PAPER Special Section on Knowledge-Based Software Engineering

Privacy-Aware Best-Balanced Multilingual Communication

Mondheera PITUXCOOSUVARN[†a)], *Nonmember*, **Takao NAKAGUCHI**[††], **Donghui LIN**[†], *Members*, *and* **Toru ISHIDA**[†††], *Fellow*

SUMMARY In machine translation (MT) mediated human-to-human communication, it is not an easy task to select the languages and translation services to be used as the users have various language backgrounds and skills. Our previous work introduced the best-balanced machine translation mechanism (BBMT) to automatically select the languages and translation services so as to equalize the language barriers of participants and to guarantee their equal opportunities in joining conversations. To assign proper languages to be used, however, the mechanism needs information of the participants' language skills, typically participants' language test scores. Since it is important to keep test score confidential, as well as other sensitive information, this paper introduces agents, which exchange encrypted information, and secure computation to ensure that agents can select the languages and translation services without destroying privacy. Our contribution is to introduce a multi-agent system with secure computation that can protect the privacy of users in multilingual communication. To our best knowledge, it is the first attempt to introduce multi-agent systems and secure computing to this area. The key idea is to model interactions among agents who deal with user's sensitive data, and to distribute calculation tasks to three different types of agents, together with data encryption, so no agent is able to access or recover participants' score.

key words: *secured implementation, user privacy, multilingual communication support*

1. Introduction

In multilingual communication, machine translation(MT) is a useful tool to overcome the language barrier. There are many MT services available with varied quality, but sometimes users' foreign language skill can yield better results in terms of communication if the MT quality is too low. For example, when there are two users and they have English as a shared language but with different proficiency, they can choose to use MT or English for communication. If the MT services they use have low quality, using English could be a better communication channel. When there are more users with various language skills, it becomes more difficult to decide what languages and which services to be used.

In 2018, we have already proposed a solution called best-balanced machine translation method [1]. Since it is difficult for a human to decide what languages should be used when there is a group of people who speak different languages and have different levels of proficiency, the method suggests what languages should be used in multilingual communication when machine translation services exist. In order to calculate and suggest the best languages to be used, MT quality and users' language test scores, i.e. TOEIC, TOEFL, have to be shared between Personal Agents and an Optimizing Agent.

The original proposal made users hesitate to disclose their test scores which might give a negative impression of the service. Even though language scores do not represent users, there are users that are not comfortable with sharing their scores. As is true for other personal information, it is important to protect test score confidentiality. The issue of disclosing the test scores of every user has been raised and there are many organizations that place importance on the confidentiality of test scores. An example includes Education Testing Service (ETS), a non-profit organization that administers international tests, including, TOEFL, TOEIC, etc. ETS pays strict attention to confidentiality so private and personal information, including score data, must be kept confidential unless there is the informed consent of the individual is given. In the U.S., the Family Educational Rights and Privacy Act (FERPA) also requires that written permission from the students' parent or eligible student be given before releasing any data from a student's record, except for some special conditions [2]. We can infer that it is critical to treat language test scores as confidential.

The issue with the previous version of the best-balanced machine translation model is that distributing the test scores to permit calculation violates confidentiality. Calculating the best-balanced language combination has specific procedures and its characteristic creates challenges in protecting user privacy. Our research problem is how to calculate the best-balanced language set without disclosing nor distributing language scores from each user's *PersonalAgent*(*PA*), the agent who deals with each user's personal data and activities.

Multi-agent systems and decentralization offer a great many applications. They have been used not only for general problem solving [3], [4], but also for private information protection [5], [6]. With multi-agent systems and data encryption, we can introduce secure computing into human agent interaction. As far as we can tell, this study is the first trial to apply secure computation to a multilingual communication support system. In this paper, we propose a solution

Manuscript received July 7, 2019.
Manuscript revised January 27, 2020.
Manuscript publicized March 18, 2020.

[†] The authors are with the Department of Social Informatics, Kyoto University, Kyoto-shi, 606–8501 Japan.
[††] The author is with the Kyoto College of Graduate Studies for Informatics, Kyoto-shi, 606–8225 Japan.
[†††] The author is with the School of Creative Science and Engineering, Waseda University, Tokyo, 169–8050 Japan.
a) E-mail: mondheera@ai.soc.i.kyoto-u.ac.jp

DOI: 10.1587/transinf.2019KBP0008

that protects test score data privacy from disclosure by user's PA while retaining the language optimization based on best-balanced machine translation. Our main solution is to divide the calculation tasks into small pieces and distribute optimization tasks among different types of agents and use data encryption to support human-human communication; the original method uses one agent for optimization in order to hide sensitive data from all other agents. With this solution, our main contribution is to introduce a multi-agent system with secure computing to multilingual communication with privacy protection.

2. Related Work

Several studies use agents to support human-human communication in various ways. Agents are used to support remote collaboration; for example Vartiainen et al. [7] proposed the concept of a mobile tele-presence robot to support collaboration. Some research tackles agents for group discussion and meeting support. A researcher group [8] tried to create a conversational interface that allows agents and people to communicate smoothly by studying head movements in multi-party meetings. Another group [9] aimed at building an agent that can participate in group discussions to improve the communication skill of users; they proposed a model to determine attention targets for the agent in group discussions.

Each existing paper focuses on different parts of support for collaboration, including user experience and conversation training. Our work, however, focuses on using a multi-agent system to support intercultural collaboration and multilingual communication with strong user data privacy. In this paper, we use multiple agents to create a privacy-aware system that can select the languages to be used by each user by combining the existing concept of best-balanced machine translation with secure computation techniques.

Since private and personal data must be protected while still allowing it to be processed with some other data, various method have been proposed that make use of the data without violating data privacy. Jian and Bhandare [10] proposed a method to preserve privacy based on min max normalization transformation in data mining.

K-anonymity [11] has been widely used to protect data privacy, especially for data publication; it usually employs data suppression and generalization. Since different kinds of data have different characteristics, the calculations used are varied. For example, test scores of students have their own characteristic, Yi [12] proposed a method to publish test scores in a location-based service while protecting students privacy protected by using K- anonymity. Some researchers use cryptography techniques to protect user's privacy. When the raw data is online, it can leak. Many studies using encryption to protect data confidentiality. Popa et al. [13] proposed CryptDB, a system that uses encrypted database queries to protect sensitive data. They execute SQL queries over encrypted data and also link encryption keys to user passwords so even the database admin cannot access the data.

Besides protecting user's data, Yokoo et al. [6] proposed a method to protect the private information by using multi-party techniques. Their method utilizes a public key encryption scheme that allows the information to be computed cooperatively but blocks any link back to the agents.

There exist various methods to protect data privacy while making use of the data. However, each method suits only specific types of data and specific situations, for instance data suppression is suitable for some data publication processes but not those that need to process real or specific data. None of the stated methods is suitable to keep language score data private while enabling best-balanced calculation. The challenge is that using only data encryption and decryption or purely secure computation is not enough to hide user scores from the other agents, since the processes of calculation are fairly complex and there is some chance that an agent can guess user language levels.

3. Best-Balanced Machine Translation

Since language differences are one of the biggest barriers in inter-cultural collaboration, MT has been used in multilingual communication in various projects and for different purposes, from non-profit, research, to for profit use [14]. Our original published work [1], showed that the best balance machine translation makes the best use of both MT services and human language skill.

MT can cause communication balance problems. For example, Fig. 1 displays the difficult situation possible in multilingual communication. Given a simple conversation between user *j* with good English skill and user *i* with limited English skill, choosing the best communication method is not complicated. Using MT is a good option as one of the users cannot communicate well in English. Later, user *k* with fair English skill joins the conversation, it becomes more difficult to determine the best method of communication. It is possible to use a shared foreign language, English, MT, or a combination of both options. If English is used as the medium for this conversation, it might cause difficulties for user *i* whose English skill is limited. Using MT could be a good alternative. However, the other two partici-

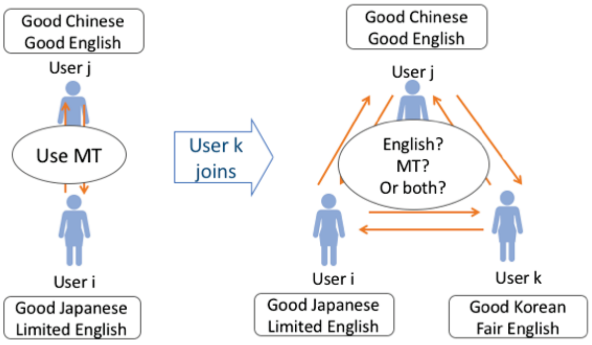


Fig. 1 Situation when BBMT could be useful.

pants have good enough English skill to communicate which might yield better results than using MT.

English skill that considered includes $WritingSkill(WS)$ and $ReadingSkill(RS)$, normalized to the range of 0 to 1. English skills were measured using normalized standard test scores from TOEIC, TOEFL, or IELTS. Test scores were converted into Common European Framework of Reference for Languages(CEFR) [15] which is an international standard for English language ability. CEFR has six levels: A1(Basic), A2(Basic), B1(Independent), B2(Independent), C1(Proficient), and C2(Proficient). Score conversion is done with data from ETS [16] for TOEIC and TOEFL and data from Cambridge assessment [15] for IELTS. The conversion matrix used for our calculation is as follows: 1 for C1 and above, 0.75 for B2, 0.5 for B1, 0.25 for A2, and 0 for A1 and lower. If the user does not have complete score data; for example if the user has TOEIC reading and listening test scores, but did not take the TOEIC writing test, the overall score is used to estimate the missing score

The original method requires user language skill or scores to be sent from each user's *PersonalAgent* to the *OptimizingAgent*; MT service quality levels are taken from the Language Grid [17] and used compute the optimal language set. After sharing the information and calculation results, the *OptimizingAgent* can suggest to each user what language is the best to optimize the quality of messages (QoM), using given the known user language skills and MT quality.

To compute the the optimal language set, first, a list of language combinations is made by the *OptimizingAgent* from the languages each user knows. If there are three users and each user can speak two languages as in Fig. 1, there are eight possible combinations. Let *ja*, *ko*, and *zh* represent Japanese Korean, and Chinese language, respectively. Under the assumption that English can be used by everyone to some degree, possible combinations, C_1 to C_8 for the communication of the three users are as follows:

$C_1 = (ja, ko, zh)$, $C_2 = (ja, ko, en)$, $C_3 = (ja, en, zh)$, $C_4 = (ja, en, en)$, $C_5 = (en, ko, zh)$, $C_6 = (en, ko, en)$, $C_7 = (en, en, zh)$, $C_8 = (en, en, en)$

The values in the bracket are the languages of the first user, the second user, and the third user respectively.

If there are n users, each combination consists of $n(n-1)/2$ QoM pairs. For example, $C1$ consists of threeaveraged QoM pairs ($AvgQoM$) including (ja, ko), (ko, zh), and (zh, ja). C1 utilizes three pairs or six of MT services, including ($MT_{ja,ko}, MT_{ko,ja}$), ($MT_{ko,zh}, MT_{zh,ko}$), and ($MT_{zh,ja}, MT_{ja,zh}$). The Optimizing agent calculates the QoM. $QoM(P_i, MT_{i,j}, P_j)$ represents the quality of the message that user P_i who used language L_i sends to user P_j, who uses language L_j via a machine translation service $MT_{i,j}$. $MT_{i,j}$ represents MT service that translates messages from language L_i to language L_j. We consider the input language writing skill of the message sender, machine translation accuracy of $MT_{i,j}$, and output language reading skill of the message receiver. $QoM(P_i, MT_{i,j}, P_j)$ can be calculated as follows:

$$QoM(P_i, MT_{i,j}, P_j) = writing\_skill(P_i, L_i) \times MTQ(i, j) \times reading\_skill(P_j, L_j) \tag{1}$$

ALGORITHM 1: Best-balance machine translation

```
Input  : $P_\mathrm{i}$: User $i$ (1 <= i <= N, n is the number of users)
         $L_\mathrm{i}$: Language $i$ used by user $P_\mathrm{i}$
         $WS(P_\mathrm{i}, L_\mathrm{i})$: Writing skill of $P_\mathrm{i}$ when using language $L_\mathrm{i}$
         $RS(P_\mathrm{i}, L_\mathrm{i})$ : Reading skill of $P_\mathrm{i}$ when using language $L_\mathrm{i}$
         $MTQ(MT_\mathrm{i\,j})$ : $MT$ accuracy in translating from $L_\mathrm{i}$ to $L_\mathrm{j}$
Output: Best-balanced language combination $BBC$
Generate the set of all possible language combinations $LC$
forall language combination $C_k$ in $LC$ do
    forall language pair $< L_i\ L_j >$ in $C_k$ do
        $QoM(P_\mathrm{i}, MT_\mathrm{ij}, P_\mathrm{j}) \leftarrow$
          $WS(P_\mathrm{i}, L_\mathrm{i}) \times MTQ(MT_\mathrm{i,j}) \times RS(P_\mathrm{j}, L_\mathrm{j})$;
        $QoM(P_\mathrm{j}, MT_\mathrm{ji}, P_\mathrm{i}) \leftarrow$
          $WS(P_\mathrm{j}, L_\mathrm{j}) \times MTQ(MT_\mathrm{j,i}) \times RS(P_\mathrm{i}, L_\mathrm{i})$;
        $AvgQoM_\mathrm{ij} \leftarrow$
          $(QoM(P_\mathrm{i}, MT_\mathrm{ji}, P_\mathrm{j}) + QoM(P_\mathrm{j}, MT_\mathrm{ji}, P_\mathrm{i}))/2$;
    end
end
Acquire the set of Pareto optimal language combination $PLC$;
$m \leftarrow$ number of acquired language combinations in $PLC$;
if $m = 1$ then
    $BBC \leftarrow$ the only language combination in $PLC$;
else
    forall Pareto optimal language combination $C_k$ in $PLC$ do
        Compute the variance between each $AvgQoM$ in $C_k$;
    end
    $BBC \leftarrow$ language combination with minimum variance;
end
```

This model shows that the writing skill of the sender, reading skill of the receiver and accuracy of machine translation impact message the quality of message. Therefore, selecting the most appropriate language pair is critical.

Since the QoM indicate the one way quality of message, QoM from user P_i to P_j and the QoM from user P_j to P_i could be different. To select languages for the best-balanced communication channel, average values of the QoMs from both sides (AvgQoM) are used in the computation as shown in Algorithm 1.

To further illustrate best balance language combination selection, an example is shown in Table 1. Each combination has three $AvgQoM$ values, calculated for each communication channel (each pair of users) using Eq. (1). $AvgQoM$ in the table is the example value from the original work and is used to simply explain the process of calculation. The best balance combination in this case is C_4, which is the only Pareto optimal combination. A combination is called Pareto optimal when it is impossible to make a better $AvgQoM$, without making another $AvgQoM$ worse.

In many cases, there could be more than one Pareto optimal combination. The best-balanced combination can be determined by using variance to evaluate the differences among the $AvgQoMs$. Lower difference indicates higher conversation equality.

This computation procedure makes it necessary to for-

Table 1 An example of *AvgQoM*table.

Combi-nation /User	i → j j → i	j → k k → j	k → i i → k
C_1	ja→zh zh→ja 0.7125	zh→ko ko→zh 0.5406	ko→ja ja→ko 0.7719
C_2	ja→en en→ja 0.7906	en→ko ko→en 0.625	ko→ja ja→ko 0.7719
C_3	ja→zh zh→ja 0.7125	zh→en en→zh 0.5703	en→ja ja→en 0.7906
C_4	ja→en en→ja 0.7906	en→en en→en 0.875	en→ja ja→en 0.7906
C_5	en→zh zh→en 0.4047	zh→ko ko→zh 0.5406	ko→en en→ko 0.4688
C_6	en→en en→en 0.75	en→ko ko→en 0.625	ko→en en→ko 0.4688
C_7	en→zh zh→en 0.4047	zh→en en→zh 0.5703	en→en en→en 0.5
C_8	en→en en→en 0.75	en→en en→en 0.875	en→en en→en 0.5

ward users' language scores to the *OptimizingAgent* to compute *QoM*.

4. Privacy-Aware Best-Balanced Machine Translation

4.1 Agents and Their Roles

In this section, we propose a multi-agent system to optimize the languages that could yield the best balance in MT mediated communication. Our method includes three types of agent, *PersonalAgent*, *ProcessManagingAgent*, and *SelectionAgent*, rather than just the *PersonalAgents* and *OptimizingAgent* of the original method. Tasks done by the *OptimizingAgent* in the original method are distributed among all agents. The major functions of this system are secured *AvgQoM*calculation done by *PersonalAgents*, and key handling and secure language combination selection by the *SelectionAgent*. *ProcessManagingAgent* does the remaining tasks that cannot be done by the other two agents for security reasons, including creating language combinations and managing the data. In this section, we describe the agents we introduce and their roles using the role schema from Gaia methodology [18]. The role schemas include protocols, which are linked to the activities and interactions. Activities are tasks done by the agent itself, without interaction, and are underlined. The interactions among agents are shown in Fig. 2. The arrows on the diagram represent data transmission among the three different types of agents, including function name, links to the agent role schemas (data sent is shown in the parentheses).

4.1.1 Personal Agent

In the original calculation, *AvgQoM* is calculated by one single agent so this agent needs to see every user's score. User's personal computer, or the *PersonalAgent* is not involved in the calculation. In this proposal, the *PersonalAgent* is responsible for the user's data and takes part in computing AvgQoM, so user's score is not sent to any other agent.

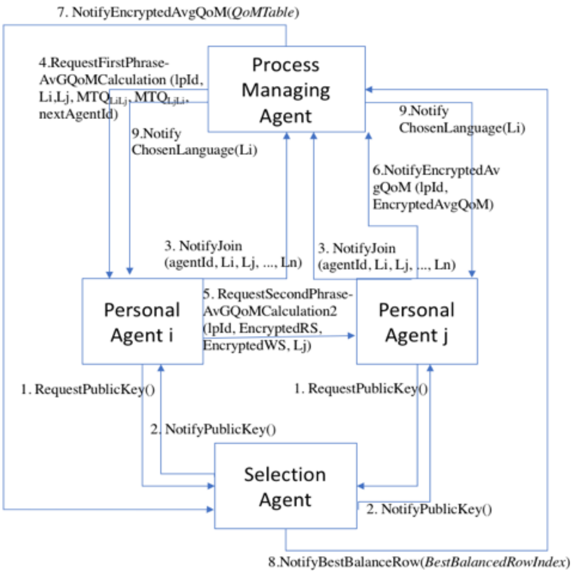

7. NotifyEncryptedAvgQoM(QoMTable)
Process Managing Agent
4.RequestFirstPhrase-AvGQoMCalculation (lpId, Li,Lj, MTQ$_{LiLj}$, MTQ$_{LjLi}$, nextAgentId)
9.Notify ChosenLanguage(Li)
9.Notify ChosenLanguage(Li)
6.NotifyEncryptedAvgQoM (lpId, EncryptedAvgQoM)
3. NotifyJoin (agentId, Li, Lj, ..., Ln)
3. NotifyJoin (agentId, Li, Lj, ..., Ln)
Personal Agent i
5. RequestSecondPhrase-AvGQoMCalculation2 (lpId, EncryptedRS, EncryptedWS, Lj)
Personal Agent j
1. RequestPublicKey()
1. RequestPublicKey()
2. NotifyPublicKey()
Selection Agent
2. NotifyPublicKey()
8.NotifyBestBalanceRow(BestBalancedRowIndex)

Fig. 2 Interactions and data sharing among agents in privacy-aware BBMT calculation.

ALGORITHM 2: Firstphrase AvgQoMCalculation

```
Input : $IpId, Li, Lj, MTQ_{LjLi}, nextAgentId$
Output: $EncryptedWS, EncryptedRS$
Calculate
    $EncryptedWS = MTQ_{LiLj} * Encrypt(WS_{PiLi})$;
Calculate
    $EncryptedRS = MTQ_{LjLi} * Encrypt(RS_{PiLi})$;
Request $SecondPhrase-$
    $AvGQoMCalculation(lpId, EncryptedRS, EncryptedWS, Lj)$
    to $PersonalAgent$ of the language pair whose
    $id = nextAgentId$ ;
```

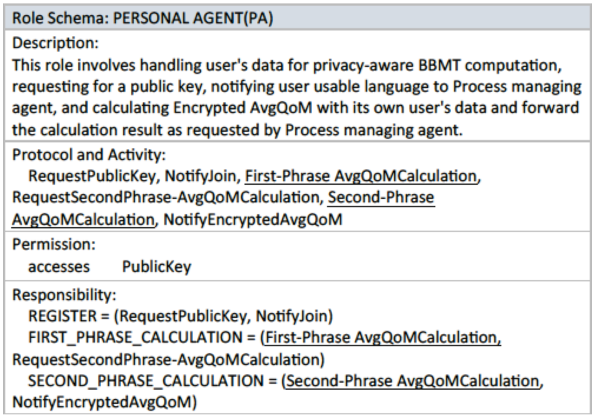
To calculate *AvgQoM*without disclosing user language score, encryption is used and the calculation is done by two *PersonalAgent*s, the score owners. Each *AvgQoM* consists of two parts; *FirstPhraseAvgQoMCalculation* done by the *PersonalAgent* of the first score owner and *SecondPhraseAvgQoMCalculation* done by the *PersonalAgent* of the second score owner. The following procedure is used where the number in parenthesis indicates the process in Fig. 2.

The first calculation is decided and requested by *ProcessManagingAgent*(*PMA*). The first agent encrypts its user *writingskill*(*WS*) and *readingskill*(*RS*) of the requested language L_i and multiplied by given MT quality score (*MTQ*). After the first calculation is done, the first agent requests the second calculation to the next agent identified earlier by *PMA* using encrypted reading skill(*EncryptedRS*) and encrypted writing skill(*EncryptedWS*).

The second agent also encrypts its own RS and multiplies it by first agent's EncryptedWS and encrypts its WS then multiplies the results by the first agent's EncryptedRS. After both calculation phrases are executed,

ALGORITHM 3: Secondphrase AvgQoMCalculation

```
Input  : $IpId, EncryptedWS, EncryptedRS$
Output: $EncryptedAvgQoM$
Calculate $EncryptedAvgQoM =$
  $(EncryptedWS * Encrypt(RS_{UjLj}) +$
  $EncryptedRS * Encrypt(WS_{UjLj}))/2$ ;
Send
  $NotifyEncryptedAvgQoM(lpId, EncryptedAvgQoM)$
  to Process Managing Agent;
```



Role Schema: PERSONAL AGENT(PA)

Description:
This role involves handling user's data for privacy-aware BBMT computation, requesting for a public key, notifying user usable language to Process managing agent, and calculating Encrypted AvgQoM with its own user's data and forward the calculation result as requested by Process managing agent.

Protocol and Activity:
RequestPublicKey, NotifyJoin, First-Phrase AvgQoMCalculation, RequestSecondPhrase-AvgQoMCalculation, Second-Phrase AvgQoMCalculation, NotifyEncryptedAvgQoM

Permission:
accesses PublicKey

Responsibility:
REGISTER = (RequestPublicKey, NotifyJoin)
FIRST_PHRASE_CALCULATION = (First-Phrase AvgQoMCalculation, RequestSecondPhrase-AvgQoMCalculation)
SECOND_PHRASE_CALCULATION = (Second-Phrase AvgQoMCalculation, NotifyEncryptedAvgQoM)

Fig. 3 Role schema of *PersonalAgent*.

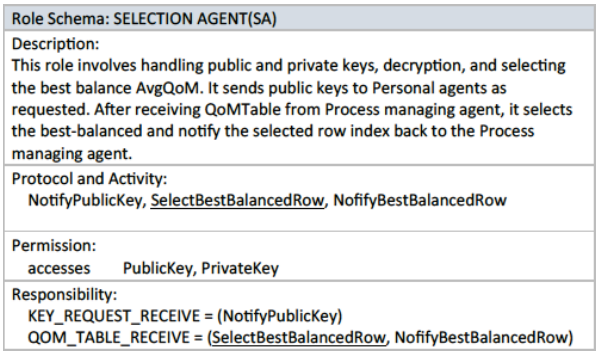
the second agent sends the encrypted $AvgQoM$ back to the $ProcessManagingAgent$ with reference $IpId$.

4.1.2 Selection Agent

Selection agent creates a public key for encryption and a private key for decryption. When a $PersonalAgent$ joins the system and requests a public key, $SelectionAgent$ sends a public key to the $PersonalAgent$. In this system, $PersonalAgent$s use a public key for secured $AvgQoM$ calculation and $SelectionAgent$ has both public key and private key; $ProcessManagingAgent$ has no key.

The keys are handled by $SelectionAgent$ because when an agent knows the raw value of $AvgQoM$ and knows who $AvgQoM$belongs to, it is possible to guess the value of user scores. For example, if the $AvgQoM$ between user i and user j is very low, the agent who knows the $AvgQoM$ can imply that machine translation quality is low and that both user i and user j have low skill in that particular language. $SelectionAgent$ can see both encrypted $AvgQoMs$ and $AvgQoMs$, but this agent does not know who owns which $AvgQoM$ since all the encrypted $AvgQoMs$ are forwarded from $ProcessManagingAgent$ in $QoMTable$, a table contains all encrypted or decrypted $AvgQoM$, without any information about user, $PersonalAgent$, or combination.

4.1.3 Process Managing Agent



Role Schema: SELECTION AGENT(SA)

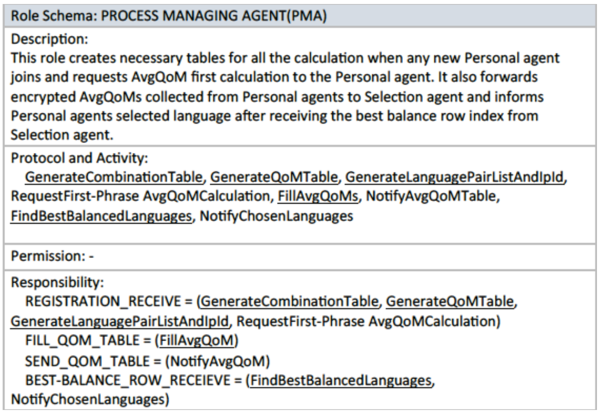
Description:
This role involves handling public and private keys, decryption, and selecting the best balance AvgQoM. It sends public keys to Personal agents as requested. After receiving QoMTable from Process managing agent, it selects the best-balanced and notify the selected row index back to the Process managing agent.

Protocol and Activity:
NotifyPublicKey, SelectBestBalancedRow, NofifyBestBalancedRow

Permission:
accesses PublicKey, PrivateKey

Responsibility:
KEY_REQUEST_RECEIVE = (NotifyPublicKey)
QOM_TABLE_RECEIVE = (SelectBestBalancedRow, NofifyBestBalancedRow)

Fig. 4 Role schema of *SelectionAgent*.



Role Schema: PROCESS MANAGING AGENT(PMA)

Description:
This role creates necessary tables for all the calculation when any new Personal agent joins and requests AvgQoM first calculation to the Personal agent. It also forwards encrypted AvgQoMs collected from Personal agents to Selection agent and informs Personal agents selected language after receiving the best balance row index from Selection agent.

Protocol and Activity:
GenerateCombinationTable, GenerateQoMTable, GenerateLanguagePairListAndIpId, RequestFirst-Phrase AvgQoMCalculation, FillAvgQoMs, NotifyAvgQoMTable, FindBestBalancedLanguages, NotifyChosenLanguages

Permission: -

Responsibility:
REGISTRATION_RECEIVE = (GenerateCombinationTable, GenerateQoMTable, GenerateLanguagePairListAndIpId, RequestFirst-Phrase AvgQoMCalculation)
FILL_QOM_TABLE = (FillAvgQoM)
SEND_QOM_TABLE = (NotifyAvgQoM)
BEST-BALANCE_ROW_RECEIEVE = (FindBestBalancedLanguages, NotifyChosenLanguages)

Fig. 5 Role schema of *ProcessManagingAgent*.

This agent generates tables and creates language combinations based on user languages as notified by $PersonalAgent$. It can see the encrypted $AvgQoMs$ data from $PersonalAgents$, but without decryption key, it is unable to determine the real $AvgQoM$ values. It also handles index, as reference codes, to distribute calculations and selection tasks and to keep $SelectionAgent$ from linking its data back to any user or combination of users.

4.1.4 Computation Flow

When a user joins this system, her/his $PersonalAgent$ sends a request for a public key to $SelectionAgent(SA)$. Upon receiving the request, SA sends the public key to the $PersonalAgent$. Each $PersonalAgent$ also notifies $ProcessManagingAgent$ when it joins the system with its id called agentId, and the languages that its user can employ, displayed as languages L_i and L_j.

Upon receiving the notification, $ProcessManagingAgent$ receives the notification, the agent creates a table called Combination Table which contains all possible language combinations based on users' usable languages. Each row contains an ID called $combinationID$, username and language of each user. For example, $(i, Li), (j, Lj)$ is a language combination for user i using language L_i and user j using language L_j to communicate. In case, the third user, user k, who speaks language L_k joins, one of the possible combinations would be $(i, L_i), (j, L_j), (k, L_k)$.

When there are 2 users and both user i and user j can

Table 2 An example of combination table.

Combination ID	Language and User
C_1	$(i, L_1), (j, L_1)$
C_2	$(i, L_1), (j, L_2)$
C_3	$(i, L_1), (j, L_3)$
C_4	$(i, L_2), (j, L_1)$
C_5	$(i, L_2), (j, L_2)$
C_6	$(i, L_2), (j, L_3)$
C_7	$(i, L_3), (j, L_1)$
C_8	$(i, L_3), (j, L_2)$
C_9	$(i, L_3), (j, L_3)$

use languages L_1, L_2, and L_3, the combination table will contain information as shown in Table 2. The total number of combinations equals the number of language(s) used by user *i* multiplied by the number of language(s) used by user *j*. If there are more than two users, the multiplication continues to the number of languages used by the third user, fourth user, until the last user language(s) is(are) considered.

Next, *ProcessManagingAgent* also creates a blank *QoMTable* to store encrypted *AvgQoM* when received. The table height equals the heights of Combination Table and the table width is equal $n * (n - 1)/2$ when n is the number of users. This table will store encrypted *AvgQoM* values for each link between pairs of agents. If there are three users, there are three links among the users, including links between user *i* and user *j*, between user *i* and user *k*, and between user *j* and user *k*. Thus there will be three encrypted *AvgQoMs* for each combination.

Besides Combination Table and *QoMTable*, the *ProcessManagingAgent* also creates *LanguagePairList*, including language pairs whose *AvgQoM* needed to be calculated. A language pair is a pair of language that allow two users to communicate. For example $(i, L_1), (j, L_1)$ is a language pair when both user *i* and *j* can use language L_1 to communicate. When there are two users, this Language Pair List contains all the language pairs in the *CombinationTable*, but when there are three or more users, some language pairs in the *CombinationTable* are redundant, and only unique redundant language pairs appear in the Language Pair List. For three-user communication, each combination contains three language pairs including the language pairs of user *i* and user *j*, user *i* and user *k*, user *j* and user *k*. Each language pair in this list also has an ID called *lpId* generated by the *ProcessManagingAgent*, in order to link the language pair with its location in *QoMTable*.

Every language pair in the *LanguagePairList* needs to have *AvgQoM* calculated, so the *ProcessManagingAgent*, sends one message per on language pair in the list to the first *PersonalAgents* of the pair. The *ProcessManagingAgent* also has a list of machine translation quality or *MTQ*. $MTQ(L_i, L_j)$ values representing machine translation quality translating from language *Li* to language *Lj*. For instance, for the language pair $(i, L_i), (j, L_j)$, the *ProcessManagingAgent* sends a message to *PersonalAgenti*, requesting for the first phrase of *AvgQoM*calculation. This message includes *lpId* of the pair, language L_i, language L_j, $MTQ(L_iL_j)$, $MTQ(L_jL_i)$ and the *nextAgentId* which is the identification of the second agent in the language pair.

When the *PersonalAgent* of user *Pi* receives the request for *AvgQoM*calculation, it calculates *Encryptedwritingskill*(*EncryptedWS*) by using the public key to encrypt the value of user i's writing skill in language L_i or $WS(P_iL_i)$), then multiplies the value by $MTQ(L_iL_j)$. *Encryptedreadingskill*(*EncryptedRS*) is calculated by encrypting user *i*'s reading skill of language L_i or $RS(P_iL_i)$ multiplied by $MTQ(L_iL_j)$. Both *EncryptedWS* and *EncryptedRS* are encrypted values and need the private key, held only by the *SelectionAgent*, to read the real values. Hence, there is no agent can read the pure data of this value, since this data is not sent to the *SelectionAgent* either.

After the first calculation part is done by the first *PersonalAgent* PA_i, the first agent sends a request for the second calculation to the second *PersonalAgent* PA_j of the language pair. The message includes *lpId*, *EncryptedWS*, *EncryptedRS*, and Language *Lj*. When the second agent receives the request, it calculates Encrypted *AvgQoM*by using the public key to encrypt and multiply own reading skill of language *Lj* (RS_{UjLj}) to *EncryptedWS* and own writing skill of language *Lj* (WS_{UjLj}) by *EncryptedRS*, then average these values. This whole process of calculation also needs the public key and the result value is an encrypted value of *AvgQoM* called *EncryptedAvgQoM*. When the calculation is done, the second agent of the language pair sends *EncryptedAvgQoM* together with the reference *lpId* back to the *ProcessManagingAgent*.

Processmanagingagent collects Encrypted *AvgQoM*values from *PersonalAgents* and enters them into previously created *QoMTable* based on lpId. The *QoMTable* has no index that allows reference to any combination or user. The *ProcessManagingAgent* sends this *QoMTable* to *SelectionAgent*.

Selection agent selects the Best balance row by, first, decrypting the encrypted *AvgQoMs* to pure *AvgQoMs* with its private key, then select the row that has Pareto optimal value. After decryption, *SelectionAgent* can see the only numbers on the table, there is no reference to any combination or user. If there are several Pareto optimal roles, this agent calculates the variance among the values in the Pareto optimal roles, and selects the row with the least variance as $Best - balancedRow$. After that, *SelectionAgent* sends the index of the $Best - balancedRow$ back to the *ProcessManagingAgent*.

After *ProcessManagingAgent* receives the row index, it uses the row index to search for the information to see which language combination is linked to the Best balance row and what language each user should use in that combination. It notifies each *PersonalAgent* of the recommended language for that user based on the best-balanced combination selected.

5. Discussion

5.1 Security Argument

In this section, we discuss privacy protection. The information that we are trying to protect here is user language score(s) held by each *PersonalAgent*, including reading skill and writing skill of its user. In addition to user language skills, *AvgQoM* should also be treated as fairly sensitive information. An agent who knows the *AvgQoM* owners of low AvgQoM, can guess that both user scores are low. Thus another security requirement is, any agent that knows pure *AvgQoM*value must not know which agents' skills were used to calculate the value.

We investigate situations when each agent, one by one, acts as an adversary that wants to violate user privacy, given the information known to that agent, called its view. The view of an agent is all the information that is visible to that agent, including the information it owns, and information from the other agents or the other source, see Table 3. Information that each agent owns or created by itself is underlined. Information without underline is information received from other agents. We assume that each agent is Honest-but-Curious(HbC). HbC agents follow the steps of the protocol but try to learn as much information as possible [19].

When we consider the *PersonalAgent* of user i, or PA_i, as an HbC adversary, PA_i should get no information about another *PersonalAgent*'s private information other than what is trivially derivable from its own input and the final language outcome. From *PersonalAgent*'s view in Table 3, data related to another agent, PA_j, is encrypted, including *EncryptedRS* and *EncryptedWS*, hence no private information leaks to PA_j.

Considering *SelectionAgent* as an HbC adversary, we try to prove that, given the view of the protocol, there is no way for this agent to recover the inputs from *PersonalAgent*(*s*). The worst case in terms of information security is when there are only two users and each user speaks only one language. *SelectionAgent* knows the pure or decrypted *AvgQoM*value which is calculated by the following equation.

Table 3 View of each agent

Personal Agent (PA) View	Selection Agent (SA) View	Process Managing Agent (PMA) View
-Own language data -Public key for encryption -Some MTQ -Encrypted RS and Encrypted WS -Suggested language to be used	-Public and private key -Decrypted AvgQoMs -Best balanced row index in *QoMTable* -User(s) joined the system -All encrypted AvgQoM in the QoM table	-Combination Table -QoM Table -All MTQ -User(s) joined the system -User language(s) -All encrypted AvgQoM -Best balanced row index in *QoMTable*

$$\begin{aligned} AvgQoM(P_i, MT_{i,j}, P_j) = \{[(WS(P_i, L_i) \times MTQ(i,j) \times \\ RS(P_j, L_j)] + [WS(P_j, L_j) \times \\ MTQ(j,i) \times RS(P_i, L_i)]\}/2 \end{aligned} \qquad (2)$$

In this equation, there are 6 variables unknown to *SelectionAgent*, so we can make an information theoretic argument that, *SelectionAgent* cannot recover the scores from each *PersonalAgent*. However, in reality when each user speaks only one language, there is no need to look for the best-balanced language combination from the beginning. *AvgQoM* and all language scores obviously should be high since everybody use her/his native language. But when user(s) speak more than one language, the complexity barrier is even higher for *SelectionAgent*, since it does not know whose data and what language each *AvgQoM* linked to, as the *QoMTable*, sent from *ProcessManagingAgent*, contains only *EncryptedAvgQoM* values. The *QoMTable* contains no user data, *PersonalAgent* data, or language combination data. With regard to *ProcessManagingAgent*, if this agent gets plain AvgQoM, it implies violation of semantic security. However, all *AvgQoM* values seen by *ProcessManagingAgent* are encrypted and this agent does not have the private key for decryption. Hence, *ProcessManagingAgent* cannot violate the semantic security. Moreover, if there are two *EncryptedAvgQoMs* that link to the *best* − *balancedRowindices*, and *ProcessManagingAgent* can identify which value is associated with which index, there is a violation of encryption indistinguishability. However, different *EncryptedAvgQoMs*, such as higher *AvgQoM* in the *Best* − *balancedrow*, can also give the same best-balanced index result. As a result, encryption indistinguishability is not violated.

Using the view of each agent of the protocol, we can conclude that, this protocol satisfies our privacy-protection objective. However, some actions that lead to information leakage cannot be prevented. Natural leakage of information could happen, regardless of how secure the system is. Even though the system is ideally trusted and secured, the leakage is going to happen unavoidably and naturally. In our case, when a malicious *PersonalAgent* joins the system numerous times, it could change its data input little by little, and look for a corresponding change in recommended language result. This might allow the *PersonalAgent* to guess the level of language ability of another user, especially when there are only two users in the system.

5.2 Implementation

Privacy protection can be proved via the security argument, however, it cannot prove that the new proposed method can be processed successfully and yield the same result as the original proposal. To examine the computation result, we implemented the privacy-aware version on a MacBook Pro(15-inch, 2017) with Intel Core i7 2.9GHz and 16GB Memory. The program is run on one process per user on the same machine and connected them over network protocol (UDP). We found that this version gave the same results as the original BBMT for the same inputs.

The proposal has longer calculation time when there are many users and each user speaks many languages due to combinatorial explosion. However, in real settings, best-balanced machine translation is created to be used for multilingual chat and is normally used by 3-4 users where each user speaks 2-3 languages. Thus the real computation time is acceptable. For instance, with 4 users, each speaking 2-3 languages, our computer took around 1.8 seconds to finish the calculation needed by our proposal.

To meet different implementation environments, some adjustments might be needed. While some encryption algorithms, which offer fully homomorphic encryption, allow multiplication of two encrypted values. However, this technology is relatively new and only available when implemented on C++. Many encryption algorithms, such as Paillier, do not support multiplication of encrypted values but support the multiplication of one encrypted value and a raw value. In this case, instead of encrypting reading skill and writing skill, encrypting machine translation quality and multiply by own reading skill and own writing skill could be an option.

6. Conclusion

The original method of calculating best-balanced machine translation combinations requires users to disclose their language test scores. Since there are some users who do not wish to share their language scores, language scores should be treated as private information. This paper proposed a protocol to compute best-balanced language combination without disclosing user's language scores. We combined cryptography techniques and a multi-agent system to create a 3-agent-type system with a privacy-aware protocol, since existing privacy protection methods are not suitable for implementing best-balanced machine translation. Our method enables the calculation for the best-balanced language recommendation while ensuring the privacy of language scores. This was confirmed by a rigorous security argument.

Our contribution is to introduce secure computation to protect user's private information in multilingual collaboration, as we aim to emphasize the importance of user's data privacy in human-computer interaction and computer-mediated intercultural collaboration. Because it is significant to treat test scores confidential, as they are user private information, our proposal ensures that user language scores will not be disclosed except for natural leakage, which is unavoidable.

Acknowledgments

This research was partially supported by a Grant-in-Aid for Scientific Research (A) (17H00759, 2017–2020) and (B) (18H03341, 2018–2020) from Japan Society for the Promotion of Science (JSPS). We thank Prof. Masayuki Abe who provided insight and expertise in secure computation that greatly assisted the research.

References

[1] M. Pituxcoosuvarn and T. Ishida, "Multilingual communication via best-balanced machine translation," New Generation Computing, vol.36, no.4, pp.349–364, Aug. 2018.
[2] U.D. of Education, "Family educational right and privacy act," 2018.
[3] G. Anders, F. Siefert, J.P. Steghöfer, and W. Reif, "A decentralized multi-agent algorithm for the set partitioning problem," Int. Conf. Principles and Practice of Multi-Agent Systems, pp.107–121, Springer, 2012.
[4] T. Matsui, M. Silaghi, K. Hirayama, M. Yokoo, and H. Matsuo, "Distributed search method with bounded cost vectors on multiple objective dcops," Int. Conf. Principles and Practice of Multi-Agent Systems, pp.137–152, Springer, 2012.
[5] R. Greenstadt, J.P. Pearce, and M. Tambe, "Analysis of privacy loss in distributed constraint optimization," AAAI, pp.647–653, AAAI, July 2006.
[6] M. Yokoo, K. Suzuki, and K. Hirayama, "Secure distributed constraint satisfaction: Reaching agreement without revealing private information," Int. Conf. Principles and Practice of Constraint Programming, pp.387–401, Springer, 2002.
[7] E. Vartiainen, V. Domova, and M. Englund, "Expert on wheels: an approach to remote collaboration," Proc. 3rd Int. Conf. Human-Agent Interaction, pp.49–54, ACM, 2015.
[8] R. Ishii, S. Kumano, and K. Otsuka, "Prediction of next-utterance timing using head movement in multi-party meetings," Proc. 5th Int. Conf. Human Agent Interaction, pp.181–187, ACM, Oct. 2017.
[9] S. Kimura, H.H. Huang, Q. Zhang, S. Okada, N. Ohta, and K. Kuwabara, "Proposal of a model to determine the attention target for an agent in group discussion with non-verbal features," Proc. 5th Int. Conf. Human Agent Interaction, pp.195–202, ACM, Oct. 2017.
[10] Y.K. Jain and S.K. Bhandare, "Min max normalization based data perturbation method for privacy protection," Int. J. Computer & Communication Technology, vol.2, no.8, pp.45–50, 2011.
[11] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems, vol.10, no.05, pp.571–588, 2002.
[12] T. Yi, M. Shi, and Z. Hong, "Privacy protection method for test score publishing," 2015 7th Int. Conf. Information Technology in Medicine and Education (ITME), pp.516–520, IEEE, 2015.
[13] R.A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," Proc. Twenty-Third ACM Symposium on Operating Systems Principles, pp.85–100, ACM, Oct. 2011.
[14] T. Ishida, Y. Murakami, D. Lin, T. Nakaguchi, and M. Otani, "Language service infrastructure on the web: the language grid," Computer, vol.51, no.6, pp.72–81, June 2018.
[15] A. Cambridge, "International language standards," Oct. 2019. Available at http://www.cambridgeenglish.org/exams-and-tests/cefr.
[16] ETS, "Compare toefl scores," Oct. 2019. Available at https://www.ets.org/toefl/institutions/scores/compare/.
[17] T. Ishida, The language grid: Service-oriented collective intelligence for language resource interoperability, Springer Science & Business Media, 2011.
[18] M. Wooldridge, N.R. Jennings, and D. Kinny, "The gaia method-

ology for agent-oriented analysis and design," Autonomous Agents and multi-agent systems, vol.3, no.3, pp.285–312, 2000.

[19] J. Wang and Z.A. Kissel, Introduction to network security: theory and practice, John Wiley & Sons, 2015.

Mondheera Pituxcoosuvarn is a Ph.D. student at the Graduate School of Informatics, Kyoto University after attaining a Master's degree there. She holds a B.S. majored in Computer Engineering from King Mongkut's University of Technology Thonburi, Thailand. Her research interests include intercultural collaboration, design methods and creativity.

Takao Nakaguchi is an associate professor in The Kyoto College of Graduate Studies for Informatics, Japan.He holds a Ph.D. in Informatics from Department of Social Informatics of Kyoto University, Japan, and a Master of Science in Information Technology in the Kyoto College of Graduate Studies for Informatics, Japan. He was a co-developer and a principal developer in IPA Exploratory Software project from 2000 to 2001 and 2002, respectively. His research interests include services computing, intercultural collaboration and multiagent systems.

Donghui Lin holds a Ph.D. in Informatics from Department of Social Informatics of Kyoto University, Japan, and a M.E. degree from Department of Computer Science and Engineering of Shanghai Jiao Tong University, China. He was a researcher of National Institute of Information and Communications Technology (NICT), Japan during 2008 to 2011. He was an assistant professor in Department of Social Informatics, Kyoto University during 2012 to 2018. Currently he is an associate professor in Department of Social Informatics, Kyoto University. His research interests include services computing, intercultural collaboration, artificial intelligence, and multiagent system.

Toru Ishida is a professor of Faculty of Science and Engineering, Waseda University. His academic background includes a professor of of Kyoto University from 1993 to 2019. He experienced a vice president of IEICE, and a member of the Science Council of Japan. He was a co-founder of the Department of Social Informatics, Kyoto University, and the Kyoto University Design School. His research interest lies with Autonomous Agents and Multi-Agent Systems and modeling collaboration within human societies. His projects include Community Computing, Digital City, Intercultural Collaboration and the Language Grid. He is a fellow of IEEE, IEICE and IPSJ.