International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 7, Issue 2, March - April 2018 ISSN 2278-6856

Real-time De-identification of Healthcare Data Using Ephemeral Pseudonyms

[1]Ashish Shukla, [2]Mohit Kumar Sahni, [3]Sourav Aggarwal, [4]Bipin Kumar Rai
[1] [2] [3]Student, Computer Science & Engineering Department, ABES IT, Ghaziabad
[4]Research Scholar, Banasthali University & Associate Professor, Information Technology Department, ABES IT, Ghaziabad

Abstract: ***Information explosion is radically changing our perception of the surroundings and healthcare data is at the core of it. The nature of healthcare data being extremely sensitive poses a threat of invasion of privacy of individuals if stored or exported without taking proper security measures. De-identification involves pseudonymization or anonymization of data which are methods to disassociate an individual's identity temporarily or permanently respectively. These methods can be used to provide secrecy to user's healthcare data. A commonly overlooked weakness of Pseudonymization technique is Inference attacks. This paper discusses an approach to de-identify Enterprise Healthcare Records (EHR) using chained hashing for generating short-lived pseudonyms to minimize the effect of inference attacks and also outlines a re-identification mechanism focusing on information self-determination.***

Keywords:De-identification, Electronic Healthcare Records, Pseudonymization, Inference Attack.

1. INTRODUCTION

Electronic Health Records (EHRs) provides us many advantages such as better communication between healthcare services and patients, no-need of carrying previous reports, reduced costs of treatment and also serves as a repository to retrieve data for research purpose. Healthcare data is inherently extremely sensitive by its nature. The leakage of the same can result in social as well as economic losses to the individual. Thus securing EHR is extremely important. Securing data follows two approaches namely Encryption and de-identification. Although Encryption is the conventional and most reliable way of assuring the data security it has significant drawbacks like the overhead of decrypting data for any analysis or real-life usage. An alternative approach is de-identification of data which is essentially disassociation of personal identifiers from data. It should be noted that de-identification is not a technique of securing data itself, instead, it is a technique of protecting an individual's privacy. De-identification follows two approaches Anonymization and Pseudonymization.

1.1. Anonymization - Anonymization is a de-identification technique that dis-associates all identifiers from the data. For example, creating a teaching file for radiological images illustrating a specific condition requires anonymization of the data.[1] Here the important point is that there is no requirement to be able to identify the patient later so all traces of the patient should be removed and the data is made fully anonymous by manually reviewing the files and their fields to determine which fields are required for instructional purposes and which required fields can be used for re-identification of patient. In practice, such fields are rewritten to retain useful meaning while not disclosing any private information.[1]

Anonymization has following three principles-

Let there be a relation $T(a_1,a_2 \ldots\ldots, a_d)$ for which Q_T is the set of Quasi-identifiers for relation T. where for $i = (1, \ldots, m)$ $a_i \in Q_T$.Then,

1.1.1. k-anonymity[2] - Qt_i for $t_i \in T$ should be indistinguishable from at least k-1, $t_j \in T$ *where* $j \in (1,\ldots,d)$ *and* $j \mathrel{!=} i$. The process of enforcing k-anonymity is called k-anonymization in which T is partitioned into groups g_j *such that* $j \in (1\ldots h)$ *and* $| g_j | < k$, *here* $|x|$ *means the size of* x.tuples in $g_j$are made identical to the Q_T in process of k-anonymization.

1.1.2. l-diversity[2] - Only providing k-anonymity may cause inference of an individual's values in the sensitive values (SA), this is called value disclosure. To prevent value disclosure each anonymized group must contain at least *l well-represented* values. Here well-represented value means *distinct* and leads to the principle called *distinct l-diversity*. which requires each anonymized group to contain at least l distinct SA values.

1.1.3. Recursive (c, l) diversity[2] - Given parameters c,l, which are specified by data publishers, a group g_j is (c,l)-diverse when $r_1 < c \times (r_l + r_{l+1} + \ldots + r_n)$, where r_i,$i \in \{1,\ldots,n\}$ is the number of times the i-th frequent SA value appears in g_j, and n is the domain size of g_j. T is (c,l)-diverse when every g_j, j = 1,...,h is (c,l)-diverse.

1.2. Pseudonymization - Pseudonymization is a de-identification technique in which we introduce a pseudonym in place of the attributes that directly or indirectly identify an individual. IHE defines it as a technique that uses *controlled replacements to allow longitudinal linking and authorized re-identification.*[1]Let there be a relation $T(a_1,a_2 \ldots\ldots, a_d)$ for which Q_T is the set of Quasi-identifiers for relation T. where for $i = (1, \ldots, m)$ a_i

$\in Q_T$ then pseudonymization is essentially replacing Q_T *with* P_T *where* $P_T = (P_1, P_2\ \ldots, P_m)$ Keeping another relation $P_T \cdot Q_T$ for re-identification.
The definitions of de-identification techniques itself clarify that being unable to re-associate data with any individual Anonymization is not suitable for all the purposes in EHR. It is the reason why Pseudonymization is often the recommended process for providing privacy to users. Pseudonymization is also advised to be used by EU General Data Protection Regulation (GDPR) which will be enforced on May 25, 2018.
Few significant pseudonymization approaches are following -

1.2.1. Peterson's approach[6] - Robert L Peterson suggested a key-based approach to provide access control and encryption of medical information. The patient holds a Personal Key (PEK). This approach also involves assigning a static pseudonym to the individuals. There exists a Global Key (GK) which uniquely identifies the patient in the pseudonymized records when used jointly with PEK. The records are secured by encryption on database using PEK thus the entire security of information is revolving around the encryption of information. If PEK is stolen then this approach is rendered ineffective against attackers.

1.2.2. Slamanig and Stingl's Approach[7] - This approach suggests storage of User Information and Medical Data on different databases. These two are mapped with the help of some central components. Same as Peterson's approach, Slamanig's approach also suggests storing data in encrypted form and giving the encryption key to the patient. It focuses on access control as well but doesn't ensure the security of data if the data is to be shared with a 3rd party (e.g. for research purpose).
Similar approaches were suggested by Pommerening and Thielscher as well.[8] All of the approaches seem to be greatly affected by the problem of inference attacks as the used pseudonyms are persistent and eventually start to work as a unique identifier as the patient's information grows larger. Thus a need for *variable* or *ephemeral* pseudonyms arises to weaken the inference attacks.

1.3. Pseudonym Generation Techniques - Primarily we use two pseudonym generation techniques namely *Hashing*and *Tokenization*, Hashing is computationally more expensive and leaves no traceback of the information it has been generated from whereas tokenization is a method that creates a pseudonym that retains the data it originated from and requires much less computation. Although tokenization and hashing both have their respective use cases but generally tokenization increases the possibilities of inference attacks.

1.4. Real-time de-identification - Real-time de-identification refers to de-identification of data as it streams. This is a basic requirement if we are dealing with data that needs to be de-identified as it's generated and EHR falls under such category. It's hard to create a secure mechanism for such cases as it involves dealing with relations having varying attributes. To resolve this there must exist a standard API or protocol that has values in a predefined format.

1.5. Inference Attacks and Pseudonymization - Pseudonymized data is prone to inference attacks. The biggest loophole being *persistent pseudonym* usage. Inference attacks relate to data mining techniques. If an adversary can *infer* the identity associated with some pseudonymized data with high confidence then the data is said to be leaked. As pseudonymization is not a technique of encryption and rather relies on hiding the identity of individuals, it is highly liable to this attack. Statistical frequency analysis attacks are a very basic example of inference attacks. Dataset aggregation techniques are also used heavily by attackers in order to derive an inference from existing datasets.
If there is a relation $T(a_1,a_2 \ldots\ldots, a_d)$. for which Q_T is the set of Quasi-identifiers for relation T. and there exists another relation $D(d_1,d_2 \ldots\ldots, d_d)$ which contains identification information about the individuals belonging to relation T.
if for $i = (1, \ldots, m)\ a_i \in Q_T$ *and* $a_i \in D$ then we can associate an identity based on the other attributes in the same tuple belonging to D.
One such example for EHR is evident with D_T as Voter List. If the pseudonymization was done on basis of YOB, ZIP, and Sex then for a particular state the total number of possible pseudonyms can be in the range of 10,000s.[3]
Which is significantly low and the actual identity can be derived using further inferences. This particular inference attack was exploited heavily and caused the creation of HIPAA (Health Insurance Portability and Accountability Act of 1996). Nevertheless, inference attacks are still prevalent as although the process of formation of pseudonyms has significantly changed but the underlying loopholes remain the same and the persistence in pseudonyms poses a wide threat to user's privacy.
Based on these facts it's obvious that intuitive pseudonymization methods are almost certain to fail in order to provide privacy. Successful pseudonymization requires a deep knowledge of the data.[4] It is necessary to design models keeping in mind that other datasets may be used in association with the existing records to derive identities.

2. Proposed Solution
The solution assumes that there exists an authorized body that regulates the identification information and provides a unique identifier for each resident. Let the identifier be represented by U_i , The patient is represented by $t_i \in T$ where T is set of all patient's identification records. The system consists of 3 Nodes namely Accession Node, Key Node and Data Node. Accession Node enrolls the user in Healthcare system only once. It extracts $Qt_i = (q_{1i}, q_{2i} \ldots, q_{ni})$ (*Quasi Specifiers for* t_i) from t_i and transmits it to Key Node. Key Node applies '*Ephemeral Pseudonym Generation algorithm - Initialize*' (EPGA-Init) on Qt_i which produces g_i (*i*th *group*) and gu_i (*unique ID in* g_i) for t_i

and initializes a report_schema for insertion of records in form of record IDs in Data Node corresponding to a H_i which is a hash of gu_i and g_i. Another relation is maintained for retrieval of gu_i through mapping of biometrics of patients at Key Node. Whole communication on the network is protected using ECDH (Elliptic Curve Diffie Hellman). There should exist a mapping of E_i (Ephemeral IDs) corresponding to each H_i, E_i will be used by healthcare services to insert and retrieve data for a patient. E_i will be generated for de-identification purposes in EPGA-D. Each E_i is only one time usable thus it gives a strong protection against caching of pseudonyms and makes it hard to infer the identity of an individual from records. To reassociate the identity of individuals with U_i the user must provide his consent by providing the gu_i.

2.1. Ephemeral Pseudonym Generation Algorithm - EPGA is divided in to three parts i.e. Initialize (EPGA-Init), De-identification (EPGA-D) and Re-identification (EPGA-R).

2.1.1. EPGA-Init - EPGA-Init Algorithm generates a global pseudonym H_i against which we will store the report_schema which will contain the Record IDs of the reports and other de-identified documents. In EPGA-Init *generalize_or_suppress* function returns the generalized form of an identifier else a null string if identifier should be suppressed. H_m is a highly collision resistant Hashing algorithm (e.g. SHA256). Kg_i stands for ith group's key. The *getLast* function takes the argument as group id and returns the de-serialized object associated with that gid else returns Null if group id doesn't exist in Key Node's Database.

- **EPGA-Init(Qt_i):**

```
1. $gQt_i \leftarrow generalize\_or\_suppress(q_i : q_i \in Qt_i)$
2. $Kg_i \leftarrow$ '\0'
3. $Kg_i \,||\, q_i : q_i \in gQt_i$
4. $g_i \leftarrow H_m(Kg_i)$
5. $count \leftarrow getLast(g_i)$
6. $gu_i \leftarrow randomize\_count(count)$
7. $H_i \leftarrow HMAC(g_i \,||\, gu_i, key = Kg_i)$
8. return $H_i$, $gu_i$
```

To define *getLast* function we assume that there must exist a cQueue associated with each group id in Key Node's database which stores the counts of revoked gu_i corresponding to g_i to avoid overflow in group unique ID's counts. *randomize_count* takes *count* as seed and maps the count to another number within a defined prime number's range. It only introduces randomness in generated group unique IDs.

- **getLast(g_i):**

```
1. retrieve $g_i$ row from database.
2. if $g_i$ doesn't exist in database:
       2.a. $g_i$.count = 0
       2.b. return 0
3. cQueue $\leftarrow$ deserialize($g_i$.cQueue)
4. if cQueue is null:
       4.a.return $g_i$.count+1
5. else:
       5.a. count $\leftarrow$ dequeue(cQueue)
       5.b. serialize(cQueue)
       5.c. update $g_i$.cQueue in database
       5.d. return count
```

2.1.2. EPGA-D - EPGA-D Algorithm de-identifies the streaming data and fulfills the purpose of real-time de-identification of streaming data. If the data is being produced by a producer on a stream processing platform e.g. Kafka in a predefined format e.g. FHIR (Fast Healthcare Interoperability Resources) then we can apply EPGA-D on producer-end if the producer is reliable else on consumer-end on Data Node to de-identify data in real-time. The de-identification of a patient report is partially influenced by *safe harbor method*[5] which suggests suppression of 18 identifiers like Names, Locations, Dates directly relating to an individual, Telephone numbers, Fax numbers etc. The key difference being that EPGA-D assigns a short-lived pseudonym as the report's ID called Ephemeral ID (E_i) along with suppression of identifiers suggested in safe harbor method. The Ephemeral ID is generated by user's consent on report producer's end after providing gu_i. Upon receiving the pseudonymized data with E_i on Data Node, the Data Node generates a random identifier RH_i and replaces E_i with RH_i. RH_i is updated in the report_schema corresponding to the patient's H_i who generated the E_i.
In order to generate E_i patient can send the request for the generation of E_i to Key Node through an authenticated medium by providing his gu_i and U_i.

- **createEi(gu_i , U_i):**

```
1. retrieve $Qt_i$ from identification body through $U_i$.
2. $gQt_i \leftarrow generalize\_or\_suppress(q_i : q_i \in Qt_i)$
3. $g_i \leftarrow H_m(concat(q_i) : q_i \in gQt_i)$
4. creates a random identifier $E_i$ and associate it with the $H_i$.
5. return $E_i$
```

We further subdivide the EPGA-D algorithm into two parts i.e. @Producer and @Consumer where Producer is the segment that should be used on the stream's end which produces the de-identified report and Consumer is the stream's end which receives the de-identified report i.e. Data Node.

@Producer

- **EPGA-D(Report):**

```
1. Request patient to generate $E_i$.
2. $E_i \leftarrow createEi(gu_i , U_i)$
3. gReport $\leftarrow$ generalize_or_suppress(Report)
4. gReport.id $\leftarrow E_i$
5. Stream gReport on data pipeline.
```

@Consumer

- **EPGA-D(Report):**

```
1. create random identifier $RH_i$.
2. $E_i \leftarrow$ Report.id
3. Request $H_i$ corresponding to $E_i$ from Key Node.
4. On receiving $H_i$ request Key Node deletes $E_i$ from the map and returns corresponding $H_i$.
5. Report.id $\leftarrow RH_i$
6. Save $RH_i$ in report_schema corresponding to $H_i$
```

2.1.3. EPGA-R - EPGA-R Algorithm re-associates the identity of an individual with a Report with the explicit consent of the patient. The patient generates a short-lived one-time usable Ephemeral Group Unique ID (*Egu_i*) by providing his *gu_i*, *U_i* and Lifetime of *Egu_i*. In case the patient does not provide the lifetime of *Egu_i* a default timeout must be set up to prevent misuse of *Egu_i* through malevolent attempts.

In order to generate *Egu_i* patient can send the request for the generation of *E_i* to Key Node through an authenticated medium by providing his *gu_i*, *U_i* *and* optionally the time to live *(ttl)* for *Egu_i*.

- **createEgui(gu_i , U_i , ttl = default_time):**

```
1. retrieve $Qt_i$ from identification body through $U_i$.
2. $gQt_i \leftarrow$ generalize_or_suppress($q_i$: $q_i \in Qt_i$ )
3. $g_i \leftarrow H_m(concat(q_i) : q_i \in gQt_i)$
4. creates a random identifier $Egu_i$ and associate it with the $H_i$.
5. Set ttl of Egui.
6. return $Egu_i$
```

Let us assume there exists a 'Service' which wants to re-identify the patient.

- **EPGA-R(gu_i , U_i):**

```
1. Service requests patient to generate $E_i$.
2. $Egu_i \leftarrow$ createEgui($gu_i$ , $U_i$, optional_ttl )
3. Service requests patient to provide $U_i$.
4. Service sends $U_i$ and $Egu_i$ to DataNode.
5. Data Node requests Key Node to return $H_i$ corresponding to $Egu_i$ and $U_i$ .
6. KeyNode returns the $H_i$ to Data Node and deletes $Egu_i$.
7. DataNode returns requested data associated with $H_i$ to the Service.
```

3. Conclusion

EPGA can be used to implement real-time de-identification of healthcare data. It provides the patient information self-determination as EPGA-D and EPGA-R both revolve around the group unique ID *gu_i* which is exclusively known to user. *gu_i* works as a proof-of-consent for the algorithm.

EPGA-D provides a fairly complex relation between report ID and *H_i* which makes it hard to find a straight relation between reports and patient pseudonyms making inference attacks less effective. To reduce the effect of inference attacks even more we can split report_schemas on distributed resources which will require inference from multiple sources making it even harder to identify patient through inference attacks.

The stored pseudonyms are never shared with any of the third-party services in the whole mechanism instead a short-lived pseudonym is shared which makes caching of pseudonym corresponding to *U_i* ineffective.

4. Future Work

Based on the algorithm we can create an architecture for scalable EHR using appropriate messaging queues and stream processing platforms. Although the proposed solution provides a robust mechanism for de-identification of data but it lacks the safe storage of data. An adversary's malevolent attempt can be aimed at destroying the integrity of the data which would render the de-identified data useless for the patient. Perhaps a blockchain based immutable storage can address this problem but the proposed solution lacks it.

Appendix

T - Relation containing all patients.
D - Relation containing de-identification information of all patients.
P - Relation containing pseudonyms for all patients.
t_i - i^{th} patient belonging to relation T
U_i- Basic identity information of t_i.
g_i- Group ID of t_i.
gu_i- Unique ID in group for t_i.
Qt_i- List of Quasi Specifiers for t_i.
gQt_i- Generalized or suppressed list of Quasi Specifiers for t_i.
Egu_i- Ephemeral Unique ID in group for t_i.
H_i- Globally Unique ID for t_i to map Report IDs.
RH_i- Unique Global ID for i^{th} report.
H_m- Highly collision resistant Hashing algorithm
|| - Concatenation symbol.
E_i- Ephemeral ID for i^{th} report.
HMAC - Hash based Message Authentication Coding function.
Kg_i- Key for creating H_i through HMAC for i^{th} patient.

REFERENCES

[1] IHE IT Infrastructure Technical Committee, Integrating the healthcare enterprise (IHE IT Infrastructure Book), June 6,2014, pp. 170.
[2] Aris Gkoulalas-Divanis Grigorios Loukides, Overview of patient Data Anonymization, September 13, 2012, pp. 9-11.
[3] Latanya Sweeney, Only You, Your Doctor, and Many Others May Know, Sept. 29, 2015.
[4] Phil Factor, Pseudonymization and the Inference Attack (Redgate Hub), August 01, 2017.
[5] Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule , September 4, 2012.

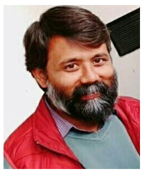
[6] Peterson, R.L., Encryption system for allowing immediate universal access to medical records while maintaining complete patient control over privacy. US Patent Application Publication, No.: US 2003/0074564 A1 , 2003.
[7] Daniel slamanig, Christian stingl , 'Privacy aspect of e-health' the 3rd international conference on availability, reliability and security, IEEE computer society, 2008.
[8] Bipin Kumar Rai, Dr. A.K. Srivastava, Pseudonymization Techniques for Providing Privacy and Security in EHR, IJETTCS, July, 22, 2017.

AUTHORS

Ashish Shukla is an undergraduate Computer Science & Engineering student pursuing B.Tech at ABES IT, Ghaziabad. His primary area of interest is Information Security and Data Sciences. (ash2shukla@gmail.com)

Mohit Kumar Sahni is an undergraduate Computer Science & Engineering student pursuing B.Tech at ABES IT, Ghaziabad. His primary area of interest is Big Data and Data Analytics. (mohitkumarsahni@gmail.com)

Sourav Aggarwal is an undergraduate Computer Science & Engineering student pursuing B.Tech at ABES IT, Ghaziabad. His primary area of interest is Deep Learning and Data Science. (srvaggarwal96@gmail.com)



Bipin Kumar Rai, received the B.Tech(CSE) from UPTU (BIT Muzaffarnagar) Lucknow, UP and M.Tech(CSE) from RGPV Bhopal, (SSSIST, Sehore) MP in 2004 and 2009, respectively. During 2004-2006 & 2008-2014 he taught in different engineering colleges. He is with ABES IT as Associate Professor now. His primary area of interest is Information Security.(bipinkrai@gmail.com)