Cloud-Based Privacy Aware Preference Aggregation Service

Sourya Joyee De, Asim K. Pal

Cloud-based Privacy Aware Preference Aggregation Service

Sourya Joyee De and Asim K. Pal

Management Information Systems
Indian Institute of Management Calcutta, Joka, D. H. Road, Kolkata 700 104, India.
sjoyeede@gmail.com, asim@iimcal.ac.in

Abstract. Each day newer security and privacy risks are emerging in the online world. Users are often wary of using online services because they are not entirely confident of the level of security the provider is offering, particularly when such services may involve monetary transactions. Often the level of security in the algorithms underlying online and cloud-based services cannot be controlled by the user but is decided by the service provider. We propose a cloud-based Privacy Aware Preference Aggregation Service (PAPAS) that enables users to match preferences with other interested users of the service to find partners for negotiation, peer-groups with similar interests etc while also allowing users the ability to decide the level of security desired from the service, especially with respect to correct output and privacy of inputs of the protocol. It also lets users express their level of trust on the provider enabling or disabling it to act as a mediating agent in the protocols. Along with this we analyze the security of a preference hiding algorithm in the literature based on the security levels we propose for the PAPAS framework and suggest an improved version of the multi-party privacy preserving preference aggregation algorithm that does not require a mediating agent.
Keywords: Security, privacy, preference aggregation, cloud computing, multi-party computation

1 Introduction

The use of online services such as auctions, banking, shopping etc are ever-increasing. The advent of the cloud has led to the growth of these services by giving service providers access to so-called unlimited computing power, storage and the benefits of pay-per-use. However, newer security and privacy risks are also emerging each day. Users are often wary of using online services because they are not entirely confident of the level of security the provider is offering, particularly when such services may involve monetary transactions. Online services such as auctions etc hardly provide users the choice of controlling their security requirements for the underlying algorithms. Users must solely depend on the reputation of the service provider to accept that the outcome is correct and his private inputs have not been revealed to anybody else. Intense research in secure multi-party computation in the past decades has been able to provide some solutions in this respect [2, 3, 4, 6, 8, 9, 10]. However,

adfa, p. 1, 2011.

online services still do not allow users to choose their desired level of security. It may vary from user to user and the type of service being provided. In this age of rapidly growing incidents of information security breaches, providing users the ability to specify the desirable level of security can become a critical success factor for a service provider. Moreover, different users may also have different levels of trust on service providers, in our case a CSP, based on the purpose of use of such services [5].

In this paper we propose a privacy-aware preference aggregation service based on cloud which allows users to choose the level of security they desire from the underlying preference aggregation protocols as well as specify its level of trust for the cloud service provider, hence allowing or disallowing it to act as mediating agents in the preference aggregation protocols. Preference aggregation can be an important service before two or more persons enter into a negotiation process. Before a negotiation starts, users are interested in knowing the range of persons who share a common preference with them and with whom negotiations can be entered into. They would also like to find out the common set of alternatives X in which all the negotiating parties are interested. The final interest would be to find out PO, the pareto-optimal subset of X, which will form the basis of negotiation. Similarly, persons interested in finding friend groups or peers with similar interests can use a privacy aware preference aggregation service to do so without revealing their preferences to each other before the outcome is known. The work of [4] has dealt with the problem of privacy preserving preference aggregation (although called preference hiding scheme in [4]). Our work is entirely based on the protocols they suggest. The privacy-aware preference aggregation service (PAPAS) that we propose introduces the concept of allowing users to select desired level of security from the aggregation service and uses a modified version of preference hiding algorithm of [4]. Apart from formulating the PAPAS framework, we extensively analyze the security flaws of the algorithm in [4] under the security levels we propose in this framework and then suggest our version of the preference aggregation algorithm.

Why it is necessary to give users the freedom to choose security levels of an algorithm? Would they not simply go for the highest possible security level? If a certain protocol has to be run by the user itself, on its own device, then for him efficiency of the protocol may be a great concern, especially if he is using a resource constrained device. He will therefore make a trade-off between how efficient he wants the protocol to be and how much security is required for the purpose for which he is using the protocol. If a lower level of security satisfies his requirement, then he will hardly want to run a highly secure protocol that consumes all his resources. If the protocol is run on third-party resources (such as pay-per-use clouds), then cost will be a consideration for the user. If by choosing a lower level of security, his purpose is served then he will not want to go for a highly secure protocol that costs him much more. Therefore it has been our aim to provide users with some choice about how secure he wants the protocols to be, taking into account issues such as cost and efficiency.

2 Privacy Aware Preference Aggregation Service (PAPAS)

Let us look at some more examples to convince ourselves about the need of PAPAS.

Bob is interested in buying a red Chevrolet Optra but Cathy, the car-dealer wants to sell the white variety of this car model that has been in her showroom for quite a while. Now both of them wish to know whether their preferences match and further negotiations are possible or not. But, Cathy does not want to reveal her compulsion about selling the white car because that may lead Bob to try negotiating a lower price for the car. Similarly Bob is unwilling to reveal his preference of a red car as Cathy may take advantage of his preference by demanding a higher price. Bob may even wish to search for car-dealers who have selling preferences that match his.

The movie theatre is going to play six movies A, B, C, D, E and F in the weekend. Alice, Melissa and Peter would like to watch a movie together but each of them has a different preference of movies. Alice's preference can be expressed as follows: $D > E > F > B > C > A$ where '$>$' indicates 'preferred over'. Similarly, Melissa's preference is: $E > F > D > B > C > A$ and Peter's preference is: $D > C > E > F > A > B$. However, none of them wants to reveal his/her preference to anybody else. Now, the three friends want to find out the movies that they all can watch together such that no other choice would make any of them better off without making at least one of the others worse off. This means that finding the Pareto Optimal set of movies will solve their problem. So, if movie D is preferred over movie F by two of them and not by the third, then the aggregated preference says that D and F are incomparable. This is an all-or-nothing scenario where the overall preference is 'indifferent' or 'incomparable' if different participants have different preferences over the same alternatives and the overall preference is same as the preference of the participants when all of them have the same preference over same alternatives. After aggregation of preferences of all individuals for different pairs of alternatives, a suitable comparison rule (for e.g. the comparison algorithms in [4]) can find out the desired Pareto-optimal set of movies.

The next example looks at the allocation of apartments to members of a cooperative housing society which has built a complex of apartments. The members can negotiate on prices. The negotiation process becomes simpler and faster if redundant options are removed through preference aggregation and obtaining the Pareto-optimal set.

For yet another example consider a builder who has a few alternative sites each having its pros and cons. There are several possibilities of constructions (amenities and apartment sizes, etc.) along with cost and time implications for each of these sites. The builder wants to find the customer preferences before he goes on to select a particular plan for a particular site. The problem becomes more complex as he does not have any fixed set of customers to talk to. This problem can possibly be reduced to some repeated applications of preference aggregation algorithm.

In both the above scenarios, a Privacy Aware Preference Aggregation Service (PAPAS) allows the user to come in contact with other users of the service with the same preferences as him/her without revealing his/her preferences to any other user or the cloud. This cloud-based service brings the advantage of being able to establish

contact with other users of the service who have similar preferences but were previously unknown to this user while the cloud can execute the preference aggregation algorithms either as a mediating agent or perform multi-party computations on behalf of the participants. The second example may involve running a preference aggregation algorithm several times (for aggregating preferences for pair-wise alternatives) in which case the cloud is a potential resource provider. So participants with resource constrained devices such as smart phones can easily use PAPAS whenever they wish and wherever they are.

PAPAS compares different participants' or Decision Making Agents' (DMA) preferences of one alternative a over another alternative b and finds whether all of them prefer a over b (denoted by $a > b$) or b over a (denoted by $a < b$) or some of them prefer a over b and some prefer b over a (denoted by $a \sim b$) without revealing any individual's preference to others, except what may be inferred from the output itself. We call this service 'privacy aware' because the user has the flexibility to choose the level of privacy-protection he desires from the service. A privacy aware service may protect the privacy of the user's preference from adversaries of different strengths as specified by the user. As an example, we may say that in the two illustrative scenarios above, Bob may desire a higher level of privacy of personal data than Alice as he is using the service for a purpose that may lead to a large monetary transaction as opposed to Alice's case. PAPAS enables users from different parts of the world to come in contact with each other based on their preferences in a particular context and then engage in further preference aggregation among themselves either with or without the help of the service provider as a mediating agent (MA). In this paper we have only considered 'strict' preference for the preference hiding algorithm. This may not be much meaningful if there are a large number of participants. There two possibilities, either the common set of alternatives is too small and hence of little interest, or the set is not so small, but then there will be hardly any alternative which is not in the Pareto-optimal set, which also is of little consequence. In such situations we need a relaxed constraint on the 'preference' condition, e.g. majority rule. Further as already mentioned, PAPAS can also include additional services like finding intersections, comparison schemes, participants offering alternatives (with linguistic support from the service provider for uniform framing of the alternatives, for removing redundant alternatives, etc.), allowing time zone to connect people from different places, and validating the participants (through registration and authentication).

2.1 PAPAS Framework

The PAPAS framework has three layers: 1) User Interface; 2) Interpretation Middleware and 3) Operations. The first module is a user facing module that enables the user to interact with the service by allowing them to enroll in the service, specify different requirements and inputs and obtain outputs. The Interpretation Middleware interprets all forms of user specifications to enable the right choice of protocols and participants. Thirdly, the Operations layer provides the complete set of Privacy Preserving Preference Aggregation (PPA) protocols and participant choice of which only some are triggered by the Interpretation Middleware.

User Interface. The user of a preference aggregation service is concerned about the security of the preference aggregation protocol especially the privacy of his preferences. The user may also like to choose whether the cloud should participate as a mediating agent (MA) during the execution of PPA protocol depending on how much it trusts the cloud. Apart from these, the user may also like to specify an initial quality of the other participants or DMAs based on some attributes. Users interact with PAPAS through the user interface which contains the following sub-modules: 1) Enrollment; 2) Security Requirement Statement; 3) Initial Participant Requirement Statement and 4) Preference I/O (and also inputting the alternatives with linguistic support for uniform formulation of alternatives and eliminating removing alternatives).

The options for user's specifications or requirements must be simply and comprehensively expressed so that the user has no difficulty in understanding or choosing the right one. We note here that this idea is very similar to a Service Level Agreement (SLA) but is not the same. As we have already seen, PAPAS can be used for a wide range of purposes and the security and participant requirements for each use may widely vary with the particular instance of use. The user specifies its security requirements a Security Requirement Statement (SRS). Each SRS is specific to the current use of the service and is valid till the current use is over. Similarly, the user may often wish to set a quality of participants he desires for preference aggregation. For example, Bob may want to aggregate his preferences with only a well-known car dealer in city X instead of any car dealer anywhere in the world. Therefore an initial filter for participants based on certain attributes is fixed using the Initial Participant Requirement Statement. The Enrollment module enables enrollment of new users to PAPAS whereas the Preference I/O module takes the user's preference as input and provides the user with the desired output of preference aggregation.

Interpretation Middleware. The specifications in the SRS are interpreted by a Security Requirements Interpretation Service (SRIS) to translate user security requirements to protocol security requirements. Protocol security requirements enables the Interpretation Middleware to trigger the right PPA protocol based on strength of adversarial model it uses. Similarly the Participant Requirements Interpretation Service (PRIS) translates participant requirements so that enrolled participants can be filtered and enrolled users satisfying the requirements can be allowed to participate in the particular user instance of a chosen PPA protocol. Protocol security requirements enables the Interpretation Middleware to trigger the right PPA protocol based on strength of adversarial model it uses.

Operations. This layer consists of the most important requirements for the PAPAS framework: the complete set of PPA protocols and the choice of participants as gathered from the user by Interpretation Middleware. This layer is responsible for the execution of the triggered PPA protocol with chosen participants and communicating the outcome back to the User Interface.

2.2 The User Perspective

In this section we describe the user's view of the security requirements. Therefore we first express the adversarial models from the users' viewpoint both with respect to other participants and the CSP which may act as the mediating agent (MA) depending on user specification. Next, we specify the attributes for initial participant selection.

User-specified Security Requirements for DMAs. We consider that DMAs do not deviate arbitrarily from the protocol i.e. behave maliciously in true sense of the term as used in the literature of SMC. However, they may perform certain actions that are not attributed to semi-honest behavior. When a ***DMA*** is semi-honest, it does not deviate from the protocol and gathers information about user inputs by keeping track of intermediate steps [2]. Sometimes ***DMA*** s may collude with others or generation of fake random numbers instead of random ones when so desired by the protocol. Moreover, an adversary can corrupt ***DMA*** s adaptively i.e. one by one as the protocol proceeds and can thus gain more information than a non-adaptive adversary. Therefore, we allow the users to base their security requirements on whether 1) *DMA*s collude; 2) ***DMA*** s generate fake random numbers instead of random numbers or 3) ***DMA*** s are adaptively corrupted. However, we do not require the users to understand these concepts and therefore we represent the adversarial models in terms of the strength of security a user may desire. So users are only required to choose from among the three options 1) Semi-honest ***DMA*** which indicates the lowest security level; 2) Medium Semi-honest ***DMA*** that indicates a medium security level and 3) Strong semi-honest ***DMA*** which indicates a strong security level. We shall indicate in a later section how these user security requirements are mapped to protocol security requirements by the SRIS component of the Interpretation Middleware.

User-specified Security Requirements for Cloud (MA). The user's main concern is the privacy of his preference from other users or DMAs. However since the CSP offers PAPAS with regard to preference hiding, users impose some minimal levels of trust for the cloud. Accordingly, the user may either require the CSP to act as an MA in the PPA protocol used or require the CSP to only use such PPA protocols that do not require an MA. In the later case, the Virtual Machines (VMs) in the CSP engage in a multi-party computation without any mediating VM (referred to as VM MA) while in the former case they take the help of a VM MA. When the CSP acts as an MA, the cost to users will be much less than when several VMs engage in multi-party computation. In both cases, the user tries to protect itself from ***DMA*** s of varied level of corruption but finally the user has to make a trade-off between his trust on the CSP as an MA and the costs involved in choosing the CSP to act as an MA and using VMs for multi-party computations.

Thus for the SRS, the user must specify two requirements on the CSP. First, whether it wishes the CSP to participate as an MA or not and second, if the CSP is required to participate as an MA then what level of trust the CSP is assigned i.e. 1) semi-trusted MA and 2) untrusted MA. We must mention here that the Interpretation Middleware is always assumed trusted in the sense that it will always interpret user requirements correctly and trigger the right protocol according to user requirements. It

will not try to manipulate the user security requirements, for e.g., by setting the CSP to be semi-trusted MA when actually the user chose untrusted MA.

When the cloud is chosen to act as an MA, then all ***DMA***s must provide their inputs to the MA after suitable modifications and the MA after computing the aggregated preference will declare the output to the ***DMA***s. On the other hand, when the cloud cannot act as an MA, each ***DMA*** is assigned a VM. The VMs, on behalf of the ***DMA***s, exchange messages and perform the required computations. In this scenario, each VM is assumed to have the same adversarial characteristics as the ***DMA*** to which it is assigned.

User-specified Initial Participant Requirement. The user must specify what kind of participants it wants to match its preference with on the basis of different attributes. However, the attributes will depend on the nature of use of PAPAS. If the user is using PAPAS to search for car-dealers with similar preferences then the attributes required by him to filter his initial participants will be much different from the ones he requires if he is using PAPAS for movie preference matching. Therefore, the user has to first specify the broad class of participants he is looking for. During enrollment all users are provided the option to include himself in different classes of participants based on his interest in searching and being searched for preference matching. Once this broad class is specified, the user gets options suitable to the class he specified. If he specified 'car-dealers' as his interest then the attribute options he gets are location, reputation etc of car-dealers and he must specify the required level for each of these attributes.

2.3 The Protocol Perspective

The PPA protocols used in PAPAS can tolerate one of the following adversarial behaviors: 1) there is no collusion i.e. only one ***DMA*** acts as a non-adaptive adversary and it generates random numbers when required by the protocol (NC/R/NA); 2) a single ***DMA*** or a group of ***DMA*** s (i.e. collusion does not matter) is controlled by the adversary who is non-adaptive and the adversary generates numbers of its choice when actually the protocol requires random number generation (NR/NA); 3) there is collusion i.e. a group of ***DMA*** s is controlled by the adversary who is non-adaptive and the adversary generates random numbers when required by the protocol (C/R/NA); 4) there is collusion i.e. a group of ***DMA*** s is controlled by the adversary who is adaptive and the adversary generates random numbers when required by the protocol (C/R/A) and 5) a single ***DMA*** or a group of ***DMA*** s (i.e. collusion does not matter) is controlled by the adversary who is adaptive and the adversary generates numbers of its choice when actually the protocol requires random number generation (NR/A).

Table 1. Mapping Table from User Security Requirements for ***DMA*** to Protocol Security Requirements for ***DMA***

User Security Requirements for ***DMA*** s	Protocol Security Requirements for ***DMA*** s

Semi-honest ***DMA***	NC/R/NA
Medium Semi-honest ***DMA***	C/R/NA, NR/NA
Strong Semi-honest ***DMA***	C/R/A, NR/A

The SRIS component of the Interpretation Middleware uses the following mapping table to translate user security requirements for ***DMA*** s to protocol security requirements for ***DMA*** s.

The PRIS component of Interpretation Middleware finds out the subset of enrolled users with which a user can participate in a PPA protocol based on the attributes specified by the user. So it defines which ***DMA*** s can participate in a particular use by a particular user of a PAPAs service.

3 Privacy Preserving Preference Aggregation Protocols

Parties (persons, organizations etc) in a negotiation process want to reach a consensus without revealing their constraints on the acceptability and desirability of the available choices thus preventing unacceptable information disclosure about the business operations or strategies that they have adopted. If the information exchange essential to the negotiation process occurs through electronic media, then the process is called e-negotiation. Negotiators may also take help of support tools to arrive at a better decision or the whole process of negotiation may be fully automated [7].

[4] attacks the problem of finding the Pareto-optimal frontier in multi-party negotiations allowing minimum information disclosure using the solutions to secure multi-party computation problems in set theory, linear programming etc. Therefore, various privacy preserving algorithms for the different steps of solving this problem have been discussed. The steps are: 1) finding out the feasible space by using intersection algorithms and 2) finding the Pareto-optimal subset. The algorithms for the latter step consists of two major components 1) Preference Hiding Schemes and 2) Comparison Schemes. After the Comparison Scheme (that repeatedly calls the Preference Hiding Protocol), a set of non-dominated alternatives i.e. the Pareto-optimal set is arrived at. The algorithms have been designed considering that the negotiating parties i.e. the decision making agents (***DMA*** s) and the mediating parties i.e. the mediating agents (MA) in the negotiation process are semi-honest. There is no extensive discussion on the effect of collusions etc on the algorithms or different possible security scenarios. We take this opportunity to use the preference hiding scheme proposed by [6] and propose a Cloud-based Privacy Aware Preference Aggregation Service by taking into account the several possible adversary models the user may like to consider while using such a cloud-based service.

3.1 XOR-based Preference Aggregation Protocol

The XOR-based Preference hiding Algorithm proposed by [4] is not secure under the adversarial models we have considered as there is either partial or full information disclosure (See Appendix for details). We present here a security analysis of their

protocol based on the adversary models we have proposed and suggest a modification of this algorithm that will remove the drawbacks. The main motivation behind the modification is to remove the role of the central ***DMA*** which proves to be a notorious give-away.

Before we analyze the security of the algorithm considering the adversarial models as proposed in section 2.3, we provide a brief overview of the algorithm. One of the participants i.e. DMAs (call it the central DMA or DMA^c) generates random representations e_1^i for the preference relation $a > b$ and e_2^i for the preference relation $a < b$ for each DMA^i $(i = 1 \dots \dots m)$ distributes these representations to the corresponding DMA. Next each DMA^i generates $m - 1$ random numbers and by using these random numbers as masks, computes $h_+ = \bigoplus_i e_1^i$ to indicate $a > b$ and $h_- = \bigoplus_i e_2^i$to indicate $a < b$ in a distributed manner. Afterwards, depending on its preference, DMA^i sets the value of its preference e^i to be either e_1^i or e_2^i. The earlier step using random numbers as mask is repeated to calculate in a distributed manner the aggregated preference i.e. $h = \bigoplus_i e^i$. We shall see that information disclosure occurs mainly because of the fact that the single DMA, the central DMA or DMA^c, generates the random pairs of binary vectors that correspond to the choices of each DMA. This makes DMA^c powerful in terms of the information about how each DMA's choice is represented. Notably, the other DMA s only know the representation of their own choices and nothing about how other DMA s' choices are represented. For DMA^c, deducing some information about the choice of at least some of the other DMAs, is not very difficult as we will show below. Also, when DMA^c retains the right to generate the random representations of the preferences, it may as well calculate the values of h_+ and h_- itself and distribute it to the other DMA s. The only advantage of each DMA calculating it on its own is that DMA^c would not be able to manipulate this calculation somehow or give incorrect or different values for h_+ and h_- to the DMA s. The authors in [4] do not clearly mention the adversary model they have considered.

We must note here that DMA^c should either be elected or randomly selected at the beginning of each instance of the algorithm (although authors in [4] do not mention this). In the whole negotiation process, the comparison of alternatives occurs a number of times and each time the preference hiding algorithm is to be used to secretly compute the overall preference of the DMA s. We may have to either assume that an adversary controlling a fixed number of DMA s exists before the negotiation process begins (non-adaptive) or that the adversary can select whom to corrupt as the protocol proceeds. The adversary's success lies in its ability to either begin with such a set of parties one of whom will be selected as DMA^c or corrupt the party chosen as DMA^c later on. If a DMA is randomly selected to be DMA^c, then beginning with a fixed set of corrupted parties does not guarantee that will belong to that set. If a DMA is elected to the position of DMA^c, then the adversary may influence the election process in such a way so that one of the colluding DMA s gets selected as DMA^c. In case of adaptive adversary, however, corrupting the central DMA later is easier than in the non-adaptive case.

For the rest of the analysis, we consider that there are a total of m DMA s of which c collude and b bits are used to represent the choices of each DMA. For results, we shall see that whenever we consider that adversaries can generate fake random num-

bers without any other deviation from the protocol, full information disclosure occurs. Also since the success of the adversary depends on whether it is able to corrupt DMA^c, the adaptive adversary gains a clear advantage over non-adaptive ones. Collusion helps the adversary to some extent when it is absolutely non-deviating (i.e. C/R/NA). Below we present the analysis for each adversary model.

NC/R/NA. In this case, DMA^c will generate the binary vectors randomly but will still be able to know some information about the choices of some of the other DMA s. We show this by an example (see Appendix). However, DMA^c will not know beforehand for which of the DMAs it will be able to know the exact choice. This is a disadvantage for it. However, if DMA^c is chosen randomly from the set of DMAs, the chances of the adversary of corrupting DMA^c becomes very low. The adversary gains no information by corrupting any other DMA.In the worst case, full information disclosure is possible (for explanation, see Appendix).

NR/NA. In this scenario, the adversary (if it is DMA^c) instead of generating random representations of preferences, can construct them in such a way so as to help it in maximizing its information gain. We see that in this scenario causing full information disclosure is very easy and it is independent of the presence of collusion. In this case also, such disclosure is possible only if the adversary is able to corrupt DMA^c chances of which are very low for reasons we have already mentioned.

C/R/NA. The collusion benefits only when DMA^c is a part of it. A collusion of c DMAs including DMA^cmakes it easier for the adversary to derive information about the exact choice of at least some of the non-colluding DMAs. So we see here that with collusion that includes DMA^c, there is quite a significant information disclosure. In the worst case, full information disclosure is possible.

C/R/A. This case is similar to C/R/NA with the exception that the adversary is now able to choose whom to corrupt at any point during the protocol execution which implies that it is able to corrupt DMA^c almost certainly once the latter has been selected randomly. Information disclosure is same as that of C/R/NA.

NR/A. This case is similar to NR/NA with the exception that the adversary is now able to choose whom to corrupt at any point during the protocol execution which implies that it is able to corrupt DMA^c almost certainly once the latter has been selected randomly. This situation is definitely the worst possible scenario as it leads to full information disclosure with very high probability. The information disclosure is independent of whether there is collusion or not.

Now we propose our version of the XOR-based Preference Aggregation Protocol that we use as one of the PPA protocols in PAPAS. We use the same notations as in [6].

Our XOR-based Preference Aggregation Protocol without MA

Input: Alternatives a and b
Output: $a > b$ or $a < b$ or $a \sim b$
Condition: No mediators
Set up Phase: Each DMA^i generates a pair of b-bit binary vectors $(e_1^i, e_2^i), i = 1, \ldots\ldots, m$. The number of bits denoted by b is fixed previously depending on the number of DMAs and is sufficiently large. This vector pair is to be kept secret by each DMA.
Calculation of h_+ and h_-:
The DMAs need to calculate $h_+ = \bigoplus_i e_1^i$ to indicate $a > b$ and $h_- = \bigoplus_i e_2^i$to indicate $< b$. This can be done as follows:

Each DMA^i

1. Generates $m-1$ random vectors $r_1^i, \ldots\ldots\ldots, r_{m-1}^i$.
2. Finds $r^i = e_1^i \oplus r_1^i \oplus r_2^i \oplus \ldots\ldots\ldots \oplus r_{m-1}^i$.
3. Sends randomly one vector p^{ij} from the set of vectors $\{r_1^i, \ldots\ldots\ldots, r_{m-1}^i, r^i\}$ to each DMA^j where $j = 1, \ldots\ldots, m; j \neq i$ and retains one with itself. No vector is sent to more than one DMA.
4. Finds XOR of the vectors received in the previous step i.e. it obtains $\alpha^i = \bigoplus_j p^{ji}$ and sends it to all other DMAs.
5. Finds XOR of the vectors received in the previous step i.e. it obtains $h_+ = \bigoplus_i \alpha^i$.

The above process is repeated for calculation of h_-.

Preference Aggregation Phase:

Each DMA^i performs the following:

1. Finds $e^i = \begin{cases} e_1^i \; when \; v^i(a) \geq v^i(b) \\ e_2^i \; when \; v^i(a) < v^i(b) \end{cases}$ where (e_1^i, e_2^i) is a random pair of binary vectors received in the Set up phase.
2. Generates $m-1$ random vectors $r_1^i, \ldots\ldots\ldots, r_{m-1}^i$.
3. Finds $r^i = e^i \oplus r_1^i \oplus r_2^i \oplus \ldots\ldots\ldots \oplus r_{m-1}^i$.
4. Sends randomly one vector p^{ij} from the set of vectors $\{r_1^i, \ldots\ldots\ldots, r_{m-1}^i, r^i\}$ to each DMA^j where $j = 1, \ldots\ldots, m; j \neq i$ and retains one with itself. No vector is sent to more than one DMA.
5. Finds XOR of the vectors received in the previous step i.e. it obtains $\alpha^i = \bigoplus_j p^{ji}$ and sends it to all other DMAs.
6. Finds XOR of the vectors received in the previous step i.e. it obtains $h = \bigoplus_i \alpha^i$.
7. If $h = h_+$ then $a > b$ else if $h = h_-$ then $a < b$ else $a \sim b$.

When more comparisons are to be made then this whole process is to be repeated.

Security Analysis. The adversary can take advantage of the XOR-based algorithm by [4] because of the use of DMA^c which assumes the important role of generating the representation of the choices of each of the DMAs. The role of DMA^c has been removed in our algorithm and each DMA is now allowed to randomly generate its own binary vector i.e. the representation of its own choice. This ensures that none of the DMAs will have with it all the representations and hence knowing the individual choices becomes impossible even when some of them collude. No information disclosure takes place under adaptive or non-adaptive adversaries.

4 Conclusion and future work

In this paper we have presented a Cloud-based Privacy Aware Preference Aggregation Service that allows enrolled users to search for other unknown users of the service and find out whether preferences in a certain context match. The preference matching can occur among multiple users depending upon the use-case and the user has the flexibility to choose the security level of the protocols used to prevent privacy-breach of his preferences and the types of participants preferred for the preference aggregation. Our framework may be looked upon as a precursor to a cloud based Privacy Aware Negotiation-as-a-Service that will allow users who previously did not know each other to negotiate on their preferences to reach a consensus without revealing their preferences to anyone. Algorithms of different security level, efficiency and with different preference aggregation rules can be integrated into this service. We are working towards this end.

Existing online auction services such as eBay.com etc do provide a certain platform of negotiation between buyers and sellers but they focus on a very specific use i.e. buying and selling of goods and do not allow the flexibility to users to choose the security level of the service. In contrast, PAPAS is a highly flexible cloud-based service which allows users to choose their desired level of security and types of participants they want to interact with apart from providing a generalized arena for preference aggregation on any use-case starting from choice of movies, restaurants among friends to buying and selling of cars, real estate etc. The process of choosing security levels for algorithms must be made comprehensive for users so that users with little or no knowledge about security can effectively express their choices.

References

1. Goldreich, O. (2004). Foundations of Cryptography Volume II Basic Applications. Cambridge, UK: Cambridge University Press.
2. Chakraborty, S., Sehgal, S. K. & Pal, A. K. (2005). Privacy preserving e-negotiation Protocols based on secure multiparty computation. In *Proceedings of the IEEE SoutheastCon 2005 (pp. 455-461).* Springer.

3. Saroop, A., Sehgal, S. K. & Ravikumar, K. (2007). Multi-Attribute Auction Format for Procurement with Limited Disclosure of Buyer's Preference Structure. In *Decision Support for Global Enterprises Annals of Information Systems Vol. 2 (pp. 257-267)*. Springer.
4. Sehgal, S. K. & Pal, A. K. (2004). Finding Pareto Optimal Set of Distributed Vectors with Minimum Disclosure. In *Proceedings of the 6th International Workshop on Distributed Computing (pp. 144-149)*. Springer.
5. De, S., Saha, S. & Pal, A. K. (2013). Achieving Energy Efficiency and Security in Mobile Cloud Computing. In *Proceedings of the 3rd International Conference on Cloud Computing and Services Sciences CLOSER 2013*. SciTePress.
6. Du, W., & Zhan, Z. (2002). A Practical Approach to Solve Secure Multi-party Computation Problems. In *NSPW'02 Proceedings of the 2002 workshop on New security paradigm* . (pp. 127-135).ACM.
7. Bichler, M., Kersten, G., & Strecker, S. (2003). Towards a Structured Design of Electronic Negotiations. In *Group Decision and Negotiation Vol. 12, No. 4* (pp. 311-335).
8. Naor, M., Pinkas, B., & Sumner, R. (1999). Privacy Preserving Auctions and Mechanism Design. *Proceedings of the 1st ACM Conference on Electronic Commerce (pp. 129-139)*. ACM.
9. Cramer, R., Gennaro, R., & Schoenmakers, B. (1997). A secure and Optimally Efficient Multi-Authority Election Scheme. In *European Transactions on Telecommunications Vol. 8, Issue 5* (pp. 481-490).
10. Kamara, S., Raykova, M. (2011). Secure Outsourced Computation in Multi-tenant Cloud. In *WCSC'11, IBM Workshop on Cryptography and Security in Clouds*.

Appendix

Here we present in details the security analysis of XOR-based preference hiding scheme of [4] under NC/R/NA and NR/NA we propose. We have not shown the rest for lack of space.

NC/R/NA. Let us suppose that there are ten DMAs among which one has been elected or randomly chosen as DMA^c. Now, DMA^c (say DMA^5 is DMA^c) generates the binary vector pair $(e_1^i, e_2^i), i = 1, \ldots\ldots, m$ randomly. We assume that each vector is of 10-bit length.

After DMA^c randomly distributes these vector pairs to each of the DMAs, the following is the random allocation:

Table 2. Randomly Generated Vectors for Input Representation as distributed to each DMA by DMA^c

DMA	e_1^i	e_2^i
DMA^1	00 1001 1011	01 0101 0101
DMA^2	01 1110 1001	11 1010 1111
DMA^3	00 0011 0101	10 0110 1001
DMA^4	11 1111 1000	10 1010 0000

DMA^5 (DMA^c)	11 0110 0101	00 0010 1100
DMA^6	10 1100 1010	01 1111 1010
DMA^7	10 0000 0011	00 0101 0011
DMA^8	10 0101 1100	01 1100 0000
DMA^9	01 1111 1001	11 0111 0001
DMA^{10}	00 1000 0001	00 1001 0111

h_+ and h_- are calculated as follows:

$$h_+ = \bigoplus_i e_1^i = 10\ 0011\ 0111$$
$$h_- = \bigoplus_i e_2^i = 01\ 1001\ 0000$$

At the end of the algorithm, each of the DMAs comes to know about $h = \bigoplus_i \alpha^i = \bigoplus_i e^i$

Let us now tabulate the choices of the DMAs.

Table 3. Choices of each DMA

DMA	Choices
DMA^1	00 1001 1011
DMA^2	01 1110 1001
DMA^3	10 0110 1001
DMA^4	11 1111 1000
DMA^5 (DMA^c)	11 0110 0101
DMA^6	01 1111 1010
DMA^7	00 0101 0011
DMA^8	01 1100 0000
DMA^9	11 0111 0001
DMA^{10}	00 1000 0001

Given these choices, we have $h = \bigoplus_i \alpha^i = \bigoplus_i e^i = 00\ 0001\ 1111$.

Now given the values of h and $\left(e_1^i, e_2^i\right)$ we show below that it is possible for DMA^c to find out the exact choices for at least some of the DMAs. The following table shows the bits (represented by X) where the representations of each of the choices of each DMA (except DMA^c) vary. Also we may consider that this table is available to DMA^c.

Table 4. Adversary's Analysis Table

DMA	Varying bits in Choice
DMA^1	0X XX01 XXX1
DMA^2	X1 1X10 1XX1
DMA^3	X0 0X1X XX01
DMA^4	1X 1X1X X000
DMA^5 (DMA^c)	11 0110 0101
DMA^6	XX 11XX 1010

DMA^7	X0 0X0X 0011
DMA^8	XX X10X XX00
DMA^9	X1 X111 X001
DMA^{10}	00 100X 0XX1
Value of h	**00 0001 1111**

Using the property of XOR that odd number of true values (1s) when XORed gives true value (1), DMA^c can deduce the following from the table above: 1) the second bit of one or all of DMA^1, DMA^2 and DMA^{10} are 1; 2) the third bit of DMA^1, DMA^2,DMA^3, DMA^8 and DMA^{10} are all 0 or that of two or four of them are 1; 3) the fourth bit of three or all of DMA^1, DMA^3, DMA^4, DMA^8 and DMA^9 are 1 etc.

Although DMA^c comes to know these pieces of information, it is difficult for it to know the exact choice of any of the other DMAs. Since the above table can be derived only by DMA^c (because only DMA^c knows all the options of all the DMAs), no other DMA can derive these pieces of information.

The worst case here will arise when the adversary generates the representations of choices such that within each pair, the choices differ by a single bit and across pairs the position of difference varies. However, since the pairs have to be randomly generated, the probability of being able to generate the representations according to the above condition will be $\frac{b.(b-1).....(b-m-1)}{b^m}$ where m is the number of DMAs and b is the number of bits used in the representations of choices. We must note here that $2^b >> 2m$.

NR/NA. In this scenario instead of generating $(e_1^i, e_2^i), i = 1, \ldots\ldots, m$ randomly, the adversary (if it is DMA^c) can construct the vectors in such a way so as to help it in maximize its information disclosure. We give an example below.

Let us consider that the adversary constructs and distributes the binary vectors according to the following table:

Table 5. Vectors for Input Representation

DMA	e_1^i	e_2^i
DMA^1	01 0000 0000	11 0000 0000
DMA^2	00 1000 0000	01 1000 0000
DMA^3	00 0100 0000	00 1100 0000
DMA^4	00 0010 0000	00 0110 0000
DMA^5 (DMA^c)	00 0000 0000	11 1111 1111
DMA^6	00 0001 0000	00 0011 0000
DMA^7	00 0000 1000	00 0001 1000
DMA^8	00 0000 0100	00 0000 1100
DMA^9	00 0000 0010	00 0000 0110
DMA^{10}	00 0000 0001	00 0000 0011

$$h_+ = \bigoplus_i e_1^i = 01\ 1111\ 1111$$
$$h_- = \bigoplus_i e_2^i = 01\ 1111\ 1110$$

Let the actual choices of the DMAs be as follows:

Table 6. Choice of each DMA

DMA	Choice
DMA^1	01 0000 0000
DMA^2	01 1000 0000
DMA^3	00 1100 0000
DMA^4	00 0010 0000
DMA^5 (DMA^c)	00 0000 0000
DMA^6	00 0011 0000
DMA^7	00 0001 1000
DMA^8	00 0000 1100
DMA^9	00 0000 0110
DMA^{10}	00 0000 0001
Value of h	**00 0100 0011**

Given these choices, we have $h = \bigoplus_i \alpha^i = \bigoplus_i e^i = 00\ 0100\ 0011$.

Now, as before, the adversary deduces the following table from its knowledge about the representation of the choices of each of the DMAs.

Table 7. Adversary's Analysis Table

DMA	Choice
DMA^1	X1 0000 0000
DMA^2	0X 1000 0000
DMA^3	00 X100 0000
DMA^4	00 0X10 0000
$\boldsymbol{DMA^5}$ **(**$\boldsymbol{DMA^c}$**)**	**00 00X1 0000**
DMA^6	00 0000 0000
DMA^7	00 000X 1000
DMA^8	00 0000 X100
DMA^9	00 0000 0X10
DMA^{10}	00 0000 00X1
Value of h	**00 0100 0011**

Now, it becomes easy for the adversary to deduce the exact choice made by each of the other DMAs using the property of XOR mentioned previously. So the adversary comes to know 1) the eighth bit of DMA^1 is 0 and its choice is e_1^1. Hence DMA^1 prefers$a > b$; 2) The seventh bit of DMA^2 must be 1 and its choice is e_2^2. So it prefers $a < b$. Similarly, the adversary comes to know the choices of each of the DMAs.