

PROBABILISTIC ENCRYPTION METHOD FOR QUERYING DATABASE

FIELD OF THE INVENTION

[0001] The present invention relates to a new probabilistic encryption method, particularly used in querying a database.

BACKGROUND

[0002] Information Technology (IT) outsourcing, including cloud computing and database as-a-service, offers many advantages like enhanced data availability, ability to provision resources accordingly to usage demand, and economies of scale. Nevertheless, adoption has been slowed by persistent concerns over control and protection of data, as well as conflicting legislative frameworks (e.g. U.S. Patriot Art versus EU Data Protection Art).

[0003] A method of preserving data privacy is by applying a deterministic encryption scheme to the data record values (or keys) before storing them in the un-trusted or “outsourced” servers, and applying the same deterministic encryption scheme to the query values before sending them from the client machines to the un-trusted servers. Therefore the un-trusted servers only see the encrypted data record values and encrypted query values and never see the actual data record values or actual query values. Since any query value always produces the same encrypted query value, the server can determine whether a query value matches a data record value from their encrypted values. However, this method of preserving data privacy is not secure. For example, it allows other parties to deduce whether two query values are the same, and whether two data record values are the same.

[0004] Concerns over data control and protection may be mitigated if a probabilistic encryption scheme is applied to the data record values and query values before they leave the client’s machines, so multiple encryptions of one value can produce different encrypted values. The object of the invention is thus to provide a solution that overcomes the above disadvantages or at least provide a new method for querying a database based on probabilistic encryption.

SUMMARY OF INVENTION

[0005] According to a first aspect of the invention, a method for querying a database to return a data record based on a query is described where the query has a query value and the data record has a data record value. The method comprises the steps of composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements, and generating a public key. The method further comprises the steps of producing an encrypted data record value by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup, producing an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup and performing a matching operation on the encrypted data record value and the encrypted query value. The matching operation comprises the steps of performing a mathematical operation on the encrypted data record value and the encrypted query value, removing the element of the second subgroup from the encrypted data record value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the data record value and the query value, using the public key to identify when the data record value is equal to the query value in the function, and returning the data record.

[0006] In another embodiment, the step of composing a composite group having a first subgroup, a second subgroup and a third subgroup further comprises composing a bilinear mapping on the composite group and the step of performing a mathematical operation on the encrypted data record value and the encrypted query value comprises applying bilinear mapping on the encrypted data record value and the encrypted query value.

[0007] In another embodiment, the step of producing an encrypted data record value by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup comprises the steps of encrypting the data record value as an exponent in the first element of the first subgroup, encrypting the first random factor as an exponent in the element of the second subgroup, and multiplying the first element of the first subgroup and the element of the second subgroup to produce the encrypted data record value.

[0008] In another embodiment, the step of producing an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup comprises the steps of encrypting the query value as an exponent in the second element of the first subgroup, encrypting the second random factor as an exponent in the element of the third subgroup, and multiplying the second element of the first subgroup and the element of the third subgroup to produce the encrypted query value.

[0009] According to a second aspect of the invention, a method for querying a database to return a key interval based on a query is described where the query has a query value and the key interval has an anchor value and a key interval length. The method comprises the steps of partitioning the database into a plurality of key intervals and composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements. The method further comprises the steps of generating a public key and producing an encrypted key interval value for each of the plurality of key intervals by mapping the anchor value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup. The method further comprises the steps of generating a plurality of additional query values based on the query value and a query counter, the query counter being dependent on the key interval length, and producing an encrypted query value for the query value and each additional query value by mapping the query value and the query counter to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup, and performing a matching operation on each encrypted key interval value and on each encrypted query value. The matching operation comprises the steps of performing a mathematical operation on the encrypted key interval value and the encrypted query value, removing the element of the second subgroup from the encrypted key interval value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the query value, the query counter and the anchor value, using the public key to identify when the anchor value is equal to the difference between the query value and the query counter in the function, and returning the key interval.

[0010] In another embodiment, the step of composing a composite group having a first subgroup, a second subgroup and a third subgroup further comprises composing a bilinear mapping on the composite group and the step of performing a mathematical operation on the

encrypted key interval value and the encrypted query value comprises applying bilinear mapping on the encrypted key interval value and the encrypted query value.

[0011] In another embodiment, the key interval further comprises a lower value and an upper value.

[0012] In another embodiment, the anchor value is the lower value and the step of generating a plurality of additional query values based on the query value and a query counter comprises the step of generating a plurality of additional query values by subtracting the query counter from the query value.

[0013] In another embodiment, the anchor value is the upper value and the step of generating a plurality of additional query values based on the query value and a query counter comprises the step of generating a plurality of additional query values by adding the query counter to the query value.

[0014] According to a third aspect of the invention, a system for querying a database to return a data record based on a query is described where the query has a query value and the data record has a data record value. The system comprises a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements, and a server for housing the database, the database for storing at least one encrypted data record value and the public key, the at least one encrypted data record value produced by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup. Where the client machine produces an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup, and sends the encrypted query value to the server. And where the server performs a matching operation on the at least one encrypted data record value and the encrypted query value by performing a mathematical operation on the at least one encrypted data record value and the encrypted query value, removing the element of the second subgroup from the at least one encrypted data record value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the data record value and

the query value, using the public key to identify when the data record value is equal to the query value in the function, and returning the data record to the client machine.

[0015] In another embodiment, the composite group has a bilinear mapping and the mathematical operation is bilinear mapping.

[0016] In another embodiment, for the at least one encrypted data record value, the data record value is mapped to the first element of the first subgroup by encrypting the data record value as an exponent in the first element of the first subgroup, and the first random factor is mapped to an element of the second subgroup by encrypting the first random factor as an exponent in the element of the second subgroup.

[0017] In another embodiment, for the encrypted query value, the query value is mapped to the second element of the first subgroup by encrypting the query value as an exponent in the second element of the first subgroup, and the second random factor is mapped to an element of the third subgroup by encrypting the second random factor as an exponent in the element of the third subgroup.

[0018] According to a fourth aspect of the invention, a system for querying a database to return a key interval based on a query is described where the query has a query value and the key interval has an anchor value and a key interval length. The system comprises a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements, and a server for housing the database, the database for storing the public key and at least one encrypted key interval value, the at least one encrypted key interval value produced by mapping the anchor value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup. Where the client machine generates a plurality of additional query values based on the query value and a query counter, the query counter being dependent on the key interval length, and produces an encrypted query value for the query value and each additional query value by mapping the query value and the query counter to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup, and sends each encrypted query value to the server. And where the server performs a matching operation on the at least one encrypted key interval value and each encrypted query value by performing a mathematical operation on the at least one

encrypted key interval value and the encrypted query value, removing the element of the second subgroup from the at least one encrypted key interval value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the query value, the query counter and the anchor value, using the public key to identify when the anchor value is equal to the difference between the query value and the query counter in the function, and returning the key interval to the client machine.

[0019] In another embodiment, the composite group has a bilinear mapping and the mathematical operation is bilinear mapping.

[0020] In another embodiment, the key interval further comprises a lower value and an upper value.

[0021] In another embodiment, the anchor value is the lower value and the client machine generates the plurality of additional query values by subtracting the query counter from the query value.

[0022] In another embodiment, the anchor value is the upper value and the client machine generates the plurality of additional query values by adding the query counter to the query value.

[0023] According to a fifth aspect of the invention, a method for querying a database to return a bottom tier key interval based on a query is described where the query has a top query value and a bottom query value. The method comprises the steps of partitioning the database into a plurality of top tier key intervals, each top tier key interval having a top anchor value and a top key interval length, and partitioning each top tier key interval into a plurality of bottom tier key intervals, each bottom tier key interval having a bottom anchor value and a bottom key interval length. The method further comprises the steps of composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements, and generating a public key. The method further comprises the steps of producing an encrypted top tier key interval value for each of the plurality of top tier key intervals by mapping the top anchor value to a first element of the first subgroup and mapping a first random factor to a first element of the second subgroup, and producing an encrypted

bottom tier key interval value for each of the plurality of bottom tier key intervals by mapping the bottom anchor value to a second element of the first subgroup and mapping a second random factor to a second element of the second subgroup. The method further comprises the steps of generating a plurality of additional top query values based on the top query value and a top query counter, the top query counter being dependent on the top key interval length, and generating a plurality of additional bottom query values based on the bottom query value and a bottom query counter, the bottom query counter being dependent on the bottom key interval length. The method further comprises the steps of producing an encrypted top query value for the top query value and each additional top query value by mapping the top query value and the top query counter to a third element of the first subgroup and mapping a third random factor to a first element of the third subgroup, and producing an encrypted bottom query value for the bottom query value and each additional bottom query value by mapping the bottom query value and the bottom query counter to a fourth element of the first subgroup and mapping a fourth random factor to a second element of the third subgroup. The method further comprises performing a first matching operation on each encrypted top tier key interval value and on each encrypted top query value. The first matching operation comprises the steps of performing a mathematical operation on the encrypted top tier key interval value and the encrypted top query value, and removing the first element of the second subgroup from the encrypted top tier key interval value. The first matching operation further comprises removing the first element of the third subgroup from the encrypted top query value to obtain a first function which contains a random multiple of the difference between the top query value, the top query counter and the top anchor value, and using the public key to identify a matched top tier key interval when the top anchor value is equal to the difference between the top query value and the top query counter in the first function. The method further comprises the step of performing a second matching operation on each encrypted bottom tier key interval value in the matched top tier key interval and on each encrypted bottom query value. The second matching operation comprises the steps of performing a mathematical operation on the encrypted bottom tier key interval value and the encrypted bottom query value, and removing the second element of the second subgroup from the encrypted bottom tier key interval value. The second matching operation further comprises removing the second element of the third subgroup from the encrypted bottom query value to obtain a second function which contains a random multiple of the difference between the bottom query value, the bottom query counter and the bottom anchor value, and using the public key to identify when the bottom anchor

value is equal to the difference between the bottom query value and the bottom query counter in the second function, and returning the bottom tier key interval.

[0024] According to a sixth aspect of the invention, a system for querying a database to return a bottom tier key interval based on a query is described where the query has a top query value and a bottom query value. The system comprises a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements. The system further comprises a server for housing the database, the database being partitioned into a plurality of top tier key intervals, each top tier key interval having a top anchor value and a top key interval length and further partitioned into a plurality of bottom tier key intervals, each bottom tier key interval having a bottom anchor value and a bottom key interval length. The database stores the public key and at least one encrypted top tier key interval value and at least one encrypted bottom tier key interval value, where the at least one encrypted top tier key interval value is produced by mapping the top anchor value to a first element of the first subgroup and mapping a first random factor to a first element of the second subgroup, and the at least one encrypted bottom tier key interval value is produced by mapping the bottom anchor value to a second element of the first subgroup and mapping a second random factor to a second element of the second subgroup. Where the client machine generates a plurality of additional top query values based on the top query value and a top query counter, and a plurality of additional bottom query values based on the bottom query value and a bottom query counter, the top query counter being dependent on the top key interval length and the bottom query counter being dependent on the bottom key interval length. And where the client machine produces an encrypted top query value for the top query value and each additional top query value by mapping the top query value and the top query counter to a third element of the first subgroup and mapping a third random factor to a first element of the third subgroup, and produces an encrypted bottom query value for the bottom query value and each additional bottom query value by mapping the bottom query value and the bottom query counter to a fourth element of the first subgroup and mapping a fourth random factor to a second element of the third subgroup. Where the server performs a first matching operation on each encrypted top tier key interval value and on each encrypted top query value by performing a mathematical operation on the encrypted top tier key interval value and the encrypted top query value, removing the first element of the second subgroup from the encrypted top tier key interval value, removing the first element of the third subgroup from the encrypted top query

value to obtain a first function which contains a random multiple of the difference between the top query value, the top query counter and the top anchor value, using the public key to identify a matched top tier key interval when the top anchor value is equal to the difference between the top query value and the top query counter in the first function. And where the server performs a second matching operation on each encrypted bottom tier key interval value in the matched top tier key interval and on each encrypted bottom query value by performing a mathematical operation on the encrypted bottom tier key interval value and the encrypted bottom query value, removing the second element of the second subgroup from the encrypted bottom tier key interval value, removing the second element of the third subgroup from the encrypted bottom query value to obtain a second function which contains a random multiple of the difference between the bottom query value, the bottom query counter and the bottom anchor value, using the public key to identify when the bottom anchor value is equal to the difference between the bottom query value and the bottom query counter in the second function, and returning the bottom tier key interval.

[0025] The invention will now be described in detail with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] The accompanying figures illustrate disclosed embodiment(s) and serve to explain principles of the disclosed embodiment(s). It is to be understood, however, that these drawings are presented for purposes of illustration only, and not for defining limits of the application.

[0027] Figure 1 is a flow chart illustrating an exemplary method for querying a database with probabilistic encryption.

[0028] Figure 2 is a flow chart illustrating an exemplary query method for querying a database with probabilistic encryption, and where the database being queried has a key interval indexing structure.

[0029] Exemplary, non-limiting embodiments of the present application will now be described with references to the above-mentioned figures.

DETAILED DESCRIPTION

[0030] In the specification herein, a query is an instruction sent by a user to a database. A query can be for example “retrieve the data records of all employees who earn an annual salary of S\$200,000”. A query has a query value and in this illustration, the query value is 200,000. A data record is data that is stored in a database. In this illustration, a data record can have the employee’s name: “John Fisher”, the number of years the employee has been with the company: “10” and the employee’s annual salary: “S\$200,000”. The data record has a data record value or key, and in this illustration, the data record value is 200,000.

[0031] Figure 1 shows a method for querying a database with probabilistic encryption, in accordance with a preferred embodiment of the invention.

[0032] Referring to step 101 in figure 1, a composite group G with bilinear mapping $e: G \times G \rightarrow G_T$ is composed. The composite group G consists of a first subgroup G_1 with order p_1 , a second subgroup G_2 with order p_2 , and a third subgroup G_3 with order p_3 , i.e. $G = G_1 \times G_2 \times G_3$. G_1 has a generator g_1 , G_2 has a generator g_2 , and G_3 has a generator g_3 . The key domain of the data records is $[L, U)$. p_1, p_2, p_3 can be prime numbers where p_1 is larger than $U - L$. Under the group law of G , any two elements a and b of G can be combined to form another element $a \cdot b$ in G .

[0033] The first subgroup G_1 has a plurality of elements and g_1 is the generator i.e. $G_1 = \{g_1, g_1^2, g_1^3, \dots, g_1^{p_1}\}$ and $g_1^{p_1} = 1$. The second subgroup G_2 has a plurality of elements and g_2 is the generator i.e. $G_2 = \{g_2, g_2^2, g_2^3, \dots, g_2^{p_2}\}$ and $g_2^{p_2} = 1$. The third subgroup G_3 has a plurality of elements and g_3 is the generator i.e. $G_3 = \{g_3, g_3^2, g_3^3, \dots, g_3^{p_3}\}$ and $g_3^{p_3} = 1$, where 1 is the identity element of G . For any natural number x and y , $e(g_1^x, g_1^y) = e(g_1, g_1)^{xy}$, $e(g_2^x, g_2^y) = e(g_2, g_2)^{xy}$, $e(g_3^x, g_3^y) = e(g_3, g_3)^{xy}$, and $e(g_1^x, g_2^y) = 1$, $e(g_1^x, g_3^y) = 1$, $e(g_2^x, g_3^y) = 1$.

[0034] In step 102, a public key P is generated by multiplying $e(g_2, g_2)$ and $e(g_3, g_3)$ i.e. $P = e(g_2, g_2)e(g_3, g_3)$. Public key P is then stored in the un-trusted server.

[0035] In step 103, for a data record with a data record value l , two encrypted data C and C' (which under the group law of G are elements of G) are generated as follows. Firstly,

random factors c_1 and c_2 are generated. Let c_1 be any random integer in the range $[1..p_1]$ and let c_2 be any random integer in the range $[1..p_2]$. Then compute C where $C = g_1^{-l} g_2^{c_2} g_3$ where g_1^{-l} is an element of the first subgroup G_1 , $g_2^{c_2}$ is an element of the second subgroup G_2 , g_3 is an element of the third subgroup G_3 . As can be seen in the first encryption data C , the data record value l is mapped to the element g_1^{-l} . More specifically in this case, the data record value l is encrypted as the exponent in g_1^{-l} . The random factor c_2 is mapped to the element $g_2^{c_2}$. More specifically in this case, the random factor c_2 is encrypted as the exponent in $g_2^{c_2}$. One skilled in the art can appreciate that there are other ways the data record value l and random factor c_2 can be encrypted in an element.

[0036] Then compute C' where $C' = g_1^{c_1} g_2^{\frac{1}{(c_2+1)}-1} g_3$ and where $g_1^{c_1}$ is an element of the first subgroup G_1 , $g_2^{\frac{1}{(c_2+1)}-1}$ is an element of the second subgroup G_2 , g_3 is an element of the third subgroup G_3 . As can be seen in the second encryption data C' , the random factor c_1 is mapped to the element $g_1^{c_1}$. More specifically in this case the random factor c_1 is encrypted as the exponent in $g_1^{c_1}$. The random factor c_2 is mapped to the element $g_2^{\frac{1}{(c_2+1)}-1}$. More specifically in this case the random factor c_2 is encrypted as the exponent in $g_2^{\frac{1}{(c_2+1)}-1}$. One skilled in the art can appreciate that there are other ways the random factor c_1 and random factor c_2 can be encrypted in an element.

[0037] The two encrypted data C and C' are then arranged as an encrypted data record value tuple $\langle C, C' \rangle$. The encrypted data record value tuple $\langle C, C' \rangle$ is then deposited along with the encrypted data record in a database in the un-trusted server.

[0038] In step 104, user enters a query having a query value q into the client machine. Two encrypted data Q and Q' (which under the group law of G are elements of G) are generated as follows. Firstly, random factors u_1 and u_2 are generated. Let u_1 be any random integer in the range $[1..p_1]$ and let u_2 be any random integer in the range $[1..p_3]$. Then compute Q where $Q = g_1^q g_2 g_3^{u_2}$ and where g_1^q is an element of the first subgroup G_1 , g_2 is an element of the second subgroup G_2 , $g_3^{u_2}$ is an element of the third subgroup G_3 . As can be seen in the

first encrypted data Q , the query value q is mapped to the element g_1^q . More specifically in this case the query value q is encrypted as the exponent in g_1^q . The random factor u_2 is mapped to the element $g_3^{u_2}$. More specifically in this case the random factor u_2 is encrypted as the exponent in $g_3^{u_2}$. One skilled in the art can appreciate that there are other ways the query value q and random factor u_2 can be encrypted in an element.

[0039] Then compute Q' where $Q' = g_1^{u_1} g_2 g_3^{\frac{1}{u_2+1}-1}$ and where $g_1^{u_1}$ is an element of the first subgroup G_1 , g_2 is an element of the second subgroup G_2 , $g_3^{\frac{1}{u_2+1}-1}$ is an element of the third subgroup G_3 . As can be seen in the second encrypted data Q' , the random factor u_1 is mapped to the element $g_1^{u_1}$. More specifically in this case the random factor u_1 is encrypted as the exponent in $g_1^{u_1}$. The random factor u_2 is mapped to the element $g_3^{\frac{1}{u_2+1}-1}$. More specifically in this case the random factor u_2 is encrypted as the exponent in $g_3^{\frac{1}{u_2+1}-1}$. One skilled in the art can appreciate that there are other ways the random factor u_1 and random factor u_2 can be encrypted in an element.

[0040] The two encrypted data Q and Q' are then arranged as an encrypted query value tuple $\langle Q, Q' \rangle$. The encrypted query value tuple $\langle Q, Q' \rangle$ is then sent to the un-trusted server.

[0041] In step 105, the un-trusted server then performs the matching operation $\frac{e(QC, Q'C')}{P}$ to determine whether the query value q matches the data record value l . If $\frac{e(QC, Q'C')}{P} = 1$, then the query value q matches or is equal to the data record value l .

[0042] The matching operation involves applying bilinear mapping to combinations of the encrypted query value tuple $\langle Q, Q' \rangle$ with the encrypted data record value tuple $\langle C, C' \rangle$, and then dividing by the public key P :

$$\Rightarrow \frac{e(QC, Q'C')}{e(g_2, g_2)e(g_3, g_3)}$$

$$\begin{aligned}
& \Rightarrow \frac{e(g_1^{q-l} g_2^{1+c_2} g_3^{u_2+1}, g_1^{u_1+c_1} g_2^{\frac{1}{(c_2+1)-1}} g_3^{\frac{1}{(u_2+1)-1+1}})}{e(g_2, g_2)e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1^{q-l} g_2^{1+c_2} g_3^{u_2+1}, g_1^{u_1+c_1} g_2^{\frac{1}{(c_2+1)}} g_3^{\frac{1}{(u_2+1)}})}{e(g_2, g_2)e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1, g_1)^{(q-l)(u_1+c_1)} e(g_2, g_2)^{\left(\frac{1+c_2}{c_2+1}\right)} e(g_3, g_3)^{\left(\frac{u_2+1}{u_2+1}\right)}}{e(g_2, g_2)e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1, g_1)^{(q-l)(u_1+c_1)} e(g_2, g_2)e(g_3, g_3)}{e(g_2, g_2)e(g_3, g_3)} \\
& \Rightarrow e(g_1, g_1)^{(q-l)(u_1+c_1)}
\end{aligned}$$

*The element with the random factor c2 and the element with the random factor u2 are removed

[0043] When query value q is equal to data record value l ,

Then $e(g_1, g_1)^{(l-l)(u_1+c_1)}$

$$\Rightarrow e(g_1, g_1)^{(0)(u_1+c_1)}$$

$\Rightarrow e(g_1, g_1)^0 = 1$ and therefore the query value q matches the data record value l and the encrypted data record is then returned to the client machine.

[0044] This encryption method is advantageous as it is probabilistic. This is because u_1 and u_2 are random values that are generated each time a query value q is encrypted, therefore for the same query value q , queried at different times, the encrypted query value tuple $\langle Q, Q' \rangle$ will be different. Thus, the un-trusted server will see different encrypted query value tuples, despite the fact that the same query value q was queried twice in succession. The un-trusted server will therefore be unable to deduce the pattern and logic of the encryption and the actual value of the query value q will be hidden from the un-trusted server. The problem solved by this invention is thus that the encryption is probabilistic and non-deterministic as the same query will have different encrypted versions at different times.

[0045] Despite the fact that the same query value q will have different encrypted versions at different times, the matching operation $\frac{e(QC, Q'C')}{P}$ is able to determine if a query value q matches a data record value l . This is due to the specific structure of the matching operation, encrypted data record value tuple $\langle C, C' \rangle$, encrypted query value tuple $\langle Q, Q' \rangle$ which removes the element with the random factor c_2 and removes the element with the

random factor u_2 , and using the public key P to result in a “match” ($\frac{e(QC, Q'C')}{P} = 1$) when the query value q is equal to the data record value l .

[0046] The above method for querying a database with probabilistic encryption can be applied to an un-trusted server with a database with a multiple key interval indexing structure. In a multiple key interval indexing structure, the data records are partitioned such that groups of data records fall into disjoint key intervals based on their data record values. The un-trusted server thus has to determine if a query value falls within a key interval or an interval of data record values. If it is determined that a query value falls within a specific key interval, the data records whose data record values fall within that key interval are returned to the client machine and to the user. The other key intervals and their corresponding data records therein, can be ignored. This constitutes a one-tier indexing scheme.

[0047] A two-tier indexing scheme can be formulated by further partitioning the key interval into smaller key intervals, to form a top tier (tier 1) and bottom tier (tier 2) structure. At runtime, the user supplies two encrypted representations of the query value, one on its leading bits(i.e. first query value), the second on the entire query value (i.e. second query value). At the un-trusted server, the protocol is executed on the encrypted representation of the first query value to identify the matching key interval in the top tier, and within it, the encrypted representation of the second query value is used to identify the matching key interval in the bottom tier. The data records whose data record values fall within the matching key interval in the bottom tier are then returned to the user.

[0048] One skilled in the art can appreciate how the above extension can be applied repeatedly, leading to a multi-tiered index tree. Through it, the un-trusted server can perform record retrieval with a complexity that is logarithmic to the size of the key range.

[0049] Figure 2 shows how the method for querying a database with probabilistic encryption can be incorporated in a system that has an un-trusted server with a database that has a multiple key-interval indexing structure. Referring to step 201 in figure 2, a composite group G with bilinear mapping $e: G \times G \rightarrow G_T$ is composed. The composite group G consists of a first subgroup G_1 with order p_1 , a second subgroup G_2 with order p_2 , and a third subgroup G_3 with order p_3 , i.e. $G = G_1 \times G_2 \times G_3$. G_1 has a generator g_1 , G_2 has a generator g_2 , and G_3 has

a generator g_3 . The key domain of the data records is $[L, U)$. p_1, p_2, p_3 can be prime numbers where p_1 is larger than $U - L$. Under the group law of G , any two elements a and b of G can be combined to form another element $a \cdot b$ in G .

[0050] The first subgroup G_1 has a plurality of elements and g_1 is the generator i.e. $G_1 = \{g_1, g_1^2, g_1^3, \dots, g_1^{p_1}\}$ and $g_1^{p_1} = 1$. The second subgroup G_2 has a plurality of elements and g_2 is the generator i.e. $G_2 = \{g_2, g_2^2, g_2^3, \dots, g_2^{p_2}\}$ and $g_2^{p_2} = 1$. The third subgroup G_3 has a plurality of elements and g_3 is the generator i.e. $G_3 = \{g_3, g_3^2, g_3^3, \dots, g_3^{p_3}\}$ and $g_3^{p_3} = 1$, where 1 is the identity element of G . For any natural number x and y , $e(g_1^x, g_1^y) = e(g_1, g_1)^{xy}$, $e(g_2^x, g_2^y) = e(g_2, g_2)^{xy}$, $e(g_3^x, g_3^y) = e(g_3, g_3)^{xy}$, and $e(g_1^x, g_2^y) = 1$, $e(g_1^x, g_3^y) = 1$, $e(g_2^x, g_3^y) = 1$.

[0051] In step 202, a public key P is generated by multiplying $e(g_2, g_2)$ and $e(g_3, g_3)$ i.e. $P = e(g_2, g_2)e(g_3, g_3)$. Public key P is then stored in the un-trusted server.

[0052] In step 203, for each key interval I_i , where $I_i = [l_i, u_i)$, two encrypted data C_i and C_i' (which under the group law of G are elements of G) are generated. Each key interval I_i has a lower value l_i , and an upper value $u_i - 1$ (u_i is not part of the key interval). The length of a key interval is d . Another way of putting it is that d is the maximum number of distinct data record values in a key interval. Therefore, $d = u_i - l_i$. All key intervals in the same database have the same key interval length d . C_i and C_i' are generated as follows. Firstly, random factors $c_{i,1}$ and $c_{i,2}$ are generated. Let $c_{i,1}$ be any random integer within the set $[1..p_1]$ and let $c_{i,2}$ be any random integer within the set $[1..p_2]$. In this preferred embodiment, lower value l_i is used as the anchor value i.e. the value of the key interval I_i that is to be encrypted. Alternatively, upper value $u_i - 1$ can be used as the anchor value.

[0053] Then compute C_i where $C_i = g_1^{-l_i} g_2^{c_{i,2}} g_3$ and where $g_1^{-l_i}$ is an element of the first subgroup G_1 , $g_2^{c_{i,2}}$ is an element of the second subgroup G_2 , g_3 is an element of the third subgroup G_3 . As can be seen in the first encrypted data C_i , the anchor value (which is l_i in this embodiment) is mapped to the element $g_1^{-l_i}$. More specifically in this case the anchor value is encrypted as the exponent in $g_1^{-l_i}$. The random factor $c_{i,2}$ is mapped to the element $g_2^{c_{i,2}}$. More specifically in this case the random factor $c_{i,2}$ is encrypted as the exponent in $g_2^{c_{i,2}}$. One

skilled in the art can appreciate that there are other ways the anchor value and random factor $c_{i,2}$ can be encrypted in an element.

[0054] Then compute C_i' where $C_i' = g_1^{c_{i,1}} g_2^{\frac{1}{(c_{i,2}+1)}-1} g_3$ and where $g_1^{c_{i,1}}$ is an element of the first subgroup G_1 , $g_2^{\frac{1}{(c_{i,2}+1)}-1}$ is an element of the second subgroup G_2 , g_3 is an element of the third subgroup G_3 . As can be seen in the second encrypted data C_i' , the random factor $c_{i,1}$ is mapped to the element $g_1^{c_{i,1}}$. More specifically in this case the random factor $c_{i,1}$ is encrypted as the exponent in $g_1^{c_{i,1}}$. The random factor $c_{i,2}$ is mapped to the element $g_2^{\frac{1}{(c_{i,2}+1)}-1}$. More specifically in this case the random factor $c_{i,2}$ is encrypted as the exponent in $g_2^{\frac{1}{(c_{i,2}+1)}-1}$. One skilled in the art can appreciate that there are other ways random factor $c_{i,1}$ and random factor $c_{i,2}$ can be encrypted in an element.

[0055] The two encrypted data C_i and C_i' are then arranged as an encrypted key interval tuple $\langle C_i, C_i' \rangle$.

[0056] In step 204, the encrypted key interval tuples $\langle C_i, C_i' \rangle_{i=1}^n$ are then shuffled to conceal their relative order. In other words, the arrangement of the tuples is re-ordered such that tuples of consecutive key intervals I_i are no longer in sequence. The shuffled encrypted key interval tuples $\langle C_i, C_i' \rangle_{i=1}^n$ are then deposited in a database in the un-trusted server. This step of shuffling is advantageous as it preserves the privacy of the tuples.

[0057] In step 205, user enters a query with query value q into the client machine and the client machine generates $d-1$ additional query values with the values $q-1, q-2, \dots, q-d+1$. For example, if query value $q = 100$ and $d = 5$, the client will generate 4 additional query values 99, 98, 97, 96.

[0058] Let j represent a member of a query value set containing query value q and the additional query values. For example, $j=0$ refers to query value q . If $j > 0$, this means that it is one of the additional query values. Therefore, for $j=0$ to $(d-1)$, each member j of the query value set will have a value of $q-j$. In step 206, for each member j in the query value set, two

encrypted data Q_j and Q_j' (which under the group law of G are elements of G) are generated as follows. Firstly, random factors $u_{j,1}$ and $u_{j,2}$ are generated. Let $u_{j,1}$ be any random integer in the range $[1..p_1]$ and let $u_{j,2}$ be any random integer in the range $[1..p_3]$. Then compute Q_j where $Q_j = g_1^{q-j} g_2^{u_{j,2}}$ and where g_1^{q-j} is an element of the first subgroup G_1 , g_2 is an element of the second subgroup G_2 , $g_3^{u_{j,2}}$ is an element of the third subgroup G_3 . As can be seen in the first encrypted data Q_j , the value of each member j of the query value set (i.e. $q-j$) is mapped to element g_1^{q-j} . More specifically in this case the value of each member j of the query value set is encrypted as the exponent in g_1^{q-j} . The random factor $u_{j,2}$ is mapped to the element $g_3^{u_{j,2}}$. More specifically in this case the random factor $u_{j,2}$ is encrypted as the exponent in $g_3^{u_{j,2}}$. One skilled in the art can appreciate that there are other ways the value of each member j of the query value set and random factor $u_{j,2}$ can be encrypted in an element.

[0059] Then compute Q_j' where $Q_j' = g_1^{u_{j,1}} g_2 g_3^{\frac{1}{u_{j,2}+1}-1}$ and where $g_1^{u_{j,1}}$ is an element of the first subgroup G_1 , g_2 is an element of the second subgroup G_2 , $g_3^{\frac{1}{u_{j,2}+1}-1}$ is an element of the third subgroup G_3 . As can be seen in the second encrypted data Q_j' , the random factor $u_{j,1}$ is mapped to the element $g_1^{u_{j,1}}$. More specifically in this case the random factor $u_{j,1}$ is encrypted as the exponent in $g_1^{u_{j,1}}$. The random factor $u_{j,2}$ is mapped to the element $g_3^{\frac{1}{u_{j,2}+1}-1}$. More specifically in this case the random factor $u_{j,2}$ is encrypted as the exponent in $g_3^{\frac{1}{u_{j,2}+1}-1}$. One skilled in the art can appreciate that there are other ways random factor $u_{j,1}$ and random factor $u_{j,2}$ can be encrypted in an element.

[0060] The two encrypted data Q_j and Q_j' are then arranged as an encrypted query value tuple $\langle Q_j, Q_j' \rangle$.

[0061] In step 207, the encrypted query value tuples $\langle Q_j, Q_j' \rangle_{j=0}^{d-1}$ are shuffled to conceal their relative order. In other words, the arrangement of the tuples is re-ordered such that tuples of consecutive query values are no longer in sequence. The shuffled encrypted

query tuples $\langle Q_j, Q_j' \rangle_{j=0}^{d-1}$ are then sent to the un-trusted server. This step of shuffling is advantageous as it preserves the privacy of the tuples.

[0062] In step 208, the un-trusted server performs the matching operation $\frac{e(Q_j C_i, Q_j' C_i')}{P}$ for each member j in the query set and each key interval I_i . The matching operation determines whether the value of a member j in the query set (q-j) is equal to the anchor value of key interval I_i . If $\frac{e(Q_j C_i, Q_j' C_i')}{P} = 1$, then the value of a member j in the query set is equal to the anchor value of key interval I_i .

[0063] The matching operation involves applying bilinear mapping to combinations of the encrypted query value tuple $\langle Q_j, Q_j' \rangle$ with the encrypted key interval tuple $\langle C_i, C_i' \rangle$, and then dividing by the public key P:

$$\begin{aligned}
& \Rightarrow \frac{e(Q_j C_i, Q_j' C_i')}{e(g_2, g_2) e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1^{q-j-l_i} g_2^{1+c_{i,2}} g_3^{u_{j,2}+1}, g_1^{u_{j,1}+c_{i,1}} g_2^{\frac{1}{(c_{i,2}+1)}-1} g_3^{\frac{1}{(u_{j,2}+1)}-1+1})}{e(g_2, g_2) e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1^{q-j-l_i} g_2^{1+c_{i,2}} g_3^{u_{j,2}+1}, g_1^{u_{j,1}+c_{i,1}} g_2^{\frac{1}{(c_{i,2}+1)}} g_3^{\frac{1}{(u_{j,2}+1)}})}{e(g_2, g_2) e(g_3, g_3)} \\
& \Rightarrow \frac{e(g_1, g_1)^{(q-j-l_i)(u_{j,1}+c_{i,1})} e(g_2, g_2)^{\left(\frac{1+c_{i,2}}{c_{i,2}+1}\right)} e(g_3, g_3)^{\left(\frac{u_{j,2}+1}{u_{j,2}+1}\right)}}{e(g_2, g_2) e(g_3, g_3)} \quad \begin{array}{l} \text{*The element with the random factor } c_{i,2} \text{ and} \\ \text{the element with the random factor } u_{j,2} \text{ are} \\ \text{removed} \end{array} \\
& \Rightarrow \frac{e(g_1, g_1)^{(q-j-l_i)(u_{j,1}+c_{i,1})} e(g_2, g_2) e(g_3, g_3)}{e(g_2, g_2) e(g_3, g_3)} \\
& \Rightarrow e(g_1, g_2)^{(q-j-l_i)(u_{j,1}+c_{i,1})}
\end{aligned}$$

[0064] When the value of a member j in the query set (q-j) is equal to the anchor value of key interval I_i ,

$$\begin{aligned}
& \Rightarrow e(g_1, g_2)^{(l_i-l_j)(u_{j,1}+c_{i,1})} \\
& \Rightarrow e(g_1, g_1)^{(0)(u_{j,1}+c_{i,1})}
\end{aligned}$$

$\Rightarrow e(g_1, g_1)^0 \Rightarrow 1$ and therefore the value of one of the members j is equal to the anchor value of key interval I_i . In other words, the query value q falls within the key interval I_i . The data records in key interval I_i are then returned to the client machine.

[0065] This querying method is advantageous as the matching involves only the encrypted anchor value of the key intervals, and the encrypted query values of the members in the query set (i.e. encrypted query value q and the encrypted additional query values). As the client machine sends encrypted query value q , along with the encrypted additional query values to the un-trusted server, the un-trusted server does not know which the “real” encrypted query value is, and therefore the “real” encrypted query is “masked” from the un-trusted server. All the un-trusted server knows is that there are multiple encrypted query values, and there is, or there isn’t a match with the encrypted anchor value of the key intervals.

[0066] Alternatively, in another embodiment, the upper value u_i-1 (instead of the lower value l_i) of key interval I_i is used as the anchor value. In this embodiment, the additional query values generated from query value q will then have the following values: $q+1, q+2, \dots, q+d-1$. For example, if query value $q = 100$ and $d = 5$, the client will generate 4 additional query values 101, 102, 103, 104. Let j represent a member of a query value set containing query value q and the additional query values. Then $j=0$ refers to query value q . If $j > 0$, this means that it is one of the additional query values. Therefore, for $j=0$ to $(d-1)$, each member j of the query value set will have a value of $q+j$. One skilled in the art would appreciate that this embodiment will work in a similar manner to the preferred embodiment described herein.

[0067] It will be apparent that various other modifications and adaptations of the application will be apparent to the person skilled in the art after reading the foregoing disclosure without departing from the spirit and scope of the application and it is intended that all such modifications and adaptations come within the scope of the appended claims.

[0068] In the application, unless specified otherwise, the terms “comprising”, “comprise”, and grammatical variants thereof, are intended to represent “open” or “inclusive” language such that they include recited elements but also permit inclusion of additional, non-explicitly recited elements.

CLAIMS

1. A method for querying a database to return a data record based on a query, the query having a query value, the data record having a data record value, the method comprising the steps of:-

composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

generating a public key;

producing an encrypted data record value by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup;

producing an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup;
and

performing a matching operation on the encrypted data record value and the encrypted query value wherein the matching operation comprises the steps of:-

performing a mathematical operation on the encrypted data record value and the encrypted query value;

removing the element of the second subgroup from the encrypted data record value;

removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the data record value and the query value;

using the public key to identify when the data record value is equal to the query value in the function; and

returning the data record.

2. The method of claim 1 wherein the step of composing a composite group having a first subgroup, a second subgroup and a third subgroup further comprises composing a bilinear mapping on the composite group and wherein the step of performing a mathematical operation on the encrypted data record value and the encrypted query value comprises applying bilinear mapping on the encrypted data record value and the encrypted query value.

3. The method of claims 1 or 2 wherein the step of producing an encrypted data record value by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup comprises the steps of:

encrypting the data record value as an exponent in the first element of the first subgroup;

encrypting the first random factor as an exponent in the element of the second subgroup;

multiplying the first element of the first subgroup and the element of the second subgroup to produce the encrypted data record value.

4. The method of anyone of claims 1 to 3 wherein the step of producing an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup comprises the steps of:-

encrypting the query value as an exponent in the second element of the first subgroup;

encrypting the second random factor as an exponent in the element of the third subgroup;

multiplying the second element of the first subgroup and the element of the third subgroup to produce the encrypted query value.

5. A method for querying a database to return a key interval based on a query, the query having a query value, the key interval having an anchor value and a key interval length, the method comprising the steps of:-

partitioning the database into a plurality of key intervals;

composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

generating a public key;

producing an encrypted key interval value for each of the plurality of key intervals by mapping the anchor value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup;

generating a plurality of additional query values based on the query value and a query counter, the query counter being dependent on the key interval length;

producing an encrypted query value for the query value and each additional query value by mapping the query value and the query counter to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup;

and

performing a matching operation on each encrypted key interval value and on each encrypted query value wherein the matching operation comprises the steps of:-

performing a mathematical operation on the encrypted key interval value and the encrypted query value;

removing the element of the second subgroup from the encrypted key interval value;

removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the query value, the query counter and the anchor value;

using the public key to identify when the anchor value is equal to the difference between the query value and the query counter in the function; and

returning the key interval.

6. The method of claim 5 wherein the step of composing a composite group having a first subgroup, a second subgroup and a third subgroup further comprises composing a bilinear mapping on the composite group and wherein the step of performing a mathematical operation on the encrypted key interval value and the encrypted query value comprises applying bilinear mapping on the encrypted key interval value and the encrypted query value.

7. The method of claims 5 or 6 wherein the key interval further comprises a lower value and an upper value.

8. The method of claim 7 wherein the anchor value is the lower value and wherein the step of generating a plurality of additional query values based on the query value and a query counter comprises the step of generating a plurality of additional query values by subtracting the query counter from the query value.

9. The method of claim 7 wherein the anchor value is the upper value and wherein the step of generating a plurality of additional query values based on the query value and a query counter comprises the step of generating a plurality of additional query values by adding the query counter to the query value.

10. A system for querying a database to return a data record based on a query, the query having a query value, the data record having a data record value, the system comprising :-

a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

a server for housing the database, the database for storing at least one encrypted data record value and the public key, the at least one encrypted data record value produced by mapping the data record value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup;

wherein the client machine produces an encrypted query value by mapping the query value to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup, and sends the encrypted query value to the server;

and wherein the server performs a matching operation on the at least one encrypted data record value and the encrypted query value by performing a mathematical operation on the at least one encrypted data record value and the encrypted query value, removing the element of the second subgroup from the at least one encrypted data record value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the data record value and the query value, using the public key to identify when the data record value is equal to the query value in the function, and returning the data record to the client machine.

11. The system of claim 10 wherein the composite group has a bilinear mapping and wherein the mathematical operation is bilinear mapping.

12. The system of claims 10 or 11 wherein for the at least one encrypted data record value, the data record value is mapped to the first element of the first subgroup by encrypting the data record value as an exponent in the first element of the first subgroup, and the first random factor is mapped to an element of the second subgroup by encrypting the first random factor as an exponent in the element of the second subgroup.

13. The system of anyone of claims 10 to 12 wherein for the encrypted query value, the query value is mapped to the second element of the first subgroup by encrypting the query value as an exponent in the second element of the first subgroup, and the second random factor is mapped to an element of the third subgroup by encrypting the second random factor as an exponent in the element of the third subgroup.

14. A system for querying a database to return a key interval based on a query, the query having a query value, the key interval having an anchor value and a key interval length, the system comprising :-

a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

a server for housing the database, the database for storing the public key and at least one encrypted key interval value, the at least one encrypted key interval value produced by mapping the anchor value to a first element of the first subgroup and mapping a first random factor to an element of the second subgroup;

wherein the client machine generates a plurality of additional query values based on the query value and a query counter, the query counter being dependent on the key interval length, and produces an encrypted query value for the query value and each additional query value by mapping the query value and the query counter to a second element of the first subgroup and mapping a second random factor to an element of the third subgroup, and sends each encrypted query value to the server;

and wherein the server performs a matching operation on the at least one encrypted key interval value and each encrypted query value by performing a mathematical operation on the at least one encrypted key interval value and the encrypted query value, removing the element of the second subgroup from the at least one encrypted key interval value, removing the element of the third subgroup from the encrypted query value to obtain a function which contains a random multiple of the difference between the query value, the query counter and the anchor value, using the public key to identify when the anchor value is equal to the difference between the query value and the query counter in the function, and returning the key interval to the client machine.

15. The system of claim 14 wherein the composite group has a bilinear mapping and wherein the mathematical operation is bilinear mapping.

16. The system of claims 14 or 15 wherein the key interval further comprises a lower value and an upper value.

17. The system of claim 16 wherein the anchor value is the lower value and wherein the client machine generates the plurality of additional query values by subtracting the query counter from the query value.

18. The system of claim 16 wherein the anchor value is the upper value and wherein the client machine generates the plurality of additional query values by adding the query counter to the query value.

19. A method for querying a database to return a bottom tier key interval based on a query, the query having a top query value and a bottom query value, the method comprising the steps of:-

- partitioning the database into a plurality of top tier key intervals, each top tier key interval having a top anchor value and a top key interval length;

- partitioning each top tier key interval into a plurality of bottom tier key intervals, each bottom tier key interval having a bottom anchor value and a bottom key interval length;

- composing a composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

- generating a public key;

- producing an encrypted top tier key interval value for each of the plurality of top tier key intervals by mapping the top anchor value to a first element of the first subgroup and mapping a first random factor to a first element of the second subgroup;

- producing an encrypted bottom tier key interval value for each of the plurality of bottom tier key intervals by mapping the bottom anchor value to a second element of the first subgroup and mapping a second random factor to a second element of the second subgroup;

- generating a plurality of additional top query values based on the top query value and a top query counter, the top query counter being dependent on the top key interval length;

- generating a plurality of additional bottom query values based on the bottom query value and a bottom query counter, the bottom query counter being dependent on the bottom key interval length;

- producing an encrypted top query value for the top query value and each additional top query value by mapping the top query value and the top query counter to a third element of the first subgroup and mapping a third random factor to a first element of the third subgroup;

- producing an encrypted bottom query value for the bottom query value and each additional bottom query value by mapping the bottom query value and the bottom query

counter to a fourth element of the first subgroup and mapping a fourth random factor to a second element of the third subgroup;

performing a first matching operation on each encrypted top tier key interval value and on each encrypted top query value wherein the matching operation comprises the steps of:-

performing a mathematical operation on the encrypted top tier key interval value and the encrypted top query value;

removing the first element of the second subgroup from the encrypted top tier key interval value;

removing the first element of the third subgroup from the encrypted top query value to obtain a first function which contains a random multiple of the difference between the top query value, the top query counter and the top anchor value;

using the public key to identify a matched top tier key interval when the top anchor value is equal to the difference between the top query value and the top query counter in the first function;

performing a second matching operation on each encrypted bottom tier key interval value in the matched top tier key interval and on each encrypted bottom query value wherein the matching operation comprises the steps of:-

performing a mathematical operation on the encrypted bottom tier key interval value and the encrypted bottom query value;

removing the second element of the second subgroup from the encrypted bottom tier key interval value;

removing the second element of the third subgroup from the encrypted bottom query value to obtain a second function which contains a random multiple of the difference between the bottom query value, the bottom query counter and the bottom anchor value;

using the public key to identify when the bottom anchor value is equal to the difference between the bottom query value and the bottom query counter in the second function; and

returning the bottom tier key interval.

20. A system for querying a database to return a bottom tier key interval based on a query, the query having a top query value and a bottom query value, the system comprising :-

a client machine for generating a public key and composing a composite group, the composite group having a first subgroup, a second subgroup and a third subgroup, each subgroup having a plurality of elements;

a server for housing the database, the database being partitioned into a plurality of top tier key intervals, each top tier key interval having a top anchor value and a top key interval length and further partitioned into a plurality of bottom tier key intervals, each bottom tier key interval having a bottom anchor value and a bottom key interval length, the database for storing the public key and at least one encrypted top tier key interval value and at least one encrypted bottom tier key interval value, where the at least one encrypted top tier key interval value is produced by mapping the top anchor value to a first element of the first subgroup and mapping a first random factor to a first element of the second subgroup, and the at least one encrypted bottom tier key interval value is produced by mapping the bottom anchor value to a second element of the first subgroup and mapping a second random factor to a second element of the second subgroup;

wherein the client machine generates a plurality of additional top query values based on the top query value and a top query counter, and a plurality of additional bottom query values based on the bottom query value and a bottom query counter, the top query counter being dependent on the top key interval length and the bottom query counter being dependent on the bottom key interval length, and the client machine produces an encrypted top query value for the top query value and each additional top query value by mapping the top query value and the top query counter to a third element of the first subgroup and mapping a third random factor to a first element of the third subgroup, and produces an encrypted bottom query value for the bottom query value and each additional bottom query value by mapping the bottom query value and the bottom query counter to a fourth element of the first subgroup and mapping a fourth random factor to a second element of the third subgroup;

and wherein the server performs a first matching operation on each encrypted top tier key interval value and on each encrypted top query value by performing a mathematical operation on the encrypted top tier key interval value and the encrypted top query value, removing the first element of the second subgroup from the encrypted top tier key interval value, removing the first element of the third subgroup from the encrypted top query value to obtain a first function which contains a random multiple of the difference between the top query value, the top query counter and the top anchor value, using the public key to identify a matched top tier key interval when the top anchor value is equal to the difference between the top query value and the top query counter in the first function, and the server performs a second matching operation on each encrypted bottom tier key interval value in the matched top tier key interval and on each encrypted bottom query value by performing a mathematical operation on the encrypted bottom tier key interval value and the encrypted bottom query

value, removing the second element of the second subgroup from the encrypted bottom tier key interval value, removing the second element of the third subgroup from the encrypted bottom query value to obtain a second function which contains a random multiple of the difference between the bottom query value, the bottom query counter and the bottom anchor value, using the public key to identify when the bottom anchor value is equal to the difference between the bottom query value and the bottom query counter in the second function, and returning the bottom tier key interval.