# (12) UK Patent Application (19) GB (11) 2 419 000

(43) Date of A Publication

12.04.2006

(21) Application No:

0422170.1

(22) Date of Filing:

06.10.2004

(71) Applicant(s):

Hew lett-Packard Development Company L.P., 20555 S.H.249, Houston, Texas 77070, **United States of America** 

(72) Inventor(s): Liqun Chen

(74) Agent and/or Address for Service: Hew lett-Packard Limited IP Section, Filton Road, Stoke Gifford, BRISTOL, BS34 8QZ, United Kingdom

G06F 1/00 (2006.01) H04L 29/06 (2006.01) H04L 9/00 (2006.01)

(52) UK CL (Edition X ): **G4A** AAP A23B A23X H4P PPEB

(56) Documents Cited:

None

(58) Field of Search: UK CL (Edition X ) G4A, H4P

INT CL7 G06F, H04L

Other: WPI, EPODOC, INSPEC, TXTE

- (54) Abstract Title: Proving relationships between data
- (57) A method for a data provider to provide and prove an association between a first data value A and a second data value B, comprising: computing evidence E of the form

 $E = H(A \mid\mid B)^{k \cdot (p-1)/q} \mod p,$ 

where H is a secure hash function, of an association between A and B; ensuring that the verifier has A, B and E; and running an interactive proof with the verifier to convince the verifier that the evidence is valid and that the provider knows the value of k without disclosing the value of k to the verifier.

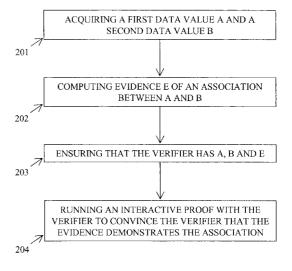


Figure 2

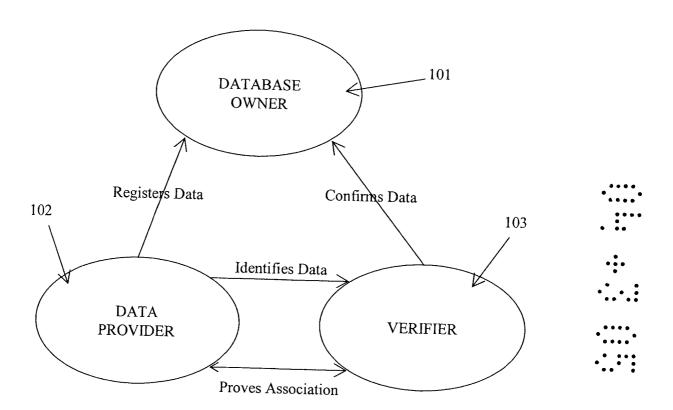


Figure 1



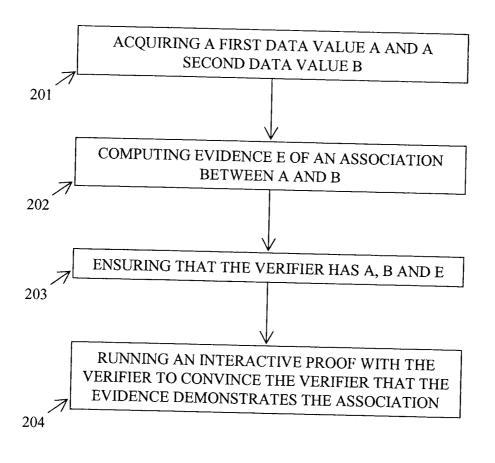


Figure 2

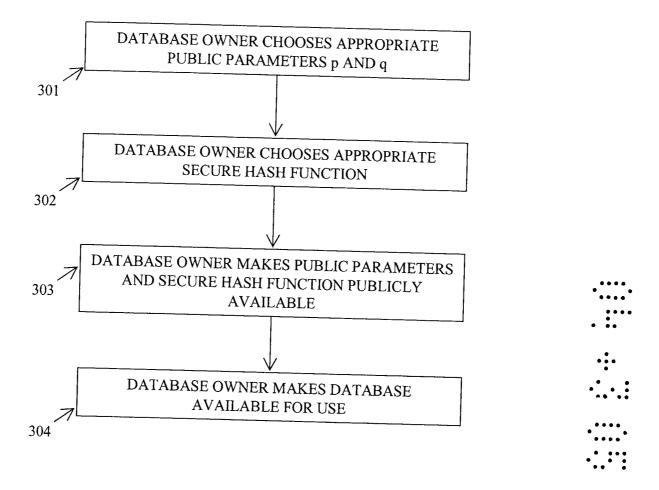


Figure 3



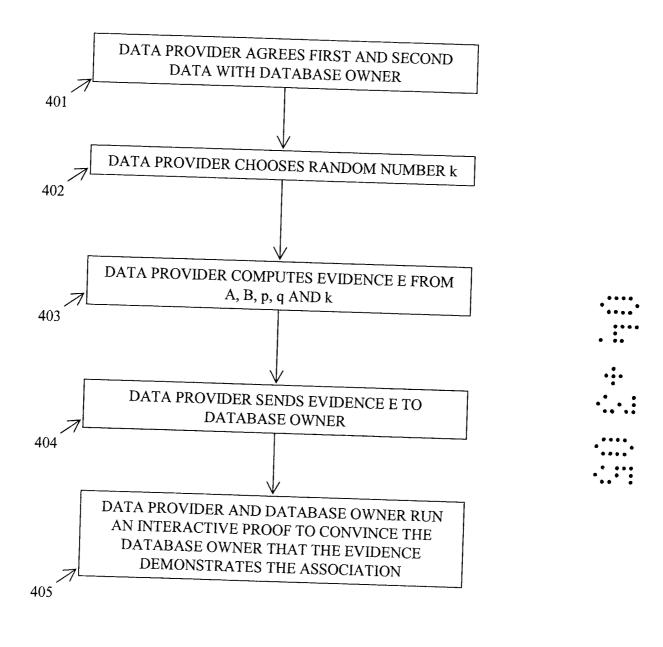


Figure 4



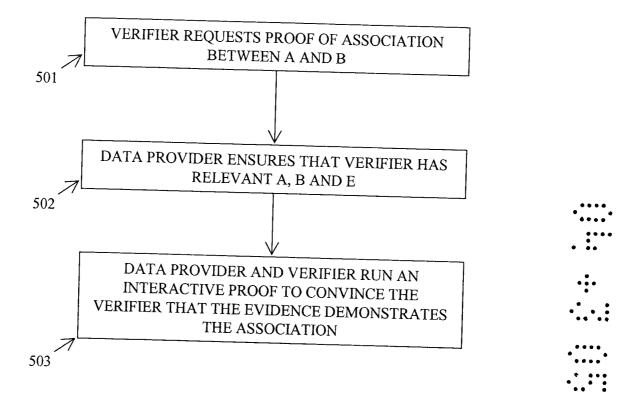


Figure 5



## PROVING RELATIONSHIPS BETWEEN DATA

### Field of the Invention

~ <del>~</del>~~~

5 The invention relates to the proving of relationships between data.

## Background to the Invention

Circumstances arise when there is an association between two or more pieces of data, and the association is sufficiently sensitive that some form of privacy control is needed in respect of the association. This typically requires placing privacy controls on at least one of the pieces of data. Maintaining such privacy controls particularly if the association is merely one of many similar associations – may be very onerous.

15

20

25

## Summary of the Invention

In a first aspect, the invention provides a method for a provider to provide and prove an association between a first data value A and a second data value B, comprising: computing evidence E of the form

$$E = H(A \mid\mid B)^{k \cdot (p-1)/q} \bmod p$$

where H is a secure hash function, of an association between A and B; ensuring that the verifier has A, B and E; and running an interactive proof with the verifier to convince the verifier that the evidence is valid and that the provider knows the value of k without disclosing the value of k to the verifier.

#### Description of Drawings

30 Specific embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, of which:

Figure 1 shows a relationship between parties providing a context in which embodiments of the invention may be used;



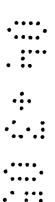


Figure 2 shows a method for a provider to provide and prove an association between a first data value A and a second data value B according to an embodiment of the invention;

Figure 3 shows the elements of an algorithm for setting up a database for holding data values in accordance with an embodiment of the invention;

Figure 4 shows the elements of a protocol for registering data in the database set up as in Figure 3 in accordance with an embodiment of the invention; and

Figure 5 shows the elements of a protocol for proving the link between a first data value A and a second data value B held in the database set up as in Figure 3 in accordance with an embodiment of the invention.

## Description of Specific Embodiments

10

15

20

25

30

Figure 1 shows a context relevant for application of embodiments of the invention. Figure 1 represents three parties and their relationships and interactions. A database owner 101 maintains a database which holds public data. The database owner 101 is trusted to hold reliable data by users of the database. Data in the database is submitted to the database owner for entry by data providers 102. A data provider 102 provides first data and second data, and also third data which provides evidence of an association between the first data and the second data, this data being registered with the database owner 101 and entered onto the database. The database contains no indication of any relationship between any of the first data, the second data and the third data. A verifier 103 wishes to establish whether there is an association between the first data and the second data (having learned at least one of these data from another source) and contacts the data provider. The data provider 102 enables the verifier 103 to check that the first data, second data and third data are all held in the database. The data provider 102 then interacts with the verifier 103 to prove that the third data evidence shows an association between the first data and the second data, but without enabling the verifier 103 to be able to convince others.

This context may be relevant where the first data represents a name of a product vendor and the second name represents a data value identifying product produced by that product vendor. In one example, these products are computational components used for trusted computing. A computer platform may ship with one of several components that perform the same trust function, or dynamically provide the

option to use one of multiple components that perform the same trust function. If a component can identify itself via a number, the platform can customise itself according to the idiosyncrasies of the component actually fitted and/or in use. Thus using the numbers provides the opportunity for a platform to be built to use components from multiple vendors, for example, and software could automatically adapt to the particular component that was fitted.

Such vendors may wish to preserve their anonymity in general yet be able to prove an association with a number when/if necessary. A database constructed in the manner indicated above allows this to be achieved. The database is publicly available. The database does not disclose the link between a specific name and a specific number, even though all of the vendor's names, numbers and the evidence linking them appear in the database at the same time (the effectiveness of the anonymity that this provides is limited by the size of the database). The database owner himself need not be aware of any linkage between A, B and E – this can simply be forgotten after A, B and E are loaded into the database. A vendor, who is the owner of a specific name and number, can prove the link between the name and number to any verifier in a private matter. Except for the vendor, no one else is able to prove the link between the name and number. After the vendor proves a link to a verifier, the verifier is convinced of the proof but is not able to transfer his knowledge of the proof to a third party.

10

15

20

25

30

Steps carried out by the data provider – product vendor in this example – are set out in Figure 2. The first data value A and the second data value B are acquired 201 – these could be generated by the data provider himself, or by agreement with the database provider. The data provider then computes 202 evidence E of an association between the first data value A and the second data value B. These are placed in the database by the database owner (who is first satisfied that the evidence does indeed demonstrate the association). When a verifier appears requiring proof of the association, the data provider ensures 203 that the verifier knows the relevant values of A, B and E – which the verifier can check in the database to confirm that they are a possible related set of values – and then runs 204 an interactive proof with the verifier to convince the verifier that the evidence does indeed demonstrate the association.

Two exemplary schemes are described to implement this approach. Both exemplary schemes have an initial setup algorithm ("Setup"), followed by a



registration protocol ("Register") and a proof protocol ("Prove"), the elements of each being set out in Figures 3, 4 and 5 respectively.

#### Example 1

5

10

#### Setup

Setup is discussed with reference to Figure 3. To setup the database, the database owner does the following. He first chooses (301) two public parameters p and q, such that both p and q are large primes and  $q \mid (p-1)$  is an integer. A recommended (but not limited) size of p is at least 1024-bit and the size of q is 160-bit. He then chooses (302) a highly secure hash function, H, such as SHA-1. He then makes (303) the primes and hash function identifier publicly available. At this point, the database can be made available (304) for use.

#### 15 Register

This is shown in Figure 4. When a data provider registers itself with the database owner for the purpose of adding information of the data provider into the database, the data provider and the database owner first negotiate 401 a name A and a random number B for the data provider. Both name and number are unique and have selected sizes (for example, that A is at most 20 bytes and B is a fixed 16 bits). The scheme proposed here does not limit the size of a name string or a number string. Authentication between the database provider and the database owner would be expected in most practical arrangements implementing this approach — this is outside the scope of the present protocol, however.

25

20

Based on the agreed name and number, the data provider performs as follows:

- Chooses 402 a random number  $k \in [1, q-1]$ .
- Computes a base value

$$g = H(A \mid\mid B)^{(p-1)/q} \bmod p$$

- where A||B| denotes concatenation between A and B.
  - Computes 403 evidence

$$E = g^k \mod p$$

Stores k in secret.



- Optionally, stores g. (The alternative is to re-compute g every time needed).
- Sends 404 E to the database owner.

The database owner and the data provider then interact 405 as follows to convince the

5 database owner that the evidence E demonstrates an association between A and B.

Upon receipt of the evidence E, the database owner performs as follows:

- Chooses a random number  $a \in ]1, q-1]$ .
- Computes a challenge

$$C = H(A || B)^{a*(\rho-1)/q} \bmod \rho$$

Sends C to the data provider.

Upon receipt of the value C, the data provider performs as follows:

- Chooses a random number  $b \in ]1, q-1]$ .
- Computes a response

15 
$$D = C^{k} * H(A || B || 3)^{b*(p-1)/q} \mod p$$

(it should be noted that instead of 3, any other convenient non-zero value can be used).

- Send D to the database owner.
- Upon receipt of the value D, the database owner releases a to the data provider.

Upon receipt of the value a, the data provider performs as follows:

Checks if

$$C = g^a \mod p$$

- 25 holds.
  - If the check fails, aborts the protocol.
  - Otherwise releases the value b to the database owner.

Upon receipt of the value b, the database owner performs as follows:

30 • Checks if

$$D = E^a * H(A || B || 3)^{b*(p-1)/q} \mod p$$

holds.



- If the check fails, rejects the evidence E.
- Otherwise publishes A, B and E in the database.

#### **Prove**

This is shown in Figure 5. When a data provider needs to prove the link between A and B to a verifier, the data provider first indicates his A, B and E to the verifier (the verifier will generally know B – in the case of a trusted component discussed above, the value may be held as an identifier on the trusted component). Then the data provider convinces the verifier of the evidence. This part is very similar to the register protocol.

The verifier first performs as follows:

- Chooses a random number  $x \in ]1, q-1]$ .
- Computes a challenge

$$U = H(A \mid\mid B)^{x \cdot (p-1)/q} \bmod p$$

Sends U to the data provider.

Upon receipt of the value U, the data provider performs as follows:

- Chooses a random number  $y \in ]1, q-1]$ .
- 20 Computes a response

$$V = U^k * H(A || B || 4)^{y*(p-1)/q} \mod p$$

(As for D in the register protocol, it should be noted that instead of 4, any other convenient non-zero value can be used).

• Sends V to the verifier.

25

30

15

Upon receipt of the value V, the verifier releases x to the data provider.

Upon receipt of the value x, the data provider performs as follows:

Checks if

$$U = g^x \mod p$$

holds.

- If the check fails, aborts the protocol.
- Otherwise releases the value y to the verifier.



Upon receipt of the value b, the verifier performs as follows:

Checks if

$$V = E^x * H(A || B || 4)^{y*(\rho-1)/q} \mod p$$

5 holds.

- If the check fails, rejects the proof.
- Otherwise accepts the proof.

### Example 2

10

15

#### <u>Setup</u>

Reference can again be made to Figure 3. As for Example 1, to setup the database, the database owner does the following. He first chooses (301) two public parameters p and q, such that both p and q are large primes and  $q \mid (p-1)$  is an integer. A recommended (but not limited) size of p is at least 1024-bit and the size of q is 160-bit. He then chooses (302) a highly secure hash function, p, such as SHA-1. He then makes (303) the primes and hash function identifier publicly available. At this point, the database can be made available (304) for use.

#### 20 Register

Reference can again be made to Figure 4. As for Example 1, when a data provider registers itself with the database owner for the purpose of adding information of the data provider into the database, the data provider and the database owner first negotiate 401 a name A and a random number B for the data provider. Both name and number are unique and have selected sizes (for example, that A is at most 20 bytes and B is a fixed 16 bits). The scheme proposed here does not limit the size of a name string or a number string. Authentication between the database provider and the database owner would be expected in most practical arrangements implementing this approach – this is outside the scope of the present protocol, however.

30

25

Based on the agreed name and number, the vendor performs as follows:

- Chooses 402 a random number  $k \in ]1, q-1]$ .
- Computes 403 evidence



$$E = H(A \mid\mid B)^{k \cdot (p-1)/q} \bmod p$$

where A||B denotes concatenation between A and B.

- Sends 404 E to TCG.
- Stores *k* in secret.

5

After receiving the evidence E, TCG computes

$$g = H(A \mid\mid B)^{(p-1)/q} \bmod p$$

TCG and the vendor then run 405 an interactive Discrete Logarithm Proof (DLP) protocol namely DLP(p, g, E) to convince TCG that  $E = g^k \mod p$  and the vendor has knowledge of the value k.

There exist a number of secure protocols to achieve such a function. One such is described in D. Chaum, "Zero-knowledge undeniable signatures", in Advances in Cryptology – Proceedings of Crypto'90, Lecture Notes in Computer Science 473, pages 458-464, Springer-Verlag, 1991. This protocol will be referred to as DLP(p,g,E) and involves two parties, Prover and Verifier.

Prover and Verifier share the values p, g and E. Prover knows the discrete logarithm  $k = \log_g E$ . Prover wants to convince Verifier of Prover's knowledge of this value, but (i) does not want to disclose the value k to Verifier, and (ii) does not want Verifier to convince others of this fact after the proof.

DLP(p, g, E) works as follows:

25

15

- 1. Verifier randomly chooses  $a \in [1, q-1]$ , computes  $X = g^a \mod p$ , and sends X to Prover.
- 2. Prover randomly chooses  $b \in ]1, q-1]$ , computes  $Y = X*g^b \mod p$  and  $Z = Y^k \mod p$ , and then sends Y and Z to Verifier.
- 30 3. Verifier releases the value a to Prover.
  - 4. Prover checks if  $X = g^a \mod p$  holds. If the check fails, Prover aborts the protocol. Otherwise Prover releases the value b to Verifier.



5. Verifier checks if both  $Y = X*g^b \mod p$  and  $Z = E^a*E^b \mod p$  hold. If any of these checks fails, Verifier rejects the proof. Otherwise Verifier accepts the proof.

### 5 Prove

This is again shown in Figure 5. When a data provider needs to prove the link between A and B to a verifier, the data provider first indicates his A, B and E to the verifier (the verifier will generally know B – in the case of a trusted component discussed above, the value may be held as an identifier on the trusted component).

Then the data provider convinces the verifier of the evidence. This is achieved in a similar way to the register protocol.

More specifically, the verifier computes

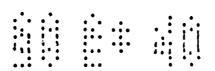
$$g = H(A \mid\mid B)^{(p-1)/q} \bmod p$$

15

The vendor and verifier then run a DLP protocol DLP(p, g, E) to prove that E is evidence of the link between A and B exactly as shown above in respect of the Register protocol.



20



#### **CLAIMS**

1. A method for a provider to provide and prove an association between a first data value A and a second data value B, comprising:

computing evidence E of the form

$$E = H(A || B)^{k \cdot (p-1)/q} \bmod p$$

5

15

25

where H is a secure hash function, of an association between A and B; ensuring that the verifier has A, B and E; and

running an interactive proof with the verifier to convince the verifier that the evidence is valid and that the provider knows the value of k without disclosing the value of k to the verifier.

- 2. A method as claimed in claim 1, further comprising placing A, B and E on a database maintained by a database owner trusted by the verifier.
- 3. A method as claimed in claim 2, wherein the first data value A and the second data value B are agreed between the provider and the database owner.
- 4. A method as claimed in claim 3 or claim 4, wherein the provider runs an interactive proof with the database owner to convince the database owner that the provider knows the value of k without disclosing the value of k to the database owner.
  - 5. A method as claimed in any preceding claim, where the or each interactive proof is a discrete logarithmic proof.
  - 6. A method for a database owner to maintain a database, comprising:

choosing and making public a first prime p, a second prime q and a secure hash function H;

obtaining first data values A and second data values B from data 30 providers;

obtaining evidence E from the data providers;

confirming that the evidence E has been validly generated from the first prime p, the second prime q, the secure hash function H and the



respective first data value A and the respective second data value B in an interactive proof with the data provider; and

making each first data value A, second data value B and evidence E publicly available in the database without indicating any relationship between them.

7. A method as claimed in claim 6, wherein first data values A and second data values B are agreed between the database owner and the respective data providers.

10

5









**Application No:** 

GB0422170.1

**Examiner:** 

Mr Adam Tucker

Claims searched:

1-5

Date of search: 24 February 2005

## Patents Act 1977: Search Report under Section 17

#### Documents considered to be relevant:

Relevant to claims	Identity of document and passage or figure of particular relevance
	None

#### Categories:

- X Document indicating lack of novelty or inventive step
- Y Document indicating lack of inventive step if combined with one or more other documents of same category.
- & Member of the same patent family
- A Document indicating technological background and/or state of the art.
- P Document published on or after the declared priority date but before the filing date of this invention.
- E Patent document published on or after, but with priority date earlier than, the filing date of this application.

#### Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKCX:

G4A; H4P

Worldwide search of patent documents classified in the following areas of the IPC07

G06F; H04L

The following online and other databases have been used in the preparation of this search report

WPI, EPODOC, INSPEC, TXTE