MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

A Distance-sensitive Attribute Based Cryptosystem for Privacy-Preserving Querying

Sun, W.; Rane, S.

TR2012-054 July 2012

Abstract

We propose an attribute-based cryptosystem in which decryption is conditional on the distance between attributes. Alice constructs a cipher text that consists of an encrypted message and a hidden attribute vector. Bob is able to decrypt Alice's message if and only if his attribute vector is within a specified maximum distance from Alice's attribute vector. We provide constructions for Euclidean and Hamming distances. The cryptosystem has advantages for privacy preserving querying. In particular, all parties can broadcast their respective cipher texts or store them on a database server. Then, a client – not necessarily belonging to the original set of parties – can independently and privately query the database server for cipher texts whose attributes are within some small distance from its own attribute. We describe an application of this cryptosystem in which a customer obtains recommendations from other customers of a movie rental company in a privacy-preserving manner.

IEEE International Conference on Mjltimedia and Expo (ICME)

A DISTANCE-SENSITIVE ATTRIBUTE BASED CRYPTOSYSTEM FOR PRIVACY-PRESERVING QUERYING

Wei Sun and Shantanu Rane

Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.
{weisun,rane}@merl.com

ABSTRACT

We propose an attribute-based cryptosystem in which decryption is conditional on the distance between attributes. Alice constructs a ciphertext that consists of an encrypted message and a hidden attribute vector. Bob is able to decrypt Alice's message if and only if his attribute vector is within a specified maximum distance from Alice's attribute vector. We provide constructions for Euclidean and Hamming distances. The cryptosystem has advantages for privacy preserving querying. In particular, all parties can broadcast their respective ciphertexts or store them on a database server. Then, a client – not necessarily belonging to the original set of parties – can independently and privately query the database server for ciphertexts whose attributes are within some small distance from its own attribute. We describe an application of this cryptosystem in which a customer obtains recommendations from other customers of a movie rental company in a privacy-preserving manner.

Index Terms— Attribute-based encryption, Bilinear maps, Privacy preserving querying

1. INTRODUCTION

Consider a movie rental company that wants to provide a service using which a customer, Bob, can obtain recommendations for movies to watch based on ratings he has provided for movies in the past. A logical solution is to recommend movies from the viewing profile of a customer whose movie ratings closely match the ratings assigned by Bob. However, in the interest of privacy, none of the customers should discover each other's movie viewing profile or preferences. Privacy preserving querying scenarios of this sort are expected to become increasingly common with the growing prevalence of cloud computing. Computing the distance between the movie ratings of two customers without revealing the individual ratings falls under the realm of secure multiparty computation [1, 2]. As long as the function to be computed, in this case the distance between rating vectors, can be expressed as an algebraic circuit, there exists a protocol to compute it while satisfying the privacy requirements of all parties. In practice, however, such a generalized protocol is extremely complex in terms of computation and communication overhead. Therefore, depending upon the specific function of interest, many efficient specialized protocols have been developed.

One line of work that has received increased attention in recent years is the application of public-key homomorphic cryptosystems for computing distance functions in the encrypted domain. Depending on the encrypted-domain computation that these cryptosystems allow, they can be classified into additively homomorphic [3, 4], multiplicatively homomorphic [5] and doubly homomorphic [6] cryptosystems. Such cryptosystems have been applied for privacy preserving clustering [7, 8], secure distance computation [9] and, indeed, for private movie recommendations [10]. Many of these privacy preserving protocols operate in two stages; the distance or correlation between data entities is computed securely in the first stage, and data retrieval based on the distance criterion occurs in the second stage. While these protocols are efficient for a single data retrieval request, they do not scale for a very large number of users sequentially or simultaneously querying a database. For example, if a second customer, Charlie, wants to retrieve movie recommendations, the entire protocol must be executed again, using the encryption/decryption key pair of Charlie. It would be preferable, if all parties used *public encryption parameters* to store or update their data securely on a server, after which, any user(s) can retrieve anonymous recommendations from the server *using user-specific decryption keys* calculated from their movie ratings. Our goal is to construct a cryptosystem that makes this possible. To do this, we depart from the encrypted-domain homomorphic computation paradigm. Instead, we consider cryptosystems which allow decryption conditioned on some mathematical property of the data, such as a distance between movie ratings. We achieve this by constructing an Attribute Based Encryption (ABE) scheme.

In a conventional cryptosystem, when Alice needs to transmit a message securely to Bob, she must encrypt it either with a symmetric key known to her and Bob, or with Bob's public key. Instead, in an ABE system such as [11, 12, 13], Alice obtains some public encryption parameters from a Key Authority and generates a ciphertext that contains two entities: the encryption of the message m and a so-called attribute vector $\mathbf{x}$. The encryption can only be reversed by a decryption

key that satisfies a mathematical condition on the attribute $\mathbf{x}$ of Alice, and the attribute $\mathbf{y}$ (say) of Bob. In order to perform decryption, Bob applies to the Key Authority for a decryption key which is a function of his attribute vector $\mathbf{y}$. In one example of ABE, Bob can decrypt m if and only if $\mathbf{x} \perp \mathbf{y}$, or equivalently $\mathbf{x}^T\mathbf{y} = 0$ [12].

In some ABE systems [11], Alice's attribute $\mathbf{x}$ is a publicly readable portion of the ciphertext; thus anybody can download the ciphertext but can only decrypt the message if they possess a compliant attribute. In other ABE systems [12], the attribute $\mathbf{x}$ is not publicly readable, i.e., the ciphertext consists of a hidden or encrypted attribute in conjunction with the encrypted message. This has the advantage that an adversary who obtains the ciphertext cannot determine the criteria to be satisfied by the intended recipient of the message. This is of practical importance because it may be necessary in some applications to protect both the attribute vector and the message being transmitted. In this case, the decryption function verifies whether the decryptor's attribute is compliant without revealing the encryptor's attribute, and if compliant, reveals the message m.

The remainder of this paper is organized as follows: Section 2 reviews bilinear mappings and their properties, and the computational hardness assumptions that underpin the proposed work. In Section 3, we construct an attribute based cryptosystem in which decryption is possible only if the squared Euclidean distance, i.e., the squared ℓ_2 distance between the attributes of the encryptor and decryptor are below a threshold. Section 4 presents a variant of the proposed scheme in which the decryption is conditional on the Hamming distance between binary attribute vectors. In Section 5, we show an application of this cryptosystem to a privacy-preserving movie recommendation system and compare and contrast this ABE-based recommender system with previously proposed recommender systems that exploit homomorphic encryption. Section 6 concludes the paper.

2. THEORETICAL BACKGROUND

2.1. Bilinear Groups of Composite Order

We review the mathematical properties of bilinear groups of composite order, particularly when the group order N is a product of three primes [12]. As explained in the later sections, these properties drive the construction of the proposed cryptosystem, and are used to prove the correctness of the algorithm.

Let $N = pqr$, where p, q, r are three distinct prime numbers. Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of order N. Then, the mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map if the following conditions are satisfied: (1) $e(x^\alpha, y^\beta) = e(x, y)^{\alpha\beta}$ for $x, y \in \mathbb{G}$ and $\alpha, \beta \in \mathbb{Z}$, and (2) If g is a generator of $\mathbb{G}$, then $e(g, g)$ is a generator of $\mathbb{G}_T$. Now, consider the cyclic groups $\mathbb{G}_p$, $\mathbb{G}_q$ and $\mathbb{G}_r$ with orders p, q and r respectively and generators g_p, g_q and g_r respectively. Then $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$ and any element $x \in \mathbb{G}$ can be represented as $x = g_p^\alpha g_q^\beta g_r^\gamma$, where $\alpha, \beta, \gamma \in \mathbb{Z}$. From the definition of a bilinear map, if g generates $\mathbb{G}$ then g^{pq} is a generator of $\mathbb{G}_r$, g^{qr} is a generator of $\mathbb{G}_p$, and g^{pr} is a generator of $\mathbb{G}_q$. Below, we present some well-known properties of the bilinear maps $e(\cdot, \cdot)$ that will be useful later in the paper:

$$
\begin{aligned}
e(g_p^\alpha, g_q^\beta) &= 1 \\
e(g_p^\alpha g_q^{\beta'}, g_q^\beta) &= e(g_q^{\beta'}, g_q^\beta) \\
e(g_p^\alpha, g_p^{\beta'} g_p^\beta) &= e(g_p^\alpha, g_p^{\beta'}) \cdot e(g_p^\alpha, g_p^\beta) \\
e(g_p^\alpha g_q^\beta, g_p^{\alpha'} g_q^{\beta'}) &= e(g_p^\alpha, g_p^{\alpha'}) \cdot e(g_q^\beta, g_q^{\beta'})
\end{aligned}
$$

Proving the above properties is a straightforward application of the definition of the bilinear mapping given earlier, and the properties of generators of multiplicative cyclic groups, which allows us to express $g_p \equiv g^{qr}$, $g_q \equiv g^{pr}$, and $g_r \equiv g^{pq}$.

2.2. Security Assumptions

In this paper, we describe in detail the construction of a distance-attribute-based cryptosystem and provide a sketch of the proof of security. The detailed security proof is deferred to a later work. At this stage, we note that breaking the proposed cryptosystem reduces to solving two problems (described below) that are regarded as computationally intractable. Consider an integer $N = pqr$ for large prime numbers p, q, r and a cyclic group $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$. Then, the security of our cryptosystem is based on the following problems.

1. **Subgroup Decision Problem:** It is computationally hard to distinguish elements of the subgroup $\mathbb{G}_p \times \mathbb{G}_q$ from an element of the group $\mathbb{G}$. In other words, it is computationally hard to determine whether an element is drawn from a uniform distribution on $\mathbb{G}$, or from a uniform distribution on the subgroup $\mathbb{G}_p \times \mathbb{G}_q$.
2. **Pairing Diffie-Hellman Problem:** Consider a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose $\bar{g}$ as one element from the set $\{g_p, g_q, g_r\}$. Suppose that $e(\bar{g}, \bar{g})^v$ is given and an integer u is chosen at random. Then, it is computationally hard to distinguish $e(\bar{g}, \bar{g})^{uv} \in \mathbb{G}_T$ from a randomly chosen element of $\mathbb{G}_T$. Another way of stating this is that, given $e(\bar{g}, \bar{g})^v$, it is computationally hard to obtain v.

Both these assumptions are related to the computational intractability of finding non-trivial prime factors of N. For a detailed discussion of proving security of an ABE system using the above hardness assumptions, please refer to [12].

3. ABE CONDITIONED ON ℓ_2 DISTANCE

An ABE system involves the interplay of three kinds of entities: (1) one or more encryptors (2) one or more decryptors (3) a Key Authority. In our treatment, we employ an

$$\text{Ciphertext, } \xi(m,\mathbf{x}): \quad (A_0, \{A_i\}_{i=1}^n, \bar{A}, B, C) = \left((g_q c)^{\gamma}, \{(g_q c)^{-\gamma x_i} s_i\}_{i=1}^n, (g_q c)^{-\gamma \sum_{i=1}^n x_i^2} s_0, g_p^{\delta}, m \cdot e(g_p, a)^{\delta} \right) \tag{1}$$

$$\text{Pre-decryption Key: } \quad (K_0, \{K_t^{(1)}, K_t^{(2)}\}_{t=0}^{\tau}) = \left(g_p^{\alpha} g_q^{\beta}, \{a^{-1} g_p^{\alpha - \rho_t - 2\sum_{i=1}^n \alpha_i + 2\alpha \sum_{i=1}^n z_i}, g_p^{\rho_t} g_q^{\beta \sigma_t}\}_{t=0}^{\tau} \right) \tag{2}$$

$$\text{Pre-decryption Key: } \quad (K_0, \{K_t^{(1)}, K_t^{(2)}\}_{t=0}^{\tau}) = \left(g_p^{\alpha} g_q^{\beta}, \{a^{-1} g_p^{n\alpha - \rho_t - 2\sum_{i=1}^n \alpha_i + 2\alpha w(\mathbf{z})}, g_p^{\rho_t} g_q^{\beta \sigma_t}\}_{t=0}^{\tau} \right) \tag{3}$$

$$\text{Decryption Function: } \quad D_{\ell_2} = C \cdot e(B, K_t^{(1)}) \cdot e(A_0, K_t^{(3)}) \cdot e(A_0 B, K_t^{(2)}) \cdot \left(e(\bar{A}B, K_0)\right)^{-1} \cdot \left(\prod_{i=1}^n e(A_i B, K_i) \right)^2 \tag{4}$$

$$\text{Decryption Function: } \quad D_H = C \cdot e(B, K_t^{(1)}) \cdot e(A_0, K_t^{(3)}) \cdot e(A_0 B, K_t^{(2)}) \cdot \left(\prod_{i=1}^n e(A_i B, K_0) \right)^{-1} \cdot \left(\prod_{i=1}^n e(A_i B, K_i) \right)^2 \tag{5}$$

encryptor, Alice, and a decryptor, Bob. Decryption is allowed on the condition that the squared ℓ_2 distance between the attributes $\mathbf{x}$ of Alice and $\mathbf{y}$ of Bob is less than a threshold τ.

Setup: The Key Authority generates large prime numbers p, q, r and two cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of order $N = pqr$. As above, there are cyclic groups $\mathbb{G}_p$, $\mathbb{G}_q$ and $\mathbb{G}_r$ with orders p, q and r respectively and generators g_p, g_q and g_r respectively. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a non-degenerate bilinear map[1]. The Key Authority randomly chooses $a \in \mathbb{G}_p$ and $c \in \mathbb{G}_r$, and outputs public parameters $(N, g_p, g_r, g_q c, e(g_p, a))$, and retains a private master key (p, q, r, g_q, a, c). The Key Authority and Bob publicly agree on a distance threshold τ.

Encryption: Alice has a message m to be encrypted and an integer attribute vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ to be hidden in the ciphertext. She randomly chooses $\delta, \gamma \in \mathbb{Z}$ and $s_i \in \mathbb{G}_r, i = 0, 1, 2, ..., n$ and computes the ciphertext as shown in (1). Note that the parameters s_i, δ and γ are chosen at random for each encryption and, as we shall see, they are not needed by the decrypting party. These parameters ensure that the encryption of the same message m is different every time, i.e., the ciphertext is semantically secure.

Decryption Key Generation: Bob has an integer attribute vector, $\mathbf{y} = (y_1, y_2, ..., y_n)$. To hide $\mathbf{y}$ from the Key Authority, he randomly chooses an integer vector $\mathbf{z} = (z_1, z_2, ..., z_n)$ and sends $\mathbf{z} + \mathbf{y}$ and $\sum_{i=1}^n z_i$ to the Key Authority. The Key Authority randomly chooses integers $\alpha, \beta, \alpha_i, i = 1, 2, ..., n$ and σ_t, ρ_t, $t = 0, 1, ..., \tau$, and generates a "pre-decryption key". This is given by (2) and the following relations:

$$K_i' = g_p^{\alpha_i} g_q^{\beta(y_i + z_i)} \text{ for } i = 1, 2, ..., n$$

$$K_t'^{(3)} = g_p^{\alpha} g_q^{\beta(\sum_{i=1}^n (y_i + z_i)^2 - t - \sigma_t)} \text{ for } t = 0, 1, ..., \tau$$

Then, using the pre-decryption key and his knowledge of $\mathbf{y}$ and $\mathbf{z}$, Bob obtains the decryption key, which is given by (2) and the following relations:

$$K_i = K_i' K_0^{-z_i} \text{ for } i = 1, 2, ..., n \tag{6}$$

$$K_t^{(3)} = K_t'^{(3)} K_0^{-\sum_{i=1}^n (z_i^2 + 2 z_i y_i)} \text{ for } t = 0, 1, ..., \tau \tag{7}$$

Decryption: Given the attribute $\mathbf{y}$, Bob evaluates the expression (4) repeatedly for $t = 0, 1, ...\tau$ and stops if he recovers the message m. The message m is designed in a special way to ensure that Bob can verify that he has succeeded in recovering it. Details of this are explained below.

Proof of Correctness: We now evaluate the right hand side of (4) to show that the algorithm is correct. Throughout, we will use the properties of bilinear maps from Section 2. Let us evaluate some terms in expression (4) separately. First note that, for $t = 0, 1, ..., \tau$, we have

$$e(A_0, K_t^{(3)}) = e\left((g_q c)^{\gamma}, g_p^{\alpha - \sum_{i=1}^n (z_i^2 + 2 z_i y_i)} g_q^{\beta(\sum_{i=1}^n y_i^2 - t - \sigma_t)} \right)$$

$$= e(g_q, g_q)^{\gamma \beta (\sum_{i=1}^n y_i^2 - t - \sigma_t)} \tag{8}$$

$$e(A_0 B, K_t^{(2)}) = e\left((g_q c)^{\gamma} g_p^{\delta}, g_p^{\rho_t} g_q^{\beta \sigma_t} \right)$$

$$= e(g_p^{\delta}, g_p^{\rho_t}) \cdot e(g_q, g_q)^{\gamma \beta \sigma_t} \tag{9}$$

$$\left(e(\bar{A}B, K_0)\right)^{-1} = e(g_p^{\delta}, g_p^{-\alpha}) \cdot e(g_q, g_q)^{\gamma \beta \sum_{i=1}^n x_i^2} \tag{10}$$

[1] There exist algorithms based on elliptic curves to generate groups of composite order and bilinear mappings using these groups. Examples include the Weil pairing and Tate pairing [14].

Further, for each $i = 1, 2, ..., n$,

$$\begin{aligned} e(A_iB, K_i) &= e\left(g_p^{\delta}(g_q c)^{-\gamma x_i} s_i, g_p^{\alpha_i - \alpha z_i} g_q^{\beta y_i}\right) \\ &= e(g_p^{\delta}, g_p^{\alpha_i - \alpha z_i}) \cdot e(g_q^{-\gamma x_i}, g_q^{\beta y_i}) \\ &= e(g_p^{\delta}, g_p^{\alpha_i - \alpha z_i}) \cdot e(g_q, g_q)^{-\gamma\beta x_i y_i}, \end{aligned}$$

so the product term in the decryption expression becomes

$$\left(\prod_{i=1}^{n} e(A_iB, K_i)\right)^2 = e(g_p^{\delta}, \prod_{i=1}^{n} g_p^{2(\alpha_i - \alpha z_i)}) \cdot e(g_q, g_q)^{-2\gamma\beta \mathbf{x}^T \mathbf{y}}$$

$$= e\left(g_p^{\delta}, g_p^{2\sum_{i=1}^{n}(\alpha_i - \alpha z_i)}\right) \cdot e(g_q, g_q)^{-2\gamma\beta \mathbf{x}^T \mathbf{y}} \quad (11)$$

Upon evaluating the product of the right-hand sides of (8),(9),(10) and (11) using the properties of bilinear maps, we obtain

$$V := e\left(g_p^{\delta}, g_p^{\rho_t - \alpha + 2\sum_{i=1}^{n}(\alpha_i - \alpha z_i)}\right) \cdot e(g_q, g_q)^{\gamma\beta(\|\mathbf{x}-\mathbf{y}\|_2^2 - t))}$$

Substituting the value of V in expression (4), we have

$$\begin{aligned} D_{\ell_2} &= C \cdot e(B, K_t^{(1)}) \cdot V \\ &= m \cdot e(g_p, a)^{\delta} \cdot e\left(g_p^{\delta}, a^{-1} g_p^{\alpha - \rho_t - 2\sum_{i=1}^{n}(\alpha_i - \alpha z_i)}\right) \cdot V \\ &= m \cdot e(g_q, g_q)^{\gamma\beta(\|\mathbf{x}-\mathbf{y}\|_2^2 - t))} \quad (12) \end{aligned}$$

Thus, the message m is unmasked by the decrypting party if and only if $\|\mathbf{x} - \mathbf{y}\|_2^2 = t$. Otherwise, the result D_{ℓ_2} is just an element of $\mathbb{G}_T$. A practical issue here is to embed a *publicly known* pattern in the message m such that a decryptor can actually verify that he has decrypted m. One way to accomplish this is to left-shift the digits of m, and append a publicly known pattern of digits. For example, if the true message is "7984", then Alice actually encrypts $m =$**7984**29742, where the last 5 digits are used to verify correct decryption. The underlying assumption is that, because the ciphertext field is so large, it is extremely unlikely that Bob's calculation of D_{ℓ_2} will return a value with the chosen 5-digit pattern for $\|\mathbf{x} - \mathbf{y}\|_2^2 \neq t$. Note that the above calculation of V and D_{ℓ_2} must be performed for all integers $t \in [0, \tau]$. If Bob discovers the pattern in the last 5 digits of D_{ℓ_2}, he declares that decryption was successful for some $t \leq \tau$, removes the 5-digit pattern, and recovers the message "7984". If he does not discover the 5-digit pattern in D_{ℓ_2} for any $t \in [0, \tau]$, then decryption is deemed unsuccessful based on the ℓ_2 distance condition on the attributes.

Proof of Security (Sketch): We briefly sketch the proof of security, which we shall detail in a later work. In particular, we consider "selective" security, wherein the adversary first generates example attributes, and a simulator chooses encryption parameters in response to this action. Now, consider the following two-person game, which challenges the adversary to distinguish between ciphertexts based on two attributes that he has generated. This proof strategy is similar to that used in [12] :

1. The adversary chooses three attribute vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{v}$ such that $\|\mathbf{x} - \mathbf{v}\|_2^2 = \|\mathbf{y} - \mathbf{v}\|_2^2$, and sends $\mathbf{x}$ and $\mathbf{y}$ to the simulator.

2. Given public encryption parameters, the simulator chooses random variables $s_i^{(\mathbf{x})}, s_i^{(\mathbf{y})} \in \mathbb{G}_r$ for all $i = 0, 1, 2, ..., n$, and generates ciphertexts $\xi(m, \mathbf{x})$ and $\xi(m, \mathbf{y})$ respectively, i.e., it encrypts the same message m using two different attribute vectors.

3. The simulator express the two ciphertexts in the form $\xi(m, \lambda)$ parameterized by a Bernoulli-0.5 random variable λ such that

$$\xi(m, \lambda) = \begin{cases} \xi(m, \mathbf{x}), & \text{if } \lambda = 0 \\ \xi(m, \mathbf{y}), & \text{if } \lambda = 1. \end{cases}$$

The simulator tosses a fair coin, records the result λ, and sends $\xi(m, \lambda)$ to the adversary. Now, whether the adversary can successfully decrypt the ciphertext or not, he has no strategy better than random guessing to distinguish whether the attribute used to construct $\xi(m, \lambda)$ was $\mathbf{x}$ or $\mathbf{y}$.

4. ABE CONDITIONED ON HAMMING DISTANCE

In this section, we describe a variant in which the attributes $\mathbf{x}$ and $\mathbf{y}$ are binary and decryption is allowed provided the Hamming distance $H(\mathbf{x}, \mathbf{y})$ between the attributes is less than a threshold τ.

Setup: This step is identical to that in Section 3.

Encryption: Alice has a message m to be encrypted and a *binary* attribute vector $\mathbf{x} = (x_1, x_2, ..., x_n)$ to be hidden in the ciphertext. She randomly chooses $\delta, \gamma \in \mathbb{Z}$ and $s_i \in \mathbb{G}_r, i = 1, 2, ..., n$ and computes the ciphertext as $(A_0, \{A_i\}_{i=1}^n, B, C)$, where $A_0, \{A_i\}_{i=1}^n, B$ and C are exactly as given in (1). Note that this ciphertext does not contain the $\bar{A}$ term from (1).

Decryption Key Generation: Bob also has a *binary* attribute vector, $\mathbf{y} = (y_1, y_2, ..., y_n)$. To hide $\mathbf{y}$ from the Key Authority, he randomly chooses a *binary* vector $\mathbf{z} = (z_1, z_2, ..., z_n)$ and sends $\mathbf{z} \oplus \mathbf{y}$ and $w(\mathbf{z}) = \sum_{i=1}^n z_i$ to the Key Authority. Here $\oplus$ denotes addition modulo 2 and $w(\mathbf{z})$ is the Hamming weight of $\mathbf{z}$. The Key Authority randomly chooses integers $\alpha, \beta, \alpha_i, i = 1, 2, ..., n$ and σ_t, ρ_t, $t = 0, 1, ..., \tau$, and generates a "pre-decryption key". This is given by (3) and the

following relations:

$$K_i' = g_p^{\alpha_i} g_q^{\beta(y_i+z_i)} \text{ for } i = 1, 2, ..., n.$$
$$K_t'^{(3)} = g_p^{\alpha} g_q^{\beta(w(\mathbf{y}\oplus\mathbf{z})-t-\sigma_t)} \text{ for } t = 0, 1, ..., \tau.$$

Then, using the pre-decryption key and his knowledge of $\mathbf{y}$ and $\mathbf{z}$, Bob obtains the decryption key, which is given by (3) and the following relations:

$$K_i = K_i' K_0^{-z_i} \text{ for } i = 1, 2, ..., n.$$
$$K_t^{(3)} = K_t'^{(3)} K_0^{-w(\mathbf{z})+2\mathbf{z}^T\mathbf{y}} \text{ for } t = 0, 1, ..., \tau.$$

Decryption: Given the binary attribute $\mathbf{y}$, Bob evaluates the expression (5) repeatedly for $t = 0, 1, ...\tau$ and stops if he recovers the message m. Again, the message m is designed as in Section 3 to ensure that Bob knows when he has recovered m rather than a random element of $\mathbb{G}_T$.

Proof of Correctness: We now evaluate the right hand side of (5) to show that the algorithm is correct. Using the properties of bilinear maps from Section 2 and proceeding as in Section 3, we can similarly show that

$$D_H = m \cdot e(g_q, g_q)^{\gamma\beta(H(\mathbf{x},\mathbf{y})-t))} \tag{13}$$

Thus, the message m is unmasked and discovered by the decrypting party if and only if $H(\mathbf{x}, \mathbf{y}) = t$. Otherwise, the result D_H is just an element of $\mathbb{G}_T$. The sketch of the security proof is similar to that given in Section 3.

5. PRIVATE MOVIE RECOMMENDER SYSTEM

We apply the proposed attribute-based cryptosystem to a privacy preserving movie recommender system. The participants include a Key Authority that generates public encryption parameters and pre-decryption keys, a set of users who consent to store their movie ratings and recommendations in encrypted form on a database server, and a user who seeks to obtain recommendations based on his previous viewing preferences. The parties are assumed to be "honest-but-curious", i.e., they will follow the rules of the protocol, but will attempt to glean as much information as possible about data owned by the other parties during each step of the protocol.

5.1. Notation and Problem Statement

As shown in Fig. 1, let KA be the Key Authority, S be the database server, $U_i, i = 1, 2, ..., M$ be the M users who provide encrypted movie ratings and recommendations to a customer, Bob. Let the $\mathbf{x}^{(i)}$ be the n-length movie ratings vector of the i^{th} user, and $\mathbf{y}$ be the n-length movie ratings vector of Bob. Since movie raters typically assign a 1-to-5 star rating, let $x_j^{(i)}, y_j \in \{1, 2, 3, 4, 5\}$ for $j = 1, 2, ..., n$. For simplicity of exposition, assume that all the participants have watched

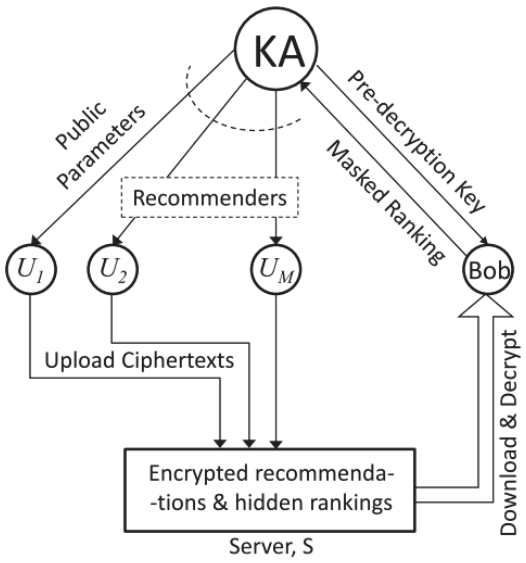

Fig. 1. Setup of a privacy-preserving movie recommender system using the proposed ABE system, showing the directions of data-flow among the various participants.

all n movies indexed by $j = 1, 2, ..., n$, i.e., they have a valid rating for each movie. Let m_i be the movie recommended by user i. We assume that each user recommends a single movie, but this is not a binding requirement; the system easily extends to the case of a vector of movie recommendations. The goal is for Bob to obtain recommendations from users who satisfy the ℓ_2 distance condition on the attributes, i.e., that $\|\mathbf{x}^{(i)} - \mathbf{y}\|_2^2 \leq \tau$. For users that do not satisfy this condition, Bob should obtain neither the rating vectors nor the recommendations. We utilize the ABE cryptosystem of Section 3.

5.2. A Movie Recommendation Protocol

Setup: The Key Authority selects large prime numbers and cyclic groups of composite order as explained earlier. It generates the public parameters and a private master key.

Encryption of Ratings & Recommendations: Each user, U_i performs the function of Alice, i.e., it generates a ciphertext of the form of (1) using its message $m = m_i$ and attribute vector $\mathbf{x} = \mathbf{x}^{(i)}$. The ciphertexts are stored on the server without identifying the users. This is accomplished, for example, by the server randomly permuting the user indices.

Decryption Key Generation for Bob: Using his movie rating (attribute) vector $\mathbf{y}$ and a randomly chosen masking vector $\mathbf{z}$, Bob interacts with the Key Authority to obtain decryption keys given by (2), (6) and (7).

Decryption: Bob downloads the ciphertext of the users U_i. Then, for every $i \in \{1, 2, ..., M\}$ and for every

$t \in \{0, 1, ..., \tau\}$, Bob attempts to decrypt m_i. According to the proof of correctness in Section 3, he succeeds only if the ℓ_2 distance criterion on the rating vectors is satisfied, i.e., he obtains a movie recommendation only from those users whose ratings for the n previously watched movies are τ-close to his own ratings.

Salient features: A noteworthy difference between the proposed protocol and those employing homomorphic functions, e.g., [10], is that encryption does not have to be repeated for a new user. Any user can just download the ciphertexts from the server and try to decrypt them, accessing the recommendations only from users whose ratings are τ-close to his own. In other words, every user U_i can broadcast his ciphertext, assured of the fact that (a) his exact movie preferences, given by the rating vector will not be revealed to others, not even to those users U_j for which $\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|_2^2 \leq \tau$, and (b) his recommendations will be anonymously revealed to the users with similar preferences. Another attractive feature is that the protocol does not require multiple rounds of communication; there is one setup phase, one encryption phase and a download-and-decrypt phase. By design, Bob does not reveal his movie preferences to the Key Authority or to any of the recommenders. Furthermore, if one more movie is appended to the rating vector, say x_{n+1}, a user can easily modify his ciphertext in (1) by uploading to the server $(g_q c)^{-\gamma x_{n+1}} s_{n+1}$ to update A_{n+1} and $(g_q c)^{-\gamma x_{n+1}^2} s_0$ to update $\bar{A}$.

Limitations: The main limitation of this protocol is that, Bob must carry out $O(\tau)$ decryptions per user, one decryption for each $t = 0, 1, ..., \tau$. This is a direct consequence of the way the ciphertext is designed, i.e., the way in which the message m in (1) is unmasked if the ℓ_2 distance equals t for some $t \in \{0, 1, ..., \tau\}$. This reveals the distance between the attributes and places a limitation on the value of τ or more generally on the range of distances that can be tested. It would be preferable if the threshold condition, $\|\mathbf{x}^{(i)} - \mathbf{y}\|_2^2 \lesseqgtr \tau$, could be tested over all $t \in [0, \tau]$ using only one decryption. For the Hamming distance metric, this is achievable but with a vastly larger ciphertext [15]. In our protocol, decryption complexity of Bob can be reduced by evaluating (10) and (11) only once, as the expressions do not depend on t. However, incorporating a more efficient way to test the threshold condition on the attributes is an interesting avenue for future research.

6. CONCLUSIONS

We presented an attribute-based cryptosystem in which decryption is possible if and only if the decryptor's attribute vector is within a specified ℓ_2 distance from the encryptor's attribute vector. We presented a variant of the system for binary attributes satisfying a Hamming distance criterion. The decryption algorithm recovers the encrypted message using properties of bilinear maps on cyclic groups of composite order. Using this cryptosystem, a privacy-preserving movie recommender system was constructed using which a customer exploits the ℓ_2 distance condition to recover movie recommendations from other customers who have similar viewing preferences. Different from previous attribute-based cryptosystems, the customer can use additive masking to obtain a decryption key without revealing his attribute to the key authority. The proposed cryptosystem affords an implementation of privacy preserving querying with a "encrypt once, decrypt many times" feature. This is an advantage over two-party secure querying based on homomorphic cryptosystems, where encryption must necessarily be repeated using the unique public key of a particular querying client.

7. REFERENCES

[1] R. Cramer, "Introduction to secure computation," *Lectures on Data Security - Modern Cryptology in Theory and Practice*, vol. 1561, pp. 16–62, March 1999.

[2] A. C-C. Yao, "How to Generate and Exchange Secrets," in *Proc. 27th Annual Symposium on Foundations of Computer Science (SFCS)*, Washington, DC, USA, 1986, pp. 162–167, IEEE Computer Society.

[3] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *Advances in Cryptology, EUROCRYPT 99*. 1999, vol. 1592, pp. 233–238, Springer-Verlag, LNCS.

[4] I. Damgård and M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System," in *4th International Workshop on Practice and Theory in Public Key Cryptosystems*, Cheju Island, Korea, Feb. 2001, pp. 119–136.

[5] T. El Gamal, "A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 4, pp. 469–472, Jul. 1985.

[6] C. Gentry, "Computing arbitrary functions of encrypted data," *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, March 2010.

[7] M. Shaneck, Y. Kim, and V. Kumar, "Privacy preserving nearest neighbor search," in *Proc. of the Sixth IEEE Intl. Conf. Data Mining - Workshops*, Washington, DC, USA, 2006, pp. 541–545.

[8] Y. Qi and M. Atallah, "Efficient privacy-preserving k-nearest neighbor search," *Intl. Conference on Distributed Computing Systems*, vol. 0, pp. 311–319, 2008.

[9] M. Atallah and J. Li, "Secure outsourcing of sequence comparisons," *International Journal of Information Security*, vol. 4, no. 4, pp. 277–287, Oct. 2005.

[10] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, "Efficiently Computing Private Recommendations," in *International Conference on Acoustic, Speech and Signal Processing-ICASSP*, Prague, Czech Republic, May 2011, pp. 5864–5867.

[11] A. Sahai and B. Waters, "Fuzzy Identity Based Encryption," in *Proc. Eurocrypt'05*, Aarhus, Denmark, May 2005, pp. 457–473.

[12] J. Katz, A. Sahai, and B. Waters, "Predicate Encryption Supporting Disjunction, Polynomial Equations, and Inner Products," in *Proc. Eurocrypt'08*, Istanbul, Turkey, April 2008, pp. 146–162.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT*, 2010, pp. 62–91.

[14] J. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, Springer-Verlag, 1986.

[15] D. Cheung, N. Mamoulis, W. Wong, S.-M. Yiu, and Y. Zhang, "Anonymous fuzzy identity-based encryption for similarity search," in *ISAAC (1)*, 2010, pp. 61–72.