ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20181218005415
Original scientific paper

Efficient Q-Value Zero-Leakage Protection Scheme in SRS Regularly Publishing Private Data

Zongmin CUI, Lungui ZHANG, Bin WU, Zhiqiang ZHAO, Zhuolin MEI, Zongda WU

Abstract: Spontaneous Reporting System (SRS) has been widely established to collect adverse drug events. Thus, SRS promotes the detection and analysis of ADR (adverse drug reactions), such as the FDA Adverse Event Reporting System (FAERS). The SRS data needs to be provided to researchers. Meanwhile, the SRS data is publicly available to facilitate the study of ADR detection and analysis. In general, SRS data contains private information of some individual characteristics. Before the information is published, it is necessary to anonymize private information in the SRS data to prevent disclosure of individual privacy. There are many privacy protection methods. The most classic method for protecting SRS data is called as PPMS. However, in the real world, SRS data is growing dynamically and needs to be published regularly. In this case, PPMS has some shortcomings in the memory consumption, anonymity efficiency, data update and data security. To remove these shortcomings, we propose an Efficient Q-value Zero-leakage protection Scheme in SRS regularly publishing private data, called EQZS. EQZS can deal with almost all of potential attacks. Meanwhile, EQZS removes the shortcomings of PPMS. The experimental results show that our scheme EQZS solves the problem of privacy leakage in SRS regularly publishing private data. Meanwhile, EQZS significantly outperforms PPMS on the efficiency of memory consumption, privacy anonymity and data update.

Keywords: data anonymity; privacy protection; Q-value zero-leakage; regularly publishing private data; spontaneous reporting system

1 INTRODUCTION

Although drugs are essential for people to cure diseases, taking drugs may be accompanied by serious adverse drug reactions (ADR). A new drug has to go through a series of clinical trials before it enters the market. Unfortunately, the number of volunteers involved in drug trials is quite limited. Thus, it is difficult to prior collect all ADR in the pre-marketing stage. Most developed countries have established spontaneous reporting system (SRS) to collect ADR. SRS supports the analysis and detection of ADR data, such as the US Food and Drug Administration (FDA)'s FAERS [1], the UK's Yellow Card scheme [2], Canada's MedEffect [3], and so on. Moreover, some countries even publicly publish their SRS data to promote ADR researches [4].

However, the published SRS data also brings about some privacy threats. The SRS data [5] is a type of microdata that contains sensitive individual health information. The SRS data is associated with specific patients. For example, George gets liver cancer. He does not want anyone to know it. That is, it is private information. One day, George has ADR after taking a new drug. The ADR data is then published in SRS. Consequently, the adversary can get George's private information by the ADR data. Therefore, the patient's ADR should be protected to prevent this disclosure of identity and private information [6].

In general, publisher removes identity attributes before the data is published, such as Name, SSN, Phone, etc. However, even if all identity attributes have been removed, the adversary can still get the private information [7]. The adversary links the published data with external data (such as voter lists) through quasi-identity attributes, such as Gender, Job, Age, ZIP code, etc. Therefore, many researches anonymize the SRS data before publishing it, such as k-anonymity [8].

In [9], Lin etc. point out that traditional anonymity methods are not applicable in SRS data. This is due to some characteristics of SRS data, including a large number of individual records, multi-value sensitive attributes, rare events, etc. Recently, a privacy protection model called MS has been proposed to anonymize SRS data to prevent the disclosure of individual privacy [10]. In the real world, new ADR data is published in SRS at any time. Thus, countries like the USA and Canada publish SRS data periodically (for example, quarterly) to handle this dynamically increasing datasets. However, MS is designed to be used in a single static publishing environment. Therefore, MS is very clumsy to handle a series of dynamic datasets. Once new data is published, continuous data publishing method [11] needs to anonymize all data. Thus, the data update is too costly. Dynamic data publishing method [12] only anonymizes unmodified data. Then an adversary could use the modified data to steal the privacy information. Therefore, this method is weak to against BFL-attacks (i.e. Backward-attack, Forward-attack, and Latest-attack [13]). To remove this issue, Wang etc. propose a method called PPMS [13]. PPMS can anonymize SRS data in the periodical data publishing scenario. Meanwhile, PPMS prevent the disclosure of personal sensitive information caused by BFL-attacks. However, PPMS has three shortcomings as follows. (1) Inefficient. When PPMS queries data with an old CaseID, it needs to filter the data to determine which data needs to be anonymized. Thus, it needs to develop a set of storage spaces to store the data that needs to be anonymized. These operations waste computing and storage resources. (2) The cost of updating is high. Necessarily, there is the latest case table to be published. When a new table is published, PPMS gradually anonymize the related data from the latest table to the oldest table. As a result, almost all of tables need to be re-anonymized. Therefore, the cost of such data updates is high. (3) Insecure. *Q*-values are the data which are valuable to the adversary. Thus, only *Q*-values need to be anonymized. PPMS use the coarse-grained judgment to expand the data range. Thus, after the anonymity, some attributes of *Q*-value are protected by very good anonymity. However, some other attributes of *Q*-value have reduced anonymity effect, which results in new security risks.

To remove these issues, we propose an Efficient *Q*-value Zero-leakage protection scheme in SRS regularly publishing private data, which is called as EQZS. EQZS can find out the data that needs to be anonymized without the filtering operation. Thus, we effectively decrease the requirements of computing and storage resources. Meanwhile, EQZS anonymizes the related data from the oldest table to the latest table. Therefore, we almost only anonymize the new published table. Thus, we greatly reduce the cost of data updates. Finally, through the fine-grained judgment to expand the data range, we ensure that each attribute of *Q*-value is protected by the maximum anonymity. Therefore, we enhance the security.

1.1 Research Motivation

Usually each record of SRS data contains a CaseID to track the subsequent behavior of the event [14]. All records with the same CaseID point to the same patient. Before publication, all records are anonymized for protecting individual privacy. Unfortunately, the CaseID also provides a useful link for adversary through a series of anonymous datasets. By the link, the adversary can exclude those records that they do not want to steal.

The following examples reflect our challenges. For better illustration, let us consider the three consecutive quarters of the published SRS datasets in Tabs. 1(a), 1(b) and 1(c). Each quarter's data satisfies *k*-anonymity ($k = 3$). In another word, Fig. 1 shows three anonymized case tables: R_0, R_1 and R_2 (i.e. TableID = 0, 1 and 2).Each table contains five attributes: LineID, CaseID, Sex, Age and Disease. LineID is for our fine-grained judgment. CaseID and Disease are public for ADR. Only Sex and Age can be anonymized. Therefore, Sex and Age are the Q-value. Each line of a case table is a case. We use $r_{i,j}$ to denote a case, where *i* denotes the TableID and *j* denotes the LineID. For example, $r_{1,2}$ = {14, ANY, [20, 30], HIV}. In addition, we use $r_{i,j}$. Attribute to specifically denote a data. For example, $r_{1,2}$·CaseID = 14, $r_{1,2}$·Sex = ANY, etc. Besides, we use CaseID$_i$ to denote all CaseIDs in case table R_i.

Attack 1: Backward-attack. We assume an adversary knows that his neighbor Kitty had a Q-value {female, 22}. Meanwhile, he also knows that Kitty suffered from some adverse drug reactions in Quarter 2. First, the adversary links the Q-value with Tab. 1(b). Then he learns that Kitty's records are in CaseIDset {11, 14, 17}. Second, he links the possible CaseID set {11, 14, 17} with previous published Tab. 1(a). In Tab. 1(a), the Sex with CaseID = 11 is male. However, the Sex of Kitty is female. Therefore, he can exclude the CaseID 11 from the possible CaseID set. This makes the original 3-anonymous information records to be 2-anonymous by Tab. 1(b). This makes it easier for adversary to steal Kitty's information.

Attack 2: Forward-attack. Based on the above operation, the adversary knows that Kitty's information is in the possible CaseID set {14, 17}. Now, the adversary can further use this possible CaseID set to link to the subsequently published SRS data. In Tab. 1(c), the sex with CaseID = 17 is male. Thus, the adversary can exclude the CaseID 17 from the possible CaseID set. That is, he concludes that Kitty's CaseID is 14.

Attack 3: Latest-attack. We assume an adversary knows that his neighbor Tony's *Q*-value is {male, 23}. Meanwhile, he also knows that Tony had adverse drug reactions in Quarter 3 first time. This means that Tony's CaseID is definitely a new CaseID in Quarter 3. That is, Tony's CaseID cannot appear in any previously published data. First, the adversary links the *Q*-value with Tab. 1(c). He gets the possible CaseID set {17, 3, 21}. However, CaseIDs 17 and 3 appear in Tab. 1(b). Therefore, he concludes that Tony's CaseID is 21.

Table 1 The published SRS datasets in the three consecutive quarters

(a) R_0: QUARTER 1				
LineID	CaseID	Sex	Age	Disease
1	11	Male	[25, 30]	Fever
2	8	Male	[25, 30]	Flu
3	6	Male	[25, 30]	Diabetes
4	14	Female	[25, 30]	HIV
5	15	Female	[25, 30]	Flu
6	16	Female	[25, 30]	Diabetes
(b) R_1: QUARTER 2				
LineID	CaseID	Sex	Age	Disease
1	11	ANY	[20, 30]	Fever
2	14	ANY	[20, 30]	HIV
3	17	ANY	[20, 30]	Diabetes
4	3	Male	[25, 30]	Flu
5	19	Male	[25, 30]	Flu
6	20	Male	[25, 30]	Fever
7	31	Male	[25, 30]	HIV
8	42	Male	[25, 30]	Flu
(c) R_2: QUARTER 3				
LineID	CaseID	Sex	Age	Disease
1	33	Female	[20, 30]	Flu
2	9	Female	[20, 30]	Diabetes
3	35	Female	[20, 30]	HIV
4	5	Female	[25, 30]	Flu
5	47	Female	[25, 30]	Fever
6	17	Male	[15, 30]	Diabetes
7	3	Male	[15, 30]	Flu
8	21	Male	[15, 30]	HIV

From the above examples, we can find that even if the published table has been anonymized, it is still going to be cracked by BFL-attacks. Therefore, we need to propose a secure and efficient privacy protection model to re-anonymize the *Q*-value to improve the security.

1.2 Our Contributions

Our contributions are illustrated as follows.

(1) We propose a new privacy protection model. Our model directly expands the data range during the comparison judgment. When a data's range does not need to be expanded, we directly exit the loop. That is, we omit the filtering operation. Thus, we improve the computing efficiency and reduce the storage cost.

(2) We propose a new anonymity frame. Our frame gradually anonymizes the related *Q*-value from the oldest table to the latest table. Thus, when a new table is published, we almost only need to anonymize this new table. Therefore, we greatly reduce the cost of data updates.

(3) We propose a new fine-grained judgment method. Our method does not anonymize the entire line of *Q*-value. We make a fine-grained judgment of a single attribute of a *Q*-value. After the fine-grained judgment, we ensure

that each attribute of a *Q*-value is protected by the maximum anonymity. Therefore, we enhance the security.

The rest of this paper is organized as follows. In Section 2, we review the related works. In Section 3, we study the three attack modes of BFL. Section 4 provides our core algorithm against BFL-attacks. In Section 5, we compare the EQZS with existing methods by experiments. In Sections 6, we conclude this paper.

2 RELATED WORKS

2.1 Continuous Data Publishing

Continuous data publishing means that if a data owner wants to publish data, he needs to publish all collected data, even if some data has been published [15]. In other words, if a data owner has previously collected a set of data D_1 at timestamp t_1 and published R_1 (i.e. the anonymized version of D_1), the data owner then collects a new set of data D_2 at timestamp t_2. Then he should publish R_2 (the anonymized version of D_2) as an anonymized version of all collected data (i.e. $D_1 \cup D_2$). In general, the published version R_i ($i \geq 1$) should be the anonymized version of $D_1 \cup D_2 \cup ... \cup D_i$.

The definition of continuous data publishing shows that privacy protection models [11, 16] proposed in this scenario have a common problem. They need to anonymize all collected data. Once new data is published, they need to anonymize all data to ensure that the private data will not be leaked. Thus, the data update is too costly. Our scheme EQZS gradually anonymizes the data from oldest timestamp to latest timestamp. Thus, when a new dataset is published, we almost only need to anonymize this new dataset. That is, we greatly reduce the cost of data updates.

2.2 Dynamic Data Publishing

In a dynamic data publishing scenario, data owner can insert and/or delete data from the original dataset [17]. If a data owner collects a dataset D_1 at timestamp t_1 and publishes R_1 (i.e. the anonymized version of D_1), he continuously collects new data and inserts them into D_1 during the time period $[t_1, t_2)$. In addition, he may delete and update some data from D_1. Finally, he obtains an updated version D_2 of D_1 at timestampt$_2$. Then the version R_2 published at timestamp t$_2$ is the anonymized version of D_2. In fact, the version R_i published at timestamp t_i is the anonymized version of D_i.

According to the definition of dynamic data publishing, when the published data is modified, the published version is modified accordingly. If privacy protection methods [12, 18] proposed in this scenario do not anonymize the modified published version, the adversary can steal privacy data from the modified published version. If these methods anonymize the modified published version, the data update is too costly. In our scheme EQZS, we use a fine-grained judgment to anonymize all modified published version by pre-finding old CaseIDs. Our anonymity frame provides higher efficiency and security than these methods.

2.3 Regularly Data Publishing

PPMS [13] is the most classic privacy protection method proposed for the regularly data publishing scenario. Meanwhile, PPMS is the most relevant method to our challenges. In this method, data owner regularly publishes new data (such as SRS data). A data owner collects a dataset D_1 during the time period $[t_0, t_1)$ and publishes R_1 (i.e. the anonymized version of D_1) at timestamp t_1. Next, he collects a dataset D_2 during the time period $[t_1, t_2)$ and publishes R_2. R_1 and R_2 may have the same CaseID, i.e. $r_{1,i}$·CaseID = $r_{2,j}$·CaseID. Then $r_{1,i}$·CaseID is the old CaseID of $r_{2,j}$·CaseID.

However, PPMS has four shortcomings as follows.

(1) High memory consumption. In Tab. 1, $r_{0,1}$·CaseID = 11 = $r_{1,1}$·CaseID and $r_{0,4}$·CaseID = 14 = $r_{1,2}$·CaseID. Thus, $r_{0,1}$·CaseID and $r_{0,4}$·CaseID are old cases. PPMS needs to filter CaseIDs to find the old CaseID. Based on the old CaseID, PPMS can find the *Q*-value that needs to be anonymized. Therefore, PPMS needs to store $r_{1,1}$ and $r_{1,2}$ into memory. However, these two data both do not need to be anonymized. Our scheme omits the filter operation. We do not store these two data into memory. We only store the data that needs to be anonymized into memory. Thus, we decrease the memory consumption.

(2) Low computing efficiency. PPMS needs to run a loop for the filter operation. We omit the unnecessary loop to enhance the computing efficiency.

(3) High update cost. PPMS compares R_i with R_{i-1} to possibly anonymize R_i and R_{i-1}. Next, PPMS compares R_{i-1} with R_{i-2} to possibly anonymize R_{i-1} and R_{i-2}, and so on until the initial table R_1 is compared. As a result, when a new table R_{i+1} is published, PPMS needs to re-anonymize almost all of tables. Thus, the cost of data updates is very high. Our scheme EQZS gradually possibly anonymizes related *Q*-values from R_1 to $R_{i.}$ That is, we firstly compare R_1 with R_2 to possibly anonymize R_1 and R_2. Next, we compare R_2 with R_3 to possibly anonymize R_2 and R_3, and so on until the latest table R_i is possibly anonymized. Therefore, when a new table R_{i+1} is published, we only need to possibly anonymize R_{i+1} and R_i. Therefore, we greatly improve the efficiency of data update.

(4) Not secure enough. When the range of a *Q*-value $\{r_{i,j}\cdot q_1, r_{i,j}\cdot q_2\}$ needs to be expanded by an old *Q*-value $\{r_{i-1,k}\cdot q_1, r_{i-1,k}\cdot q_2\}$, PPMS directly assigns $r_{i-1,k}\cdot q_1$ to $r_{i,j}\cdot q_1$ and $r_{i-1,k}\cdot q_2$ to $r_{i,j}\cdot q_2$ (i.e. $r_{i,j}\cdot q_1 = r_{i-1,k}\cdot q_1$ and $r_{i,j}\cdot q_2 = r_{i-1,k}\cdot q_2$). If $r_{i-1,k}\cdot q_1 > r_{i,j}\cdot q_1$ and $r_{i-1,k}\cdot q_2 > r_{i,j}\cdot q_2$, this is certainly fine. However, if $r_{i-1,k}\cdot q_1 > r_{i,j}\cdot q_1$ and $r_{i-1,k}\cdot q_2 < r_{i,j}\cdot q_2$, $r_{i,j}\cdot q_1$ is well protected anonymously but the range of $r_{i,j}\cdot q_2$ is narrowed. That is, PPMS's coarse-grained judgment results in new security risks. Our fine-grained judgment protects each attribute of a *Q*-value by maximum anonymity. Therefore, we enhance the security.

3 BFL-ATTACKS

Definition 1. (Coverage of *Q*-value). Given case table R_i, we need to compare R_i with R_{i-1} to find R_i's *Q*-value that needs to be anonymized. $r_{i,j}$ denotes a case, where *i* is the TableID and *j* is the LineID. The range of $r_{i,j}$'s attribute *q* is denoted by $r_{i,j}\cdot q$. If $r_{i,j}$·CaseID = $r_{i-1,k}$·CaseID and $r_{i,j}\cdot q < r_{i-1,k}\cdot q$, we define that $r_{i-1,k}\cdot q$

covers $r_{i,j}$·q. In other words, $r_{i,j}$·q has a smaller range and has a worse anonymity effect. In this situation, the range of $r_{i,j}$·q needs to be expanded by $r_{i-1,k}$·q (i.e. $r_{i,j}$·q=$r_{i-1,k}$·q).

For example, $r_{2,6}$·CaseID = 17 = $r_{1,3}$·CaseID and $r_{2,6}$·Sex = Male < $r_{1,3}$·Sex = ANY, so $r_{2,6}$·Sex needs to be expanded by $r_{1,3}$·Sex (i.e. $r_{2,6}$·Sex = ANY).

The core of BFL-attacks is to exclude cases from case tables by coverage judgment. When the number of excluded case is big enough, the adversary can steal the privacy information.

3.1 Backward-attack

If an old Q-value cannot cover the new Q-value, the adversary can exclude the old case from case table.

Definition 2. (Backward-attack). R denotes all case tables in the system. If there are $r_{i,j} \in R \wedge r_{i-1,k} \in R \wedge r_{i,j}$·CaseID = $r_{i-1,k}$·CaseID $\wedge$ $r_{i,j}$·$q \geq r_{i-1,k}$·q, Backward-attack happens. In this situation, the adversary can exclude $r_{i-1,k}$ from R.

3.2 Forward-attack

If a new Q-value cannot cover the old Q-value, the adversary can exclude the new case from case table.

Definition 3. (Forward-attack). R denotes all case tables in the system. If there are $r_{i,j} \in R \wedge r_{i+1,k} \in R \wedge r_{i,j}$·CaseID = $r_{i+1,k}$·CaseID $\wedge$ $r_{i,j}$·$q \geq r_{i+1,k}$·q, Forward-attack happens. In this situation, the adversary can exclude $r_{i+1,k}$ from R.

3.3 Latest-attack

The adversary knows that a patient had ADR at timestamp t_i first time. Thus, his cases definitely do not exist in previously published version. Given a possible CaseID $r_{i,j}$·CaseID of the patient, if previously published version includes $r_{i,j}$·CaseID, $r_{i,j}$ can be excluded.

Definition 4. (Latest-attack). R denotes all case tables in the system. If there are $r_{i,j} \in R \wedge r_{i-1,k} \in R \wedge r_{i,j}$·CaseID = $r_{i-1,k}$·CaseID $\wedge$ $r_{i,j}$ shouid be published at timestamp t_i first time, Latest-attack happens. In this situation, the adversary can exclude $r_{i,j}$ from R.

4 EQZS

4.1 Core Idea

Our core idea is composed of the following seven parts.

(1) **Against Backward-attack**. If the old Q-value cannot cover the new Q-value, the old case can be excluded from the possible case set. Thus, if we assign the larger range of new Q-value to the old Q-value, the old Q-value then can cover the new Q-value. That is, we effectively stop Backward-attack.

(2) **Against Forward-attack.** If the new Q-value cannot cover the old Q-value, the new case can be excluded from the possible case set. Thus, if we assign the larger range of old Q-value to the new Q-value, the new Q-value then can cover the old Q-value. That is, we effectively stop Forward-attack.

(3) **Against Latest-attack**. Possible CaseID set CaseID$_i$ should be published at timestamp t_i first time. If CaseID$_i$ includes old CaseID, Latest-attack happens. To against Latest-attack, we firstly remove the leaked old CaseID from CaseID$_i$. Secondly, we add new secure CaseID into CaseID$_i$. That is, we create a new case set that has k cases. Then we anonymize this new case set by k-anonymity. It is clear that we can effectively stop Latest-attack.

(4) **Low memory consumption.** We omit the filter operation. We only store the data that needs to be anonymized into memory to decrease the memory consumption.

(5) **High computing efficiency**. We omit the unnecessary loop to enhance the computing efficiency.

(6) **Low update cost**. When a new table is published, we almost only need to possibly anonymize the new table to improve the efficiency of data update.

(7) **High Security**. Our fine-grained judgment protects each attribute of a Q-value by maximum anonymity to enhance the security.

4.2 Algorithm

Algorithm 1 shows our core idea. The algorithm takes an old table set R, the table number x, anonymity number k and Q-value as input. Meanwhile, it takes the new anonymized table set R_{new} as output.

Phase 1 (Steps 1-20). This phase finds out all the data that might be attacked by BFL-attacks. First, Algorithm 1 finds out all old CaseIDs (Steps 5-7). Second, we find out all cases that need to be possibly anonymized (Steps 8-10). Third, Algorithm 1assigns the larger range of new Q-value to the old Q-value against Backward-attack (Steps 13-14).

Phase 2 (Steps 21-29). This phase is used to against forward-attack. If the range of old Q-value is larger than new Q-value (Step 24), we assign the larger range of old Q-value to the new Q-value (Step 25). Meanwhile, the case in this phase needs to be protected by k-anonymity.

Phase 3 (Steps 30-47). This phase is used to against and Latest-attack. If OC=Ø (Step 32), the ranges of all attacked Q-values have been expanded. Thus, we do not need to use k-anonymity. Algorithm 1 ends. If there is no enough case for k-anonymity (Step 32), Algorithm 1 also ends. If Algorithm 1 does not end, first, we randomly select a secure case set R_n from Table R_i(Step 34). Meanwhile, the number of R_n is k. Therefore, we can use k-anonymity to protect new case set (Steps 35-40).

We take Tab. 1 as an example to illustrate Algorithm 1. Thus, we run EQZS(R, 3, 3, "Sex, Age").

Comparison 1. We compare R_1 with R_0. First (Steps 5-7), Algorithm 1 finds an old CaseID set OC = {$r_{1,1}$·CaseID = 11, $r_{1,2}$·CaseID = 14}. Second (Steps 8-10), $r_{1,1}$·Sex = ANY > $r_{0,1}$·Sex = Male and $r_{1,1}$·Age = [20, 30] > $r_{0,1}$·Age = [25, 30]. Thus, $r_{0,1}$·Sex = $r_{1,1}$·Sex = ANY and $r_{0,1}$·Age = $r_{1,1}$·Age = [20, 30] (Steps 13-14). By the same way, the ranges of $r_{0,4}$ need to be expanded too. That is, OA=Ø.

Comparison 2. We compare R_2 with R_1. First(Steps 5-7), we find an old CaseID set OC={$r_{2,6}$·CaseID = 17, $r_{2,7}$·CaseID = 3}. Second(Steps 8-12), $r_{2,6}$·Sex = Male < $r_{1,3}$·Sex = ANY and $r_{2,7}$·Sex = Male = $r_{1,4}$·Sex $\wedge$ $r_{2,7}$·Age =

[15, 30] > $r_{1,4}$·Age = [25, 30]. Thus, $r_{1,4}$·Age = $r_{2,7}$·Age = [15, 30] (Steps 13-14), *OA* = {$r_{2,6}$} and *OC* = {$r_{2,7}$·CaseID}. Third (Steps 21-29), as $r_{2,6}$·Sex = Male < $r_{1,3}$·Sex = ANY and $r_{2,6}$·Age = [15, 30] > $r_{1,3}$·Age = [20, 30]. Therefore, $r_{2,6}$·Sex = $r_{1,3}$·Sex = ANY. However, the range of $r_{2,6}$·Age does not need to be expanded. Fourth(Steps 30-47), we randomly select a secure case set R_n={$r_{2,1}$, $r_{2,3}$, $r_{2,8}$}. Then, $r_{2,1}$·Sex = $r_{2,6}$·Sex = ANY, $r_{2,1}$·Age = $r_{2,6}$·Age = [15, 30], $r_{2,3}$·Sex = $r_{2,6}$·Sex = ANY, $r_{2,3}$·Age = $r_{2,6}$·Age = [15, 30] and $r_{2,8}$·Sex = $r_{2,6}$·Sex = ANY.

All re-anonymized information has been highlighted by blue color in Tab. 2.

```
Algorithm 1: EQZS
Input: R, x, k, Q-value
Output: $R_{new}$
1   OC=Ø// A set of old CaseIDs
2   OA=Ø// A set of cases that need to be possibly anonymized
3   i=1
4   While i<x do
5     For all $r_{i,j}$.CaseID∈$CaseID_i$ do
6       If $r_{i,j}$.CaseID∈$CaseID_{i-1}$∧$r_{i,j}$.CaseID==$r_{i-1,p}$.CaseID then
7         OC=OC∪$r_{i,j}$.CaseID
8         For all q∈Q-value do
9           If $r_{i,j}$.q<$r_{i-1,p}$.q then
10            OA=OA∪$r_{i,j}$
11            Remove $r_{i,j}$.CaseID from OC
12            Exit
13          Else
14            $r_{i-1,p}$.q=$r_{i,j}$.q
15          End if
16        End for
17      End if
18    End for
19    i ++
20  End while
21  If OA≠ Ø then
22    For all $r_{i,j}$∈OA do
23      For all q∈Q-value do
24        If $r_{i,j}$.CaseID==$r_{i-1,p}$.CaseID∧$r_{i,j}$.q<$r_{i-1,p}$.q then
25          $r_{i,j}$.q=$r_{i-1,p}$.q
26        End if
27      End for
28    End for
29  End if
30  i=1
31  ON=Ø                //New case sets for k-anonymity
32  While OC≠Ø∧(ON∪OA)⊂$R_i$∧OA≠Ø do
33    If ∃$r_{i,j}$∈OA then
34      Randomly select a case set $R_n$ from Table $R_i$∧$R_n$ has k cases∧$R_n$∩(OC∪OA∪ON)== Ø
35      For all $r_{i,p}$∈$R_n$ do
36        For all q∈Q-value do
37          If $r_{i,j}$.q>$r_{i,p}$.q then
38            $r_{i,p}$.q=$r_{i,j}$.q
39          End if
40        End for
41      End for
42      ON= ON∪$R_n$∪$r_{i,j}$
43      Reomve $r_{i,j}$ from OA
44    End if
45  End while
46  Let the new anonymized table set be $R_{new}$
47  Return$R_{new}$
```

Table 2 The re-anonymized Tab. 1

(a) R_0: QUARTER 1

LineID	CaseID	Sex	Age	Disease
1	11	**ANY**	**[20, 30]**	Fever
2	8	Male	[25, 30]	Flu
3	6	Male	[25, 30]	Diabetes
4	14	**ANY**	**[20, 30]**	HIV
5	15	Female	[25, 30]	Flu
6	16	Female	[25, 30]	Diabetes

(b) R_1: QUARTER 2

LineID	CaseID	Sex	Age	Disease
1	11	ANY	[20, 30]	Fever
2	14	ANY	[20, 30]	HIV
3	17	ANY	[20, 30]	Diabetes
4	3	Male	**[15, 30]**	Flu
5	19	Male	[25, 30]	Flu
6	20	Male	[25, 30]	Fever
7	31	Male	[25, 30]	HIV
8	42	Male	[25, 30]	Flu

Table 2 The re-anonymized Tab. 1 (continuation)

(c) R_2: QUARTER 3

LineID	CaseID	Sex	Age	Disease
1	33	**ANY**	**[15, 30]**	Flu
2	9	Female	[20, 30]	Diabetes
3	35	**ANY**	**[15, 30]**	HIV
4	5	Female	[25, 30]	Flu
5	47	Female	[25, 30]	Fever
6	17	**ANY**	[15, 30]	Diabetes
7	3	Male	[15, 30]	Flu
8	21	**ANY**	[15, 30]	HIV

4.3 Analysis

The attribute number of *Q*-value is unlikely to be large. Thus, we do not consider it. We suppose that the maximum number of cases in a table is n and the number of tables in the system is m. PPMS has one more computational loop than EQZS for the filter operation. Meanwhile, we almost only anonymize the new table to greatly improve the data update efficiency. Therefore, the complexities of these two methods are shown in Tab. 3.

Table 3 The comparison of complexity

	Memory consumption	Anonymity efficiency	Data update
PPMS [13]	O(2×*n*×*m*)	O(n^2×*m*)	O(*n*×*m*)
EQZS	O(*n*×*m*)	O(*n*×m)	O(*n*)

We need to prove that our anonymity by comparing R_i and R_{i-1} is secure enough to against BFL-attacks.

Theorem 1. (Anonymity transitivity). Given a random attribute *q* and three cases $r_{i-1,j}$, $r_{i,p}$ and $r_{i+1,h}$ with the same CaseID from three tables R_{i-1}, R_i and R_{i+1} respectively, after our re-anonymity, if $r_{i-1,j}\cdot q \le r_{i,p}\cdot q$ and $r_{i,p}\cdot q \le r_{i+1,h}\cdot q$, obviously there must be $r_{i-1,j}\cdot q \le r_{i+1,h}\cdot q$. Meanwhile, each *Q*-value is protected by the *k*-anonymity. Therefore, we can ensure the security without comparing R_{i+1} with R_{i-1}.

5 EXPERIMENTS

To verify the practicability and efficiency of our scheme in the anonymous SRS dataset, we compare our EQZS with the most classic and most relevant method PPMS through the following experiments.

5.1 Experimental Setup

Visual C++ 6.0 is used in all experiments. The experiment uses a computer with a 3.4GHZ dual-core CPU and 32GB memory. The *k*-anonymity number is *k* = 6. We use the real case data in our affiliated hospital. Each Q-value has 5 attributes in our case tables. When we compare EQZS with PPMS, the two methods are always under the same dataset.

In our system, the case number n is from 2T (T denotes Thousand) to 10T and table number m is from 100 to 500. The specific parameters are shown in Tab. 4.

Table 4 The specific parameters

n	2T	4T	6T	8T	10T
m	100	200	300	400	500

5.2 Memory Consumption

The experimental results are shown in Fig. 1, where the *X*-axis represents the table number m, the *Y*-axis represents the case number n (whose unit is T) and the *Z*-axis represents the memory consumption (whose unit is MB).

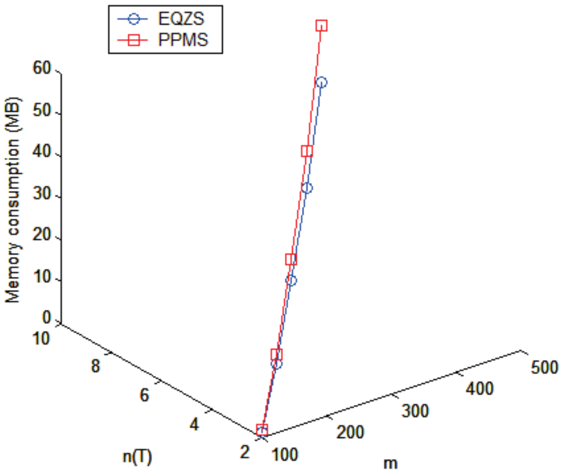


Figure 1 The comparison of memory consumption

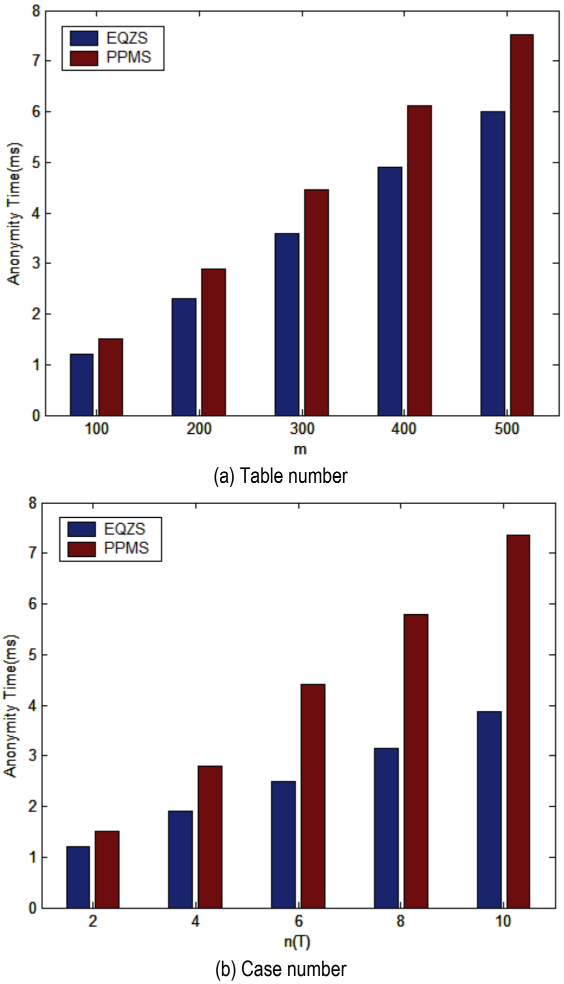
When *m* = 100 and *n* = 2T grow to *m* = 500 and *n* = 10T, the memory consumption of PPMS is almost 1.37-fold that of EQZS. PPMS needs to filter the data to determine which data needs to be anonymized. Thus, it needs to develop a set of storage spaces to store the data. However, we omit the filter operation. Thus, we pay a fewer memory consumption than PPMS.

5.3 Anonymity Efficiency

In this section, we focus on the anonymity efficiency by anonymizing all tables in the system. We use anonymity time to represent the running time. Thus, the experimental results are shown in Fig. 2. The *Y*-axes represents the anonymity time whose unit is millisecond (ms). The *X*-axes of Fig. 2(a) and 2(b) represent table number m and case number n respectively. In Fig. 2(a), *n* = 2T and in Fig. 2(b), *m* = 100.

(1) Fig. 2(a). PPMS filters some data that needs to be anonymized. Our scheme omits this filter operation. Therefore, in the process of m growing from 100 to 500, EQZS consumes less anonymity time than PPMS. Through calculations, we find that the anonymity time of EQZS is only 79.95% of PPMS.

(2) Fig. 2(b). When the case number n grows from 2T to 10T, the growth rate of EQZS's anonymity time is lower than PPMS. First, PPMS needs to compare the data in the case table to achieve the filter operation. Second, PPMS use the filtered data for the next phase of anonymity operations. However, we omit the filter operation. Therefore, more cases bring more differences between PPMS and EQZS.



(a) Table number

(b) Case number

Figure 2 The comparison of anonymity efficiency

5.4 Data Update

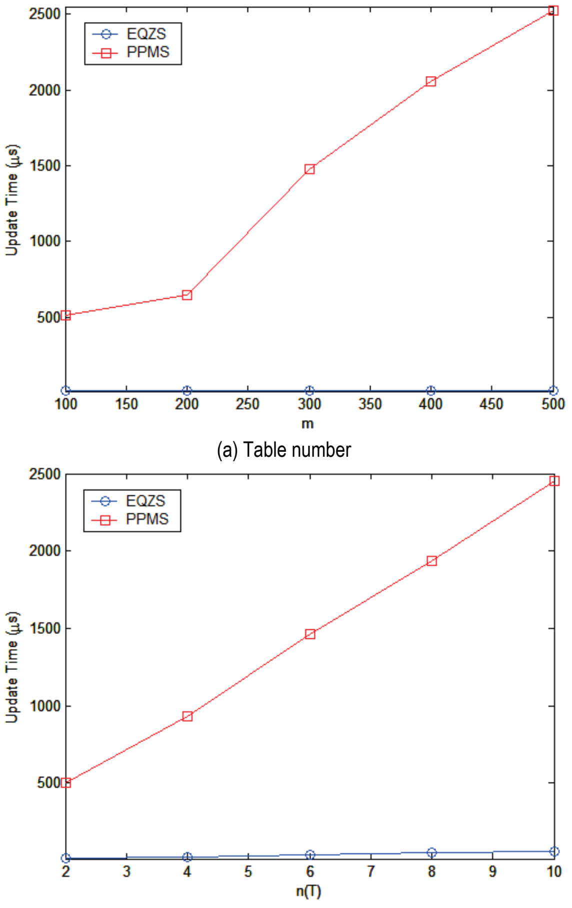
When a new table is published, the update time for the anonymity is shown in Fig. 3. Where the *Y*-axes represents the update time whose unit is microsecond (μs). The *X*-axes of Fig. 2(a) and 2(b) represent table number *m* and case number *n* respectively. In Fig. 3(a), *n* = 2T and in Fig. 3(b), *m* = 100.

(1) Fig. 3(a). When table number m grows from 100 to 500, EQZS's update time barely changes. However, PPMS's update time linearly increases with *m*. When a new table is published, PPMS needs to anonymize almost all tables and EQZS only needs to anonymize the new table. Therefore, EQZS has a lot more performance than PPMS on data update.

(2) Fig. 3(b). In the process of n growing from 2T to 10T, EQZS only needs to judge two tables with 2T to 10T

cases. However, PPMS needs to judge almost all tables. Thus, our scheme has better data update performance than PPMS

Besides, comparing the experimental results in Figs. 2 and 3, we find that the running time of data update is obviously fewer than initial re-anonymity. This is because many Q-values do not need be re-anonymized again for data update after the initial re-anonymity.



(a) Table number

(b) Case number

Figure 3 The comparison of data update

6 CONCLUSIONS

BFL-attacks exist in many of the existing data (for example, SRS, electronic health Records (EHR), etc.). The existing methods for BFL-attack have some shortcomings in memory consumption, anonymity efficiency, data update and anonymity security. To remove these shortcomings, we propose an efficient *Q*-value zero-leakage protection scheme in SRS regularly publishing private data. Our scheme omits the filter operation to reduce the memory consumption and improve the anonymity efficiency. Meanwhile, we almost only anonymize the new published table to improve the efficiency of data update. In addition, we show a fine-grained judgment frame to enhance the security. Though the experiment analysis and security proof, we can draw the following conclusions.

(1) The comprehensive performance of our scheme is significantly better than that of most existing methods.

(2) Since each attribute of our *Q*-values is protected by maximum anonymity, we have a better privacy protection effect than most of existing methods.

(3) Because of its excellent data update performance, our scheme can be used in big data environments dynamically.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No. 61762055); the Jiangxi Provincial Natural Science Foundation of China [No. 20181BAB202014]; and the Key Program of Zhejiang Provincial Natural Science Foundation of China (No. LZ18F020001), the Humanities and Social Sciences Foundation of Colleges and Universities in Jiangxi Province (No. TQ18111) and the Science and Technology Research Project of Jiangxi Education Department (No. GJJ180904).

7 REFERENCES

[1] Gupta, M. & Vujcic, B. (2018). F21. Standard Antidepressant Therapy in Posttraumatic Stress Disorder (PTSD) is associated with a Higher Odds of Externalizing Behaviors and Homicidal Ideation: Results from the US Food and Drug Administration (FDA) Adverse Events Reporting System (FAERS). *Biological Psychiatry, 83*(9), S245. https://doi.org/10.1016/j.biopsych.2018.02.634

[2] Rebecca, J. T., Rosanne, J. B., Stephen, J. K., Sarah, J. P., Simon, J. P., & David, A. M. (2018). Awareness and compliance with pharmacovigilance requirements amongst UK oncology healthcare professionals. *Ecancermedical-science*, 12, 809. https://doi.org/10.3332/ecancer.2018.809

[3] Xiao, C., Li, Y., Baytas, I., Zhou, J., &Wang, F. (2018). An MCEM Framework for Drug Safety Signal Detection and Combination from Heterogeneous Real World Evidence. *Scientific reports, 8*(1), 1806. https://doi.org/10.1038/s41598-018-19979-7

[4] Masuka, J. T., Chipangura, P., Nyambayo, P. P., Stergachis, A., & Khoza, S. (2018). A Comparison of Adverse Drug Reaction Profiles in Patients on Antiretroviral and Antitubercular Treatment in Zimbabwe. *Clin Drug Investig, 38*(1), 9-17. https://doi.org/10.1007/s40261-017-0579-z

[5] Raschi, E., Poluzzi, E., Salvo, F., Pariente, A., De Ponti, F., Marchesini, G., & Moretti, U. (2018). Pharmacovigilance of sodium-glucose co-transporter-2 inhibitors: What a clinician should know on disproportionality analysis of spontaneous reporting systems. *Nutrition, Metabolism and Cardiovascular Diseases*, *28*(6), 533-542. https://doi.org/10.1016/j.numecd.2018.02.014

[6] Ndagije, H. B., Nambasa, V., Manirakiza, L., Kusemererwa, D., Kajungu, D., Olsson, S., & Speybroeck, N. (2018). The Burden of Adverse Drug Reactions Due to Artemisinin-Based Antimalarial Treatment in Selected Ugandan Health Facilities: An Active Follow-Up Study. *Drug safety*, 1-13. https://doi.org/10.1007/s40264-018-0659-x

[7] Nisa, Z. U., Zafar, A., & Sher, F. (2018). Assessment of knowledge, attitude and practice of adverse drug reaction reporting among healthcare professionals in secondary and tertiary hospitals in the capital of Pakistan. *Saudi Pharmaceutical Journal, 26*(4), 453-461. https://doi.org/10.1016/j.jsps.2018.02.014

[8] Wang, Y., Cai, Z., Chi, Z., Tong, X., & Li, L. (2018). A differentially k-anonymity-based location privacy preserving for mobile crowdsourcing systems. *Procedia Computer Science*, 129, 28-34. https://doi.org/10.1016/j.procs.2018.03.040

[9] Lin, W. Y. & Yang, D. C. (2013). On privacy-preserving publishing of spontaneous ADE reporting data. *IEEE*

International Conference on Bioinformatics and Biomedicine (BIBM). https://doi.org/10.1109/BIBM.2013.6732760

[10] Lin, W. Y., Yang, D. C., & Wang, J. T. (2016).Privacy preserving data anonymization of spontaneous ADE reporting system dataset. *BMC medical informatics and decision making, 16*(1), 58. https://doi.org/10.1186/s12911-016-0293-4

[11] Cao, Y., Xiong, L., Yoshikawa, M., Xiao, Y., & Zhang, S. (2018). ConTPL: controlling temporal privacy leakage in differentially private continuous data release. *Proceedings of the VLDB Endowment, 11*(12), 2090-2093. https://doi.org/10.14778/3229863.3236267

[12] Du, M., Wang, Q., He, M. & Weng, J. (2018). Privacy-Preserving Indexing and Query Processing for Secure Dynamic Cloud Storage. *IEEE Transactions on Information Forensics and Security, 13*(9), 2320-2332. https://doi.org/10.1109/TIFS.2018.2818651

[13] Wang, J. T. & Lin, W. Y. (2017). Privacy Preserving Anonymity for Periodical SRS Data Publishing. *IEEE 33rd International Conference on Data Engineering (ICDE)*. https://doi.org/10.1109/ICDE.2017.176

[14] Wang, H. & Li, K. (2018). SRS-LM: differentially private publication for infinite streaming data. *Journal of Ambient Intelligence and Humanized Computing*, 1-14. https://doi.org/10.1007/s12652-018-0922-0

[15] Wang, T., Qin, X., Ding, Y., Liu, L., & Luo, Y. (2018). Privacy-Preserving and Energy-Efficient Continuous Data Aggregation Algorithm in Wireless Sensor Networks. *Wireless Personal Communications, 98*(1), 665-684. https://doi.org/10.1007/s11277-017-4889-5

[16] Salas, J. & Torra, V. (2018). A General Algorithm for k-anonymity on Dynamic Databases. *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, 407-414. https://doi.org/10.1007/978-3-030-00305-0_28

[17] Onashoga, S., Bamiro, B., Akinwale, A., & Oguntuase, J. (2017). KC-Slice: A dynamic privacy-preserving data publishing technique for multisensitive attributes. *Information Security Journal: A Global Perspective, 26*(3), 121-135. https://doi.org/10.1080/19393555.2017.1319522

[18] Xu, J., Palanisamy, B., Tang, Y., & Kumar, S. M. (2017). PADS: Privacy-preserving Auction Design for Allocating Dynamically Priced Cloud Resources. *IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. https://doi.org/10.1109/CIC.2017.00023

Contact information:

Zongmin CUI, Assoc. Prof., PhD
School of Information Science and Technology, Jiujiang University,
No. 551, Qianjin East Road, Jiujiang, Jiangxi 332005, China
E-mail: cuizm01@gmail.com

Lungui ZHANG, BE
School of Information Science and Technology, Jiujiang University,
No. 551, Qianjin East Road, Jiujiang, Jiangxi 332005, China
E-mail: 18270219411@163.com

Bin WU, Lecturer, PhD
School of Information Science and Technology, Jiujiang University,
No. 551, Qianjin East Road, Jiujiang, Jiangxi 332005, China
E-mail: wubincs@gmail.com

Zhiqiang ZHAO, Lecturer, PhD
School of Information Science and Technology, Jiujiang University,
No. 551, Qianjin East Road, Jiujiang, Jiangxi 332005, China
E-mail: zqzhao2000@foxmail.com

Zhuolin MEI, PhD
(Corresponding author)
School of Information Science and Technology, Jiujiang University,
No. 551, Qianjin East Road, Jiujiang, Jiangxi 332005, China
E-mail: meizhuolin@126.com

Zongda WU, Prof., PhD
Oujiang College, Wenzhou University,
Wenzhou 325035, Zhejiang, China
E-mail: zongda1983@163.com