Efficient Implementation of a Proxy-based Protocol for Data Sharing on the Cloud

Maryam Sepehri
Dipartimento di Informatica
Università degli Studi di Milano
Crema, Italy
maryam.sepehri@unimi.it

Stelvio Cimato
Dipartimento di Informatica
Università degli Studi di Milano
Crema, Italy
stelvio.cimato@unimi.it

Ernesto Damiani
EBTIC
Khalifa University of Science
Abu Dhabi, UAE
ernesto.damiani@kustar.ac.ae

ABSTRACT

In this paper, we provide a secure and efficient outsourcing scheme for multi-owner data sharing on the cloud. More in detail we consider the scenario where multiple data owners outsource their data to an untrusted cloud provider, and allow authorized users to query the resulting database, composed of the encrypted data contributed by the different owners. The scheme relies on a proxy re-encryption technique that is implemented using an El-Gamal Elliptic Curve (ECC) crypto-system.

We experimentally assess the efficiency of the implementation in terms of computation time, including the key translation process, data encryption and re-encryption modules, and show that it improves over previous proposals.

CCS Concepts

•**Security and privacy** → **Database and storage security;** *Privacy-preserving protocols;*

1. INTRODUCTION

The cloud computing paradigm offers a number of advantages in terms of economic savings and data availability when users want to share and outsource their data. Many commercial providers offer file (e.g. Dropbox, Google drive) or picture (e.g. Flickr) sharing services that users can conveniently access when they want to upload their data to the cloud and share them with other persons with different purposes.

When sensitive data are outsourced to the cloud, however, confidentiality and privacy concerns arise [17] and the need for mechanisms ensuring different security properties of the cloud infrastructure has recently emerged and has been deeply investigated [2, 6]. To prevent unauthorized access, a straightforward approach consists in encrypting the data before they are stored. Recently, a number of solutions have been presented to cope with the problem of querying encrypted databases, where the queries are processed directly on the encrypted records without the need of any decryption phase and considering different levels of granularity [3, 4, 15].

In this paper we consider the scenario where different data owners hold different portions of an horizontally partitioned database. Their goal is to allow authorized users to execute queries on the union of the databases they own, still maintaining the confidentiality of the data individually stored, and avoiding also that the other parties and the cloud provider, executing the query, access the data. Some solutions rely on order preserving encryption schemes [1], on the application of secret sharing schemes [14, 11, 13] or on the adoption of multi-party protocols to have scalable and efficient techniques to support queries on encrypted data [24, 25].

Another possible set of solutions is based on the usage of proxy re-encryption (PRE) schemes, where a semi-trusted proxy holding a re-encryption key translates a message encrypted under a public key into an encryption of the same message under a different public key. In this setting, firstly introduced by Blaze et al. in 1988 [4], the proxy is not able to learn anything about the encrypted message. Successively, Dong *et al.* [8] proposed a proxy re-encryption keyword search technique, which enables users to re-encrypt an encrypted message using different keys held by the other participants to the scheme. Their scheme generates a trapdoor for the user keyword that is used by the proxy server to find a match in the encrypted data. However, their trapdoor generation algorithm is relatively slow, due to the involvement of both the user and the proxy and the need for multiple arithmetic operations at both sides. Following Dong's protocol, Sepehri *et al.* [26] addressed the problem of privacy-preserving equality queries over horizontally partitioned data among multiple owners adopting a proxy re-encryption scheme. They experimentally implemented the key translation process, and computed the time needed to bring data encrypted with different keys under the same key, utilizing El-Gamal encryption system [9]. Compared to Dong's scheme [8], their collusion-resistant scheme showed improved efficiency in terms of computation time.

In this paper, we modify the proxy re-encryption scheme previously presented adopting an El-Gamal elliptic curve encryption system, and obtaining, as expected, a further improvement. We compare the performance of the proposed scheme with the results from the scheme adopted in [15], which is based on bilinear pairing [28], and the previous proxy based scheme [26].

SCC'17, April 02 2017, Abu Dhabi, United Arab Emirates
© 2017 ACM. ISBN 978-1-4503-4970-3/17/04...$15.00
DOI: http://dx.doi.org/10.1145/3055259.3055271

The rest of this paper is structured as follows. In the next section, we present related works and in Section 3 we introduce some basic notions for the implementation of the El-Gamal elliptic curve cryptosystem. In Section 4, we give and overview of the system and present the proposed framework in Section 5. Security and performance analysis are presented in Section 6 and 7, respectively. Finally, we draw in Section 8 some conclusions and some directions for future work.

2. RELATED WORK

In this section, we discuss some previous works related to privacy-preserving query processing and based on proxy re-encryption methods. In literature, several PRE schemes have been presented based on different cryptographic systems with special properties and security requirements, considering both the single and multi user scenario.

Mambo and Okamoto introduced a technique for delegating decryption in [22]. The idea was further extended by Blaze [4] who presented a scheme based on the El-Gamal cryptosystem and its formalization. A proxy is given a re-key that allows the translation of a message encrypted under the delegator key into an encryption of the same message under a different key. Their proxy scheme suffers from a number of security drawbacks, for instance, if the proxy colludes with one of the delegatees, they can easily learn the delegator key. Later, Ivan *et al.* [16] proposed three proxy re-encryption schemes based on El-Gamal, RSA, and IBE (ID-based encryption). In their scheme, the private key of the delegator is split into two parts, one distributed to the proxy and the other to the delegatee. Again, when the proxy colludes with the delegatee, they can retrieve the delegator private key. Following [16], in 2005 Ateniese *et al.* [3] presented a new scheme relying on bilinear pairings in which the delegator private key is protected from being disclosed by the collusion of the proxy and any delegatee. In 2008, Dong *et al.* [7], proposed a pairing-free PRE scheme in an effort to avoid the expensive bilinear pairing operations. In an attempt to partially solve the collusion problem left in [3], Libert *et al.* [19] proposed, instead of preventing the collusion of proxy and delegatee, to trace the malicious proxy after a possible collusion with one or more delegatees. Generally, most of these techniques are proposed for a single user scenario in the sense that data are encrypted with single key.

Recently, some improvements have been achieved in the context of multi-user setting for querying encrypted data [15, 26] and searching keywords [8, 27] using PRE schemes. In 2008, Dong *et al.* [8] proposed a proxy based scheme built upon El-Gama cryptosystem that allows all authorized users to share their encrypted data. Their technique uses a trapdoor generation algorithm to search encrypted data for a certain keyword through a semi-honest proxy server. They showed that the proxy encryption/decryption operations are more efficient in El-Gamal than RSA based scheme for the same security level because the exponents used in the El-Gamal scheme are smaller than those used in the RSA. However, their technique is relatively slow for running multiple arithmetic operations at the users and the proxy sides. In addition, it does not support multi-user access policies and it is not collusion safe i.e., if the user colludes with the proxy server, they can access the stored keys at the trusted key management party. In 2015, Hang *et al.* [15] introduced a secure system called $ENKI$ for executing relational operations coming from users with different access rights. In their scheme, data owners split their relations in multiple virtual relations and the proxy is used to rewrite queries and re-encrypt data with different encryption keys in order to enforce access restrictions. A related line of work for searching queries over encrypted data with different keys has been presented by Sepehri *et al.* [26]. They considered a multi-owner scenario where each user plays the role of a data owner who outsources her own encrypted data to a cloud service provider for sharing them with other authorized users. Their El-Gamal based scheme does not relay on any interactive computation algorithm and it is safe w.r.t. collusion, due to the fact that keys are shared between the proxy and the authorized user posing equality test queries. In this paper we follow the approach for data sharing on the cloud presented in [26] and use for the implementation an El-Gamal Elliptic Curve cryptosystem for processing equality test queries. We compare the performance of our ECC scheme at re-encryption step with a secure system ($ENKI$) proposed in [15], which executes relational operations while satisfying users with different access rights. Their scheme adopts a pairing based encryption for a deterministic proxy encryption method that is relatively costly due to the pairing operation.

3. PRELIMINARIES

In this section, we introduce some basic notation and terminology that will be used throughout the paper.

3.1 Elliptic Curve Cryptosystem

Elliptic Curve crypto-systems (EC)[28] have been introduced by Koblitz [18] and Miller [23] as alternative algorithms for implementing public key cryptography. EC received increasing attention in recent years due to the advantage of using shorter keys in comparison with other public key cryptosystem such as RSA. An elliptic curve $\mathbb{E}$ over a finite field F_p is an equation of the form $y^2 \bmod p = x^3+ax+b \mod p$ with the condition $4a^3 + 27b^2 \mod p \neq 0$ where p is a prime number with order n and a, b are defined on F_p. The security of EC depends on the difficulty of the elliptic curve discrete logarithm problem.

Problem 1 (Elliptic Curve Discrete Logarithm Problem (ECDLP)). Let P and Q be two points in $\mathbb{E}(F_p)$ on a elliptic curve $\mathbb{E}$ such that $dP = Q$ for some uniformly chosen random integer value d from the interval $[1, n-1]$. Given P and Q, it is computationally infeasible to obtain d when it is sufficiently large.

3.2 El-Gamal Elliptic Curve Cryptosystem

The El-Gamal Elliptic Curve (ECC) cryptosystem is more efficient than El-Gamal implemented over a multiplicative group [5]: While El-Gamal requires 1024-bit long keys, ECC achieves equal security level using a smaller key size, just 160-bit, resulting in improved speed and efficient use of power, bandwidth, and storage.

In the following, we describe a variant of ECC presented by Freeman [12] to encrypt integer numbers that will be used in our scheme.

- *Gen* (). Choose an elliptic curve $\mathbb{E}/F_p$ with a point P of prime order n, and an integer $[s] \xleftarrow{R} [1, n]$. Output $pk = (P, Q = [s]P)$ and $sk = [s]$.

- *Enc (pk, M)*. Let $pk = (P, Q)$ and interpret $[M]$ as an integer. Choose $r \xleftarrow{R} [1, n]$ and output $C = (rP, [M]P + rQ)$.

- *Dec (C, sk)*. Let $sk = [s]$ and $C = (C_1, C_2)$. Compute $C_2 - [s]C_1$ and output $Log_p(T)$.

- *Add (pk, C, C$^{'}$)*. Write $C = (C_1, C_2)$ and $C^{'} = (C_1^{'}, C_2^{'})$. Choose $t \xleftarrow{R} [1, n]$ and output $(C_1 + C_1^{'} + tP, C_2 + C_2^{'} + tQ)$.

3.3 Complexity Assumptions

Definition 1 (Negligible Function). A function f is negligible if for every polynomial $p(.)$ there exists an N such that for all integers $n > N$ it holds $f(n) < \frac{1}{p(n)}$.

Definition 2 (Elliptic Curve Decisional Deffie Hellman). Let $E(F_p)$ be an elliptic curve over F_p and let $P \in F_p$ be a point of prime order n. The elliptic curve decisional Deffie Hellman problem $ECDDH$ is hard if for all probabilistic polynomial time (PPT) adversaries $\mathcal{A}$, there exists a negligible function $negl$ such that

$$|\Pr[\mathcal{A}(E, q, P, aP, bP, abP) = 1] - \Pr[\mathcal{A}(E, q, P, aP, bP, cP) = 1| < negl(k)$$

where a,b and c are uniformly random from $[1, n-1]$.

4. SYSTEM OVERVIEW AND THREAT MODEL

In this paper we consider the multi-owner privacy-preserving query processing scenario presented in [26], where authorized users can execute equality test queries over data, encrypted with different keys, contributed by multiple data owners.

4.1 Problem Statement

The proposed scenario includes m tables $\{1, \ldots, m\}$ with $\frac{n}{m}$ records, which have been horizontally partitioned among m data owners $O = \{O_1, \ldots O_m\}$. Each table T_i has a set of searchable attributes $A_1, \ldots, A_{P_1}$ and a set of extra attributes $B_1, \ldots, B_{P_2}$, where P_1 and P_2 are not necessarily equal. For the sake of simplicity, we assume that each T_i includes one searchable and extra attribute $T_{i,A}$ and $T_{i,B}$, respectively. $T_{i,A}$ and $T_{i,B}$ take their values from given set of values $V_{i,A}$ and $V_{i,B}$. For each value $v \in V_{i,A}$, $ext(v)$ denotes the value occuring in $V_{i,B}$ where $T_{i,A} = v$.

Given an equality test query, spanning the entire database, like $T_A = v$ over the union of owners' tables (T) stored at the cloud service provider, the result is a set of values from T_B whose corresponding values of attribute A are equal to v. The execution of the query should satisfy the following properties:

- *Data confidentiality*: During the query processing phase, database contents are disclosed neither to the cloud-based database server nor to the user posing the query;

- *Query privacy*: Query result is disclosed neither to the cloud server nor to data owners;

- *Query anonymous result*: The user does not learn which data owner the returned data belonged to.

4.2 System Overview

Here, we give an overview of our system model and discuss the security assumptions for the different parties involved in the framework. As shown in Figure 1, five parties are involved: Data owners, Proxy, Key administrator, Authorized users and the Cloud service provider.

1) *Data owners* locally encrypt their data corresponding to a horizontal partition of a common database and outsource them to a proxy server.

5) *Key administrator (KA)* generates a master key and random keys for the owners and users and accordingly computes keys for the proxy and the cloud service provider.

3) *Proxy server* translates owners' data encrypted with different keys to data encrypted under the same key.

2) *Authorized users* submit queries over relations composed of the union of the owners' tables.

4) *Cloud service provider* stores data contributed by the owners and executes search queries on behalf of the users.

4.3 Threat Model

We assume that key administrator is fully trusted and goes offline after the key distribution phase. The proxy and the cloud servers are semi-trusted i.e., they fairly execute protocols, but also they may try to learn information about the owners' data and the content of user queries. We further assume that there exists a secure communication channel between any pair of participating parties. It is worth pointing out that without colluding with the proxy server, the user and the cloud service provider together cannot access the master key and correspondingly the key of each data owner.

5. A FRAMEWORK FOR MULTI-OWNER PRIVACY-PRESERVING EQUALITY QUERIES

In this section, we provide a complete description of our proxy re-encryption scheme which is based on the adoption of an El-Gamal elliptic curve public cryptosystem. We consider the problem of executing equality test queries in multi-owner setting [26]. We use our ECC implementation to translate data owners' encrypted values to the encryption of the same values, so that users who do not hold the owners' key can access query results. This proxy based mechanism for delegating to authorized users the data owners' capability of decrypting their ciphertext will be called key translation. The proposed scheme has three phases, namely: *Key Generation*, *Key Translation* and *Query search*.

Phase 1- Key Generation. The key generation algorithm uses a fully trusted party to generate random keys for each party of the system.

Phase 2- Key Translation. The key translation is composed of *data encryption* and *data re-encryption* operations.

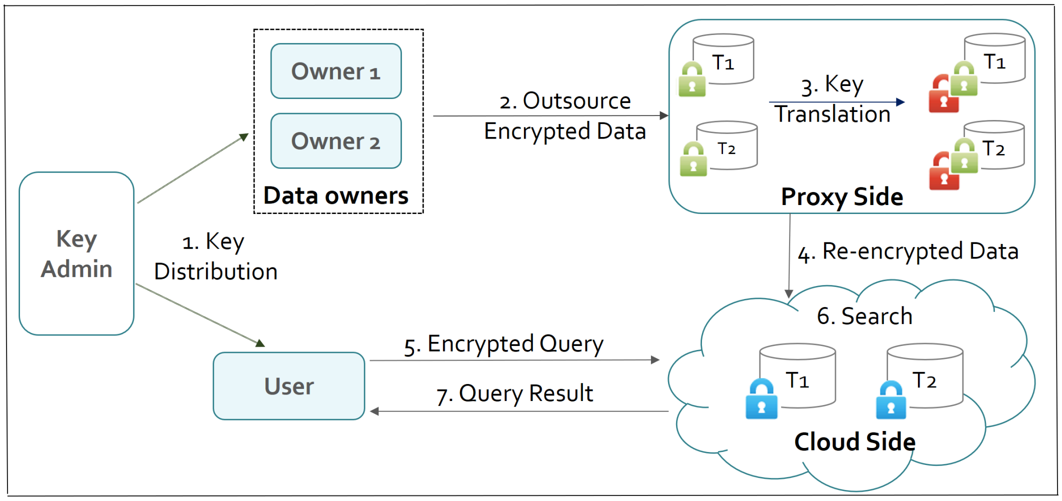


Figure 1: The proposed framework for proxy-based outsourcing scheme with 2 data owners and 1 authorized user. Owners send their encrypted data to the proxy who sets up for key translation by re-encrypting the data with the proxy key for each owner. The user submits her query in encrypted form to the cloud service provider, who executes the query over the union of data and sends back the encrypted results to the user. The user can access the result using her own key

During this phase the data encrypted under different keys are translated into data encrypted under the same key.

Phase 3- Query Search. The cloud service provider executes computation on ciphertexts to find match data corresponding to the user query.

The Algorithm (ECC Implementation of Equality test Queries).

Input.
i) m data owners, each holding a table T_i with one searchable attribute $T_{i,A}$ and one extra attribute $T_{i,B}$, and

ii) An equality test value v input from user j.

Output.
Return a set of extra values $ext(v)$ whose corresponding searchable attributes values are equal to the user value v.

I. Phase 1 (Key Generation). Let $\mathbb{E}$ be an elliptic curve over F_q with a point P of prime order n. The algorithm works as follow:

1. Choose an integer $s \overset{R}{\longleftarrow} [1, n]$ and output $pk = (\mathbb{E}, q, P)$ and $K_M = s$ as public and master key, respectively

2. KA chooses a random number r and distributes it on a secure channel to the owners and the users

3. On input master key K_M, the KA outputs a private key for each owner and each user, and accordingly for the proxy server and the cloud service provider

- For each data owner i, $1 \leq i \leq m$, the KA does the following:

  3-1. Generates a random key $K_{O_i} \overset{R}{\longleftarrow} [1, n]$
  3-2. On input K_M and owner identity i, KA outputs a re-encryption key as $K'_{O_i} = K_M - K_{O_i}$ and sends (i, K'_{O_i}) securely to the proxy server.

- For each user U_j, $1 \leq j \leq m_u$, the KA does the following:

  3-3. Generates $K_{U_j} \overset{R}{\longleftarrow} [1, n]$,
  3-4. On input K_M and user identity j, KA Computes $K'_{U_j} = K_M - K_{U_j}$ and divides K_{U_j} into two shares $K_{U_{j1}}$ and $K_{U_{j2}}$ such that $K_{U_j} = K_{U_{j1}} + K_{U_{j2}}$. KA then sends keys $(j, K_{U_{j1}})$, $K_{U_{j2}}$ and (j, K'_{U_j}) to the proxy, the user and the cloud service provider, respectively.

II. Phase 2 (Key Translation).

- Each data owner O_i encrypts its data as the following in parallel with the other data owners:

  1. On input K_{O_i}, a common random r and a searchable value $x \in V_{i,A}$, outputs the ciphertext $C_i(x) = (C_{i1}, C_{i2}) = (rP, xP + rK_{O_i}P)$

  2. Generates a new key $\bar{K_{O_i}}$ for encrypting the values of extra attribute with a semantically secure symmetric key encryption function f

  3. On input $\bar{K_{O_i}}$ and an extra value $y \in V_{i,B}$, outputs $C_i(y) = f_{\bar{k}_{O_i}}(y)$

  4. Encrypts the key $\bar{K_{O_i}}$ with owner key K_{O_i} to output $I_i = (rP, \bar{K_{O_i}} + rK_{O_i}P)$

  5. Sends the triples $(C_i(x), C_i(y), I_i)$ to the proxy server

- Given owner's i proxy side key K'_{O_i} ciphertext received from O_i, the proxy does the following operations:

  1. Finds owner's proxy side key K'_{O_i}

  2. On input the ciphertext $C_i(x)$ and K'_{O_i}, computes $C_{i1} \cdot K'_{O_i}$ and adds the result to C_{i2} to

obtain

$$\begin{aligned}\hat{C}_i(x) = (\hat{C}_{i1}, \hat{C}_{i2}) &= (rP, xP + rK_{O_i}P + rK^{'}_{O_i}P) \\ &= (rP, xP + r(K_{O_i} + K^{'}_{O_i})P) \\ &= (rP, xP + rK_M P)\end{aligned}$$

3. On input I_i and $\bar{K}_{O_i}$, encrypts I_i with owner's key K_{O_i}

$$\begin{aligned}\hat{I}_i &= (rP, \bar{K}_{O_i} + rK_{O_i}P + rK^{'}_{O_i}P) \\ &= (rP, \bar{K}_{O_i} + r(K_{O_i} + K^{'}_{O_i})P) \\ &= (rP, \bar{K}_{O_i} + rK_M P)\end{aligned}$$

4. Sends the triple $(\hat{C}_i(x),\ C_i(y), \hat{I}_i)$ to the cloud server.

III. Query Search Phase

Upon receiving user j encrypted value v as $C_j(v) = (C_{j1}, C_{j2}) = (rP, vP + rK_{U_{j2}}P)$, the query search algorithm is executed:

1. The cloud service provider sends the user identity j to the proxy for partially decrypting data using the shared key it holds $K_{U_{j1}}$, corresponding to the user value

2. On input the user identity j, the proxy server (i) re-encrypts user value with $K_{U_{j1}}$ to get:

$$\begin{aligned}v^{'} &= (rP, vP + rK_{U_{j2}}P) \\ &= (rP, vP + rK_{U_{j2}}P + rK_{U_{j1}}P) \\ &= (rP, vP + r(K_M - K^{'}_{U_j})P)\end{aligned}$$

(ii) partially decrypts $\hat{I}_i$ with $K_{U_{j1}}$ to output:

$$I^*_i = (rP, \bar{K}_{O_i} + r(K_M - K_{U_{j_1}})P)$$

(iii) The proxy server sends $(v^{'}, I^*_i)$ to the cloud service provider

3. The cloud service provider encrypts $v^{'}$ with its own key to get $\hat{v} = (rP, vP + rK_M P)$

4. On input $(\hat{C}_i(x), C_i(y), I^*_i)$ and $\hat{v}$, the cloud server returns $C_i(y)$ whose entry of $\hat{C}_i(x)$ is equal to $\hat{v}$ and partially decrypts its corresponding I^*_i as

$$\begin{aligned}\tilde{I}_i &= (rP, \bar{K}_{O_i} + r(K_M - K_{U_{j_1}} - K^{'}_{U_j})P) \\ &= (rP, \bar{K}_{O_i} + rK_{U_{j_2}}P)\end{aligned}$$

5. The cloud service provider sends the selected $C_i(y)$ together with the corresponding $\tilde{I}_i$ to the user

6. On input $(C_i(y), \tilde{I}_i)$, the user (i) decrypts $\tilde{I}_i$ with its private key to recover the key $\bar{K}_{O_i}$ as

$$\bar{K}_{O_i} = \bar{K}_{O_i} + rK_{U_{j_2}}P - rK_{U_{j_2}}P$$

(ii) decrypts $C_i(y)$ with the obtained key $\bar{K}_{O_i}$ to get the plaintext, corresponding to the result of the query

6. SECURITY ANALYSIS

In this section, we analyze the security properties of the proposed proxy re-encryption scheme. First, we consider a passive adversary who has access to all the encrypted data of data owners. If a data owner is compromised during the attack, the adversary can learn the owner key and the common random r. Despite knowing the public point P, r and the keys obtained from the compromised owners, the adversary cannot decrypt the data because of the hardness of discrete logarithm problem on elliptic curves. For the same reason, the obtained information prevents the adversary to learn the secret key of the other owners and the master key from the public key information.

Next, we show how the proxy re-encryption scheme is indistinguishable against chosen plaintext attack $(IND-CPA)$. We mean that an adversary $\mathcal{A}$ choosing two different values m_0 and m_1 from the domain of searchable attribute, and querying an oracle cannot distinguish which ciphertext is for what value with probability non-negligibly different from $\frac{1}{2}$.

Theorem 1. Our ECC implementation is $IND-CPA$ secure against the proxy if for all PPT adversaries $\mathcal{A}$ there exists a negligible function *negl* such that

$$Succ^{\mathcal{A}}_{ECC}(k) = \mathsf{Pr}\left[b^{'} = b \;\middle|\; \begin{array}{l} (K_O, K^{'}_O) \leftarrow KeyGen(K_M, O) \\ m_0, m_1 \leftarrow \mathcal{A}^{Enc(K_O,\cdot)}(K^{'}_O) \\ b \xleftarrow{R} \{0,1\} \\ C(m_b) = Enc(K_O, m_b) \\ b^{'} \leftarrow \mathcal{A}^{Enc(K_O,\cdot)}(K^{'}_O, C(m_b)) \end{array} \right] < \frac{1}{2} + negl(k)$$

Where O is a set of data owners and $K_O, K^{'}_O$ are the set of keys for data owners and the proxy, respectively. $KeyGen()$ and $Enc()$ functions are the key generation and Encryption of key translation phase of our algorithm.

Proof. Our proof relies on the assumption that DDH is hard for any adversary to distinguish between elliptic curve group elements abP and cP given aP and bP. Let's consider a PPT adversary $\mathcal{A}^{'}$ who attempts to challenge the proxy encryption $IND-CPA$ game using $\mathcal{A}$ as subroutine. $\mathcal{A}^{'}$ does the following:

Setup: $\mathcal{A}^{'}$ is given $(\mathbb{E}, q, P, P_1, P_2, P_3)$ as input, where $P_1 = aP$, $P_2 = bP$ and $P_3 = abP$ or cP for some random a, b, $c \in [1, n]$.

- $\mathcal{A}^{'}$ sends $(\mathbb{E}, q, P)$ to $\mathcal{A}$.
- For each owner $O_i \in O$ where O is a set of all data owners, $\mathcal{A}^{'}$ chooses a random element $K^{'}_{O_i} \in [1, n]$ and computes $K_{O_i}P = aP - K^{'}_{O_i}P = (a - K^{'}_{O_i})P$.
- $\mathcal{A}^{'}$ sends $(O_i, K^{'}_{O_i})$ to $\mathcal{A}$ and keeps $(O_i, K_{O_i}P, K^{'}_{O_i})$.

Query: Whenever $\mathcal{A}$ requires oracle access to the data owner encryption algorithm, it sends m, which is a value of searchable attribute domain to $\mathcal{A}^{'}$.

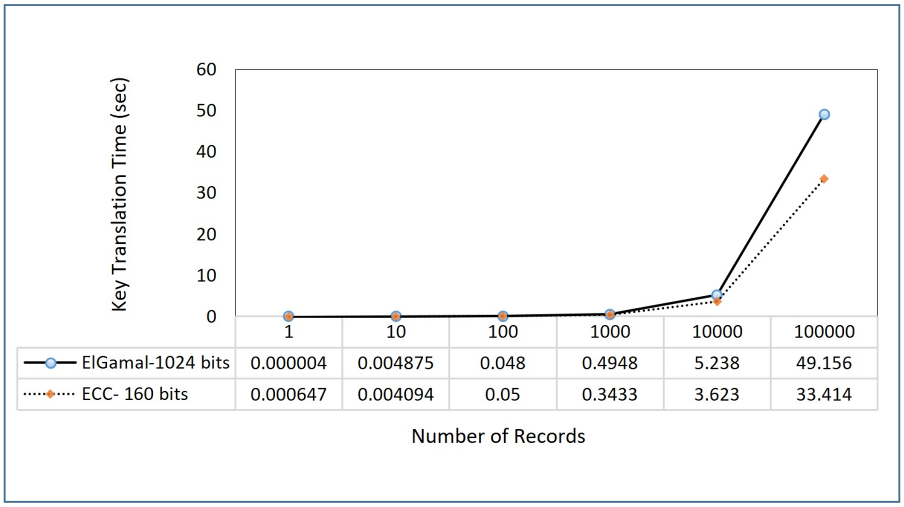
- $\mathcal{A}^{'}$ chooses a random element $r_t \in [1, n]$ and replies with $(r_t P, MP + r_t K_{O_i}P)$.

Challenge: At some point of time, $\mathcal{A}$ outputs m_0 and m_1. $\mathcal{A}^{'}$ chooses a random bit b and transmits $(P_2, m_b P - K_{O_i}P_2 + P_3)$ to $\mathcal{A}$.

Output: $\mathcal{A}$ outputs $b^{'}$. If $b = b^{'}$ holds, $\mathcal{A}^{'}$ outputs 1, otherwise outputs 0.

There are two cases to consider depending on the value of P_3:

Case 1: If $P_3 = cP$ that is a random element of elliptic group because c is chosen at random. Then, $m_b P - K_{O_i}P_2 + P_3$ is also a random element and gives no information about m_b, so



	1	10	100	1000	10000	100000
ElGamal-1024 bits	0.000004	0.004875	0.048	0.4948	5.238	49.156
ECC- 160 bits	0.000647	0.004094	0.05	0.3433	3.623	33.414

Figure 2: Comparing El-Gamal and ECC implementations of the proposed scheme with 3 data owners running in parallel

the adversary $\mathcal{A}$ must distinguish m_0, m_1 without additional information. The probability it can successfully output $b = b'$ is exactly $\frac{1}{2}$ when b is chosen uniformly random. $\mathcal{A}'$ outputs 1 iff $\mathcal{A}$ outputs $b = b'$ then we have

$$\Pr[\mathcal{A}'(E, q, P, aP, bP, cP) = 1] = \frac{1}{2}$$

Case 2: If $P_3 = abP$, then we have $(m_bP_2 - K'_{O_i}P_2 + P_3) = (m_bbP - K'_{O_i}bP + abP) = (m_bbP + bP(-K'_{O_i} + a) = (m_bbP + bP(K_{O_i} + a)$ that is a proper ciphertext under our ECC implementation. Then the probability is

$$\Pr[\mathcal{A}'(E, q, P, aP, bP, cP) = 1] = Succ^{\mathcal{A}}_{ECC}(k)$$

If DDH problem is hard then the following equation is true

$$\left|\Pr[\mathcal{A}'(E, q, P, aP, bP, abP) = 1] - \Pr[\mathcal{A}'(E, q, P, aP, bP, cP) = 1\right|$$

is negligible, then

$$\left|Succ^{\mathcal{A}}_{ECC}(k) - \frac{1}{2}\right|$$

is negligible, therefore $Succ^{\mathcal{A}}_{ECC}(k)$ is negligible close to $\frac{1}{2}$. □

Hence, without knowing owners keys, the proxy cannot distinguish the ciphertext in a chosen plaintext attack. For the sake of conciseness and clarity, we omit the security proof under IND-CPA attack against the cloud service provider that is analogous to the Theorem 1.

7. PERFORMANCE ANALYSIS

In this section we analyze the implementation of the proposed proxy based scheme using ECC and compare it with the technique described in Sepehri *et al.* [26] and in Hang *et al.* [15], which are based on the El-Gamal and pairings encryption methods, respectively.

We tested our ECC implementation and measured the time required for encryption and re-encryption operations varying the number of records from 1 to 100000. As stated in [26], the resulting time of the two operations denotes the translation key time, which is the time needed for converting data encrypted with different keys under the same key. We carried our experimental evaluation on a laptop device (Ubuntu 16.4 LTS, 2.60 GHz $8x$ Intel Core (TM) $i7-4720$ HQ CPU, 16 GB RAM). We implemented our scheme based on 160-bit ECC over prime field in C using big number integer functions of the GMP library [10]. We set up elliptic curve using Brainpoolpl60r [20] domain parameter over finite curve.

First, we compared our scheme with [26] underlying El-Gamal encryption with a 1024-bit prime p and a 160-bit prime q. We analyzed the changes in the performance of the two protocols by increasing the number of records, which has been horizontally partitioned among 3 data owners. Figure. 2 shows the performance where every data owner holds a uniform number of records. Each point corresponds to the average computation time for the key translation obtained by running 10 equality test queries on encrypted data. As shown in Figure. 2, our proposal improves the performance still providing an equivalent level of security of [26].

Next, we compare our ECC implementation with [15], which adopts Type A pairing provided by the PBC library [21] with security level of 80 bits. Type A pairings are built on top of an elliptic curve $y^2 = x^3 + x$ over a finite field F_q, for some prime $q = 3$ mod 4, and have a fixed embedding degree $k = 2$. As mentioned in their paper, the time to encrypt one item is $1.56ms$ and the proxy re-encryption consumes $1.019ms$. The key translation time is $0.02s$ for one item and $200s$ for 10^5 records. Compared to [15], our ECC implementation saves $100s$ for the key translation and the obvious reason is because our scheme involves 1 multiplication and 1 addition compared to the pairing cost of [15] during proxy re-encryption process. It can be noticed that we can increase the efficiency of our scheme to save $166s$ compared to [15] if the 3 data owners run the protocol in parallel as shown in Figure. 2.

Figure. 3 shows the key translation time of our scheme compared to [26, 15] for 10^5 number of records. Clearly among the compared schemes, our ECC implementation came to be fastest for large size data, as expected.

8. CONCLUSION

We presented a proxy based protocol to execute queries over outsourced encrypted database, preserving the confidentiality of

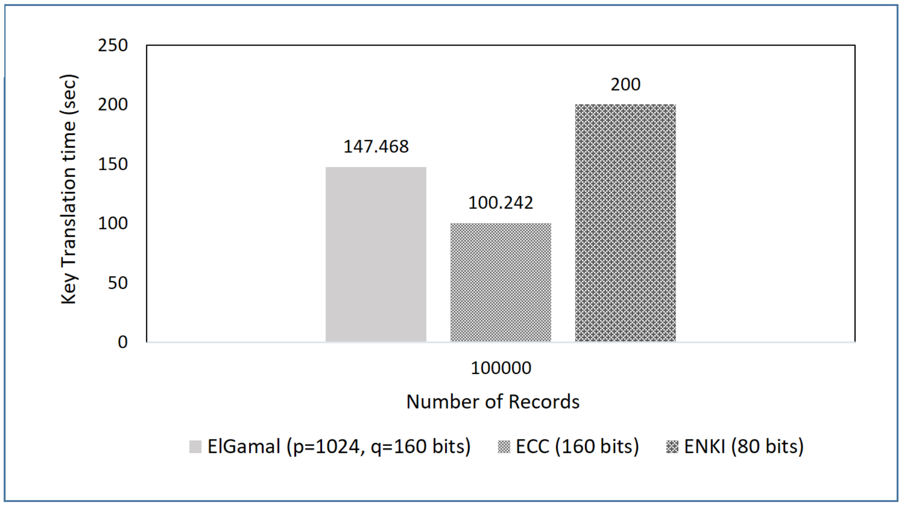


Figure 3: Comparing key translation speed of the different schemes on input size 10^5 records

owners' data and users' queries. We considered the problem of equality test queries in multi-owner scenario adopting El-Gamal elliptic curve to speed up the key translation algorithm. We analyzed the protocol in terms of security and performance, achieving encouraging results compared to previous proposals [15, 26].

In future work we plan to extend the protocol to execute other types of queries such as range queries. In addition, the proposed scheme does not provide an access control mechanism since every authorized user is allowed to search all owners databases locatedÂă at the cloud service provider. Hence, we planÂă to enforce access control policies on data according to their provenance, in order to support search queries coming from users with different access rights.

9. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD '04, pages 563–574, New York, NY, USA, 2004. ACM.

[2] C. A. Ardagna, R. Asal, E. Damiani, and Q. H. Vu. From security to assurance in the cloud: A survey. *ACM Comput. Surv.*, 48(1):2:1–2:50, 2015.

[3] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, Feb. 2006.

[4] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *In EUROCRYPT*, pages 127–144. Springer-Verlag, 1998.

[5] M. Cerveró Abelló, V. Mateu Meseguer, J. M. Miret Biosca, F. Sebé Feixas, and J. Valera Martín. An elliptic curve based homomorphic remote voting system. 2014.

[6] S. Cimato, E. Damiani, F. Zavatarelli, and R. Menicocci. Towards the certification of cloud services. In *Services (SERVICES), 2013 IEEE Ninth World Congress on*, pages 92–97, June 2013.

[7] R. H. Deng, J. Weng, S. Liu, and K. Chen. *Chosen-Ciphertext Secure Proxy Re-encryption without Pairings*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[8] C. Dong, G. Russello, and N. Dulay. Shared and searchable encrypted data for untrusted servers. *Journal of Computer Security*, 19(3):367–397, 2011.

[9] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[10] T. G. et al. GNU multiple precision arithmetic library 4.1.2, December 2002. http://swox.com/gmp/.

[11] L. Ferretti, M. Colajanni, and M. Marchetti. Distributed, concurrent, and independent access to encrypted cloud databases. *IEEE Transactions on Parallel and Distributed Systems*, 25(2):437–446, Feb 2014.

[12] D. Freeman. Homomorphic encryption and the bgn cryptography. 2011.

[13] M. Hadavi, R. Jalili, E. Damiani, and S. Cimato. Security and searchability in secret sharing-based data outsourcing. *International Journal of Information Security*, pages 1–17, 2015.

[14] M. A. Hadavi, E. Damiani, R. Jalili, S. Cimato, and Z. Ganjei. AS5: A secure searchable secret sharing scheme for privacy preserving database outsourcing. In *Data Privacy Management and Autonomous Spontaneous Security, 7th International Workshop, DPM 2012, and 5th International Workshop, SETOP 2012, Pisa, Italy, September 13-14, 2012. Revised Selected Papers*, pages 201–216, 2012.

[15] I. Hang, F. Kerschbaum, and E. Damiani. Enki: Access control for encrypted query processing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 183–196, New York, NY, USA, 2015. ACM.

[16] A. Ivan and Y. Dodis. Proxy cryptography revisited. In *in Proceedings of the Network and Distributed System Security Symposium (NDSS*, 2003.

[17] M. Kantarcioglu and C. Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1026–1037, 2004.

[18] N. Koblitz. *A Course in Number Theory and Cryptography.* Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[19] B. Libert and D. Vergnaud. *Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption*, pages 360–379. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[20] M. Lochter and J. Merkle. Elliptic curve cryptography (ecc) brainpool standard curves and curve generation, 2010.

[21] B. Lynn. Pbc library manual, 2007.

[22] M. Mambo and E. Okamoto. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts (special section on cryptography and information security). *IEICE*

transactions on fundamentals of electronics, communications and computer sciences, 80(1):54–63, jan 1997.

[23] V. S. Miller. Use of elliptic curves in cryptography. In *Lecture Notes in Computer Sciences; 218 on Advances in cryptology—CRYPTO 85*, pages 417–426, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[24] M. Sepehri, S. Cimato, and E. Damiani. A multi-party protocol for privacy-preserving range queries. In W. Jonker and M. Petkovic, editors, *Secure Data Management - 10th VLDB Workshop, SDM 2013, Trento, Italy, August 30, 2013, Proceedings*, volume 8425 of *Lecture Notes in Computer Science*, pages 108–120. Springer, 2013.

[25] M. Sepehri, S. Cimato, and E. Damiani. Privacy-preserving query processing by multi-party computation. *The Computer Journal*, 58(10):2195–2212, 2015.

[26] M. Sepehri, S. Cimato, E. Damiani, and C. Y. Yeuny. Data sharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, pages 1357–1362. IEEE, 2015.

[27] J. Shao, Z. Cao, X. Liang, and H. Lin. Proxy re-encryption with keyword search. *Inf. Sci.*, 180(13):2576–2587, 2010.

[28] J. H. Silverman. *The arithmetic of elliptic curves*. Graduate texts in mathematics. Springer, New York, London, 2009. Informations sur la publication, prÃl'face et table des matiÃÍres disponibles en ligne Ãă l'adresse http://www.math.brown.edu/ÌČjhs/AECHome.html.