A Method And System For Sharing Data

FIELD OF INVENTION

The present invention relates broadly to a method of sharing data between a first and a second party, to a system for sharing data between a first and a second party and to a computer readable data storage medium having stored thereon computer code means for instructing respective computer processors of a first party and a second party to execute a method of sharing data between the first and the second parties.

BACKGROUND

Sharing proprietary information across private databases belonging to autonomous or independent parties can be essential for decision making applications. For example, two or more countries may wish to share information of terrorist suspects. However, it is typically not feasible for one country to share the information of all its terrorist suspects with another. It is typically desired to find out the common suspects that both countries/parties are monitoring before sharing information about these suspects. In other words, one step for privacy-preserving information sharing is to allow queries to be executed across databases belonging to autonomous parties/entities to find out what records are to be shared in such a way that no other records are revealed, other than what is common among the parties/participants.

To maintain the privacy and secrecy of the databases, each of the participants encrypts its respective private dataset and then exchange the corresponding encrypted dataset/database with another party. Typically, the participating parties involved in privacy preserving information sharing protocols use commutative encryption that executes a set of instructions. Due to underlying characteristics of commutative encryption, none of the parties can sense any individual transactions or records unless these transactions are common in both databases.

It is noted that the instructions in commutative encryption are tightly coupled, meaning that the instructions are executed in ordered sequences. If the exact order is not followed, it is typically technically impossible to find the resultant intersection set. Such a technical limitation typically compels every participating party to execute the instructions/protocol in exactly the same sequence without knowing whether the other party follows it or not. Such a protocol can work with a so-called *honest-but-curious* setting where it is assumed that every party follows the protocol. As none of the participants is able to verify whether the other party has fully followed the protocol or not, it is possible for a particular participating party/site to find the resultant set without letting the other party know the common transactions in their respective private databases.

Provided below is a brief description of a typical information sharing process between two sites.

Assume that there are two sites S and R that have datasets D_S and D_R respectively. At a first step, both sites S and R apply a hash function h to their private datasets respectively, i.e. $D'_S[i]=h(D_S[i])$ and $D'_R[j]=h(D_R[j])$, and then randomly choose a secret key, ie. e_S for site S and e_R for site R. Site S then uses its secret key e_S on the hashed dataset and generates its encrypted dataset $D''_S[i]=f_\varphi(D'_S[i],e_S)$, where f is a commutative encryption function defined as $f_\varphi(x,e)=x^e \bmod \varphi$. Similarly, site R generates its encrypted dataset D''_R. Next, to carry out the actual intersection i.e. to find out the common elements, either site S or R sends its encrypted dataset to the other site. Assume that it is site S that transmits its encrypted dataset D''_S to R. Upon receiving D''_S, site R carries out two distinct tasks. Firstly, site R uses its secret key e_R to encrypt each entry $d \in D''_S$ of D''_S such that $\overline{D_S}[i]=f_\varphi(D_S''[i],e_R)$. Site R then sends a pair $\langle D''_S, \bar{D}_S\rangle$ to site S and then sends its own encrypted set D''_R to site S. Upon receiving D''_R, site S encrypts each entry $d \in D''_R$ of D"$_R$ with secret key e_S such that $\overline{D_R}[j]=f_\varphi(D_R''[j],e_S)$. Since at this stage, site S possesses the two sets that are $\bar{D}_S$ and $\bar{D}_R$, site S is able to intersect all common elements between D_S and D_R. Although site S can already obtain a resultant intersection set at this stage, site R does not have any knowledge about the common elements. In order to discover the resultant intersection set, site R is totally

reliant upon site *S*. In fact, it is possible to have a scenario whereby site *S* manipulates or deliberately misleads site *R* about the resultant intersection set, such that the benefits of mutual information sharing is only attained by site *S*. Furthermore, even if site *R* enforces site *S* to send the pair $\langle D''_R, \bar{D}_R \rangle$ to it, site *S* can still mislead site *R* if S encrypts each entry $d \in D''_R$ of D''_R with another secret number e_W such that $e_S \neq e_W$. If S uses a different secret number and sends the encrypted set back to R, one disadvantage is that R would not be able to tell that S is dishonest. In other words, R would simply have no intersection of entries (due to the different secret number used by S) and would arrive at a conclusion that there are no common elements with S. Indeed, such a scenario typically raises a critical question about the usefulness of information sharing. That is, unless all participating sites achieve the same foreseeable benefits where none of the sites are able to mislead each other, typical distrusting parties would not be willing to share their data.

Hence, there exists a need for a method of sharing data between a first and a second party, a system for sharing data between a first and a second party and a computer readable data storage medium having stored thereon computer code means for instructing respective computer processors of a first party and a second party to execute a method of sharing data between the first and the second parties that seek to address at least one of the above problems.

SUMMARY

In accordance with a first aspect of the present invention, there is provided a method of sharing data between a first and a second party, the method comprising the steps of: performing respective randomization processes on data sets of the first and second parties; performing an exchange process between the first and second parties; performing an audit trail check process at the first and second parties respectively; and proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can

determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

The respective randomization processes may comprise obfuscating the data sets using respective obfuscating numbers of the first and second parties; concatenating the obfuscated data sets with respective audit trail elements of the first and second parties; and randomly shuffling the concatenated data sets of the first and second parties.

The method may further comprise, prior to the obfuscating step, the steps of: hashing the data sets of the first and second parties; and encrypting the hashed data sets of the first and second parties.

The exchange process may comprise exchanging the randomly shuffled data sets between the first and second parties; re-encrypting the exchanged randomly shuffled data sets at the first and second parties respectively; re-obfuscating the re-encrypted data sets using the respective re-obfuscating numbers at the first and second parties; and exchanging the re-obfuscated data sets between the first and second parties.

The exchange process may further comprise generating respective temporary numbers at the first and second parties; exchanging the temporary numbers between the first and second parties; encrypting the exchanged temporary numbers at the first and second parties respectively; and wherein the re-obfuscating step is based on the encrypted temporary numbers and the respective obfuscating numbers of the first and second parties.

The audit trail check process may comprise sharing respective encrypted common trail generators between the first and second parties; sharing respective modulo function values based on the encrypted temporary numbers and the obfuscating numbers between the first and second parties; computing respective re-obfuscated audit trail sets at the first and second parties based on the shared encrypted common trail generators and modulo function values; and performing the

respective audit trail checks at the first and second parties based on the re-obfuscated audit trail sets and the re-obfuscated data sets.

The matching process may comprise sharing the respective re-obfuscating numbers between the first and second parties; verifying the respective shared re-obfuscating numbers at the first and second parties respectively; re-generating the other party's re-obfuscated data set at the first and second parties respectively based on the verified re-obfuscating numbers; and determining the common records between the first and second party based on intersecting the re-generated re-obfuscated data set of the other party with the party's own re-obfuscated data set.

In accordance with a second aspect of the present invention, there is provided a system for sharing data between a first and a second party, the system comprising means for performing respective randomization processes on data sets of the first and second parties; means for performing an exchange process between the first and second parties; means for performing an audit trail check process at the first and second parties respectively; and means for proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

The means for performing respective randomization processes may be arranged to obfuscate the data sets using respective obfuscating numbers of the first and second parties; concatenate the obfuscated data sets with respective audit trail elements of the first and second parties; and randomly shuffle the concatenated data sets of the first and second parties.

The means for performing respective randomization processes may be further arranged to hash the data sets of the first and second parties; and encrypt the hashed data sets of the first and second parties.

The means for performing an exchange process may be arranged to exchange the randomly shuffled data sets between the first and second parties; re-encrypt the exchanged randomly shuffled data sets at the first and second parties respectively; re-obfuscate the re-encrypted data sets using the respective re-obfuscating numbers at the first and second parties; and exchange the re-obfuscated data sets between the first and second parties.

The means for performing an exchange process may be further arranged to generate respective temporary numbers at the first and second parties; exchange the temporary numbers between the first and second parties; encrypt the exchanged temporary numbers at the first and second parties respectively; and wherein the re-obfuscation of the re-encrypted data sets is based on the encrypted temporary numbers and the respective obfuscating numbers of the first and second parties.

The means for performing an audit trail check process may be arranged to share respective encrypted common trail generators between the first and second parties; share respective modulo function values based on the encrypted temporary numbers and the obfuscating numbers between the first and second parties; compute respective re-obfuscated audit trail sets at the first and second parties based on the shared encrypted common trail generators and modulo function values; and perform the respective audit trail checks at the first and second parties based on the re-obfuscated audit trail sets and the re-obfuscated data sets.

The means for proceeding with performing a matching process may be arranged to share the respective re-obfuscating numbers between the first and second parties; verify the respective shared re-obfuscating numbers at the first and second parties respectively; re-generate the other party's re-obfuscated data set at the first and second parties respectively based on the verified re-obfuscating numbers; and determine the common records between the first and second party based on intersecting the re-generated re-obfuscated data set of the other party with the party's own re-obfuscated data set.

In accordance with a third aspect of the present invention, there is provided a computer readable data storage medium having stored thereon computer code means for instructing respective computer processors of a first party and a second party to execute a method of sharing data between the first and the second parties, the method comprising the steps of: performing respective randomization processes on data sets of the first and second parties; performing an exchange process between the first and second parties; performing an audit trail check process at the first and second parties respectively; and proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

Figure 1 is a schematic diagram illustrating a data matching protocol in an example embodiment.

Figure 2 is a schematic flowchart illustrating a method of sharing data between a first and a second party in an example embodiment

Figure 3 is a schematic diagram illustrating a system for sharing data between system components of a first party and system components of a second party in an example embodiment.

Figure 4 is a schematic diagram illustrating a computer system for implementing an example embodiment.

DETAILED DESCRIPTION

In an example embodiment, a method is provided for detecting whether a participant employs hidden manipulation when executing a protocol. The example embodiment can provide a capability to audit a full execution history without the need to use a trusted third party to identify if any manipulation has occurred during the course of the protocol. Thus, the example embodiment can allow a honest party to restrict other participants from obtaining any resultant intersection set if an audit trial fails.

The method of the example embodiment combines multiple distributed datasets in a privacy-preserving manner whereby each of the participating data sites match or intersect its respective dataset with the other datasets without revealing any records other than the resultant intersection set.

Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "scanning", "calculating", "determining", "replacing", "generating", "initializing", "outputting", or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the the computer system into other data similarly represented as

physical quantities within the computer system or other information storage, transmission or display devices.

The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a conventional general purpose computer will appear from the description below.

In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention.

Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the preferred method.

The invention may also be implemented as hardware modules. More particular, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. Those skilled in the art will appreciate that the system can also be implemented as a combination of hardware and software modules.

The example embodiment provides a data matching protocol that has four distinct phases: (i) randomization, (ii) exchange, (iii) audit, and (iv) matching.

During the first phase (i.e. randomization), each of the data sharing participants locally generates an encrypted dataset randomly shuffled with an audit trail set. In the second phase (i.e. exchange), the participants exchange their respective encrypted datasets and other pertinent information (such as temporary numbers, temporary secrets, encrypted obfuscated numbers and their respective re-encrypted results) with each other. In the third phase (i.e. audit), each of the participants evaluates the honesty of the other participants using the information that they have received from the other participants. If the audit phase is successful for all participants, each participant then computes the resultant intersection sets in the final phase (i.e. matching).

Figure 1 is a schematic diagram illustrating a data matching protocol in one example embodiment.

Denote S 102 and R 104 as two participating sites that have datasets D_S 106 and D_R 108 of sizes n_S and n_R respectively. Before initiating the protocol, both sites S 102 and R 104 agree on using the following: a common audit trail generator ρ, 110 which is a unique value that does not exist in D_S 106 and D_R 108, a hash function h for hashing the data in the datasets D_S 106 and D_R 108, and a relatively large prime number as a public key φ. Φ is defined to be the set of prime numbers in $\left[2,\frac{(\varphi-1)}{2}\right]$, f is defined as a commutative encryption function such that $f_\varphi(x,e)=x^e \bmod \varphi$, and g is defined as a modulo

function such that $g_\varphi(x)=x \bmod \varphi$. The auditable privacy-preserving data matching protocol of the example embodiment is described below.

At a first phase or a randomization phase/process 112, both site S 102 and site R 104 apply the hash function h to create hashed datasets D'_S 114 and D'_R 116 such that

$$D'_S[i]=h(D_S[i]),\ 1\le i\le n_S \quad (1)$$

$$D'_R[j]=h(D_R[j]),\ 1\le j\le n_R \quad (2)$$

Each site S 102 and R 104 randomly chooses a secret key, i.e. $e_S\in\Phi$ for site S 102 and $e_R\in\Phi$ for site R 104. Both sites S 102 and R 104 then encrypt their respective hashed datasets using their respective secret keys to obtain encrypted datasets D''_S (see 118) and D''_R (see 120) such that

$$D''_S[i]=f_\varphi(D'_S[i],e_S) \quad (3)$$

$$D''_R[j]=f_\varphi(D'_R[j],e_R) \quad (4)$$

Each site S 102 and R 104 then generates a relatively large prime number $z_S<\varphi|z_S\neq e_S$ for site S 102 and $z_R<\varphi|z_R\neq e_R$ for site R 104 to obtain obfuscated sets $\bar{D}_S$ and $\bar{D}_R$ as follows:

$$\bar{D}_S[i]=g_\varphi(D''_S[i]\times z_S),\ 1\le i\le n_S \quad (5)$$

$$\bar{D}_R[j]=g_\varphi(D''_R[j]\times z_R),\ 1\le j\le n_R \quad (6)$$

The numbers z_S for site S 102 and z_R for site R 104 are known as respective obfuscating numbers.

Each site S 102 and R 104 also randomly chooses a set of audit trail secret keys, i.e. E_S for site S 102 and E_R for site R 104, where $E_S\subseteq\Phi$ and $E_R\subseteq\Phi$. Denote $k_S=|E_S|$ and $k_R=|E_R|$, and $E_S=\{e_1,\ldots\ldots,e_{k_S}\}$, $E_R=\{e_1,\ldots\ldots,e_{k_R}\}$. Each site S 102 and R 104 then computes an encrypted audit trail set or audit trail elements using the common trail generator ρ, ie. A_S 126 for site S 102 and A_R128 for site R 104, as follows:

$$A_S[i] = f_\varphi(\rho, e_i),\ 1 \le i \le k_S \qquad (7)$$

$$A_R[j] = f_\varphi(\rho, e_j),\ 1 \le j \le k_R \qquad (8)$$

It will be appreciated that, as the common audit trail generator ρ 110 is a unique value that does not exist in D_S 106 and D_R 108, the elements of A_S 126 and A_R128 are elements not found in the datasets D_S 106 and D_R 108.

Each site S 102 and R 104 concatenates its respective obfuscated set (see eqns (5) and (6)) with its corresponding encrypted audit trail set (see eqns (7) and (8)) to generate a set P_S for site S 102 and a set P_R for site R 104 as follows:

$$P_S = \bar{D}_S \oplus A_S = \left(\bar{D}_S[1], \cdots, \bar{D}_S[n_S], A_S[1], \cdots, A_S[k_S]\right) \qquad (9)$$

$$P_R = \bar{D}_R \oplus A_R = \left(\bar{D}_R[1], \cdots, \bar{D}_R[n_R], A_R[1], \cdots, A_R[k_R]\right) \qquad (10)$$

Each site S 102 and R 104 then creates respective randomly shuffled obfuscated sets P'_S (see 130) and P'_R (see 132) as follows:

$$P'_S[i] = g_\varphi\left(P_S[\pi_S(i)]\right),\ 1 \le i \le n_S + k_S \qquad (11)$$

$$P'_R[j] = g_\varphi\left(P_R[\pi_R(j)]\right),\ 1 \le j \le n_R + k_R \qquad (12)$$

where π_S and π_R are random shuffling functions for the respective sites S 102 and R 104.

At a second phase or an Exchange phase/process 134, site S 102 sends P'_S to site R 104 and site R 104 in turn sends P'_R to site S102 (see 136). Each site S 102 and R 104 uses its respective secret key e_S, e_R to re-encrypt the received obfuscated set, e.g. eqn (11) and eqn (12), that it has received from the other site S 102 and R 104. In other words, site S 102 computes

$$P''_R[j] = f_\varphi\left(P'_R[j], e_S\right) \qquad (13)$$

and site R 104 computes

$$P''_S[i] = f_\varphi\left(P'_S[i], e_R\right) \qquad (14)$$

Site S 102 generates a relatively large temporary number $w_S < \varphi \mid w_S \neq e_S$, computes

$$w'_S = f_\varphi(w_S, e_S) \quad (15)$$

and sends $\langle w_S, w'_S \rangle$ to site *R* 104.

Similarly, site *R* 104 generates a large temporary number $w_R < \varphi \mid w_R \neq e_R$, computes

$$w'_R = f_\varphi(w_R, e_R) \quad (16)$$

and sends $\langle w_R, w'_R \rangle$ to site *S* 102.

Each site S 102 and R 104 then re-encrypts the respective received temporary secret, ie. for site S 102,

$$w''_R = f_\varphi(w'_R, e_S) \quad (17)$$

and for site R 104,

$$w''_S = f_\varphi(w'_S, e_R) \quad (18)$$

The sites S 102 and R 104 each holds the respective re-encrypted temporary secret w''_R and w''_S for future use. It will be appreciated that the numbers w_S and w_R are not secret. However, the results in the re-encrypted values (see eqns (17 and (18)) are secret. For example, w''_R is secret to site R 104 and w''_S is secret to site S 102.

Site S 102 encrypts the prime number/obfuscating number z_S using secret key e_S, that is,

$$z'_S = f_\varphi(z_S, e_S) \quad (19)$$

and sends z'_S to site R 104. Similarly, site R 104 encrypts the prime number/obfuscating number z_R using secret key e_R, that is,

$$z'_R = f_\varphi(z_R, e_R) \quad (20)$$

and sends z'_R to site S 102.

Site S 102 then computes

$$z''_R = g_\varphi\left(f_\varphi(z'_R, e_S) \times w''_R\right) \quad (21)$$

and site R104 computes

$$z_S'' = g_\varphi\left(f_\varphi(z_S', e_R) \times w_S''\right) \quad (22)$$

Site S102 then sends z_R'' to site R 104 and site R 104 sends z_S'' to site S 102.

Upon receiving z_S'' from site R 104, site S 102 strips off one layer of encryption from z_S'' and computes,

$$\bar{z}_S = g_\varphi\left(f_\varphi^{-1}(z_S'', e_S) \times f_\varphi(w_R, e_S)\right) \quad (23)$$

Similarly, site R 104 computes

$$\bar{z}_R = g_\varphi\left(f_\varphi^{-1}(z_R'', e_R) \times f_\varphi(w_S, e_R)\right) \quad (24)$$

Each site S 102 and R 104 generates another relatively large number/secret, ie. $x_S < \varphi \mid x_S \neq e_S$ for site S 102 and $x_R < \varphi \mid x_R \neq e_R$ for site R 104. The numbers x_S for site S 102 and x_R for site R 104 are known as respective re-obfuscating numbers. Each site S 102 and R 104 computes a new re-obfuscated hashed set as follows:

$$\bar{P}_R''[j] = h\left(g_\varphi\left(P_R''[j] \times \bar{z}_S \times x_S\right)\right) \quad (25)$$

for site S 102 and

$$\bar{P}_S''[i] = h\left(g_\varphi\left(P_S''[i] \times \bar{z}_R \times x_R\right)\right) \quad (26)$$

for site R 104.

Site S 102 then sends $\bar{P}_R''$ to site R 104 and site R 104 sends $\bar{P}_S''$ to site S 102.

At a third phase or an Audit phase/audit trail check process 138, site S 102 computes

$$\rho_S = f_\varphi(\rho, e_S) \quad (27)$$

$$t_S = g_\varphi(\bar{z}_S \times x_S) \quad (28)$$

and site R 104 computes

$$\rho_R = f_\varphi(\rho, e_R) \quad (29)$$

$$t_R = g_\varphi(\bar{z}_R \times x_R) \quad (30)$$

Site S 102 then shares/sends $\langle t_S, \rho_S \rangle$ to site R 104 and site R 104 shares/sends $\langle t_R, \rho_R \rangle$ to site S 102 (see numeral 140).

Upon receiving $\langle t_R, \rho_R \rangle$ from site R 104, site S 102 computes a re-obfuscated hashed audit trail set Ω_S as follows:

$$\Omega_S[i] = h\left(g_\varphi\left(t_R \times f_\varphi\left(\rho_R, e_i\right)\right)\right),\ 1 \le i \le k_S \qquad (31)$$

Similarly, site R 104 computes Ω_R:

$$\Omega_R[j] = h\left(g_\varphi\left(t_S \times f_\varphi\left(\rho_S, e_j\right)\right)\right),\ 1 \le j \le k_R \qquad (32)$$

Site S 102 attempts to recover the re-obfuscated hashed audit trail set (see numeral 142) from the re-obfuscated hashed data set $\bar{P}_S''$ as follows:

$$\Psi_S[i] = \bar{P}_S''[\pi_S^{-1}(n_S + i)],\ 1 \le i \le k_S \qquad (33)$$

That is, the elements of the dataset D_s are not considered and the hashed audit trail set is recovered. See the number of elements (n_S + i) for $1 \le i \le k_S$ in equation (33).

If site R 104 had executed the protocol honestly during the exchange phase 134, then site S 102 obtains $\Psi_S = \Omega_S$.

Similarly, site R 104 verifies the honesty of site S 102 (see numeral 144) by computing:

$$\Psi_R[j] = \bar{P}_R''[\pi_R^{-1}(n_R + j)],\ 1 \le j \le k_R \qquad (34)$$

and then checking whether $\Psi_R = \Omega_R$ or not.

At a fourth phase or a matching phase/process 148, only if both site S 102 and R 104 have succeeded in the audit trail checks of the audit phase 138, then the sites S 102 and R 104 transmit/share their respective random numbers/re-obfuscating numbers x_S and x_R generated during the exchange phase 134 to each other (see numeral 150).

Site S 102 verifies the integrity of x_R as follows:

(i) $v_{S1} = f_{\varphi}^{-1}\left(g_{\varphi}\left(t_R \times f_{\varphi}\left(x_R, e_S - 1\right)\right), e_S\right)$ (35)

(ii) $v_{S2} = g_{\varphi}\left(f_{\varphi}^{-1}\left(\left(t_S/(x_R)\right), e_S\right) \times x_R\right)$ (36)

It is noted that, based on the principle of $x \times x^{e_S - 1} = x^{e_S}$, e_S-*1* is derived for verification of equation (35). If site R 104 sends the correct x_R, then site S 104 obtains $v_{S1} = v_{S2}$.

Similarly, site R 104 verifies the integrity of x_S as

(i) $v_{R1} = f_{\varphi}^{-1}\left(g_{\varphi}\left(t_S \times f_{\varphi}\left(x_S, e_R - 1\right)\right), e_R\right)$ (37)

(ii) $v_{R2} = g_{\varphi}\left(f_{\varphi}^{-1}\left(\left(t_R/(x_S)\right), e_R\right) \times x_S\right)$ (38)

After verifying the integrity of x_R, site S 102 applies $\bar{z}_S$ and x_R to P_R'' and re-generates a re-obfuscated hashed set of site R 104:

$$\hat{D}_S[j] = h\left(g_{\varphi}\left(P_R''[j] \times \bar{z}_S \times x_R\right)\right),\ 1 \le j \le \left|\bar{P}_R''\right| \quad (39)$$

Finally, site S 102 intersects set $\hat{D}_S$ and $\bar{P}_S''$ to find all common records between datasets D_S and D_R, (see numeral 152), namely,

$$\left\{D_S\left[\pi_S^{-1}(i)\right] \mid \bar{P}_S''[i] = \hat{D}_S[j]\right\} \quad (40)$$

In the same manner, site R 104 finds the corresponding intersection set (see numeral 152) using the following equations:

$$\hat{D}_R[i] = h\left(g_{\varphi}\left(P_S''[i] \times \bar{z}_R \times x_S\right)\right),\ 1 \le i \le \left|\bar{P}_S''\right| \quad (41)$$

$$\left\{D_R\left[\pi_R^{-1}(j)\right] \mid \bar{P}_R''[j] = \hat{D}_R[i]\right\} \quad (42)$$

Figure 2 is a schematic flowchart 200 illustrating a method of sharing data between a first and a second party in an example embodiment. At step 202, respective randomization processes are performed on data sets of the first and second parties. At step 204, an exchange process between the first and second parties is performed. At step 206, an audit trail check process is performed at the first and second parties respectively. At step 208, only after a successful audit trail check by each party in the audit trail check process, a matching process is

performed at the first and second parties respectively and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

Figure 3 is a schematic diagram illustrating a system 300 for sharing data between system components 302 of a first party and system components 304 of a second party in an example embodiment. The system 300 implements and enables the processing and exchange of data between the parties (generally indicated at numeral 306), for example, as described above with reference to Figures 1 and 2. It will be appreciated that each of the components 302, 304 may be components of a computer system as described below. For example, each component can be implemented using a computer system 400 (schematically shown in Figure 4). It may be implemented as software, such as a computer program being executed within the computer system 400, and instructing the computer system 400 to conduct the method of the example embodiment.

The computer system 400 comprises a computer module 402, input modules such as a keyboard 404 and mouse 406 and a plurality of output devices such as a display 408, and printer 410.

The computer module 402 is connected to a computer network 412 via a suitable transceiver device 414, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN).

The computer module 402 in the example includes a processor 418, a Random Access Memory (RAM) 420 and a Read Only Memory (ROM) 422. The computer module 402 also includes a number of Input/Output (I/O) interfaces, for example I/O interface 424 to the display 408, and I/O interface 426 to the keyboard 404.

The components of the computer module 402 typically communicate via an interconnected bus 428 and in a manner known to the person skilled in the relevant art.

The application program is typically supplied to the user of the computer system 400 encoded on a data storage medium such as a CD-ROM or flash memory carrier and read utilising a corresponding data storage medium drive of a data storage device 430. The application program is read and controlled in its execution by the processor 418. Intermediate storage of program data maybe accomplished using RAM 420.

It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

For example, example embodiments are not limited to two communicating parties and can include a scenario where the number of participants are more than two. For example, if there are n parties, the communication overhead is up to n^2 because each party communicates with all other parties. With n parties, Figure 3 can be modified to comprise n system components. Further, the inventors recognise that the communication cost can be reduced if architecture such as Binary tree network topology, etc is used.

CLAIMS

1. A method of sharing data between a first and a second party, the method comprising the steps of:

performing respective randomization processes on data sets of the first and second parties;

performing an exchange process between the first and second parties;

performing an audit trail check process at the first and second parties respectively; and

proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

2. The method as claimed in claim 1, wherein the respective randomization processes comprise,

obfuscating the data sets using respective obfuscating numbers of the first and second parties;

concatenating the obfuscated data sets with respective audit trail elements of the first and second parties; and

randomly shuffling the concatenated data sets of the first and second parties.

3. The method as claimed in claim 2, further comprising, prior to the obfuscating step, the steps of:

hashing the data sets of the first and second parties; and

encrypting the hashed data sets of the first and second parties.

4. The method as claimed in claim 2, wherein the exchange process comprises,

exchanging the randomly shuffled data sets between the first and second parties;

re-encrypting the exchanged randomly shuffled data sets at the first and second parties respectively;

re-obfuscating the re-encrypted data sets using the respective re-obfuscating numbers at the first and second parties; and

exchanging the re-obfuscated data sets between the first and second parties.

5. The method as claimed in claim 4, wherein the exchange process further comprises,

generating respective temporary numbers at the first and second parties;

exchanging the temporary numbers between the first and second parties;

encrypting the exchanged temporary numbers at the first and second parties respectively; and

wherein the re-obfuscating step is based on the encrypted temporary numbers and the respective obfuscating numbers of the first and second parties.

6. The method as claimed in claim 5, wherein the audit trail check process comprises,

sharing respective encrypted common trail generators between the first and second parties;

sharing respective modulo function values based on the encrypted temporary numbers and the obfuscating numbers between the first and second parties;

computing respective re-obfuscated audit trail sets at the first and second parties based on the shared encrypted common trail generators and modulo function values; and

performing the respective audit trail checks at the first and second parties based on the re-obfuscated audit trail sets and the re-obfuscated data sets.

7. The method as claimed in claim 2, wherein the matching process comprises,

sharing the respective re-obfuscating numbers between the first and second parties;

verifying the respective shared re-obfuscating numbers at the first and second parties respectively;

re-generating the other party's re-obfuscated data set at the first and second parties respectively based on the verified re-obfuscating numbers; and

determining the common records between the first and second party based on intersecting the re-generated re-obfuscated data set of the other party with the party's own re-obfuscated data set.

8. A system for sharing data between a first and a second party, the system comprising,

means for performing respective randomization processes on data sets of the first and second parties;

means for performing an exchange process between the first and second parties;

means for performing an audit trail check process at the first and second parties respectively; and

means for proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.

9. The system as claimed in claim 8, wherein the means for performing respective randomization processes are arranged to,

obfuscate the data sets using respective obfuscating numbers of the first and second parties;

concatenate the obfuscated data sets with respective audit trail elements of the first and second parties; and

randomly shuffle the concatenated data sets of the first and second parties.

10. The system as claimed in claim 9, wherein the means for performing respective randomization processes are further arranged to,

hash the data sets of the first and second parties; and

encrypt the hashed data sets of the first and second parties.

11. The system as claimed in claim 9, wherein the means for performing an exchange process are arranged to,

exchange the randomly shuffled data sets between the first and second parties;

re-encrypt the exchanged randomly shuffled data sets at the first and second parties respectively;

re-obfuscate the re-encrypted data sets using the respective re-obfuscating numbers at the first and second parties; and

exchange the re-obfuscated data sets between the first and second parties.

12. The system as claimed in claim 11, wherein the means for performing an exchange process are further arranged to,

generate respective temporary numbers at the first and second parties;

exchange the temporary numbers between the first and second parties;

encrypt the exchanged temporary numbers at the first and second parties respectively; and

wherein the re-obfuscation of the re-encrypted data sets is based on the encrypted temporary numbers and the respective obfuscating numbers of the first and second parties.

13. The system as claimed in claim 12, wherein the means for performing an audit trail check process are arranged to,

share respective encrypted common trail generators between the first and second parties;

share respective modulo function values based on the encrypted temporary numbers and the obfuscating numbers between the first and second parties;

compute respective re-obfuscated audit trail sets at the first and second parties based on the shared encrypted common trail generators and modulo function values; and

perform the respective audit trail checks at the first and second parties based on the re-obfuscated audit trail sets and the re-obfuscated data sets.

14. The system as claimed in claim 9, wherein the means for proceeding with performing a matching process is arranged to,

share the respective re-obfuscating numbers between the first and second parties;

verify the respective shared re-obfuscating numbers at the first and second parties respectively;

re-generate the other party's re-obfuscated data set at the first and second parties respectively based on the verified re-obfuscating numbers; and

determine the common records between the first and second party based on intersecting the re-generated re-obfuscated data set of the other party with the party's own re-obfuscated data set.

15. A computer readable data storage medium having stored thereon computer code means for instructing respective computer processors of a first party and a second party to execute a method of sharing data between the first and the second parties, the method comprising the steps of:

performing respective randomization processes on data sets of the first and second parties;

performing an exchange process between the first and second parties;

performing an audit trail check process at the first and second parties respectively; and

proceeding with performing a matching process at the first and second parties respectively only after a successful audit trail check by each party in the audit trail check process and the matching process is such that each party can determine whether the other party has provided a correct re-obfuscating number for determining common records between the first and second party.