A Secrecy Criterion for Outsourcing Encrypted Databases Based on Inference Analysis

Yasunori ISHIHARA(1), Member, Yasuhiro USHIROZAKO(1,)*, Nonmember, Kengo MORI(1), Member, and Jun FURUKAWA(1), Nonmember

SUMMARY In this letter, we propose a secrecy criterion for outsourcing encrypted databases. In encrypted databases, encryption schemes revealing some information are often used in order to manipulate encrypted data efficiently. The proposed criterion is based on inference analysis for databases: We simulate attacker’s inference on specified secret information with and without the revealed information from the encrypted database. When the two inference results are the same, then secrecy of the specified information is preserved against outsourcing the encrypted database. We also show that the proposed criterion is decidable under a practical setting.

1. Introduction

Outsourcing database management is a big trend these days, due to the spread of cloud computing services. Encrypted databases are a promising technology for outsourcing databases because they can achieve efficient data manipulation while they can keep the content of the database secret from the database managers. To manipulate encrypted data efficiently, encryption schemes revealing some information are useful. CryptDB[1] is one of the famous encrypted databases using such encryption schemes. In CryptDB, for example, a deterministic encryption scheme is applied to attribute values that may be involved in equality comparison, because the same plaintexts are always converted to the same ciphertexts by a deterministic scheme.

However, there has been little formal discussion on the security of outsourcing encrypted databases using such encryption schemes. Most of the researches including [1] conclude that their encrypted databases are secure if the used encryption schemes are secure. Such discussion ignores the impact of the information revealed from the ciphertexts.

In this letter, we propose a formal secrecy criterion for outsourcing such encrypted databases. To take account of the information revealed from the ciphertexts, the proposed criterion is based on inference analysis for databases. Inference analysis is to examine the possibility of inference attacks, and an inference attack is to identify (or narrow down the candidates for) the returned value of a sensitive query using authorized views and/or general domain knowledge. Consider the candidate sets for the returned value of a specified sensitive query for the following two cases: (1) the attacker is just a database user, and (2) the attacker is a party of a database user and a manager of the encrypted database. The proposed criterion states that when the two candidate sets are the same, then secrecy of the specified information is preserved against outsourcing the encrypted database.

Next, we show that the proposed criterion is decidable under a practical setting including the following situation: databases are relational and queries are written in relational algebra including selection, projection, union, product, and difference. The decision algorithm uses a bag-based data model for incomplete information [2], which has a good closure property for forward and inverse of relational operations. The decidability of the equality of the two candidate sets stems from the decidability of Presburger arithmetic.

2. A Model of Encrypted Databases

A model of encrypted databases of this letter, which is based on the model of CryptDB[1], is shown in Fig. 1. Let $D$ be a raw database. For a database or query $X$, let $[X]$ denote the database or query obtained by encrypting all values appearing in $X$. All the interactions between users and the encrypted database $[D]$ are obtained through a secure proxy. When a user issues a query $Q_1$, the secure proxy translates it to some queries $Q'_1, \ldots, Q'_{N_1}$ and $Q'$ such that

$$Q_1(x) = Q'(Q'_1(x), \ldots, Q'_{N_1}(x)) \text{ and } Q'_1([X]) = [Q'_1(X)]$$

for each $j(1 \leq j \leq N_1)$, where $X$ denotes a formal parameter. The secure proxy then sends $[Q'_1] \ldots [Q'_{N_1}]$ to the encrypted database $[D]$. The encrypted database returns $[Q'_1(D)] \ldots, [Q'_{N_1}(D)]$ using the information revealed by the encryption schemes. Finally, the secure proxy decrypts $[Q'_1(D)] \ldots, [Q'_{N_1}(D)]$, computes the final answer $Q_1(D)$ using $Q'$, and sends it to the user.

According to CryptDB, we consider the following three encryption schemes:

- RND: Randomized. No information is revealed.
- DET: Deterministic. $a = b$ if and only if $[a] = [b]$.
- OPE: Order-preserving. $a < b$ if and only if $[a] < [b]$.

The values of the same attribute are encrypted by the same encryption scheme. On the other hand, different encryption keys are used for the different attributes. Hence, encrypted
values appearing at different columns are incomparable.

**Example 1:** Consider an encrypted database shown in Table 1. Suppose that the values of attributes Name, Room, and Salary are encrypted by DET, DET, and OPE, respectively. Let \( Q_1(X) = \sigma_{\text{Name}=\text{Alice}}(X) \), i.e., selection of tuples with Name = Alice. If a user issues \( Q_1 \), the secure proxy translates it into \( Q'_1 \) and \( Q''_1 \), where \( Q'_1(X) = Q_1(X) \) and \( Q''_1 \) is the identity query. Hence, in this case, \( [Q'_1](X) = [\sigma_{\text{Name}=\text{Alice}}](X) \). The encrypted database returns

\[
[\sigma_{\text{Name}=\text{Alice}}(D)] = [\sigma_{\text{Name}=\text{Alice}}](D) = [\sigma_{\text{Name}=\text{Alice}}](D) = \{[(\text{Alice}), (\text{A101}), (\text{200})]\}.
\]

The secure proxy decrypts it and sends \([(\text{Alice}, \text{A101}, 200)]\) to the user.

\[\square\]

3. The Proposed Secrecy Criterion

See Fig. 1 again. Suppose that queries \( Q_i \) (1 \( \leq \) \( i \) \( \leq \) \( n \)) are authorized to a user. Such authorized queries are regarded as the view definitions for the user. Using \( Q_i \), their answers \( Q_i(D) \), and the schema (i.e., the set of attribute names) \( R \) of \( D \), the user wants to identify (or narrow down the candidates for) the answer \( Q_S(D) \) to a sensitive query \( Q_S \). To achieve this, collusion with the manager of the encrypted database is effective in some cases. In this letter, we suppose that for the user, the secure proxy in Fig. 1 is a black box but all data outside the secure proxy are observable. Formally, the user can obtain the following facts (F1)–(F6) by the collusion:

(F1) correspondence between each attribute name and its ciphertext,
(F2) the entire encrypted database \([D]\),
(F3) the encryption scheme (i.e., RND, DET, or OPE) used for each attribute,
(F4) \( Q'_1(X), \ldots, Q'_N(X) \) to which \( Q_1(X) \) is translated,
(F5) the encrypted queries \([Q'_j(D)]\) corresponding to \( Q'_j \), and
(F6) the encrypted answers \([Q'_j(D)] = [Q'_j](D)]\).

Since all data outside the secure proxy are observable, it is reasonable that (F2), a part of (F5), and (F6) can be obtained. On the contrary, all of (F1), (F3), and (F4) are not necessarily obtained by the user. Moreover, for (F5), all the correspondence of \([Q'_j](D)\) and \( Q'_j \) are not necessarily obtained. However, by assuming the “worst case” in the sense that all of (F1), (F3), (F4), and (F5) are obtained by the user, our criterion will be conservative. That is, if outsourcing preserves secrecy in our criterion, then it actually does.

Let \( f \) denote a mapping from a ciphertext to the corresponding plaintext guessed by the user. These facts induce the following constraints (C1)–(C3) on \( D \) and \( f \):

(C1) \( f([\text{A}]) = A \) for each attribute name \( A \), induced by (F1);
(C2) (in-)equalities among plaintexts which are encrypted by DET or OPE in \([D]\), induced by (F2) and (F3); and
(C3) \( Q'(f([Q'_j(D)])), \ldots, f([Q'_N(D)]) = Q(D) \), induced by (F4), (F5), and (F6).

**Example 2:** Consider again the encrypted database \([D]\) and the query \( Q_1 \) in Example 1. Let \( Q_2(X) = \sigma_{\text{Name}=?}(\pi_{\text{Name},\text{Room}}(X)) \), i.e., projection of Name and Room and then selection of tuples with Room beginning by letter C. Suppose that \( Q_1 \) and \( Q_2 \) are authorized to a user. Let \( Q_S(X) = \pi_{\text{Name}}(\sigma_{\text{Room}=?}(\pi_{\text{Name},\text{Room}}(X))) \), i.e., the user wants to know the salary of David. Clearly, the user can obtain no information only from the definitions of \( Q_1 \) and \( Q_2 \) and their answers \( Q_1(D) \) and \( Q_2(D) \).

Suppose that the secure proxy translates \( Q_1 \) and \( Q_2 \) as follows:

- \( Q_1(X) = Q'(Q'_1(X)) \), where \( Q'_1(X) = Q_1(X) \) and \( Q' \) is the identity query; and
- \( Q_2(X) = Q''(Q''_1(X)) \), where \( Q''_1(X) = \pi_{\text{Name},\text{Room}}(X) \) and \( Q''(X) = \sigma_{\text{Room}=?}(X) \).

Then, by colluding with the manager of the encrypted database, the user obtains the following constraints:

(C1) \( f([\text{Name}]) = \text{Name}, f([\text{Room}]) = \text{Room}, f([\text{Salary}]) = \text{Salary} \).
(C2) \( f([\text{Alice}]) \neq f([\text{Bob}]), f([\text{Alice}]) \neq f([\text{Carol}]), \ldots, f([\text{A101}]) \neq f([\text{B202}]), \ldots, f([\text{C303}]) = f([\text{C303}]), f([200]) > f([150]), f([200]) < f([300]), \ldots \).
(C3) \( f \left( \begin{bmatrix} \text{Alice} & \text{A101} & 200 \end{bmatrix} \right) = \text{Alice} \ A101 \ 200 \)

\[ \sigma_{\text{Room}=\text{C3}??} \left( f \left( \begin{bmatrix} \text{Alice} & \text{A101} \\ \text{Bob} & \text{B202} \\ \text{Carol} & \text{C303} \\ \text{David} & \text{C303} \end{bmatrix} \right) \right) = \begin{cases} \text{Carol} & \text{C303} \\ \text{David} & \text{C303} \end{cases} \]

Now, let us see how the candidates for the answer to \( Q_S \) are narrowed down by (C1)–(C3). By the first equality of (C3), the user obtains \( f([\text{Alice}]) = \text{Alice} \), \( f([\text{A101}]) = \text{A101} \), and \( f([200]) = 200 \). Moreover, by the second equality of (C3), the user obtains that C303 appears twice. Since the values of attribute Room are encrypted by DET, the user obtains \( f([\text{C303}]) = \text{C303} \), and hence,

\[
\begin{align*}
(f([\text{Alice}]) &= \text{Carol} \land f([\text{David}]) = \text{David} \\
(f([\text{Carol}]) &= \text{Carol} \land f([\text{David}]) = \text{David} \\
(f([\text{Carol}]) &= \text{Carol} \lor f([\text{David}]) = \text{David} \\
(f([\text{Carol}]) &= \text{Carol} \lor f([\text{David}]) = \text{David} \\
(f([\text{Carol}]) &= \text{Carol} \lor f([\text{David}]) = \text{David}.
\end{align*}
\]

From \([D]\), the salary of David must be \( f([300]) \) or \( f([350]) \). The user obtains that in either case it is greater than 200, the salary of Alice, by (C2).

Let \( D \) be a database instance, \( Q_i \) (1 ≤ i ≤ n) authorized queries to a user, \( Q_S \) a sensitive query, and \( R \) the database schema of \( D \). Define \( \text{Cand}(D) = \{D' \mid Q_i(D') = Q_i(D) \text{ for all } i \} \). \( \text{Cand}(D) \) is the set of candidates for \( D \) that are consistent with the information obtained by the user solely. Also define \( \text{Cand}^+(D) = \{D' \in \text{Cand}(D) \mid D' \text{ satisfies the constraints } (C1)-(C3) \text{ for some } f \} \). \( \text{Cand}^+(D) \) is the set of candidates for \( D \) that are consistent with the information obtained by the user colluding with the manager of the encrypted database. In general, \( \text{Cand}^+(D) \) becomes smaller than \( \text{Cand}(D) \), and hence, the candidates for the value of \( Q_S \) can be narrowed down. This idea derives the following secrecy criterion:

**Definition 1:** An encrypted database \([D]\) preserves secrecy with respect to \( Q_S \) against outsourcing if \( \{Q_S(D') \mid D' \in \text{Cand}^+(D)\} = \{Q_S(D') \mid D' \in \text{Cand}(D)\} \).

4. Decidability of the Proposed Secrecy Criterion

In this section, we show that the proposed secrecy criterion is decidable in the following setting:

- (S1) Relations are implemented as bags (hence the user can obtain the number of the same tuples in an answer to a query);
- (S2) Queries consist of selection, projection, bag union, bag Cartesian product, bag difference, and unique (collapsing the same multiple tuples to one tuple); and
- (S3) The user knows the upper bound \( M \) of the total number of tuples of any bags (including intermediate ones during query processing).

Only (S3) would be restrictive from the theoretical point of view. This is necessary for ensuring the closure property explained later. From the practical point of view, (S3) is not so unnatural because the upper bound of the number of tuples can be estimated by using background knowledge.

In what follows, we first introduce CGV-bags [2], a bag-based data model for incomplete information, to represent the candidate sets for database instances or query answers. CGV-bags have the best closure property among the data models explored in [2]. Then, we show how to compute the two candidate sets in Definition 1. Lastly, we explain how to decide the equality of the two candidate sets.

4.1 CGV-Bags

In what follows, we assume that the domain of any attribute is the set \( \mathbb{N} \) of non-negative integers, that is, every attribute value is encoded by a non-negative integer.

Let \( V \) be a set of variables. Let \( - \) denote the subtraction on non-negative integers, which is defined by:

\[
- a - b = \begin{cases} a - b & \text{if } a \geq b \\ 0 & \text{otherwise.} \end{cases}
\]

A non-negative integer expression is an expression consisting of non-negative integers, variables, and operators + (addition), × (multiplication), and −. A valuation is a function from \( V \) to \( \mathbb{N} \). The domain of a valuation \( v \) is naturally extended to non-negative integer expressions as follows:

- For each constant \( a \in \mathbb{N} \), let \( v(a) = a \).
- For non-negative integer expressions \( x + y \), \( x \times y \), and \( x - y \), let \( v(x + y) = v(x) + v(y) \), \( v(x \times y) = v(x) \times v(y) \), and \( v(x - y) = v(x) - v(y) \).

Moreover, conditional expressions are macros on non-negative integer expressions defined in Table 2, where \( a \) and \( b \) are arbitrary non-negative integer expressions, and \( c_i \) is an arbitrary non-negative integer expression such that \( v(c_i) \) is equal to either 0 or 1 for any valuation \( v \).

| Conditional expression | Definition |
|------------------------|------------|
| true                   | 1          |
| false                  | 0          |
| \( a = b \)            | 1 - ((a - b) + (b - a)) |
| \( a \leq b \)         | 1 - (a - b) |
| \( \lor c_i \)         | 1 - (1 - \( c_i \)) |
| \( \land c_i \)       | \( c_i \) |
| \( \neg c_i \)        | 1 - \( c_i \) |
| if \( c_i \) then \( a \) else \( b \) | \( c_i \times a + (1 - c_i) \times b \) |

An example of a CV-bag is shown in Table 3. Intuitively,
this CV-bag says that someone $x$ has three, say, desks in room A101, and Alice has two desks in room B202 or C303. The domain of a CV-bag $E$, denoted as $dom(E)$, is the set of all tuples $u$ such that $E(u)$ is not literally 0.

**Definition 2:** A CV-bag is a pair $(E, G)$ of a CV-bag $E$ and a conditional expression $G$ called global condition. □

Let us extend the domain of a valuation $v$ to V-tuples and CV-bags as follows:

- For a V-tuple $v$ over $R$, $v(u)$ is a tuple over $R$ such that for each $A \in R$, $(v(u))(A) = v((u))(A))$.
- For a CV-bag $E$ and a tuple $t$, $v(E)(t) = \sum_{u:v(u)=t} v(E(u))$.

The set of database instances represented by $(E, G)$, denoted by $rep(E, G)$, is defined as $\{v(E) \mid v(G) = true\}$.

### 4.2 Computation of the Candidate Sets

The idea of computing $Cand(D)$ is straightforward. Similarly to the inference analysis in our previous work [3], we apply the inverse of $Q_i$ to the obtained answer $Q_i(D)$. To realize this idea, we first introduce a useful macro $eq(E_1, E_2)$ such that for any valuation $v$, $v(eq(E_1, E_2)) = true$ if and only if $v(E_1) = v(E_2)$. The formal definition is as follows:

$$eq(E_1, E_2) = \bigwedge_{F} (valid(F) \land numcheck(F, E_1, E_2)),$$

where $F$ goes through all partitions of $dom(E_1) \cup dom(E_2)$,

$$\begin{align*}
valid(F) &= \bigwedge_{F \in F} \bigwedge_{u, u' \in F} u(A) = u'(A) \land \\
numcheck(F, E_1, E_2) &= \bigwedge_{F \in F} \bigwedge_{u' \in F} \bigwedge_{u \in F} u(A) \neq u'(A) \land \\
&\bigwedge_{F, F' \in F, F+\text{F'} \in F} \bigwedge_{u \in F} u(A) = u(A).
\end{align*}$$

According to [2], all the forward operations given in (S2) are closed on CV-bags, but some inverse operations (e.g., inverse of selection) are not. However, under the assumption (S3), any inverse operation is closed if its forward version is closed. To see this, consider the inverse of selection $\sigma^{-1}$ (with some selection condition). $\sigma^{-1}(E, G)$ is represented by $(E_0, G \land eq(\sigma(E_0), E))$, where $dom(E_0)$ consists of $M$ V-tuples with distinct new variables and for each $u \in dom(E_0)$, $E_0(u) = “if \; x_u = true \; then \; 1 \; else \; 0”$ with another new variable $x_u$. Note that $E_0$ represents the set of all the bags whose total number of tuples is at most $M$.

$Cand(D)$ is the intersection of $rep(Q_i^{-1}(Q_i(D)))$ for all authorized queries $Q_i$. Intersection $\cap_{rep}$ with respect to $rep$ is easy: $(E_1, G_1) \cap_{rep} (E_2, G_2) = (E_1, G_1 \land G_2 \land eq(E_1, E_2))$.

To compute $Cand^*(D)$, we first translate (C1)–(C3) to a CV-bag. Each ciphertext in $D$ is represented by a new variable (e.g., see Table 4). A valuation is expected to map a ciphertext to its plaintext (i.e., a valuation corresponds to the mapping $f$). (C1) is used for clarify the attribute sets of the CGV-bag. (C2) is straightforwardly translated into a part of the global condition. (C3) is also translated into a part of the global condition, by using the useful macro $eq$. Finally, $Cand^*(D)$ is obtained by taking $\cap_{rep}$ of the obtained CV-bag and $Cand(D)$.

**Example 3:** Constraints (C2) and (C3) in Example 2 are translated as follows:

(C2) $w_1 \neq x_1, w_1 \neq y_1, \ldots, w_2 \neq x_2, w_2 \neq y_2, \ldots, w_2 = z_2, w_3 \geq x_3, w_3 \leq y_3, \ldots$

(C3) $eq(w_1, w_2, w_3) \rightarrow 1$

Finally, $\{Q_5(D') \mid D' \in Cand(D)\}$ and $\{Q_5(D') \mid D' \in Cand^*(D)\}$ can be represented by CGV-bags because all the forward operations given in (S2) are closed.

### 4.3 Decision of the Equality of the Candidate Sets

Let $(E, G)$ and $(E^*, G^*)$ be CV-bags representing the candidate sets $\{Q_5(D') \mid D' \in Cand(D)\}$ and $\{Q_5(D') \mid D' \in Cand^*(D)\}$, respectively. Without loss of generality, we assume that the variables appearing in $(E, G)$ and $(E^*, G^*)$ are disjoint. To see the equality of them, it suffices to check whether $G \land \forall x (G^* \lor \neg eq(E, E^*))$ is unsatisfiable, where $x$ denotes all the variables appearing in $(E^*, G^*)$. Such satisfiability check is impossible in general because multiplications of integer variables are allowed in CV-bags. However, in our context, we construct CGV-bags from definite database instances $Q(D)$. From [2] and the translation of (C1)–(C3) presented above, multiplications of integer variables are never introduced during the computation of the candidate sets. Consequently, the expression is a Presburger
formulas and hence its satisfiability is decidable.

**Theorem 1:** Under the assumptions (S1)–(S3), it is decidable whether an encrypted database \([D]\) preserves secrecy with respect to \(Q_S\) against outsourcing. □

Unfortunately, it would not be realistic to apply our algorithm every time a user issues a query and every time the database instance is updated, because satisfiability of a Presburger formula is known to be in \(3\text{EXPTIME}\) and \(2\text{EXPTIME}\)-hard. Therefore, from the practical point of view, it will be useful to propose efficiently decidable sufficient conditions for preserving secrecy and incremental algorithms with respect to database updates.

5. Related Work

In this letter, we have referred to CryptDB [1] as a model of encrypted databases. CryptDB and its successor Monsoon [4] are quite valuable because they have shown that encrypted databases with special encryption schemes such as order-preserving [5], homomorphic [6], and searchable ones [7] run reasonably fast. However, as already stated, there is no formal discussion on the impact of the information revealed from the ciphertexts.

In order to manipulate encrypted data efficiently, Hacigümiş et al. proposed to prepare appropriate indices [8]. In their method, the domain of data values are partitioned into subdomains and the index of a data value is defined as the subdomain to which the data value belongs. Given a query, the encrypted database uses only indices and computes an encrypted superset of the correct answer. Then, on the client side, the superset is decrypted and the correct answer is computed. In their paper, the efficiency of their method was evaluated, but the partial information revealed by the indices was not taken care of.

In recent years, several secrecy criteria have been well studied, especially in the context of privacy preservation. The criterion proposed in this letter follows \(k\)-secrecy proposed in our previous work [3]. \(k\)-secrecy resembles \(l\)-diversity [9] in the semantics; both focus on the number of candidates for the sensitive information. The difference is that \(l\)-diversity handles linking attacks of two relations while \(k\)-secrecy, which is not limited to relational databases, handles attacks that infer all the possible candidates. In other words, \(k\)-secrecy supposes more powerful attackers.

Differential privacy [10] is a criterion for anonymization mechanisms of statistical databases. Roughly speaking, an anonymization mechanism achieves differential privacy if for any pair of database instances differing on at most one element, the pair of anonymized data are close enough (i.e., the mechanism adds enough noise so that the pair of data are indistinguishable). The problem setting and the purpose of differential privacy are quite different from ours. However, both of the criteria are based on the difference of inferable secret information between two cases where different information are given to the attacker.

6. Conclusions

We have proposed a secrecy criterion of outsourcing encrypted databases, and shown its decidability under a practical setting. To the best of the authors’ knowledge, this is the first work that formally discusses the revealed information from outsourced encrypted databases. However, this result is just a first step toward formal security analysis of encrypted databases, and hence, there is a lot of work to do. We are planning to work on quantitative measures of secrecy and instance-independent secrecy next.

**Acknowledgments**

The authors would like to thank Associate Professor Kozo Okano of Osaka University for his kind instruction on Presburger arithmetic. The authors also appreciate the helpful and constructive comments from the anonymous reviewer.

**References**

[1] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan, “CryptDB: Protecting confidentiality with encrypted query processing,” Proc. SOSP '11, pp.85–100, 2011.

[2] A. Yamaguchi, S. Shimizu, Y. Ishihara, and T. Fujiwara, “Bag-based data models for incomplete information and their closure properties,” J. Intell. Inf. Syst., vol.30, no.2, pp.115–135, 2008.

[3] K. Hashimoto, K. Sakano, F. Takasuka, Y. Ishihara, and T. Fujiwara, “Verification of the security against inference attacks on XML databases,” IEICE Trans. Inf. & Syst., vol.E92-D, no.5, pp.1022–1032, May 2009.

[4] S. Tu, M.F. Kaashoek, S. Madden, and N. Zeldovich, “Processing analytical queries over encrypted data,” PVldb, vol.6, no.5, pp.289–300, 2013.

[5] A. Boldyreva, N. Chenette, Y. Lee, and A. O’Neill, “Order-preserving symmetric encryption,” Proc. EUROCRYPT ‘09, pp.224–241, 2009.

[6] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” Proc. EUROCRYPT ’99, pp.223–238, 1999.

[7] D.X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” Proc. S&P ’00, pp.44–55, 2000.

[8] H. Hacigümiş, B. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” Proc. SIGMOD ’02, pp.216–227, 2002.

[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “\(l\)-diversity: Privacy beyond \(k\)-anonymity,” ACM Trans. Knowl. Discov. Data, vol.1, no.1, Article no.3, March 2007.

[10] C. Dwork, “Differential privacy,” Proc. ICALP ’06, pp.1–12, 2006.