SCIS 2012 The 29th Symposium on
Cryptography and Information Security
Kanazawa, Japan, Jan. 30 - Feb. 2, 2012
The Institute of Electronics,
Information and Communication Engineers

A Private Matching Algorithm Using Predicate Encryption

Hermanto * Yoshifumi Manabe † Tatsuaki Okamoto ‡

Abstract— Matching algorithms are most commonly explained using the state of proposals between two groups such as male and female where each participant creates his/her own preference list by knowing some information regarding the other participant. However, the history of the proposals during the execution of the algorithm is known to everyone, which makes the stable matching algorithm vulnerable to manipulation.

Private matching algorithms were proposed to output a matching result without revealing any information and show only the matching result to all participants. Still, in these algorithms, both parties need to know some information regarding participants or reveal some of their own attributes before the matching begins.

To resolve the problem, we propose a new private matching algorithm using a predicate encryption, in which each male participant uses his own information and the information demanded from female participants as input, while each female participant uses her condition/demanded information as input for matching. If the inputs from both participants are satisfied each other, then the algorithm outputs the results of the matching only to those who are paired/matched.

Keywords: private matching algorithm, predicate encryption, attribute hiding, inner product.

1 Introduction

Matching algorithms are often used for matching medical students to their first hospital assignments, employers who want to find jobs in companies, and couples.

In 1962, Gale-Shapley [GS62] presented a novel framework for solving the stable matching problem. The algorithm itself is very simple, by using the lists of preference from male and female to find a stable matching result. However, in the Gale-Shapley stable matching algorithm, the history of the engagements during the execution of the algorithm is known to everyone, which negates the privacy of each rejected person and leaves the history open to all participants. This leads that the stable matching algorithm is vulnerable to manipulation [GI89]. Under certain circumstances, participants with the knowledge of the preference lists of other participants have incentives to change their own true preference list. In order to output a stable matching result without revealing any information, we need a secure protocol that shows only the matching result to all participants, which is called a private stable matching algorithm. By adopting the Gale-Shapley algorithm, Golle [Go06] and Franklin-
Gondree-Mohassel [FGM07] presented their own variant of a private stable matching algorithm.

However, in these matching algorithms [GS62] [Go06] [FGM07], there is one big problem in privacy. In order to make the preference lists, participants need to know some information about the other participants. Therefore, both parties need to reveal some of their own attribute before the matching begins. In this paper, we propose a new private matching algorithm in which both parties can find their matching result without showing their own attributes or conditions to any other participants.

In our algorithm, in finding the best mate for male and female, each male is required to input his attribute, for example, "Nationality", "Job", "Salary", "Hobby", etc; while each female is required to input her condition of male to be paired. At the same time, each female can input her attribute while each male is required to input her condition of female to be paired. E.g male A_3 has attributes "Nationality = Japan", "Salary = 4 million yen" , and "Hobby = Golf AND Travel ". On the other hand, female B_2 has the condition "Nationality = Japan OR USA", "Salary = more than 3 million yen" and "Hobby = Travel". In this example, male A_3 will be automatically coupled with female B_2.

In our matching algorithm, the conditions and attributes are represented by vectors. Below is an example of vector conversion. Considering all nationality that is demanded by female is "Japan", "USA" and "China" then the attribute of a male participant is Japan , then the attribute vector becomes $\vec{x_i}$ = (1 0 0), the attribute of a male participant is USA, then the attribute vector becomes $\vec{x_i}$ = (0 1 0), the attribute of a male participant is USA, then the attribute vector becomes $\vec{x_i}$ = (0 0 1). On the other hand, the condition of a female participant is (Japan OR USA) then

* Department of Social Informatics, Graduate School of Informatics, Kyoto University.
† NTT Communication Science Laboratories 2-4 Hikaridai, Seikacho, Souraku District, Kyoto, Japan
‡ NTT Information Sharing Platform Laboratories 3-9-11 Midoricho, Musashino city, Tokyo, Japan

the condition vector becomes $\vec{v_j}$ = (0 0 1). By doing this, when the attributes and conditions match, the inner product of attribute vector and condition vector outputs 0.

For the attributes and conditions of salary, salary of male participants and female participants are grouped like the following which depend on input of all participants.
$x_1 = 1$: Salary below 3 million.
$x_2 = 1$: Salary between 3 million and 6 million.
$x_3 = 1$: Salary between 6 million and 10 million.
$x_4 = 1$: Salary above 10 million.
For male participants with salary 2 million, then the attribute vector becomes $\vec{x_i}$ = (1 0 0 0), while the condition of a female participant is salary more than 3 million (a_2 OR a_3 OR a_4), then the condition vector becomes $\vec{v_j}$ = (1 0 0 0).

It is crucial if the conditions and attributes of each participants in the matching are exposed to everyone. That is why, we define a private matching algorithm that only output nothing more than the matching result.

We create a new algorithm that every participants can find his/her partner only when their input satisfy/match each other by adopting to Okamoto-Takashima's inner product encryption (IPE)[OT11]

In the later section, we introduce a new definition of matching algorithm in which the participants can input information instead of preference lists.

2 Related works

2.1 The Gale-Shapley Algorithm

The Gale-Shapley algorithm was proposed for solving the matching problem. In this algorithm, male and female play different roles. Simply, the algorithm can be expressed as "proposals" from male to female.

Male and female are divided into two groups, the *engaged* group and the *free* group. All male and female start from the free group. Whenever there is any male in the free group, one of them is randomly selected. Then, the selected free male proposes to the female whom he likes the most and has never proposed to before. Suppose the selected free male A proposes to B (whom he never proposed before and he likes the most). In this state there are two cases. The first case is when the female B is free. In this case, A and B are automatically paired. The second case is when the female B is already engaged to another male A'. In this case, using the female B's preference list, A and A' are compared. When A has a higher rank than A', B engages with A, while A' is divorced and grouped into the free group. On the other hand, if A has a lower rank than A', B will continue to be engaged to A', while A is considered to be rejected and regrouped again into the free group. Next, another male from the free group is randomly chosen and matched by the same way. This manner of matching repeats until all male and female are paired and no person remains in the free group. In the end, Gale-Shapley algorithm outputs a stable matching result. A matching is called stable, when there is no unmatched male and female that like each other better than their own current match.

2.2 Golle's Private Matching Algorithm

In order to retain the privacy of the participants' information, this algorithm [Go06] added independent parties called Matching Authorities (MAs). By setting the *MAs* as *honest but curious* parties, participants obtain the matching result without knowing any other information. In addition to real male and female participants, fake male are added in the matching algorithm. In this algorithm, each participant creates his/her own preference list and encrypts it. The encrypted preference list is sent to matching authorities. Matching is done by matching authorities without decrypting the lists and the result is shown to all participants. The total communication complexity of this private stable matching is $O(n^3)$.

Franklin-Gondree-Mohassel's Private Matching Algorithm

Franklin-Gondree-Mohassel showed that communication complexity of Golle's main protocol is $O(n^5)$. In addition, they also introduced their own variant protocol in which the communication complexity is reduced. As participants, fake female are added to Golle's private matching algorithm (real male, real female, and fake male). In this algorithm, each participant creates his/her preference list, encrypts it. Different from Golle's algorithm, the encrypted preference lists are sent to Database matching Authority (DBMA) which has a function to store all encrypted preference lists. Matching authorities (MAs) can only access the encrypted preference lists when matching is done. The result is then shown to all participants. The computation complexity of their private matching algorithm is $O(n^4\sqrt{\log n})$, while the number of rounds of computation is $2n^2$. We also found that number of rounds of computation in this algorithm can be simplified by changing the preference lists of fake male and fake female to $n^2 + n$ [HMO11].

2.3 Attribute based Encrytion

The application of attribute based encryption [BSW07] is mostly known in cloud computing. As more and more data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. The benefit of attribute based encryption is that we can share the encypted data with other parties, just by simply share our private key. Mainly, there are 2 types of Attribute based encryption, Key-Policy Attribute based Encryption (KP-ABE) and Ciphertext-Policy Attribute based encryption (CP-ABE) used. The primitives of attribute based encryption is also the basic for predicate encryption.

2.4 Predicate Encryption

Predicate encryption [LOS10] [OT11] is a new paradigm generalizing identity-based encryption and attribute based encryption[BSW07]. Different from the attribute-based encryption, in predicate encryption schemes, secret keys correspond to predicates and ciphertexts are associated with attributes; the secret key SK_f corresponding to a predicate f can be used to decrypt a ciphertext associated with attribute I if and only if $f(I) = 1$. Constructions of such schemes are currently known for relatively few classes of predicates. Such a scheme for predicates corresponding to the evaluation of inner products over Z_N, which enables constructions in which predicates correspond to the evaluation of disjunctions, polynomials, CNF/DNF formula, or threshold predicates, etc.

3 Definition of our matching algorithms

In this section, we explain our new matching algorithms.

3.1 Models and definition

Other than participants, our algorithm is run by numbers of independent parties (exactly four) whom we call matching authorities (MA). Participants are divided into two groups, the one with attributes, and the other with conditions. In this algorithm we will set the group with attributes as male group, while the group with conditions as female group. Each attribute correspond to a vector $\vec{x}$, condition correspond to a vector $\vec{v}$.

First of all, the participants of the matching input each of their attributes and conditions. Then the matching authorities separately run a number of distributed cryptographic protocols, which are public keys and master secret keys generation (Setup) and secret key generation (KeyGen). Next, by using the public key from the matching authorities, the male participants encrypt their attributes and display the encrypted attributes in a board that is open to all participants. Then the female participants send their input to the matching authorities and receive secret keys regarding to their own conditions.

After that, female participant obtains the encrypted attribute from the open list and decrypt them with their own secret keys. If there is a pair of a male participant and a female participant whose attributes and conditions' inner product is equal to 0, then the female can decrypt the ciphertext. The female can read the message and knows which male matches her condition. Notice that in this algorithm, the result of the matching is only known by matched female participant.

Our matching algorithm for computing a match might output more than one matching results if there are multiple male participants whose attributes matches to a female participant's conditions. This matching algorithm reveals no other information to the adversary than what the adversary can learn from that match including the conditions and attributes of the participants in the matching.

3.2 Privacy Preserving Protocol

Notations

Let i be an index of male; $i \in \{1, \ldots, n\}$. Let j be an index of female; $j \in \{1, \ldots, n\}$ (n represents the maximum number of parcipants in male/female group). Let r in MA_r be an index of matching authorities, where in our algorithm, we have 4 matching authorities.

Input submission and initialization

Matching is executed as follows :
1. Each male inputs his own attribute $\vec{x_i}$ and each female inputs her own condition $\vec{v_j}$.
2. Matching Authorities (MA_1, MA_2, MA_3, MA_4) generate each public key and each master secret key.
3. Participants divide their own input as below. Each male encrypts the divided attributes using matching authorities' public keys and outputs encrypted attributes. These encrypted attributes are shown as an open list that all participants can see.
4. Each female divides her condition and sends them to matching authorities.
5. Matching authorities receive the divided conditions and generate a secret key corresponding to each condition as below.
6. Each female receives secret keys corresponding with her own condition, and then use the encrypted attributes from the open list to find a matching result.

Dividing an attribute or condition

Given an attribute $\vec{x_i}$, each male creates random vector $\vec{x_{i_1}}$ and divides $\vec{x_i}$ into $\vec{x_{i_1}}$ and $\vec{x_{i_2}}$ by $\vec{x_{i_2}} = \vec{x_i} - \vec{x_{i_1}}$. Given a condition $\vec{v_j}$, each female creates random vector $\vec{v_{j_1}}$ and divides $\vec{v_j}$ into $\vec{v_{j_1}}$ and $\vec{v_{j_2}}$ by $\vec{v_{j_2}} = \vec{v_j} - \vec{v_{j_1}}$.

Encrypting attribute and message

Every male participant encrypts each of their divided attributes in a way such that $\vec{x_{i_1}}$ is encrypted using MA_1 and MA_2's public keys which create $C_{i_{11}}$ and $C_{i_{12}}$ respectively; $\vec{x_{i_2}}$ is encrypted using MA_3 and MA_4's public keys which create $C_{i_{21}}$ and $C_{i_{22}}$ respectively. Other than these, male participants also encrypt their message by using matching authorities' public keys to create C_{i_0}. Note that *message* is the detail information of a male, such as E-mail address, telephone number, etc, which can only be decrypted when the male is matched with a female. Then, each male outputs $(C_{i_0}, C_{i_{11}}, C_{i_{12}}.C_{i_{21}}, C_{i_{22}})$.

Creating secret keys from conditions

Female participant sends $\vec{v_{j_1}}$ to MA_1 and MA_3. Then female participant receives secret key $K_{j_{11}}$ from MA_1, secret key $K_{j_{12}}$ from MA_3 Again, female participant sends $\vec{v_{j_2}}$ to MA_2 and MA_4. Then female participant receives secret key $K_{j_{21}}$ from MA_2, secret key $K_{j_{22}}$

from MA_4. Then, each female's secret key consists of $(K_{j_{11}}, K_{j_{12}}, K_{j_{21}}, K_{j_{22}})$.

Decrypting the encrypted attributes with secret keys

Female participants takes all the encrypted attributes and messages $(C_{i_0}.C_{i_{11}}, C_{i_{12}}.C_{i_{21}}, C_{i_{22}})$ from the open list and decrypt them with their own corresponding secret keys $(K_{j_{11}}, K_{j_{12}}.K_{j_{21}}, K_{j_{22}})$. If the inner product of the attribute and the condition is 0, then the decryption can be done. Notice that, the decrypted message of the matched male is visible only to the matched female.

3.3 Matching scheme

Algorithm are given as follows:

•**Setup$_\mathbf{r}$**$(1^\tau, n)$: Done by matching authorities (MA_r)
Notice that $\mathbf{B_r}$, $\mathbf{X_r}$ are generated in each MA_r, while $\mathbf{A}$ is the same for all MA_r.
G and G_T are cyclic groups of order q.
Generator $g_1 \xleftarrow{U} G, N = 4n + 2$
$g_T := e(g_1, g_1) \neq 1 \in G_T$
$a_k := (\overbrace{0, \ldots, 0}^{\text{k-1}}, g_1, \overbrace{0, \ldots, 0}^{\text{N-k}})$
canonical base $\mathbf{A} = (a_1, \ldots, a_N)$
$X_r := (X_{r_{ij}}) \xleftarrow{U} GL(N, \mathbf{F_q})$
$GL(N, \mathbf{F_q})$ is the group of $N \times N$ invertible matrices with entries in the field of q elements, $\mathbf{F_q}$ is a finite group of order q.
$G_{ob}(1^\lambda, 4n + 2) : param_V^{'} := (q, \mathbf{V}, \mathbf{G_T}, \mathbf{A}, e) \xleftarrow{R} G_{dpvs}(1^\lambda, N), \psi \xleftarrow{U} F_q^x, \vartheta_{i,j} := \psi(\mathbf{X^T})^{-1}, g_T := e(g_i, g_i)^\psi, param_V := (param_V^{'}, g_T)$,
$return(param_V, \mathbf{B}, \mathbf{B}^*)$.

$(param_V, \mathbf{B} := (b_{r_0}, \ldots, b_{r_{4n+1}}), \mathbf{B}^* := (b^*_{r_0}, \ldots, b^*_{r_{4n+1}})) \xleftarrow{R} G_{ob}(1^\lambda, 4n + 2)$
$pk_r := \hat{\mathbf{B}}_\mathbf{r} := (b_{r_0}, \ldots, b_{r_n}, b_{r_{4n+1}})$,
$sk_r := \hat{\mathbf{B}}^*_\mathbf{r} := (b^*_{r_0}, \ldots, b^*_{r_n}, b^*_{r_{3n+1}}, \ldots, b^*_{r_{4n+1}})$
where $\mathbf{B_r} := \mathbf{X_r} \cdot \mathbf{A}$; $\mathbf{B^*_r} := (X_r^T)^{-1} \cdot \mathbf{A}$;

$$\begin{pmatrix} b_{r_0} \\ \cdot \\ \cdot \\ b_{r_{4n+1}} \end{pmatrix} = \begin{pmatrix} X_{r_{0,0}} & \ldots & X_{r_{0,4n+1}} \\ \cdot & \ldots & \cdot \\ \cdot & \ldots & \cdot \\ \cdot & \ldots & \cdot \\ X_{r_{4n+1,0}} & \ldots & X_{r_{4n+1,4n+1}} \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_N \end{pmatrix}.$$

$\mathbf{V}$ is a N-dimensional vector space;
$\mathbf{V} := \overbrace{\mathbf{G}X \ldots XG}^{\text{N}}$ over $\mathbf{F_q}$; e : $\mathbf{V}$ X $\mathbf{V} \to G_T$
$e(x, y) := \Pi_{i=1}^N e(g_{1_i}, h_{1_i}) \in G_T$ where
$x = (g_{1_1}, \ldots, g_{1_N}) \in \mathbf{V}$ and $y := (h_{1_1}, \ldots, h_{1_N}) \in \mathbf{V}$.
This is nondegenerate bilinear i.e. $e(sx, ty) = e(x, y)^{st}$ and if $e(x, y) = 1$ for all $y \in \mathbf{V}$, then $x = (0, \ldots, 0)$. For all i and j, $e(a_i, a_j) = e(g_1, g_1)^{\delta_{i,j}}$ where $\delta_{i,j} = 1$ if $i = j$, and 0 otherwise, and $e(g_1, g_1) \neq 1 \in \mathbf{G_T}$

•**KeyGen$_\mathbf{r}$** : Done by matching authorities (MA_r)
$\sigma(\sigma \xleftarrow{U} \mathbf{F_q})$ is used by all matching authorities which is acquired from either MA_0 (Special Matching Authority) or a hash function, whose descriptions are written below.
$\eta_r \xleftarrow{U} \mathbf{F_q}$.
a. For MA_1 ; Input : $(sk_1, \vec{v_{j1}})$:
Output : $(K_{j_{11}})$

$$K_{j_{11}} := (\overbrace{1}^{1}, \overbrace{\sigma\vec{v_{j1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_1}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_1^*}}$$

b. For MA_2 ; Input : $(sk_2, \vec{v_{j2}})$:
Output : $(K_{j_{12}})$

$$K_{j_{12}} := (\overbrace{1}^{1}, \overbrace{\sigma\vec{v_{j2}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_2}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_2^*}}$$

c. For MA_3 ; Input : $(sk_3, \vec{v_{j1}})$:
Output : $(K_{j_{21}})$

$$K_{j_{21}} := (\overbrace{1}^{1}, \overbrace{\sigma\vec{v_{j1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_3}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_3^*}}$$

d. For MA_4 ; Input : $(sk_4, \vec{v_{j2}})$:
Output : $(K_{j_{22}})$

$$K_{j_{22}} := (\overbrace{1}^{1}, \overbrace{\sigma\vec{v_{j2}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_4}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_4^*}}$$

where $\vec{v_{j1}} = (v_{j1_1}, \ldots, v_{j1_n})$, $\vec{v_{j2}} = (v_{j2_1}, \ldots, v_{j2_n})$,.
For bases $\mathbf{B} := (b_0, \ldots, b_N)$ and $\mathbf{B}^* := (b_0^*, \ldots, b_N^*), (x_0, \ldots, x_N)_\mathbf{B} := \Sigma_{i=1}^N x_i b_i$ and $(v_0, \ldots, v_N)_{\mathbf{B}^*} := \Sigma_{i=1}^N v_i b_i^*$.
and $(\ldots)_{B_i^*}$ represents elements in B_i base and $B_i^* base$.

• **Encryption :** Done by a male participant.
Male participant generates $\delta_1, \delta_{11}, \delta_{12}, \delta_{21}, \delta_{22}, \zeta_{11}, \zeta_{12}, \zeta_{21}, \zeta_{22} \xleftarrow{U} \mathbf{F_q}$
Input : $(pk_1, pk_2, pk_3, pk_4, \vec{x_{i_1}}, \vec{x_{i_2}}, m \in G_T)$
Output : $(C_{i_0}, C_{i_{11}}, C_{i_{12}}, C_{i_{21}}, C_{i_{22}})$

$$C_{i_0} = g_T^{\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22}} m$$

$$C_{i_{11}} := (\overbrace{\zeta_{11}}^{1}, \overbrace{\delta_1\vec{x_{i1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{0^n}^{\text{n}}, \overbrace{\delta_{11}}^{1})_{\mathbf{B_1}}$$

$$C_{i_{12}} := (\overbrace{\zeta_{12}}^{1}, \overbrace{\delta_1\vec{x_{i1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{0^n}^{\text{n}}, \overbrace{\delta_{12}}^{1})_{\mathbf{B_2}}$$

$$C_{i_{21}} := (\overbrace{\zeta_{21}}^{1}, \overbrace{\delta_1\vec{x_{i2}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{0^n}^{\text{n}}, \overbrace{\delta_{21}}^{1})_{\mathbf{B_3}}$$

$$C_{i_{22}} := (\overbrace{\zeta_{22}}^{1}, \overbrace{\delta_1\vec{x_{i2}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{0^n}^{\text{n}}, \overbrace{\delta_{22}}^{1})_{\mathbf{B_4}}$$

• **Decryption :** Done by a female participant

$$m := \frac{C_{i_0}}{\Pi_{y=1}^2 \Pi_{z=1}^2 e(C_{i_{yz}}, K_{j_{yz}})}$$

if $\vec{v_j} \cdot \vec{x_i} = 0$, then

$$\frac{C_{i_0}}{\Pi_{y=1}^{2}\Pi_{z=1}^{2} e(C_{i_{yz}}, K_{j_{yz}})}$$

$$= \frac{g_T^{\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22}} m}{g_T^{(\vec{v}\cdot\vec{x}\delta_1\sigma+\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22})}}$$

$$= \frac{g_T^{\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22}} m}{g_T^{\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22}}} = m$$

where $e(b_{m_i}, b^*_{m_j}) = g_T^{\delta_{i,j}}$.
If $i = j$ then $\delta_{i,j} = 1$; else $\delta_{i,j} = 0$.

As for σ in above, it is important that the value of σ is the same for each matching authority. In order to achieve it, one of the following two protocols is executed.

1. By using MA_0

First, MA_0 generates a pair of public key, secret key and shows his public key to all MA_r. Matching authority (MA_0) has a function to output a $Token$, by creating random σ and then using each participants' *identity*, MA_0 adds his own signature ($Sign$) to the $Token$, and then send the $Token$ to matching authorities MA_r. MA_r verifies the signature and then uses σ to generate $(K_{j_{11}}, K_{j_{12}}.K_{j_{21}}, K_{j_{22}})$

Input : Participants' ID (ID)

Output : $Token = (\sigma, ID)_{Sign}$

2. By using hash function

By using the same hash function, matching authorities (MA_1, MA_2, MA_3, MA_4) uses the same σ to create g_1^{σ}.

Input : Participants' ID (ID)

Output : $g_1^{\sigma} = H(ID)$

where H() is a hash function.

Notice that by using hash function, the scheme $KeyGen_r$ is different from the procedure shown before. We propose a new way to generate the secret keys as below.

canonical base $\mathbf{A'} = (a_1, a_2^{'}, \ldots, a_{n+1}^{'}, a_{n+2}, \ldots, a_N)$

where $a'_k := (\overbrace{0, \ldots, 0}^{\text{k-1}}, g_1^{\sigma}, \overbrace{0, \ldots, 0}^{\text{N-k}})$.

and $sk_r^{'} : \mathbf{B_r^{*'}} := (\mathbf{X_r^T})^{-1} \cdot \mathbf{A'}$,

In this term the secret keys become like below

$$K^{`} := (\overbrace{1}^{1}, \overbrace{\vec{v_{j1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_1}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_1^*}}$$

$$:= (\overbrace{1}^{1}, \overbrace{\sigma\vec{v_{j1}}}^{\text{n}}, \overbrace{0^{2n}}^{\text{2n}}, \overbrace{\eta_1}^{\text{n}}, \overbrace{0}^{1})_{\mathbf{B_1'^*}}$$

= K

4 Security

We propose security definitions from various points of views. One of the point of views is the privacy of attributes of the male participants. The other point of view is privacy of conditions of the female participants.

4.1 Security of inner product encryption (IPE).

Definition 1 Inner-product encryption scheme is adaptively attribute-hiding against chosen plaintext attacks if for all probabilistic polynomial-time adversaries Adv, the advantage of Adv in the following experiment is negligible in the security parameter.

Below is the game between Adv and Challenger.

1. Setup is run to generate keys pk and sk, and pk is given to Adv.
2. Adv may adaptively make a polynomial number of key queries for predicate vector $\vec{v}$. In response, Adv is given the corresponding key $sk_{\vec{v}} \xleftarrow{R} KeyGen(sk, \vec{v})$
3. Adv outputs challenge attribute vector $(\vec{x}^{(0)}, \vec{x}^{(1)})$ and challenge plaintexts $(m^{(0)}, m^{(1)})$, subject to the following restrictions :

a. $\vec{v} \cdot \vec{x}^{(0)} \neq 0$ and $\vec{v} \cdot \vec{x}^{(1)} \neq 0$ for all the key queried predicate vectors $\vec{v}$.

b. Two challenge plaintexts are equal ($m^{(0)} = m^{(1)}$) and any key query $\vec{v}$ satisfies one of the following conditions.

i. $\vec{v} \cdot \vec{x}^{(0)} = 0$ and $\vec{v} \cdot \vec{x}^{(1)} = 0$,

ii. $\vec{v} \cdot \vec{x}^{(0)} \neq 0$ and $\vec{v} \cdot \vec{x}^{(1)} \neq 0$.

4. A random bit b is chosen. Adv is given $c^{(b)} \xleftarrow{R} Enc(pk, m^{(b)}, \vec{x}^{(b)})$.
5. Adv may continue to issue key queries for additional predicate vectors $\vec{v}$, subject to the restriction given at step 3. Adv is given the corresponding key $sk_{\vec{v}} \xleftarrow{R} KeyGen(sk, \vec{v})$.
6. Adv outputs a bit $b^{'}$, and succeeds if $b^{'} = b$.

The advantage of Adv is described as

$$Adv^{IPE}(\tau) = Pr[b' = b] - 1/2$$

Lemma 1 The proposed IPE Scheme is adaptively attribute-hiding against chosen plaintext attacks under the DLIN assumption. [OT11]

4.2 Privacy for male participants

The first point of view is from the ciphertexts, where it is hard to guess the message and attribute for any adversary including matching authorities if there is no collusion among mathching authorities.

Definition 2 This protocol is said to be private for male participants if for all probabilistic polynomial-time adversaries Adv, the advantage of Adv in the following game is negligible.

In this game, we consider MA_1 is Adv.

1. Challenger sends pk_2, pk_3, pk_4. to Adv, while Adv sends pk_1 to the Challenger.
2. Adv sends $\vec{x}^{(0)}, \vec{x}^{(1)}$ and $m^{(0)}, m^{(1)}$ to the Challenger.
3. A random bit is chosen, Challenger split $\vec{x}^{(b)}$ into $\vec{x}_1^{(b)}$ and $\vec{x}_2^{(b)}$. $\vec{x}_1^{(b)}$ is chosen random while $\vec{x}_2^{(b)}$ is created by $\vec{x}^{(b)} - \vec{x}_1^{(b)}$. Adv is given $(C_0, C_{11}, C_{12}, C_{21}, C_{22}) \leftarrow Enc(pk_1, pk_2, pk_3, pk_4, \vec{x}_1^{(b)}, \vec{x}_2^{(b)}, m^{(b)})$.
4. Adv outputs a bit $b^{'}$, and succeeds if $b^{'} = b$.

Note that we write the separated vector $\vec{x}_{i_1}$, $\vec{x}_{i_2}$ as $\vec{x}_1^{(b)}$, $\vec{x}_2^{(b)}$ and ciphertext $(C_{i_0}, C_{i_{11}}, C_{i_{12}}.C_{i_{21}}, C_{i_{22}})$ as $(C_0, C_{11}, C_{12}, C_{21}, C_{22})$ in this game.

The advantage of Adv is described as

$$Adv^{CT}(\tau) = Pr[b' = b] - 1/2$$

Lemma 2 The proposed protocol is private for male participants under the DLIN assumption.

Proof sketch of Lemma 2

We denote the game written in Definition 2 as Game 0. Then, we define Game 1 to compare with Game 0. In Game 1, the ciphertext created from Challenger at step 4 is changed in some ways so that it is hard to guess which random bit is chosen. The detail of Game 1 is given below.

Game 1: Same like Game 0 except that C_{21} and C_{22} in ciphertext are changed like the following.

$$C_0 := g_T^{\zeta_{11}+\zeta_{12}+\zeta_{21}+\zeta_{22}} m$$

$$C_{11} := (\zeta_{11}, \delta_1 \vec{x_1}, 0^{2n}, 0^n, \delta_{11})_{\mathbf{B_1}}$$

$$C_{12} := (\zeta_{12}, \delta_1 \vec{x_1}, 0^{2n}, 0^n, \delta_{12})_{\mathbf{B_2}}$$

$$C_{21} := (\zeta_{21}, \boxed{\vec{x_{21}}}, 0^{2n}, 0^n, \delta_{21})_{\mathbf{B_3}}$$

$$C_{22} := (\zeta_{22}, \boxed{\vec{x_{22}}}, 0^{2n}, 0^n, \delta_{22})_{\mathbf{B_4}}$$

where $\vec{x_{21}}$ and $\vec{x_{22}}$ are created randomly and independently from the other variables. We can show that the difference between advantages of Game 0 and Game 1 is negligible, under the DLIN assumption.

$$Adv^{CT}(\tau) = Adv^{Game0}(\tau)$$
$$|Adv^{Game0}(\tau) - Adv^{Game1}|(\tau) < \epsilon(\tau)$$

Furthermore, the advantage of Adv in Game 1 is zero since $(C_0, C_{11}, C_{12}, C_{21}, C_{22})$ are independent from $\vec{x_b}$. Therefore, the advantage of Adv in Game 0 is negligible.

4.3 Privacy for female participants

The second point of view is the security of secret keys, i.e. when one of matching authorities is Adv, the matching authority will never know any condition from secret keys if there is no collusion among Matching Authorities.

Definition 3

This protocol is said to be private to female participants if the advantage of Adv in the following game is negligible.

In this game, we consider MA_1 is Adv.

1. Challenger sends pk_2, pk_3, pk_4. to Adv, while Adv sends pk_1 to the Challenger.
2. Adv may adaptively make a polynomial number of key queries for predicate vector $\vec{v}$. In response, Adv is given parts of the corresponding key (K_{12}, K_{21}, K_{22}), $K_{12} \xleftarrow{R} KeyGen_2(sk_2, \vec{v})$, $K_{21} \xleftarrow{R} KeyGen_3(sk_3, \vec{v})$, $K_{22} \xleftarrow{R} KeyGen_4(sk_4, \vec{v})$.
3. Adv sends $\vec{v}^{(0)}, \vec{v}^{(1)}$ to the Challenger.
4. A random bit is chosen, Challenger split $\vec{v}^{(b)}$ into $\vec{v}_1^{(b)}$ and $\vec{v}_2^{(b)}$. $\vec{v}_1^{(b)}$ is chosen random while $\vec{v}_2^{(b)}$ is created by $\vec{v}^{(b)} - \vec{v}_1^{(b)}$. Adv is given $\vec{v}_1^{(b)}$.
5. Adv outputs a bit b', and succeeds if $b' = b$.

The advantage of Adv is described as

$$Adv^{SK}(\tau) = Pr[b' = b] - 1/2$$

Lemma 3 The proposed protocol is private for female participants under the DLIN assumption.

Proof of Lemma 3

Since the information of condition vector $\vec{v}^0$, $\vec{v}^1$ from the female participant is not included in the separated secret keys $\vec{v}_1^b$ and $\vec{v}_2^b$, it is hard for any Adv to get the information regarding conditions from female participants.

That is because $\vec{v}_1^b$ is created randomly by the female participant, therefore Adv will not be able to get any information regarding $\vec{v}$.

Therefore, considering if there is no collusion for matching authorities, then the algorithm we propose is private for female participants.

5 Conclusion

We proposed a new matching algorithm that both parties can execute a matching even without knowing any information about other participants by using predicate encryption. By using this algorithm, every participant can input his/her information to be matched without revealing any of his/her information.

References

[GS62] David Gale and Lloyd Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, Vol. 69, pp.:9-15, 1962.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *19th ACM Symposium on Theory of Computing*, pp.218-229, 1987.

[GI89] Dan Gusfield and Robert Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, pp.8-18, 1989.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In *Foundations of Computer Science*, pp.364-373, 1997.

[St98] Julien P. Stern. A new efficient all-or-nothing disclosure of secrets protocol. In *ASIACRYPT'98*, LNCS Vol. 1514 pp.357-371, 1998.

[JS99] Markus Jakobsson and Claus Peter Schnorr. Efficient oblivious proofs of correct exponentiation. In *Communications and Multimedia Security*, pp.71-86, 1999.

[Pa99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, LNCS Vol. 1592 pp.223-238, 1999.

[DJ01] Ivan Damgard and Mads Jurik. A generalization, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography*, LNCS Vol. 1992 pp.119-136, 2001.

[Ne01] C.Andrew Neff. A verifiable secret shuffle and its application to e-voting. In *8th ACM Computer and Communications Security*, pp116-125, 2001.

[CD01] Ronald Cramer and Ivan Damgard. Secure distributed linear algebra in a constant number of rounds. In *CRYPTO'01*, LNCS Vol. 2139 pp.143-202, 2001.

[FPS01] Pierre-Alain Fouque, G.Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. In *Financial Crypto*, LNCS Vol. 1962, pp.90-104, 2001.

[JJR02] Markus Jakobsson, Ari Juels, and Ron Rivest. Making mix nets robust for electronic voting by randomized partial checking. In *Proc. of USENIX'02*, pp.339-353, 2002.

[ACS02] Joy Algesheimer, Jan Camenisch, and Victor Shoup. Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In *CRYPTO'02*, LNCS Vol. 2442 pp.417-432, 2002.

[Li03] Helger Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *ASIACRYPT 2003*, LNCS Vol. 2894 pp.416-433, 2003.

[GJ04] Phillippe Golle and Ari Juels. Parallel mixing. In *11th ACM Computer and Communications Security*, pp.220-226, 2004.

[DFNT05] Ivan Damgard, Matthias Fitzi, Jesper Buus Nielsen, and Tomas Toft. How to split a shared secret into shared bits in constant-round. Cryptology ePrint Archive, Report 2005/140, 2005.

[Kl05] Eike Klitz. Unconditionally secure constant round multi-party computation for equality, comparison, bits and exponentiation. Cryptology ePrint Archieve, Report 2005/066, 2005.

[Go06] Phillipe Golle. A private stable matching algorithm. *Financial Crypto*, LNCS Vol. 4107, pp.65-80, 2006.

[FGM07] Matthew Franklin, Mark Gondree, and Payman Mohassel. Improved efficiency for private stable matching. *The Cryptographers' Track at the RSA Conference*, LNCS Vol 4377, pp.163-177, 2007.

[BSW07] John Bethencourt, Amit Sahai, Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 321-334, 2007.

[KSW08] Jonathan Katz, Amit Sahai, Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Smart, N.P (ed.) EUROCRYPT 2008. LNCS, vol 4965*, pp 146-162. Springer Heidelberg (2008)

[OT09] Tatsuaki Okamoto, Katsuyuki Takashima. Hierarchical predicate encryption for inner products. In *ASIACRYPT 2009*, Springer Heideberg (2009).

[LOS10] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, Brent Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Cryptology ePrint Archive*: Report 2010/110

[OT11] Tatsuaki Okamoto, Katsuyuki Takashima. Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption. In *Cryptology ePrint Archive*. Report 2011/543

[HMO11] Hermanto, Yoshifumi Manabe, Tatsuaki Okamoto. A Simplified Private Stable Matching Algorithm. In *Symposium on Cryptography and Information Security 2011*