Pseudonymization Service for X-Road eGovernment Data Exchange Layer

Jan Willemson[1,2]

[1] Cybernetica, Ülikooli 2, Tartu, Estonia
[2] Software Technology and Applications Competence Center
Ülikooli 2, Tartu, Estonia
jan.willemson@gmail.com

Abstract. Pseudonymization is sometimes used as a light-weight alternative to fully cryptographic solutions, when information from different data sources needs to be linked in a privacy-preserving manner. In this paper, we review several previously proposed pseudonymization techniques, point out their cryptographic and design flaws. As a solution, we have developed a simple pseudonymization framework based on X-Road, a unified database access layer serving as the basis for most eGovernment services developed in Estonia. Our solution has been fully implemented and benchmarking results together with the security analysis are presented to conclude the paper.

1 Introduction

As more and more datasets become available via Internet, privacy issues concerning individuals listed in such datasets emerge. The issue is compounded if the need arises to link data from different sources into aggregated databases, which may give valuable statistical information to serve as a basis for policy decision making. For example, to plan AIDS prevention measures among the drug addicts, medical databases of HIV-infected persons and drug addicts must be linked to give an overview of the current situation.

Another example motivating our research came from the Estonian Ministry of Social Affairs being responsible for labor market policy development. In order to plan regional vocational training activities and social assistance, data from different sources (unemployment register, social security register, etc.), needed to be aggregated. While the data concerning unemployment is, legally speaking, not private, unemployed persons often still prefer to avoid unnecessary exposure.

To facilitate database linking and aggregation, we must be able to check, which records in different databases refer to the same individual. At the same time, we generally do not want full identifiability, since it

could lead to over-excessive privacy violation. In the context of policy studies, personal identifiability of individuals is not even necessary, since all we really care about is the aggregated end result showing general trends.

Several approaches can be taken to tackle this problem. It is possible to make use of theoretically well-founded cryptographic techniques like fully homomorphic encryption [11] and secret-shared multi-party computations [6, 15, 4, 5]. Unfortunately, these techniques add considerable organizational and computational overhead and assume non-standard components that commercially available database engines do not support.

Hence in practice, solutions with weaker security guarantees, but lower implementation costs are often used as trade-offs. In this paper, we are going to concentrate on *pseydonymization*, a technique, where information directly referring to the individual (like one's name or social security number) is replaced by a pseudonym. This does not guarantee full privacy, since the information remaining open can still be used to identify individuals with high probability. For example, it was recently demonstrated by Narayanan and Shmatikov that a piece of information as small and innocent as movie preferences can lead to deanonymization [16].

However, when used in conjunction with appropriate organizational measures, pseudonymization may provide a reasonable level of security against certain attacks. For example, this security level can be considered sufficient when pseudonymized databases are linked for the purposes of official statistics and the statistical analysis is performed by recognized professionals in a controlled environment.

Still, we need to remember that the security guarantee provided by pseudonymization is not a very strong one. Hence, resources invested into the pseudonymization framework should be reasonably correlated to the obtained gain. Different equilibria are possible and a solution making full use of the pre-existing infrastructure may very well pay off even if the implied security model is rather weak.

This paper describes a light-weight solution for pseudonymization implemented in Estonia. The Estonian case is somewhat specific, since there is already a state-wide database access layer called X-Road in place [12, 13, 2, 23]. The existence of X-Road allows us to assume significant parts of available infrastructure, including cryptographic key storage and management solutions, standardized XML-based data exchange format and access control mechanisms. This in turn helps to simplify the pseudonymization framework considerably when compared to the other solutions previously proposed.

The functional requirements put onto the framework were rather limited as well. Creating an aggregated database was essentially the only use case, no further linking or other operations on pseudonymized datasets were required. In fact, any usage of an aggregated statistical database for purposes other than obtaining specific statistical information is explicitly prohibited under Estonian legislation. Namely, $\S 43^1(2)$ of Public Information Act states:

> *A structured body of data processed within a database may consist exclusively of unique data contained in other databases.*

This essentially means that when a different kind of survey is needed, a new aggregated database must be created from the source datasets and not by aggregating the existing ones any further (since hereby uniqueness of the data would be violated). Hence, our proposed solution will be limited in its capabilities, but will provide an adequate level of security coupled with high performance in setup and utilization.

The paper is organized as follows. First, in Section 2 we briefly review the security model that is the basis for our pseudonymization framework. Section 3 describes and analyzes several existing solutions pointing out their major design flaws. In Section 4 we propose a simple pseudonymization scheme, argue that for practical purposes it is as secure as those introduced previously, and present implementation details together with benchmarks. Finally, Section 5 draws some conclusions and gives final remarks.

2 Security model

We assume that there exist datasets $D_1, D_2, \ldots, D_m$ containing (possibly privileged) information concerning some individuals. The owners of the datasets will subsequently be called *data donors* and when no confusion occurs, they will also be denoted by D_i. The individuals described by the datasets are identified by some standardized IDs like personal codes or social security numbers that are the same across different databases. There is also a specific party R whose intent is to perform statistical research based on information obtained from $D_1, D_2, \ldots, D_m$. Hence, he needs to link the data using the common IDs, but at the same time he is not allowed to learn the identities of the individuals.

The threat coming from R is assumed to be *passive*, i.e. R will learn the identity if he sees an ID in clear text, but he is not motivated to make an effort to disclose the individuals based on the other data he sees.

This is exactly the case with our motivating example of the labor market policy information system. As noted in the introductory section, privacy of the unemployment data is not a legal requirement, and by violating it generally not much harm can be done. However, there is no need of displaying the identities of the unemployed persons if we can avoid it.

Our model is also passive in the sense that we assume the data donors to be mutually trustful. In practice, the parties responsible for the source datasets can even coincide as in our motivating case of labour market analysis.

3 State of the Art

Pseudonym systems were introduced by Chaum in 1980s in order to allow an individual to communicate with different organizations in an unlinkable manner [8]. Several possible application areas have been studied ever since, with more interest towards databases arising in the early 2000s.

In 2004, Pommerling and Reng gave a systematic overview of different problem settings that can be addressed with pseudonymization [19]. They also presented high-level ideas for the corresponding system architectures, but their view remained far too abstract for practical implementations.

Thomas Neubauer *et al.* have developed a flexible framework for health record pseudonymization [18, 17, 20, 21]. Unfortunately, their main use case is access credential delegation from the data subject to another privileged user (say, a close relative), and the solution is not well-suited for linking different datasets.

Iacono has proposed a cryptographic pseudonymization technique using a trusted third party (TTP) and claimed to rely on the hardness of discrete logarithm problem [14]. However, he has provided no formal proofs and his scheme is in fact fundamentally flawed, as we will show in Section 3.1.

In 2007, Galindo and Verheul proposed another framework suitable for secure pseudonym-based database linking, making use of a TTP [9]. A similar, but less efficient and somewhat more *ad hoc* solution has also been implemented in Estonia as the first attempt at a pseudonymization framework. Due to its shortcomings it actually never worked. In fact, our current research was greatly motivated by the need to create a better solution. In 2010, Galindo and Verheul proposed three improved pseudonymization schemes [10]. We will describe and analyze all of these approaches in Section 3.2.

3.1 Pseudonymization Framework of Iacono

In order to set up Iacono's framework, a system-wide instance of a group with hard discrete logarithm is chosen first. The paper [14] uses a prime multiplicative group $\mathbb{Z}_p^*$ with generator g, and we will use the same setting.

Next, the database owners $A, B, \ldots$ generate private exponents $a, b, \ldots$ and compute the corresponding public keys $\alpha = g^a \bmod p$, $\beta = g^b \bmod p$, etc. Also, the TTP generates a key pair $(t, \tau = g^t \bmod p)$.

In order to pseudonymize Personal Identifiable Information PII^e of a person e, it is first being run through a deterministic function (e.g. a cryptographic hash function) h to obtain identifier ID^e. Then the pseudonym is computed as

$$g^{ID^e} \cdot \alpha \cdot \beta \cdot \ldots \cdot \tau \bmod p\,. \tag{1}$$

The term "public key" is used in a somewhat misleading way in [14], since the values $\alpha, \beta, \ldots$ are assumed to be shared only by the TTP and the respective parties $A, B, \ldots$. The "public key" τ of the TTP must be kept private to the TTP (see [14], Section 5). The pseudonymization protocol has two stages – first the database owner (say, A) computes a local pseudonym as $g^{ID^e} \cdot \alpha$, which is then sent to the TTP who adds all the missing terms to compute (1).

As the first observation, note that the private exponents are needed nowhere except for the key generation. Hence they can be disregarded altogether and the "public keys" $\alpha, \beta, \ldots$ can be generated just as random elements of $\mathbb{Z}_p^*$. Furthermore, since the TTP already knows all the keys $\alpha, \beta, \ldots$, there is no need for the database owners to send elements of the form $g^{ID^e} \cdot \alpha$ and just sending the elements g^{ID^e} is enough. This in turn means that the keys $\alpha, \beta, \ldots$ are unnecessary from the beginning and the whole protocol is equivalent to multiplying g^{ID^e} by a constant by the TTP. This is obviously not secure, e.g. the knowledge of one original-pseudonym pair immediately allows the reversal of all the pseudonyms.

It may be argued that the knowledge of $g^{ID^e} = g^{h(PII^e)}$ does not reveal the identifying information PII^e. However, this is often not the case. Note that both h and exponentiation of g are public deterministic operations, whereas the space of the values PII^e may be rather small and well-known. For example, the Estonian ID number, commonly used in public databases for personal identification, only has roughly 70 million theoretically possible values.

Hence, it is often feasible to precompute the whole table containing the pairs of values $(PII^e, g^{h(PII^e)})$ and use it to open PII^e. Since the pseudonymization must be deterministic in order to support database

linking, no randomness can be added to increase the level of security either.

3.2 Pseudonymization Frameworks of Galindo and Verheul

The basic infrastructure of Galindo and Verheul [9] is similar to that proposed by Iacono. Different database owners (called Suppliers in [9]) send the identifiers to the TTP and the latter creates the pseudonyms to be used when linking the databases at the special parties called Accumulators. However, a symmetric (block)cipher is used instead of just multiplying by a constant, preventing the most severe flaw of the Iacono's framework. Only the identifiers are sent through the TTP, the corresponding data records are transferred directly from the Suppliers to the Accumulators, and record integrity is restored by the sequence numbers.

The system previously implemented in Estonia was essentially the same as the one proposed by Galindo and Verheul, the major differences being the usage of asymmetric cryptography instead of symmetric one and restoring the record integrity using random transport identifiers instead of sequence numbers. By the specification, asymmetric encryption was used only one way and the other half of the key pair was actually deleted without ever using it. Consequently, the possibilities provided by the public key cryptography were never used and the whole system might have operated on symmetric encryption equally well. Even though public key operations were performed on a dedicated hardware security module (HSM), the overall implementation was still unusable for practical purposes. Our interviews with the representatives of the Estonian Ministry of Social Affairs revealed that the main cause of problems was the inability of the HSM to manage several concurrent pseudonymization requests coming from different data sources. This in turn lead to the need to restart the pseudonymization server or the whole HSM. The latter was especially painful, since the service was provided by the main Certification Authority in Estonia and several critical services were relying on the same hardware.

We argue that the principal flaw of both of the above-described solutions is the introduction of a TTP responsible for actual pseudonymization. One may argue that the TTP can be chosen and set up in such a way that it can be trusted to see all the IDs. On the other hand note that the identities of the persons included in the sensitive datasets may be secret themselves. This is for example the case with many medical datasets, where the inclusion in the database already means that a person has or

is suspected to have a certain medical condition. Running this kind of information about all sensitive datasets through one server creates a single point of failure, becoming a very appealing target for the potential attackers.

In [10], Galindo and Verheul propose three more advanced pseudonymization schemes. Their first scheme uses a ubiquitous TTP being responsible for both pseudonymizing and linking the databases. As a result, the TTP's workload may become unacceptably high. There is another shortcoming of the scheme not noted by the authors. Namely, when linking the pseudonymized databases, not only the IDs are sent through the TTP, but also all the data fields in clear text. This way, the TTP essentially obtains copies of all the sensitive datasets and the result is even less acceptable than the one provided by [9].

The second and third schemes presented in [10] rely on the dataset union and equijoin protocols proposed by Agrawal *et al.* [1]. The protocols of [1] make use of commutative encryption, which essentially presumes public key operations (e.g. modular exponentiation). Consequently, the last two protocols of Galindo and Verheul are computationally rather challenging, one using commutative encryption plainly and the second one being built on pairings. As it is the case with Agrawal *et al.* [1], in order to prove the security of these protocols, Galindo and Verheul rely on the Random Oracle Model [3]. Applicability of this model has been disputed in the context of cryptographic proofs [7]. The second protocol of Galindo and Verheul is vulnerable to researcher collusion, but the third one patches this flaw and hence provides a scheme enabling the computation of pseudonymized equijoins explicitly approved by the TTP.

However, as noted in Section 1, equijoin protocols can actually be considered superfluous in the context of statistical surveys. It is typically not the case that the researchers are allowed to hold pseudonymized copies of the source databases and then start linking them at will. In practice, the reason should come first and only afterwards the decision to link the source datasets should be approved. Starting a two-stage process, where researcher-specific pseudonyms are first issued and are then replaced by the pseudonyms for the aggregated database, is a waste of resources. It would be much easier to create the pseudonyms for the aggregated database from the beginning, and this is the starting point of our pseudonymization scheme presented in Section 4. We will also argue in Section 4.3 that it makes no sense to require a pseudonymization scheme to resist collusion, or in fact, any form of active attack.

4 Pseudonymization Framework for the X-Road Infrastructure

4.1 X-Road

X-Road [12, 13, 2, 23] is a middleware layer for secure eGovernment database access developed in Estonia. The core of the system lies in the creation of unified interfaces for data access, so that no changes need to be made to the existing databases and an extra *adapter server* layer is added instead. Since data is typically restricted to authorized users only, a large part of the X-Road infrastructure deals with managing and granting access credentials. Each database has a dedicated *security server* capable of digital signing, verifying other servers' signatures and encrypting the communications. In order to fulfill its purpose and serve as a cryptographic module, a security server must be sufficiently protected against physical and network attacks. Essentially, security server acts as a small-scale HSM with specific tasks.

The volume of centralized X-Road services is minimal, covering only certification, monitoring, and logging. The paradigm of minimal centralization was taken to ensure maximum performance and availability of the whole infrastructure, which could be easily jeopardized by relying on heavyweight central services. An informal requirement to minimize centralization was also put upon the design of the X-Road pseudonymization framework.

A general overview of the X-Road infrastructure is presented in Figure 1.

4.2 Pseudomymization Service for X-Road

Based on the discussion given in Sections 1, 2 and 3.2, the following design principles were taken as a starting point when building the pseudonymization service for X-Road:

- There is no central server for pseudonymization.
- Pseudonymization is implemented via symmetric encryption in security servers.
- Forming each aggregated database is a separate event that has to be granted explicitly. A different pseudonymization key is created for each aggregated database and no further linking with other aggregated databases is supported. (In fact, it is even illegal under Estonian law, see Section 1.)

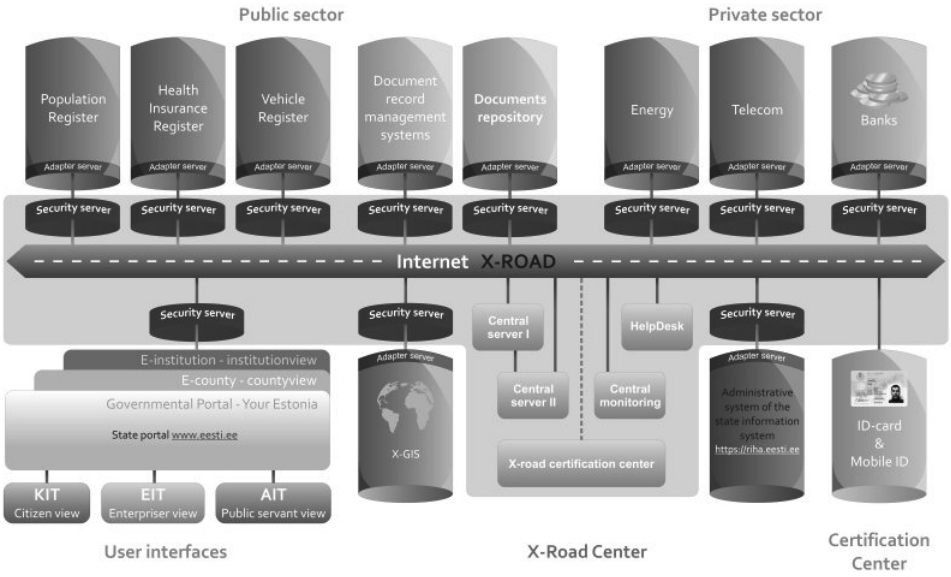

Fig. 1. Overview of the X-Road infrastructure

– Given the weak security guarantee provided by pseudonymization as such, total investment into development time should be as low as possible.

The pseudonymization service implements the following protocols.

Key generation and distribution. Let us assume that the data donors $D_1, D_2, \ldots, D_m$ want to create an aggregated research database R. Being equipped with the X-Road security server, all the donors D_i already possess an asymmetric key pair (Sig_i, Ver_i) for signing and another asymmetric key pair (Enc_i, Dec_i) for encryption. Then

1. D_1 generates a symmetric encryption key K_R.
2. In order to send the key to another D_i, the key is encrypted as $Enc_i(K_R)$ and signed as $Sig_1(Enc_i(K_R))$.
3. The encrypted and signed key is sent from D_1 to D_i.
4. D_i verifies the signature and decrypts the key K_R.

Database aggregation. Assume the database of D_i is presented in the form of records $(ID, Data(ID))$, where ID is the field we want to pseudonymize.

1. When sending data from D_i to R, the IDs are encrypted with the key K_R so that the records become $(Enc_{K_R}(ID), Data(ID))$.
2. After all the pseudonymized datasets are transmitted, R links them based on the values $Enc_{K_R}(ID)$ as identifiers.

4.3 Security Considerations

Even though pseudonymization based on a shared symmetric key provides high performance and removes the threat of direct identification from the researcher's side, there are still scenarios where the system remains vulnerable.

In order for the scheme to protect the privacy of individuals, all copies of the pseudonymization key must be kept secret. One possible threat comes from a corrupt system administrator, who may be willing to reveal the keys. However, anyone who has access to the security server, could see the private data before it gets even encrypted in the first place. Thus, no pseudonymization mechanism can protect against corrupt system administrator anyway.

One may argue (see e.g. [10]) that the data donors can be mutually distrustful. In this case leakage of the pseudonymization key may help one data donor to identify records intercepted from the communications of another. Similarly, a malicious coalition consisting of a data donor and the researcher may reveal the identities of all the persons listed in the aggregated database. However, recall that pseudonymization is by no means a strong security mechanism. A malicious party can easily obtain exposure (at least with some non-negligible probability) by using secondary identification mechanisms based on the open data fields (see [16]). Additionally, following the discussion given in Section 2, we claim that the issue of data donors cooperating maliciously with each other or with the researcher is not as substantial as sometimes presented to motivate the research (as in [10]).

Finally note that the problem of easy decryption following the leakage of a pseudonymization key can not be resolved by replacing symmetric encryption with a one-way operation like hashing or asymmetric encryption (as suggested in an earlier solution, see Section 3.2). As the space of possible candidates for ID is very limited (see Section 3.1), the party who controls the one-way pseudonymization can still actually repersonalize the pseudonyms by full inspection of the space. We conclude that choosing symmetric encryption over asymmetric one does not introduce extra vulnerabilities.

4.4 Implementation Details and Benchmarks

For pseudonymization, AES-256 block cipher in AES-CMAC mode [22] was used. Transmission of the signed and encrypted key can in principle be done over the Internet, but in our current implementation it actually happens via a physical carrier, i.e. the administrator of one security server takes the key bundle to another server on a USB memory stick.

Since all the data query responses are transformed to a standardized XML form by X-Road adapter servers, identifying the fields to pseudonymize is done by standard XPath mechanism using pugixml library[3].

All the above protocols have been implemented by Cybernetica's development team in C++ language and tested on security servers running Ubuntu Linux 10.04 on Intel Core2 8200 processors. Pseudonymization can be performed in several threads running in parallel (up to 8 in default settings). Memory requirement was within the limits of 45-55MB per thread. Overall data throughput of the security server in the setting, where pseudonymization was requested, was 120 MBps. This caused no noticeable effect on the overall throughput of the whole X-Road infrastructure and hence the introduction of pseudonymization is nearly transparent in terms of performance.

Recall that one of the design principles of our pseudonymization service was ease of implementation. This principle was followed as well – the whole implementation cycle starting from initial design and ending with testing and documentation took only 197 man-hours. We consider it to be a very good result, especially when compared to the competing original solution that was developed for over a year before abandoning.

Our pseudonymization service was included into X-Road version 5, the deployment of which in Estonia started on January 1st 2011.

5 Conclusions

In this paper we have presented an extension of the X-Road data exchange platform to support pseudonymization. The extension does not rely on any centralized services, but rather uses the existing key management capabilities of the X-Road security servers. The provided security guarantees work in the passive model, but we argue that it only makes sense to consider pseudonymization in the passive model, since open data fields do not withstand active attacks anyway. We also reviewed several

[3] `http://pugixml.org/`

previously proposed pseudonymization frameworks and pointed out their shortcomings both from a cryptographic and design perspective.

The protocols proposed in this paper have been implemented in X-Road version 5 and thoroughly tested. The testing has shown that the users experienced virtually no performance drop because of the added pseudonymization. Due to full utilization of the pre-existing X-Road infrastructure and simplicity of the proposed protocols, efforts put into the implementation were minimal and hence a good cost-benefit equilibrium of the solution was achieved.

Acknowledgements

This research has been supported by Estonian Centre of Excellence in Computer Science (EXCS), Estonian Science Foundation grant #8124 and Software Technology and Applications Competence Center. The author is grateful to Märt Laur and professor Ahto Kalja for their help in the process of preparing the manuscript.

References

1. Agrawal, R., Evfimievski, A., Srikant, R.: Information sharing across private databases. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data. pp. 86–97. SIGMOD '03, ACM, New York, NY, USA (2003)
2. Ansper, A., Buldas, A., Freudenthal, M., Willemson, J.: Scalable and eficient PKI for inter-organizational communication. In: Proceedings of the 19th Annual Computer Security Applications Conference. pp. 308–318. IEEE (2003)
3. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on Computer and communications security. pp. 62–73. CCS '93, ACM, New York, NY, USA (1993)
4. Ben-David, A., Nisan, N., Pinkas, B.: FairplayMP: a system for secure multi-party computation. In: CCS '08: Proceedings of the 15th ACM conference on Computer and communications security. pp. 257–266. ACM, New York, NY, USA (2008)
5. Bogdanov, D., Laur, S., Willemson, J.: Sharemind: A framework for fast privacy-preserving computations. In: Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings. LNCS, vol. 5283, pp. 192–206. Springer (2008)
6. Bogetoft, P., Christensen, D.L., Damgård, I., Geisler, M., Jakobsen, T.P., Krøigaard, M., Nielsen, J.D., Nielsen, J.B., Nielsen, K., Pagter, J., Schwartzbach, M.I., Toft, T.: Secure multiparty computation goes live. In: Financial Cryptography. pp. 325–343 (2009)
7. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51, 557–594 (July 2004)
8. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM 28(10), 1030–1044 (1985)

9. Galindo, D., Verheul, E.R.: Microdata sharing via pseudonymization. In: Work session on statistical data confidentiality. Manchester 17-19 December 2007. pp. 24–32. Eurostat (2009)
10. Galindo, D., Verheul, E.R.: Pseudonymized Data Sharing. In: Privacy and Anonymity in Information Management Systems: New Techniques for New Practical Problems, Advanced Information and Knowledge Processing, pp. 157–179. Springer (2010)
11. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on Theory of computing. pp. 169–178. STOC '09, ACM, New York, NY, USA (2009)
12. Kalja, A.: System integration process of government information systems. In: Kocaoglu, D., Anderson, T., Piscataway, N. (eds.) Proceedings of PICMET '03 : Portland International Conference on Management of Engineering and Technology (2003)
13. Kalja, A., Reitsakas, A., Saard, N.: eGovernment in Estonia: best practices. In: Anderson, T.R., Daim, T.U., Kocaoglu, D.F., Piscataway, N. (eds.) Technology Management : A Unifying Discipline for Melting the Boundaries. pp. 500–506 (2005)
14. Lo Iacono, L.: Multi-centric universal pseudonymisation for secondary use of the EHR. Studies in health technology and informatics 126, 239–247 (Jan 2007)
15. Malkhi, D., Nisan, N., Pinkas, B., Sella, Y.: Fairplay—a secure two-party computation system. In: SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium. pp. 287–302. USENIX Association, Berkeley, CA, USA (2004)
16. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: IEEE Symposium on Security and Privacy. pp. 111–125 (2008)
17. Neubauer, T., Heurix, J.: A methodology for the pseudonymization of medical data. International journal of medical informatics pp. 1–15 (Nov 2010), in print
18. Neubauer, T., Kolb, M.: Technologies for the Pseudonymization of Medical Data: A Legal Evaluation. In: Fourth International Conference on Systems. pp. 7–12. IEEE (2009)
19. Pommerening, K., Reng, M.: Secondary use of the EHR via pseudonymisation. Studies in health technology and informatics 103, 441–446 (Jan 2004)
20. Riedl, B., Grascher, V., Fenz, S., Neubauer, T.: Pseudonymization for improving the Privacy in E-Health Applications. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008). pp. 255–264. IEEE (Jan 2008)
21. Riedl, B., Grascher, V., Neubauer, T.: A Secure e-Health Architecture based on the Appliance of Pseudonymization. Journal of Software 3(2), 23–32 (Feb 2008)
22. Song, J., Poovendran, R., Lee, J., Iwata, T.: The AES-CMAC Algorithm. IETF RFC 4493 (June 2006), available at `http://www.ietf.org/rfc/rfc4493.txt`
23. Willemson, J., Ansper, A.: A Secure and Scalable Infrastructure for Inter-Organizational Data Exchange and eGovernment Applications. In: 2008 Third International Conference on Availability, Reliability and Security. pp. 572–577. IEEE (2008)