, there are some privacy problems with the Singapore TraceTogether app [9], and the app does not fully address users' privacy and security requirements. To protect the contact's identity, the Australian government created an app called COVIDSafe [10]. This app encrypts the contact's identity information and keeps it in their mobile phone. In Liu et al.'s scheme [11], users can store their data on their devices, similar to the one offered in the Australian contact trace scheme. Additionally, patients who are diagnosed as positive are required to decide whether or not to give their tendency information to authorities. Several of these privacy implications have been analyzed and discussed by Cho et al. [12]. The discussion includes strategies for improving privacy without reducing its utility for public health. It is obvious that this method does not provide high levels of privacy, the user has to apply for authorization daily, and the device needs to do a considerable amount of calculations, which is not very resource-friendly for low-end mobile devices.

By establishing a communications channel between general data users and positive patients, data privacy will be better protected, as only authorized users will be able to access private patient data. To accomplish this, access control needs to be fine-grained. ABE approaches are gradually emerging with more flexibility in data access control since they were generalized from Identity-Based Encryption (IBE) [13].In [14], offered an encryption scheme based on the user's roles, which is the basis for keeping confidential and sensitive data in cloud settings. In [15] suggested a privacy-aware s-health access control scheme in which a part of the access policy is concealed and the access policy attribute values are hidden in encrypted s-health records (SHR).

Additionally, in real-world cloud settings, the cloud server makes computations in response to requests for location-related data. The homomorphic encryption technique [16, 17] is commonly used in light of its simplicity and performance. The Cloud server can do calculations on encrypted data using homomorphic encryption without using the decryption key. The authors [18] suggested a delta compression-based technique to compress the geolocation data and maintain users' location privacy and confidentiality.

An enhanced security framework to protect the data of virus - infected positive patients in the cloud and block-chain architecture is proposed in [19] to reduce the computational cost of resource-limited devices by outsourcing encryption and decryption support, and verifying the accuracy of returned results in a semi-honest cloud server model. This ensures fairness between users and cloud servers and strengthens privacy preservation.

3. MODELS

3.1. System Model

This section first describes the parties required for the system and their roles, followed by a comprehensive explanation of the scheme's procedure. As shown in Fig. 1.

- Data Owner (A): A data owner should begin by authenticating himself for KGC using a zero-knowledge technique to avoid disclosure of his identity and mitigate a tracking attack. After that, he stores and shares his location information and identity in LSP. Before doing that, he gets the public key from KGC and assigns the access trees. Then, he uses the public key and access trees to encrypt his identity and location data.
- Data User (B): The data user sends an authentication request to the Key Generation Center (KGC), which verifies the user's identity before sending the associated key over a secure channel. The data owner then queries the LSP for locations using the stored ciphertext data. However, an authorized data user can obtain plain query data.
- Location Service Provider (LSP): Its principal role is the storage and processing of ciphertext.
- Key Generation Center (KGC): KGC has powerful computational abilities. At the setup phase, the KGC computes the system's public parameters and the system master key. The master key is used to generate the private key for all the parties in the system, and the public parameters are used to process system-wide operations.

Our system can be used for tracking assets in the supply chain industry and for tracing patients in the health sector. To illustrate the usage of our suggested approach, we will apply it to COVID-19 patient tracing.

Positive patients (data owners) use their mobile device's GPS module to locate location data from the previous 14 days. When a user is diagnosed as a confirmed or suspected COVID-19 patient, his encrypted identity and location information are uploaded to the server so that other users can check if there is contact with him. The data user wants to know whether he has been in contact with the confirmed user after the authority releases the news of a confirmed case, so he sends a registration request to KGC. After that, he asked the LSP server for information about the stored encrypted data. Nevertheless, only authorized contacts can get the results of location queries.

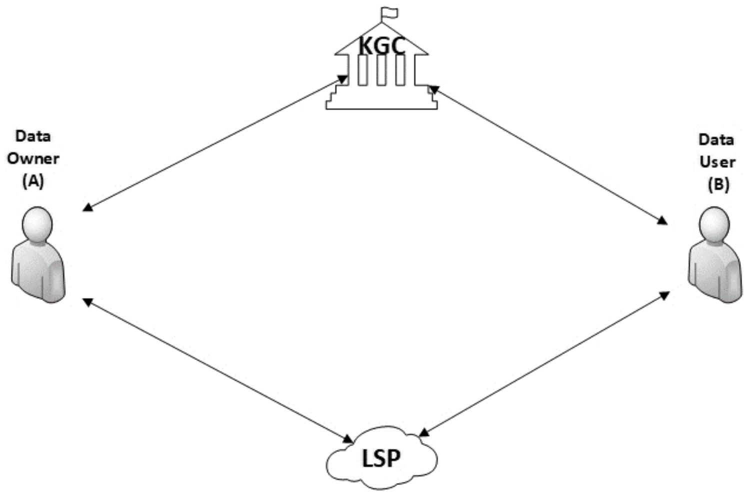

Figure1. System Model

3.2. Threat Model

Each entity's role in the threat model is defined as follows:

KGC is a fully trusted entity, and its communication with other entities is secure.

- The data owner (A) is regarded as a trustworthy entity. Data owners should perform operations following the protocol and securely protect their private keys, which means that they should not actively or passively disclose their keys to any entity.
- The data user (B) and the location service provider (LSP) are following the honest-but-curious model. Specifically, both act honestly and follow the protocol specification correctly. LSP's curious, therefore, to infer and analyze the stored data and query requests to gain illegal profits by harvesting additional information from the data.
- Data users want confidential information that is outside their authority. They may also collaborate with the location service provider.

4. PRELIMINARIES

In this section, we briefly discuss some of the preliminaries we used for our study, including the bilinear map, the ABE, the access tree, and the Scyther security verification tool.

4.1. Bilinear Pairing

Let $\mathbb{G}$be an additive cyclic group of prime order p and a generator$Q$, and $\mathbb{G}_T$ target multiplicative cyclic group of order$p$ and a generator of $e(Q,Q)$. Where e is a bilinear symmetric pairing map such that:

$$e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$$

The properties of the bilinear map e are as follows.

- Bilinearity: $\forall\, x, y \in \mathbb{Z}_p, e(x.Q, y.Q) = e(Q,Q)^{xy}$.
- Symmetry: $\forall\, u.v \in \mathbb{G}, e(U.V) = e(V,U)$.
- Non-degeneracy: $e(Q,Q) \neq 1$.

Definition (discrete logarithm assumption). The discrete logarithm assumption in a group $\mathbb{G}$ of prime order p with generator Q is defined as follows: for any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, the probability that $Pr[\mathcal{A}(Q, a.Q) = a]$ is negligible, where $Q, a.Q \in \mathbb{G}$, and$a \in \mathbb{Z}_p$.

This assumption is valid, as it is widely agreed that discrete logarithm problems are as hard as described in the above definition within a large number field. Thus a cannot be detected from$a.Q$, even if we have$Q$.

4.2. Attribute-Based Encryption

The attribute-based encryption (ABE) [20] extends the identity-based encryption scheme. The idea of ABE is to use descriptive attributes for users to gain authorized access to encrypted data. Therefore, the objective is to describe who should decrypt the data regarding an access policy over attributes. There are two types of ABE, the key policy (KP-ABE) and the ciphertext policy (CP-ABE). Within KP-ABE, secret keys for users are created based on an access tree that defines the user's scope of privileges and data encrypted over a collection of attributes. CP-ABE, however, uses access trees to encrypt data and secret keys created by users over a set of attributes.

4.3. Access tree

In ABE, the encryption policy is represented with an access treeT_P. Every non-leaf tree node is a threshold gate, and an attribute defines each leaf node. Figs. 2 and 3 display the access trees used to illustrate the operation of our suggested scheme in the healthcare sector.

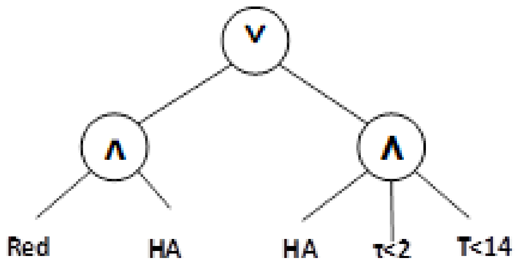

Figure2. Access tree T_{P1} for user identity

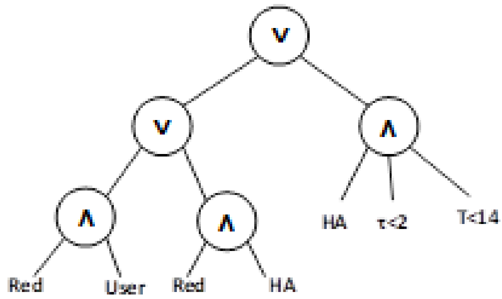

Figure3. Access tree T_{P2} for user location

This paper uses a monotonic Boolean formula to describe the access policy. A user's identity and location information are set to be visible to certain classes of data users. For example, in Figs. 2 and 3, the user's identity and location are only available to the health authority (HA) as an authorized data user (B) in two situations: if the data owner (A) has been diagnosed with COVID-19 (red) or is within two meters of an infected contact ($\tau < 2$) within the last 14 days ($T < 14$).

4.4. Scyther security Verification claims

To exam and verify security protocols, an automated application called Scyther [21] is used. Each protocol entity in Scyther is referred to as a role, and claims are used to confirm the expected security objectives of each role. The definition of a claim is the claim (R, G, and P), which states that a role R expects the parameter P to meet the security objective G. The tool reports that a given claim is false if an attack is known to exist; otherwise, it returns that the claim is OK. The suggested procedure is validated using the claims listed below.

(i) Secrecy claim: According to the definition of a secrecy claim (R, Secret, P), in this case, role R anticipates that P is a secret and that an adversary cannot read or forge it.
(ii) Authentication claim: An authentication claim is one that (R, Nisynch) this claim verifies that the intended users are corresponding with one another in the protocol's prescribed sequence.
There are other claims of authentication. First, the "Alive" claim is used for verifying that the parties detect each other. Second, the "weakness" claim that used to verify that the protocol is immune from impersonation attacks; and finally, the "agreement" claim, which used to assure that all protocol participants follow the protocol's order.

5. Our Proposed Scheme

In this section, we will present the scheme's workflow in detail.

5.1. System Initialization

(1) $Setup(1^N) \rightarrow SS, PP$: The algorithm uses the security parameter 1^N as an input. As shown in the following steps, the Key Generation Center (KGC) executes the algorithm to produce the System Secret (SS) as well as the system Public Parameter (PP) as the output.

- The KGC selects and issues a generator Q for an additive cyclic group $\mathbb{G}$ of prime order$p$.
- The KGC defines a bilinear mapping: $e: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$,where$e(Q, Q)$represents the value of the bilinear mapping in the multiplicative cyclic group$\mathbb{G}_T$.
- Besides that, the KGC selects and issues a hash function $\mathbb{H}(\cdot): \{0,1\}^* \rightarrow \mathbb{G}$.
- Then, selects$\alpha, \beta \in \mathbb{Z}_p^*$randomly and secretly, and sets α, β as the System Secret.

Finally, The KGC evaluates a public witness for the system$W_S = e(Q, Q)^\alpha$.
The output of the algorithm is $SS = \{\alpha , \beta\}$, and$PP = \{\mathbb{H}, Q, p, e, W_S, \beta. Q, e(Q, Q)\}$.

5.2. Identification

(1) $Identification(PP) \rightarrow Acceptor\ \perp$: This algorithm will be an interactive Zero-Knowledge proof between either the data owner (A) or a data user (B) as the Prover for his unique identity$ID_i$and the Location Service Provider (LSP) or KGC as the Verifier. This algorithm is

three passes, repeated iterations until LSP or KGC accepts the verification of the user's identity (A or B). Otherwise, it will output ⊥.

- Firstly, Prover had to communicate securely with KGC to get his secret key K_i corresponding to his unique identityID_i,$K_i = \beta.\mathbb{H}(ID_i)$, where $i \in \{A, B\}$.
- Then, Prover selects secretly, and randomly$r_i \in \mathbb{Z}_p^*$and computes the commitment message
- $W_i = e(Q,Q)^{r_i}$.
- After that, Prover sendsW_i, and his redundant identity $\mathbb{H}(ID_i)$ to Verifier.
- Verifier selects secretly and randomly the challengec_i, such that$c_i \in \mathbb{Z}_p^*$, and sends it toProver.
- Prover computes his response $R_i = r_i.\mathrm{Q} + c_i.K_i$ and sends it to Verifier.
- Verifier computes $e(Q, R_i)$, and accepts the identity of Prover if it equals to $W_i \cdot V_i$, Where,

$$V_i = e\big(\beta.Q, \mathbb{H}(ID_i)\big)^{c_i}$$

5.3. Encryption

(1) $Encrypt(PP, ID_i, X_i, T_{P1}, T_{P2}, K_S) \rightarrow \{CT_1, CT_2, CT_3\}$: Before uploading to the LSP server (LSP), the data owner (A) executes this algorithm to encrypt his geolocation information, whichis identified in [22],$X_i = \{x_{i1}, x_{i2}, x_{i3}\}$, and his identity$ID_A$. Additionally, A assigns two access trees T_{P1} and T_{P2} (Figs. **2** and **3**) to control the access to his encrypted identity and location information, respectively. The data owner (A) must encrypt his identity and location so that any data user, either a contact or health authority (HA), may enjoy using his location if and only if their attributes match those that the data owner who has allocated access to. There are three sections in the ciphertextCT_1,CT_2 andCT_3. The encrypting procedure consists of three basic steps.

First, encrypting the data owner's identityID_A. Then, A sets the access formula F_1 to control the access of his identity according to the access treeT_{P1}.

$$F_1 = \big\{\{Red \wedge HA\} \vee \{HA \wedge \tau \leq 2 \wedge T \leq 14\}\big\} = \big\{\{A_{11} \wedge A_{12}\} \vee \{A_{12} \wedge A_{13} \wedge A_{14}\}\big\}$$

A sets the plaintext$m_1 = ID_A$ and selects randomly, and security$t \in \mathbb{Z}_p$. Then, A uses the linear splitting for t according toF_1. The split share for each attributeA_{1j} in F_1isγ_j. The ciphertext:
$CT_1 = m_1.e(Q,Q)^{\alpha t}, t.Q\ , \gamma_j.\mathrm{Q}\ ,\ \gamma_j.h_{1j}$, where$h_{1j} = \mathbb{H}(A_{1j})$. $j \in \{1,2,3,4\}$

Secondly, A employs a homomorphic symmetric encryption approach (depending on the factorization issue) [23] to achieve location privacy, allowing the LSP server (LSP) to compare the distance between A and B (which is assigned asτ), and detect it without acquiring any information about the data owner's location.

A selects two prime numbersq_0&q_1, then computes$n = q_0 q_1$, and$\phi(n) = (q_0 - 1)(q_1 - 1)$.
Then randomly picksx, m, such that $0 < x, m < n$, and let $k_s = \langle x, m\rangle$ the symmetric key. Then selects g_0 co-prime of n.

A computes a blind factor$\left(m.g_0^{x mod \phi(n)}\right)^{-1}$.

Then encrypts his location information $X_A = \{x_{A1}, x_{A2}, x_{A3}\}$, such that

$$x_{ej} = E(x_{Aj}) = \frac{x_{Aj}}{mg_0^{x mod \phi(n)}} modn, wherej = \{1,2,3\}$$

Finally, forx_{ej}, A uses the access tree $T_{P2}$to access his encrypted location information, and randomlyselects$s \in \mathbb{Z}_p$. Then, he sets$CT_2 = \{\langle x_{e1}, x_{e2}, x_{e3}\rangle, \langle x + W_s{}^s, m \cdot W_s{}^s\rangle\}$, where$W_s{}^s = e(Q,Q)^{\alpha s}$: the ciphertext$CT_3 = $ sQ, sh_{2j}, where$h_{2j} = \mathbb{H}(A_{2j})$. A_{2j}represents the attributes in the access tree T_{P2}.

At the end, A storesCT_1, CT_2, and CT_3in the LSP server (LSP) to share them with any data user B or HA.

5.4. Inquiry

(1)$KeyGenerate(MK, PP, A_{2j}, T_{P2}) \rightarrow \{SK_2\}$: KGC uses this algorithm to assist the system's data users (contact or HA) in obtaining secret keys based on their attributes.

First, for decryptingCT_3. The KGC uses the access formula F_2 to control the access of user location according to access treeT_{P2}, which the data owner (A) sets.

$$F_2 = \{\{Red \wedge USER\} \vee \{Red \wedge HA\} \vee \{HA \wedge \tau \leq 2 \wedge t \leq 14\}\}$$
$$= \{\{A_{21} \wedge A_{22}\} \vee \{A_{21} \wedge A_{23}\} \vee \{A_{23} \wedge A_{24} \wedge A_{25}\}\}$$

Then, for any attribute $j \in A_{2j}, j = \{1,2,3,4,5\}$in access tree T_{P2}. The KGC uses the linear splitting for α according to F_2. The split share for each attribute in $F_2$is λ_j. Finally, KGC selects randomly and security $r_{2j} \in \mathbb{Z}_p$, and sets$SK_2 = \{\lambda_j Q + r_{2j} h_{2j}, r_{2j} Q\}$ to be used to decrypt CT_3.

(2) $Match(PP, SK_2, A_{2j}, CT_2) \rightarrow W_s{}^s or \perp$: By executing this algorithm, beneficiaries, only authorized data user (contact or HA), can obtain the secret parameter $W_s{}^s$. Otherwise, it will output $\perp$.

Firstly, any authorized data user (contact or HA), whose attribute A_{2j} matches the access formula F_2, can assemble the secret-sharing $\alpha$from λ_j. Then, he can compute $W_s{}^s$as follow:

$$\frac{e(\lambda_j . Q + r_{2j} . h_{2j}, s . Q)}{e(r_{2j} . Q, s . h_{2j})} = W_s{}^s$$

(3) $Operate(CT_2, W_S^S, Y) \rightarrow answer$: In this algorithm, firstly, the authorized data user (contact or HA) uses $W_s{}^s$to encrypt his location $Y_B = \{y_{B1}, y_{B2}, y_{B3}\}$as y_{ej}, after that, the LSP server operates over x_{ej} and $y_{ej}$to exam if the distance between these two locations (τ) is less than 2 meters or not. Finally, the result of this operation is encrypted by $W_s{}^s$.

Here, the location of the infected user $A$and the contact user $B$are X_i and Y_j respectively. Additionally, the infected user's location data X_i have been encrypted as$CT_2$and CT_3. In addition,CT_2 and CT_3 stored in the LSP server. The following steps can calculate the distance between the infected user and the contact:

- Contact user (B) encrypts his location as follows $y_{ej} = y_j W_S^s g_0^{W_S^s}, j = \{1,2,3\}$, then sends the result to (LSP).

- (LSP) gets $m' = mW_S^s$, and $x' = x + W_S^s$from CT_2.
- Then, (LSP) computes $K_S' = m' g_0^{x'}$.
- (LSP) computes $dis_e = \sqrt{\sum_{i=1}^{3}\left(y_{ej} - x_{ej} \cdot K_S'\right)^2}$, and sends it to the contact user (B).
- Contact user (B) detects his distance from an infected user by computing $dis = \frac{dis_e}{W_S^s g_0^{W_S^s}}$

If dis is less than 2, he has to notify HA. If $\tau < 2meter$ the HA can decrypt the contact's location and $identity$.

5.5. Decryption

(1) $KeyGenerate(MK, PP, A_{1j}, T_{P1}) \rightarrow \{SK_1\}$: After LSP determined the separation between A and B and under specific circumstances ($\tau < 2meter$), B needs local confirmation that he has decryption privileges. Only if the verification is successful can the location information and identity of A be decrypted; otherwise, because B's attribute does not meet the prerequisite for access, the ciphertext document is unavailable because he needs decryption rights.

KGC uses this algorithm to assist the system's data user (contact or HA) in obtaining secret keys based on their attributes. For any attribute$j \in A_{1j}$ in access tree T_{P1}, KGC selects randomly, and security $d_0 \in \mathbb{Z}_p$, and $r_{1j} \in \mathbb{Z}_p$. Then KGC sets $SK_1 = \{\alpha Q + d_0 Q, r_{1j} h_{1j} + d_0 Q, r_{1j} Q\}$ to be used to decrypt CT_1.

6. Security Analysis

6.1. CPA

According to a data owner's health status in our system, he can authorize the health authority or other data user to take advantage of his identity or location information. Therefore, the requests may include attackers. Furthermore, a specific pair of plaintext/ciphertext is easy for the attacker to obtain. Therefore, our scheme must be protected against the indistinguishability of ciphertext under the chosen-plaintext attack (IND-CPA).

As indicated in the previous section, the ciphertext in our proposed system consists of three partsCT_1, CT_2, and CT_3. [17] proves that the homomorphic encryption scheme used to obtain CT_2 is secure against CPA. We will confirm thatCT_1 and CT_3 are secure against CPA next.

Theorem

Assume an adversary gets $m_1$and its corresponding ciphertext CT_1. From section 5, $CT_1 = m_1.e(Q,Q)^{\alpha t}, t.Q, \gamma_j.Q, \gamma_j.h_{1j}$, we deduce that:

$$e(Q,Q)^{\alpha t} = \frac{CT_1}{m_1}$$

Assume that $F = e(Q,Q)^{\alpha t}$. It is easy to compute F. However, it is difficult to get proper $\langle \alpha, t \rangle$from$F$. Even if the attacker knows $\langle Q, t.Q, e(Q,Q)^{\alpha t} \rangle$, he can not deduce α or t due to the discrete logarithm problem. Consequently, the attacker cannot deduce the blind factor $e(Q,Q)^{\alpha t}$from CT_1. As described in section 5, blind factor $e(Q,Q)^{\alpha t}$ is the key to decrypt the user's identity. Thus, the attacker cannot decrypt extra confidential information from the already known m_1. In conclusion, our system is secure against CPA.■

CT_3 can be proved safe against CPA in the same manner as CT_1.

6.2. Formal Security Verification for the Proposed Protocol

Our proposed protocol was tested using the widely accepted tool for automatically verifying security protocols, Scyther [21]. It is capable of providing effective analysis results for complex attack scenarios. As a result, we use Scyther in order to conduct further security correctness checks on the proposed protocol. In Scyther, two roles are defined, namely, data owner A and location service provider S. Following that, each role is implemented along with its corresponding behavior. Data owner identity (IDA), system security parameters, and location data are four security parameters in the proposed protocol, which should not be disclosed. We, therefore, make secret claims about them. As part of Scyther's search pruning, we selected the option "Find all attacks" as well as the matching type "Find all types of flaws". The results of the suggested protocol's formal security verification are shown in Figs. 4 and 5, proving that it is secure.

Figure 4. Scyther results for the Identification Phase

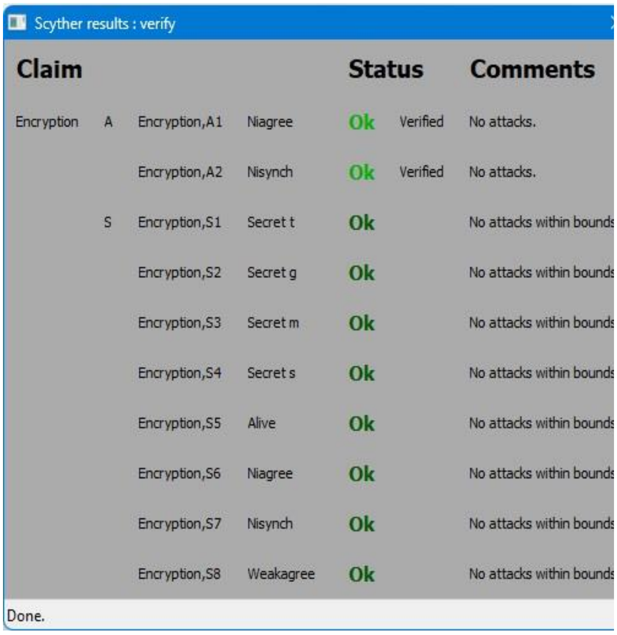

Figure 5. Scyther results for the Encryption Phase

7. Performance Analysis

In this section, we compare our suggested protocols' performance to some of the currently proposed contact tracing protocols [9, 19] described in Section 2.

7.1. Functional Analysis

We compare each of the five aspects, which are the use of an identification scheme, immunity from tracking attacks, positioning technology, power usage, and security of the technology, with those in [9], and [19], as shown in Table 1.

Table 1. Comparison of the proposed scheme with the existing contact tracking system.

Scheme	Identification	Tracking Attack	Positioning Technology	Power Usage	Security of the Technology
[9]	NO	Yes	Bluetooth	High	Low
[19]	NO	Yes	GPS, Cellular, Wifi	Medium	High
Proposed Scheme	Yes	NO	GPS	Low	High

Comparatively, we can see that the suggested scheme in [9] uses Bluetooth technology for exposure notification and does not gather any personal information about the user aside from the position and time. The power consumption is extremely high in this scheme because Bluetooth technology must be used continuously to determine whether a regular user is in touch with a positive patient. As verified in [24], attackers can use Bluetooth vulnerability in tracking the user. In addition, since all activities are carried out on the mobile terminal, its computing costs are very high, and some of their privacy is not protected. Additionally, the security level is poor. In [19], they gather location data using GPS, cellular, and WiFi technologies. They also use the outsourcing service technique to transfer some tasks from the mobile terminal to the outsourcing server, which relieves the mobile terminal's resource pressure. In that scheme, there is no identification protocol and no method used to mitigate the tracking attack from unauthorized users.

In our suggested scheme, we use GPS technology to gather the location data and use the ZKP technique to hide the identity of the data owner or data user to achieve identification without exposure to a tracking attack. We only access the scheme in this paper to ask the location service provider if we have contacted a positive patient after getting a positive patient confirmation notice. Except for the individual who has interaction with the positive patient, no one knows where the positive patient is. This plan safeguards both general users' and positive patients' privacy. Our tracing method only conducts background location knowledge uploads when charging, protecting not only the user's location privacy but also power and battery consumption. The tracing schemes necessitate that the Bluetooth feature be switched on at all times, which increases battery usage.

7.2. Computational Analysis

In addition to functionality, contact tracing and tracking protocols should consider computational efficiency. Because user devices typically have restricted resources while backend servers have abundant resources, we focus on computation on a user device. Table 2 displays the comparison between our suggested protocol and that in [19]. Table 3 summarizes the necessary operations used by the user side.

From Table 2, we can find that our scheme is more efficient than that in [19] for the decryption phase. As our scheme conducts no local differential privacy operation, in the encryption phase, it performs identity encryption and location data encryption. So, if we compare both protocols from the user's perspective, our scheme is more efficient than [19] for computation over the head of a location data encryption operation.

Table 2. Computation performance comparison.

Protocols	Identification	Encryption	Decryption
[19]	---	$T_L + T_E + T_{ME} + T_{MM} + (m+1)T_H + (2m+1)T_{EM}$	$2T_P + 2T_{MM} + T_D + T_L$
Our	$T_{ME} + T_H + T_{EA} + 2T_{EM}$	$2T_{ME} + T_{MM} + T_E + 2mT_H + (3m+2)T_{EM}$	$2T_P + 2T_{MM} + T_D$

Table 3. Notation.

Notation	Description
T_P	Bilinear Pairing
T_H	Hash Function
T_{ME}	Modular Exponential
T_{EM}	Elliptic Curve point Multiplication
T_{EA}	Elliptic Curve point Addition
T_{MM}	Modular Multiplication
T_E	Encryption
T_D	Decryption
T_L	Local Differential Privacy
m	The number of user attributes

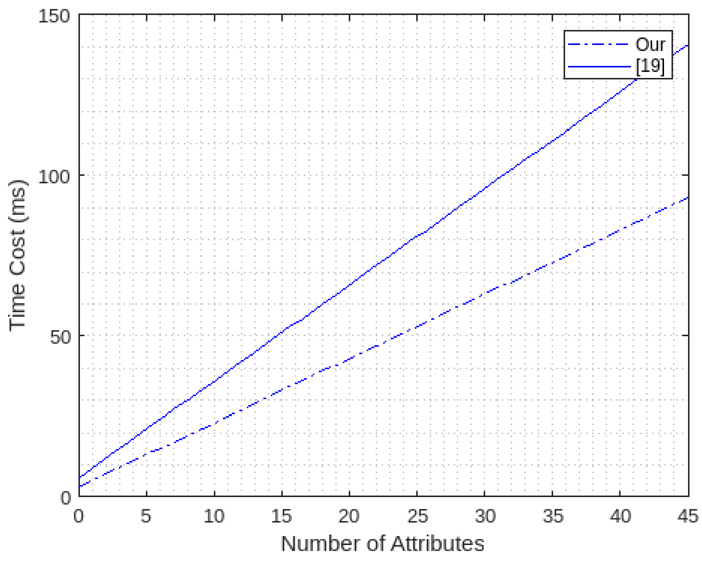

Figure 6. Location data Encryption Time of DO

Fig. 6 focuses on the complexity of encryption on the DO side. As Fig. 6 shows, the time for encryption of DO increases linearly with the number of user attributes. Under the same conditions, the scheme in [19] takes the longest because it requires more EC multiplication point operations to be executed as the number of user attributes increases than our scheme. As a result, our scheme reduces the cost of computing on DO and DU significantly, increases the efficiency of encryption and decryption, and is appropriate for resource-constrained devices.

8. CONCLUSIONS

This paper develops a novel privacy-preserving tracing and tracking scheme based on the zero-knowledge identification scheme, the homomorphic encryption algorithm, and the ABE algorithm. This scheme is for smart devices that sense location information. In comparison with existing methods, the proposed protocol conceals the identity and location of the data owner. This prevents unauthorized individuals from monitoring him. Our proposed scheme can support location service providers in performing efficient and privacy-preserving queries about location distance compared to encrypted data stored on their servers. Furthermore, by combining CP-ABE and KP-ABE, our scheme provides fine-grained control over sensitive location information and data owner identities, allowing only authorized queryers, whose attributes satisfy the access tree, to decrypt encrypted query results provided by the location service provider. As a result, both the data owner's and the data user's location information is kept private from the location service provider and unauthorized users. Additionally, the security analysis proves that it is secure against a chosen plaintext attack. Besides that, the analysis results from the Scyther tool show that our protocols can ensure the desired security features efficiently. A performance analysis indicates that the scheme has superior advantages over other approaches.

For privacy preservation, the scheme is used in contact tracing, assisting healthcare authorities in identifying individuals exposed to cases, limiting the source of transmission, and preventing the spread of diseases while protecting the privacy of those who have been contacted. Furthermore, the suggested scheme can be used to track and trace assets in the supply chain while minimizing the impact on employee privacy.

REFERENCES

[1] C. Jandl, M. Wagner, T. Moser, and S. Schlund, "Reasons and strategies for privacy features in tracking and tracing systems—a systematic literature review," Sensors, vol. 21, no. 13, p. 4501, 2021, doi: 10.3390/s21134501.

[2] V. Ang and L. K. Shar, "COVID-19 One Year on – Security and Privacy Review of Contact Tracing Mobile Apps," in IEEE Pervasive Computing, vol. 20, no. 4, pp. 61-70, 1 Oct.-Dec. 2021, doi: 10.1109/MPRV.2021.3115478.

[3] W. Zheng, C. -F. Lai, D. He, N. Kumar and B. Chen, "Secure Storage Auditing With Efficient Key Updates for Cognitive Industrial IoT Environment," in IEEE Transactions on Industrial Informatics, vol. 17, no. 6, pp. 4238-4247, June 2021, doi: 10.1109/TII.2020.2991204.

[4] Y. Liu, J. Yu, M. Yang, W. Hou, and H. Wang, "Towards Fully Verifiable Forward Secure Privacy Preserving Keyword Search for IOT Outsourced Data," Future Generation Computer Systems, vol. 128, pp. 178–191, 2022, doi: 10.1016/j.future.2021.10.009.

[5] H. Song, X. Han, J. Lv, T. Luo, and J. Li, "MPLDS: An integration of CP-abe and local differential privacy for achieving multiple privacy levels data sharing," Peer-to-Peer Networking and Applications, vol. 15, no. 1, pp. 369–385, 2021, doi: 10.1007/s12083-021-01238-8.

[6] Z. Zhao, X. Li, B. Luan, W. Jiang, W. Gao, and S. Neelakandan, "Secure internet of things (IOT) using a novel Brooks Iyengar Quantum Byzantine agreement-centeredblockchain networking (BIQBA-BCN) model in Smart Healthcare," Information Sciences, vol. 629, pp. 440–455, 2023, doi: 10.1016/j.ins.2023.01.020.

[7] M. Li, L. Zhu, Z. Zhang, C. Lal, M. Conti and F. Martinelli, "Privacy for 5G-Supported Vehicular Networks," in IEEE Open Journal of the Communications Society, vol. 2, pp. 1935-1956, 2021, doi: 10.1109/OJCOMS.2021.3103445.

[8] D. Andreoletti, O. Ayoub, S. Giordano, G. Verticale and M. Tornatore, "Network-Based Contact Tracing for Detection of Covid-19 Contagions: A Privacy-Preserving Approach," in IEEE Communications Magazine, vol. 59, no. 9, pp. 42-48, September 2021, doi: 10.1109/MCOM.001.2100015.

[9] H. Stevens and M. B. Haines, “Tracetogether: Pandemic response, democracy, and Technology,” East Asian Science, Technology and Society: An International Journal, vol. 14, no. 3, pp. 523–532, 2020, doi: 10.1215/18752160-8698301.

[10] A. Fares and M. Alanezi, "Contagious Patient Tracking Application Spotlight: Privacy and Security Rights," 2021 7th International Conference on Contemporary Information Technology and Mathematics (ICCITM), Mosul, Iraq, 2021, pp. 13-18, doi: 10.1109/ICCITM53167.2021.9677681.

[11] J. K. Liu, M. H. Au, T. H. Yuen, C. Zuo, J. Wang, A. Sakzad, X. Luo, L. Li, and K.-K. R. Choo, “Privacy-preserving contact tracing protocol for mobile devices: A Zero-knowledge proof approach,” Information Security Practice and Experience, pp. 327–344, 2021, doi: 10.1007/978-3-030-93206-0_20.

[12] H. Cho, D. Ippolito, and Y. W. Yu, "Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs", arXiv [cs.CR]. 2020, doi: 10.48550/arXiv.2003.11511.

[13] J. Li, Q. Yu, Y. Zhang, and J. Shen, “Key-policy attribute-based encryption against continual auxiliary input leakage,” Information Sciences, vol. 470, pp. 175–188, 2019, doi: 10.1016/j.ins.2018.07.077.

[14] A. Sathya and S. K. Raja, “Privacy preservation-based access control intelligence for cloud data storage in Smart Healthcare Infrastructure,” Wireless Personal Communications, vol. 118, no. 4, pp. 3595–3614, 2021, doi: 10.1007/s11277-021-08278-6.

[15] Y. Zhang, D. Zheng and R. H. Deng, "Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control," in IEEE Internet of Things Journal, vol. 5, no. 3, pp. 2130-2145, June 2018, doi: 10.1109/JIOT.2018.2825289.

[16] R. Rivest, L. Adleman, and M. Dertouzos,” On data banks and privacy homomorphisms”, Foundations of Secure Computation, vol. 4, no. 11, pp. 169–180,1976, [Online]. Available: https://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/RAD78.pdf. [Accessed: 18- March-2023].

[17] Q. Xie and L. Wang, “Privacy-Preserving Location-Based Service Scheme for Mobile Sensing Data”, Sensors, vol. 16, no. 12, p. 1993, 2016, doi: 10.3390/s16121993.

[18] V. Patil, S. Parikh, P. Singh, and P. Atrey, “ GeoSecure: Towards secure outsourcing of GPS data over cloud”, In IEEE conference on communications and network security (CNS), pp. 495-501, 2017, doi: 10.1109/CNS.2017.8228699.

[19] C. Qin, L. Wu, W. Meng, Z. Xu, S. Li, and H. Wang, “A privacy-preserving blockchain-based tracing model for virus-infected people in cloud,” Expert Systems with Applications, vol. 211, p. 118545, 2023, doi: 10.1016/j.eswa.2022.118545.

[20] Y. Song, H. Wang, X. Wei and L. Wu, “Efficient Attribute-Based Encryption with Privacy-Preserving Key Generation and Its Application in Industrial Cloud”, Security and Communication Networks, vol. 2019, pp. 1-9, 2019, doi: 10.1155/2019/3249726.

[21] T. Rabas, R. Lórencz, and J. Buček, “Verification of PUF-based IOT protocols with Avispa and Scyther,” Proceedings of the 19th International Conference on Security and Cryptography, 2022, doi: 10.5220/0011299000003283.

[22] X. Li, and T. Jung, “Search me if you can: Privacy-preserving location query service”, In Proceedings of the IEEE INFOCOM, Turin, Italy, pp. 2760–2768, 14–19 April 2013, doi: 2010.1109/INFCOM.2013.6567085.

[23] Q. Xie, and L. Wang, “Efficient privacy-preserving processing scheme for location-based queries in mobile cloud”, In Proceedings of the IEEE International Conference on Data Science in Cyberspace, Changsha, China, 13–16 June 2016, doi: 2010.1109/DSC.2016.70.

[24] A. M. Allam, “Improving the Privacy-Preserving of Covid-19 Bluetooth-Based Contact Tracing Applications Against Tracking Attacks,” International Journal of Computer Science and Information Technology, vol. 13, no. 5, pp. 49–57, Oct. 2021, doi: 10.5121/ijcsit.2021.13504.

AUTHORS

Ali M. Allam received his Ph.D. degree in Communication Engineering from Helwan University in 2008. From 2016 to current works as an associated professor in the communication department at Helwan University. His research interests include wireless communication, network security, and cryptography.