Pseudonymized Data Sharing

David Galindo and Eric R. Verheul

Abstract In this chapter pseudonymisation and pseudonym intersection algorithms are proposed and analyzed. These two procedures combined make pseudonymized data sharing possible. Pseudonymized data sharing is used by organizations, that typically do not share information, to build and provide pseudonymized copies of their private databases to third parties – called researchers. Some basic security properties are satisfied: pseudonymity, meaning that it is infeasible to relate a pseudonym to its identity; and unlinkability, meaning that it is infeasible to decide if pseudonyms belonging to different researchers correspond to the same identity. Computing the equijoin of pseudonymized databases held by researchers A and B is enabled provided that they are given proper cryptographic keys. The outcome of the equijoin protocol between A and B is that party A learns virtually nothing, while party B learns the equijoin of A and B's pseudonymized databases. We are able to prevent that malicious researchers abuse equijoin transitivity in the following sense: colluding researchers A,B,C cannot use equijoin keys for (A,B) and (B,C) to compute the equijoin of (A,C). As a prominent application of these algorithms we discuss the privacy-enhanced secondary usage of electronic health records.

1 Introduction

Let us consider databases containing sensitive and valuable data on individuals. Assume these data records consist of an identifier of the individual (e.g. name, social security number) and the data associated to the individual. This data originates from

David Galindo,
University of Luxembourg, Luxembourg, e-mail: david.galindo@uni.lu

Eric R. Verheul,
Radboud University Nijmegen & PricewaterhouseCoopers Advisory, The Netherlands, e-mail: eric.verheul@[nl.pwc.com,cs.ru.nl]

heterogenous mutually-distrustful sources, which we name *suppliers*, such as statistical offices, hospitals or insurance companies.

Subjects data records held by suppliers are the very primary ingredient for empirical research, but their release exposes the privacy of the individuals concerned. We name *researchers* the parties interested in getting access to this data for subsequent analysis. In health care, prominent scenarios include the secondary use of clinical data for research and confidential patient-safety reporting (e.g. adverse drug effects) to name but a few. The fact that statistical research is interested in collective features rather than individual distinctiveness, makes it possible to reconcile data utility and individual privacy: identifiers can be removed or encoded into a pseudonym, and subjects' data can be de-identified by using statistical disclosure control methods. Ideally, the collective features of the resulting pseudonymized de-identified data are preserved.

In this chapter we study pseudonymity in the above context. We do so from a cryptographic point of view, namely by focusing on cryptographic techniques to transform personal identifiers into pseudonyms with several properties. Thus, our techniques are necessary but not sufficient to provide pseudonymity from a system-wide perspective. The reason is simple: even though the data is pseudonymized, there is the risk that the characteristics of the data singles out a person, e.g. by a combination of profession, age and place of residence. The risk of *indirect identification*, cf. [13, 7], becomes even larger when linking several pseudonymized databases, which is our target. The issue of indirect identification, although far from trivial, it is outside the scope of this chapter. The topic is covered by an abundant literature (the interested reader is referred to [13] for an introduction to this topic, to [14] for a grasp on the state of the art, to [15] for privacy risk assessment recommendations and to [19] for an exemplification of the importance of de-identifying the individuals' private data). We briefly comment on some lines of defence against these problem. Keeping track and scrutinizing the queries by the parties, as well as query restriction techniques from the statistical database literature can help. For instance, these techniques include restricting the size of query results and keeping audit trails of all answered queries to detect possible compromises.

Security properties

To be more precise, let us consider *databases* consisting of entries of the form $(id, D(id))$, where *id* is a unique identifier field (called *identity*) and $D(id)$ is a private data field. A *pseudonymized database* is obtained by replacing the identity *id* in the database entries by a blind identifier called *pseudonym* and by modifying $D(id)$ into a pseudonymized data field $PD(id)$. Two basic security requirements apply to the pseudonymisation of identities in the context of pseudonymisation of databases. The first one, called *pseudonymity*, states that it should not be possible for any party to relate a given pseudonym with a given identity. That is, it should be infeasible to correctly answer whether a given pseudonym belongs to a given identity. The second basic security requirement is called *unlinkability*. This states that, unless explicitly

warranted, it should not be possible for two researchers to relate their pseudonyms. Or alteratively put, two researchers should not be able to correctly answer whether two of their pseudonyms belong to the same individual. This implies in particular that pseudonyms on the same identity must differ from one researcher to another. A pseudonym on identity id for researcher R_d is syntactically represented by $P(id,R_d)$. Clearly unlinkability is of paramount importance as a defence mechanism against indirect identification: it prevents researchers from correlating its databases without previous consent.

A third security requirement deals with the possibility of computing equijoins of pseudonymized databases. Let $\mathscr{I}_{R_s},\mathscr{I}_{R_d}$ be the sets of unknown identities corresponding to the pseudonyms held by researchers R_s,R_d respectively. Let

$$\left\{ \left(P(id,R_s),PD(id,R_s) \right) \right\}_{id\in\mathscr{I}_{R_s}} \text{ and } \left\{ \left(P(id,R_d),PD(id,R_d) \right) \right\}_{id\in\mathscr{I}_{R_d}}$$

be the pseudonymized databases held at a certain point in time by researchers R_s,R_d respectively. Then we say that the equijoin between R_s and R_d pseudonymized databases, with R_s playing the role of *source researcher* and R_d playing the role of *destination researcher*, equals

$$\left\{ \left(P(id,R_d),PD(id,R_d) \right) \right\} \cup \left\{ \left(P(\overline{id},R_d),PD(\overline{id},R_d)||PD(\overline{id},R_s) \right) \right\},$$

where id's are such that $id \in \mathscr{I}_{R_d}$ and $id \notin \mathscr{I}_{R_s}$, while $\overline{id}$'s are such that $\overline{id} \in \mathscr{I}_{R_d} \cap \mathscr{I}_{R_s}$. This operation is possible only if explicitly warranted and it is under the control of secret cryptographic keys. Specifically, our equijoin protocols involve two researchers R_s and R_d, where researcher R_s learns virtually nothing while R_d learns the equijoin of their pseudonymized databases. The security property we consider, called *equijoin non-transitivity*, states that researchers cannot abuse equijoin transitivity in the following sense: colluding researchers R_s,R_d,R_o cannot use equijoin keys for (R_s,R_d) and (R_d,R_o) to compute the equijoin of (R_s,R_o).

For the sake of enabling flexible equijoins, our pseudonymizing systems make use of a Trusted Third Party (TTP). This trusted party can either function as a mighty partner involve in all the security sensitive transactions in the system (i.e. pseudonymization, equijon), or alternatively as a simple key distribution center, feeding interested parties with the cryptographic keys required for the operations. Apart from that, the existence of a TTP also reflects the fact that access to pseudonymized databases as well as the allowance of operations between different databases requires previous approval by a Regulatory Privacy Body (RPB). This privacy body has two main roles. On the one hand, it ensures that the exchange of information complies with data protection legislation. On the other hand, it minimizes the risk of indirect identification, for instance by implementing defence mechanisms against it. In this work we are primarily interested in the cryptographic aspects of the pseudonymisation problem, and for this reason the functioning of the RPB is not described.

Overall the assumption on the existence of such a TTP is quite natural, even necessary, as the need to defend the system against indirect identification shows. In

fact such at TTP is included in most of the pseudonymized data sharing platforms, either implemented (see [3]) or simply proposed, we are aware of, and it is explicitly considered in the only existing standard on pseudonymized databases [15].

Relevance

Pseudonymized data sharing is in use and has been discussed in multiple venues (see for instance [1, 2, 18, 21, 6]). More importantly, the ISO standard ISO/TS 25237:2008, *Health informatics – pseudonymisation*, which has been recently released, contains principles and requirements for privacy protection in systems using pseudonymisation services for the protection of personal health information. In this chapter we provide a cryptographic mechanism for building unlikable pseudonyms sets that can be made linkable if a Trusted Third Party decides so. In this sense, our work can be seen as a cryptographic implementation of a pseudonymisation system satisfying the ISO/TS 25237:2008 requirements, yet with an enriching equijoin functionality not envisioned by the aforementioned standard.

Related work

We are not aware of any previous proposal of a cryptographic technique for building pseudonymized databases containing unlinkable pseudonyms, yet allowing secure operations on different sets of pseudonyms. In any case, some of our techniques can be seen as an extension of the work by Agrawal, Evfimievski and Srikant [4], in which protocols for secure equijoin among non-pseudonymized databases are proposed. The main tool used by Agrawal *et al.* is *commutative encryption* (see Section 3.4 for details) and a variant of Shamir's 3-pass protocol [20, 22]. We stress that the problems addressed in [4] and our work, even if related, are orthogonal. Agrawal *et al.* intersect sets containing the same identifiers, while we intersect sects containing different identifiers (which are indeed unlinkable unless some cryptographic keys are known). We are able to extend Agrawal *et al.* techniques to build a basic pseudonymisation scheme. The resulting system has however one drawback, namely, colluding researchers R_s and R_d, who are allowed to compute an equijoin of their pseudonymized databases, can manage to translate pseudonyms $P(id, R_s)$ to $P(id, R_d)$ and viceversa, for individuals outside the intersection of the databases, and therefore can abuse equijoin transitivity (cf. Theorem 5). In Section 6 we present a natural extension of our basic scheme using pairings, in which the above problem is avoided. As well as in [4], the security of our two last protocols is relative to the Random Oracle Model [9].

2 Description of a pseudonymized data sharing system

In this section we shall describe the syntax and security properties of a pseudonymized data sharing system, which comprises (at least) a pseudonymisation algorithm and an equijoin algorithm.

2.1 Syntax

Let us remind we are considering *databases* consisting of entries of the form $(id, D(id))$, where id is the identity field and $D(id)$ is the private data field. A *pseudonymized database* for a researcher R is obtained by replacing the identity id in the database entries by a blind identifier $P(id,R)$, called *pseudonym*. Each researcher has one, unique pseudonyms set. That is, for the same identifier id and different researchers R and R', $P(id,R) \neq P(id,R')$ with overwhelming probability. However, those different pseudonyms sets can be synchronized under the control of secret cryptographic keys held by a trusted service provider. Thus our pseudonymized data sharing systems make use of a TTP that sets the system up, via a 'System Setup' algorithm. As part of this, the TTP generates on request a secret cryptographic key for each researcher through a 'Researcher Key Generation' algorithm, whose output is only known to the TTP. Additional keys are output by 'Supply Keys Generation' and 'Equijoin Keys Generation' algorithms. Later, these keys are distributed to the relevant suppliers and researchers, in the case where researchers and suppliers perform themselves the pseudonymization and equijoin operations; alternatively, these keys are kept secret in the case where the TTP is in charge of running those algorithms. These additional keys enable executing the two fundamental protocols in the scheme, namely 'Researcher Supply' and 'Researcher Equijoin'.

Researcher Supply This protocol is run between a supplier S, a researcher R and eventually the TTP. At the end of the protocol, the researcher R is supplied with a pseudonymized database that originates from the supplier's private database. When the TTP is involved, we denote this protocol as $S \xrightarrow{\text{TTP}}_P R$; otherwise it is denoted $S \rightarrow_P R$. The result of the researcher supply protocol is that R possesses a pseudonymized database consisting of entries of the form $\big(P(id,R), PD(id,R)\big)$ where $PD(id)$ represents the de-identified data that the supplier is willing (or allowed) to share with the researcher on individual id. In particular, R can detect if a certain pseudonymized identity $P(id,R)$ was already present in its database, and proceed to update the associated pseudonymized data.

Researcher Equijoin This protocol is run by a source researcher R_s and a destination researcher R_d and eventually the TTP. After the protocol is completed, R_d has an equijoin of R_s and R_d pseudonymized databases, while R_s learns at most the number of entries on R_d's database. This protocol does not provide R_d with any information on individuals that do not appear in both databases. When the

TTP is involved, we denote this protocol as $R_s \xrightarrow{\text{TTP}}_{\bowtie} R_d$; otherwise it is denoted $R_s \rightarrow_{\bowtie} R_d$.

A pseudonym scheme $\mathscr{P}$ thus consists on six algorithms 'System Setup', 'Researcher Key Generation', 'Supply Keys Generation', 'Equijoin Keys Generation', 'Researcher Supply' and 'Researcher Equijoin'.

2.2 Security requirements

We assume that suppliers are honest. That is, they will not deviate from protocols, they will not collude with any other party nor try to deduce secret information from the data flow they observe. In contrast, we assume that researchers are semi-honest, namely, they will not deviate from the protocols but might try to deduce secret information they are not supposed to know. Moreover, researchers are willing to share cryptographic keys and pseudonyms sets with other researchers to deduce more information than what they are allowed to. In order to simplify security definitions and proofs, we do not consider researchers to be malicious in the traditional sense in multiparty computation [17]. That is, researchers will not abort nor use fake information as input to their protocols.

In the following we define the security requirements pseudonymity, unlinkability and equijoin non-transitivity we mentioned in the introduction. We additionally define a secure equijoin property. We formalize them in what follows. Let us stress once again that our definitions imply that no vital information is leaking from our cryptographic protocols. However we can not guarantee anything regarding the safety of the data de-identification protocols nor of the multiple linkage of pseudonymized databases.

Notation

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s_1, \ldots, s_n \xleftarrow{\$} S$ denotes the operation of picking n elements s_i of S independently and uniformly at random. Let us denote by $\mathscr{I}_{S_l}$ the set of identities held by supplier S_l. We denote by $\mathscr{P}_{R_d}$ the set of pseudonyms $P(id, R_d)$ held by researcher R_d; $\mathscr{I}_{R_d}$ the set of the corresponding unknown identities and $n_d = |\mathscr{P}_{R_d}|$ the cardinal of $\mathscr{P}_{R_d}$. We consider identities $id \in \{0,1\}^*$ be finite binary strings. Let Pset be the set of all possible pseudonyms of a pseudonyms scheme $\mathscr{P}$.

Definition 1 (Pseudonymity). Let $\mathscr{A}$ be a probabilistic polynomial-time adversary (PPT) [17]. Consider the following situation:

1. S_l and $\mathscr{A}$ run the Researcher Supply protocol $S_l \rightarrow_P R_d$ (alternatively $S_l \xrightarrow{\text{TTP}}_P R_d$)

2. $\mathscr{A}$ has been able to break the pseudonymity for pseudonyms in the set $\mathscr{P}\mathscr{I} = \{P(id_1,R_d)\ldots,P(id_t,R_d)\}$ corresponding to identities in the set $\mathscr{I} = \{id_1,\ldots,id_t\}$

We say that a pseudonym scheme $\mathscr{P}$ provides *pseudonymity* if the distributions

$$\begin{pmatrix} id_1 & \ldots & id_{n_d} \\ P(id_1,R_d) & \ldots & P(id_{n_d},R_d) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_1 & \ldots & id_t & id_{t+1} & \ldots & id_{n_d} \\ P(id_1,R_d) & \ldots & P(id_t,R_d) & Z_{t+1} & \ldots & Z_{n_d} \end{pmatrix}$$

where $Z_j \xleftarrow{\$} \mathsf{Pset}$ for $j = t+1,\ldots,n_d$ are computationally indistinguishable in $\mathscr{A}$'s view.

Definition 1 asks that PPT adversaries $\mathscr{A}$ can not distinguish between the distribution with real pseudonyms from a distribution with random values. Similarly to the definition of semantic security for encryption schemes (cf. [17] and Definition 5 in this chapter), the inability to distinguish captures the fact that pseudonyms do not reveal any information on their corresponding identities to PPT adversaries.

Definition 2 (Unlinkability). Let $\mathscr{A}$ be a PPT adversary. Consider the following situation:

1. S_l and $\mathscr{A}$ run the Researcher Supply protocol $S_l \rightarrow_P R_d$ (alternatively $S_l \xrightarrow{\text{TTP}}_P R_d$)
2. $\mathscr{A}$ gets hold of $\mathscr{P}_{R_s}$, the pseudonyms' database of R_s
3. $\mathscr{A}$ has been able to link polynomially many pseudonym pairs

$$\langle \big(P(id_1,R_s),P(id_1,R_d)\big),\ldots,\big(P(id_t,R_s),P(id_t,R_d)\big) \rangle$$

corresponding to identities in a certain set $\mathscr{I} = \{id_1,\ldots,id_t\}$

Let $\mathscr{I}_{R_s} \cap \mathscr{I}_{R_d} = \{id_1,\ldots,id_m\}$. We say that a pseudonym scheme $\mathscr{P}$ provides *unlinkability* if the distributions

$$\begin{pmatrix} id_{t+1} & \ldots & id_m \\ P(id_{t+1},R_d) & \ldots & P(id_m,R_d) \\ P(id_{t+1},R_s) & \ldots & P(id_m,R_s) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_{t+1} & \ldots & id_m \\ P(id_{t+1},R_d) & \ldots & P(id_m,R_d) \\ Z_{t+1} & \ldots & Z_m \end{pmatrix}$$

where $Z_j \xleftarrow{\$} \mathsf{Pset}$ for $j = t+1,\ldots,m$ are computationally indistinguishable in $\mathscr{A}$'s view.

Definition 2 asks that PPT adversaries $\mathscr{A}$ can not significantly better link $P(id,R_d)$ to $P(id,R_s)$ than they can link $P(id,R_d)$ to a random pseudonym. This inability ensures that pseudonyms are unlinkable by PPT adversaries.

Definition 3 (Secure Equijoin). Let R_s,R_d be semi-honest researchers running the researcher equijoin protocol $R_s \rightarrow_{\bowtie} R_d$ (alternatively $R_s \xrightarrow{\text{TTP}}_{\bowtie} R_d$). We say that a pseudonym scheme $\mathscr{P}$ provides *secure equijoin* if R_s learns nothing or alternatively the size of $\mathscr{P}_{R_d}$; R_d learns $(id,PD(id,R_s)$ for $id \in \mathscr{I}_{R_s} \cap \mathscr{I}_{R_d}$, and R_d is allowed to additionally learn the size of $\mathscr{P}_{R_s}$.

Definition 3 asks that PPT adversaries $\mathscr{A}$ only learn the minimal information that a secure equijoin protocol should disclose. We allow that the equijoin protocol might disclose the size of R_d's database to R_s, but this should be the only information R_s should apprehend. Analogously, we allow R_d to learn the size of R_s's database, and obviously the de-identified data on the individuals belonging to both databases; but no more than that.

Definition 4 (Equijoin Non-Transitivity). Suppose that:

1. R_s and R_d are allowed to compute the equijoin of their databases
2. R_d and R_o are allowed to compute the equijoin of their databases
3. R_s, R_d, R_o share their pseudonymized databases and cryptographic material

Let $\mathscr{I}_{R_s} \cap \mathscr{I}_{R_o} = \{id_1, \ldots, id_m\}$ and $\mathscr{I}_{R_s} \cap \mathscr{I}_{R_d} \cap \mathscr{I}_{R_o} = \{id_1, \ldots, id_t\}$ with $t \leq m$. We say that a pseudonym scheme $\mathscr{P}$ provides *equijoin non-transitivity* if the distributions

$$\begin{pmatrix} id_1 & \ldots & id_t & id_{t+1} & \ldots & id_m \\ P(id_1,R_s) & \ldots & P(id_t,R_s) & P(id_{t+1},R_s) & \ldots & P(id_m,R_s) \\ P(id_1,R_o) & \ldots & P(id_t,R_o) & P(id_{t+1},R_o) & \ldots & P(id_m,R_o) \end{pmatrix} \quad \text{and}$$

$$\begin{pmatrix} id_1 & \ldots & id_t & id_{t+1} & \ldots & id_m \\ P(id_1,R_s) & \ldots & P(id_t,R_s) & P(id_{t+1},R_s) & \ldots & P(id_m,R_s) \\ P(id_1,R_o) & \ldots & P(id_t,R_o) & Z_{t+1} & \ldots & Z_m \end{pmatrix}$$

where $Z_j \xleftarrow{\$} \mathsf{Pset}$ for $j = t+1, \ldots, m$ are computationally indistinguishable in R_s, R_d, R_o's view.

Definition 4 captures the fact that R_s, R_d, R_o cannot meaningfully relate pairs $\big(P(id,R_s), P(id,R_o)\big)$ for any $id \in (\mathscr{I}_{R_s} \cap \mathscr{I}_{R_o}) - (\mathscr{I}_{R_s} \cap \mathscr{I}_{R_d} \cap \mathscr{I}_{R_o})$, and thus colluding researchers can not abuse the transitivity property of equijoin. Notice that the intrinsic transitivity property of equijoin always allows R_s, R_d, R_o to compute $\mathscr{P}_{R_s} \cap \mathscr{P}_{R_d} \cap \mathscr{P}_{R_o}$.

3 Basic tools

In this section we introduce some basic cryptographic tools that we will need in our algorithms. We start by defining semantically secure symmetric encryption.

3.1 Symmetric encryption with semantic security

We adapt the classical definition of symmetric encryption [17] to what we actually need in our protocols.

Definition 5 (Semantically Secure Encryption). Let $\mathscr{E} = \big(\mathrm{Enc}_K(\cdot), \mathrm{Dec}_K(\cdot)\big)$ be a symmetric encryption scheme with secret keys belonging to a certain set $\mathscr{K}$. More precisely, let $\mathrm{Enc}_K : \{0,1\}^M \to \{0,1\}^{M'}$ with $M \leq M'$ being integers, and let $\mathrm{Dec}_K : \{0,1\}^{M'} \to \{0,1\}^M$ be such that $\mathrm{Dec}_K\big(\mathrm{Enc}_K(m)\big) = m$ for any $m \in \{0,1\}^M$. Let $\mathscr{O}_K(\cdot)$ be an encryption oracle, which when queried on $m \in \{0,1\}^M$ outputs $\mathrm{Enc}_K(m)$. We say that $\mathscr{E}$ is a *symmetric encryption scheme with semantic security* if no PPT algorithm $\mathscr{A}$ can meaningfully distinguish between the distributions $\big(m, \mathrm{Enc}_K(m)\big)$ and $\big(m, Z\big)$, where $K \xleftarrow{\$} \mathscr{K}$, $m \in \{0,1\}^M$ is a message of $\mathscr{A}$'s choosing and $Z \xleftarrow{\$} \{0,1\}^{M'}$. $\mathscr{A}$ is allowed to arbitrarily query the encryption oracle $\mathscr{O}_K(\cdot)$, except for the chosen message m.

Definition 5 requires that, for any message of the adversary's choice, it is infeasible to distinguish the encryption of this message from a random ciphertext. The importance of this definition stems from the fact that the security level it provides is the computational analogue of Shannon's perfect secrecy (see [17]) : a ciphertext $\mathrm{Enc}_K(m)$ reveals "no information" on the underlying plaintext m to an attacker which does not know the secret encryption key K.

Next we informally describe the computational assumptions we use in the sections that follow.

3.2 Decisional Diffie-Hellman assumption

Let $\mathscr{G}$ be a (cyclic) group of order q prime. The Decisional Diffie-Hellman (DDH) problem consists on distinguishing the probability distributions (u, v, u^a, v^a) and (u, v, u^a, v^r) in polynomial time, where $u, v \xleftarrow{\$} \mathscr{G}$ and $a, r \xleftarrow{\$} \mathbb{Z}_q$.

Definition 6 (Decisional Diffie-Hellman assumption). Let $\mathscr{G}$ be a group of order q prime. We say that $\mathscr{G}$ satisfies the *Decisional Diffie-Hellman assumption* if no PPT algorithm $\mathscr{A}$ can meaningfully distinguish the probability distributions (u, v, u^a, v^a) and (u, v, u^a, v^r), where $u, v \xleftarrow{\$} \mathscr{G}$ and $a, r \xleftarrow{\$} \mathbb{Z}_q$.

3.3 Pairings

Let $\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle h \rangle$ and $\mathbb{G}_3 = \langle G \rangle$ be efficiently samplable cyclic groups of order q prime. A map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ to a group $\mathbb{G}_3$ is called a *pairing* (or bilinear map), if it satisfies the following two properties:

Bilinearity: $e(g^a, h^b) = e(g,h)^{ab}$ for all integers a, b
Non-Degenerate: $e(g,h)$ has order q in $\mathbb{G}_3$.

Moreover, we assume there exists no efficiently computable homomorphism $\psi : \mathbb{G}_1 \to \mathbb{G}_2$, while an efficient homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ does exist. Such a pairing is called a Type 2 pairing [16]. We set $g = \phi(h)$.

Since $\mathbb{G}_1$ is a prime order group, we can define the DDH problem in $\mathbb{G}_1$. Thorough this chapter we assume that the DDH assumption holds in $\mathbb{G}_1$. A prominent type of groups of which it is widely believed that they satisfy the explained assumptions is presented in [11]. These groups are also used in [10, 8, 5, 12]. It is easy to see that the DDH assumption in $\mathbb{G}_1$ implies the DDH assumption in $\mathbb{G}_3$.

We need an extra final assumption. This assumption states that a variant of the Decisional Diffie-Hellman problem in Type 2 pairing groups is infeasible to solve for PPT adversaries. The problem consists on distinguishing the distributions $(g, g^a, g^b, h^b, g^{ab})$ and (g, g^a, g^b, h^b, g^r), where $g = \phi(h)$ generate $\mathbb{G}_1, \mathbb{G}_2$ respectively, and $a, b, r \xleftarrow{\$} \mathbb{Z}_q$. Notice that the problem statement does not give h out, since otherwise the problem would be trivially solvable: to distinguish whether $v = g^{ab}$ or $v = g^r$ it suffices to check whether $e(v,h) = e(g^a, h^b)$. An adversary can not compute h from g since, by assumption, there does not exist any computable isomorphism $\psi : \mathbb{G}_1 \to \mathbb{G}_2$.

Definition 7 (Asymmetric DDH assumption). Let $\langle\, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, e, \phi, g, h, q \,\rangle$ be a Type 2 pairing group. We say that such a pairing group satisfies the *Asymmetric Decisional Diffie-Hellman assumption* if no PPT algorithm $\mathscr{A}$ can distinguish the probability distributions $(g, g^a, g^b, h^b, g^{ab})$ and (g, g^a, g^b, h^b, g^r) where $g = \phi(h), h$ generate $\mathbb{G}_1, \mathbb{G}_2$ respectively, and $a, b, r \xleftarrow{\$} \mathbb{Z}_q$.

The Asymmetric DDH assumption trivially implies the DDH assumption in $\mathbb{G}_1$.

3.4 Commutative Encryption

The next primitive plays a fundamental role in two of our protocols.

Definition 8 (Commutative Encryption). A *commutative encryption* function $\mathscr{F} = \{F_k\}_{k \in \text{Keys}\,\mathscr{F}}$ is a family of computable functions $f : \text{Keys}\,\mathscr{F} \times \text{Dom}\,\mathscr{F} \to \text{Dom}\,\mathscr{F}$, defined on finite computable domains, that satisfies the properties listed below. We denote $f_a(x) := f(a,x)$.

1. Commutativity: for all $a, a' \in \text{Keys}\,\mathscr{F}$ we have $f_a \circ f_{a'} = f_{a'} \circ f_a$
2. Each $f_a : \text{Dom}\,\mathscr{F} \to \text{Dom}\,\mathscr{F}$ is a bijection
3. The inverse f_a^{-1} is computable in polynomial time given a.
4. The distribution of $\big(u, f_a(u), v, f_a(v)\big)$ is indistinguishable from the distribution of $\big(u, f_a(u), v, z\big)$, where $u, v, z \xleftarrow{\$} \text{Dom}\,\mathscr{F}$ and $a \xleftarrow{\$} \text{Keys}\,\mathscr{F}$.

Example 1. Let $\mathscr{G}$ be a group of prime order q. Let $\text{Dom}\,\mathscr{F} := \mathscr{G}$ and let $\text{Keys}\,\mathscr{F} := \mathbb{Z}_q$. Then if $\mathscr{G}$ satisfies the DDH assumption, the power function $f_a(x) = x^a \in \mathscr{G}$ is

a commutative encryption function. Properties 1,2, and 3 are trivially satisfied since $\mathscr{F}$ is the exponentiation function. Property 4 is implied by the DDH assumption. In effect, the distributions $\big(u, f_a(u), v, f_a(v)\big)$ and $\big(u, f_a(u), v, z\big)$ are precisely the distributions in the DDH assumption, since z and v^r follow the uniform distribution for $z \xleftarrow{\$} \mathscr{G}$ and $r \xleftarrow{\$} \mathbb{Z}_q$. Let us note a further property satisfied by the exponentiation function: $f_a \circ f_b = f_{ab}$ and $f_a^{-1} = f_{a^{-1}}$. We refer to this as the Property 5 of the exponentiation function. Actually, this property is exploited in our protocols, in contrast to [4], where Properties 1-4 suffice for their protocols.

3.5 Intersection protocol

Before giving out our protocols, it is helpful to recall the basic intersection protocol by Agrawal, Evfimievski and Srikant [4, Section 4]. In this protocol there are two parties, a sender S and a receiver R, who hold private databases of the form $(id, D(id))$. Let $\mathscr{I}_S, \mathscr{I}_R$ be the set of identifiers in S and R's databases respectively. At the end of the protocol S only learns $|\mathscr{I}_R|$, while R only learns $|\mathscr{I}_S|$ and $\mathscr{I}_R \cap \mathscr{I}_S$. It can later be extended to an equijoin protocol, which is slightly more technically involved. The basic ideas are the same, though.

In order to use the properties of the commutative encryption primitive, we need to map identities *id* to uniformly distributed random values. This is the reason why we need a *random oracle* [9] in these protocols. A random oracle is an artifice used in security proofs. It idealizes (in our case) a hash $H : \{0,1\}^* \to \mathrm{Dom}\,\mathscr{F}$ function, which means that $H(id)$ can be considered computed by a random oracle: every time $H(\cdot)$ is evaluated for a new identity *id*, the output $H(id)$ is distributed uniformly at random and independently from the previous output values.

In the following, if $\mathscr{I}$ is a set (list), then we denote by $H(\mathscr{I})$ the set (list) $\{H(id)\}_{id \in \mathscr{I}}$.

Intersection protocol

Input: S inputs $H(\mathscr{I}_S)$; R inputs $H(\mathscr{I}_R)$.

1. R generates a random $\kappa_R \xleftarrow{\$} \mathbb{Z}_q$ and computes the list $L_R = \langle\ \mathscr{I}_R,\ f_{\kappa_R}\big(H(\mathscr{I}_R)\big)\ \rangle$. Next, it sends S the list L_0 given by
$$L_0 = \langle\ f_{\kappa_R}\big(H(\mathscr{I}_R)\big)\ \rangle.$$
2. S generates a random $\kappa_S \xleftarrow{\$} \mathbb{Z}_q$ and send R two lists L'_0, L_1. First it sends the list L'_0 based on L_0 given by
$$L'_0 = \langle\ f_{\kappa_R}\big(H(\mathscr{I}_R)\big),\ f_{\kappa_S}\big(f_{\kappa_R}\big(H(\mathscr{I}_R)\big)\big)\ \rangle.$$

Secondly it sends R the list L_1 given by

$$L_1 = \langle\ f_{\kappa_S}\big(H(\mathscr{I}_S)\big)\ \rangle.$$

3. R transforms the list L_1 into the list

$$L_1' = \langle\ f_{\kappa_R}\big(f_{\kappa_S}\big(H(\mathscr{I}_S)\big)\big)\ \rangle.$$

4. R selects, with the help of L_R, all $id \in \mathscr{I}_R$ such that $f_{\kappa_R}\big(f_{\kappa_S}\big(H(id)\big)\big)$ appears both in L_0' and L_1'.

Output: S outputs $|\mathscr{I}_R|$; R outputs $|\mathscr{I}_S|$ and $\mathscr{I}_S \cap \mathscr{I}_R$.

The protocol is correct since, assuming there are no hash collisions and the commutativity property of the familiy $\mathscr{F}$, $id \in \mathscr{I}_S \cap \mathscr{I}_R$ iff $id \in \mathscr{I}_R$ and $f_{\kappa_R}\big(f_{\kappa_S}\big(H(id)\big)\big) = f_{\kappa_S}\big(f_{\kappa_R}\big(H(id)\big)\big)$ appears in L_0' and L_1'. Since H is modeled as a random oracle, the probability that n hash values have at least one collision equals [23]:

$$\Pr[\text{collision}] = 1 - \prod_{i=1}^{n-1} \frac{|\mathscr{G}| - i}{|\mathscr{G}|} \approx 1 - \exp\left(\frac{-n(n-1)}{2|\mathscr{G}|}\right)$$

In our protocols we use $|\mathscr{G}| \geq 2^{160}$, which renders Pr[collision] negligible. Therefore, with very high probability, the intersection protocol is correct.

4 A Pseudonym Scheme with Ubiquitous TTP

In this section we present a pseudonym scheme $\mathscr{P}^{\text{TTP}}$ and state its security properties. The description of the scheme starts by defining the system setup and key generation by the TTP and follows with the two fundamental protocols in the scheme: Researcher Supply and Researcher Equijoin. We assume that any two parties in the protocol communicate via a confidential channel. Our protocols distinguish between a sending and a receiving party which execute the protocols with the active help of the TTP. The sending and receiving parties send their inputs to the TTP. Finally the sending and receiving parties obtain their outputs from the TTP. This is why this scheme is called *ubiquitous TTP*, since the TTP is involved in all the exchanges between the parties, be it supply-to-researcher, or researcher-to-researcher.

This scheme uses a semantically secure symmetric encryption scheme (cf. Definition 5). The pseudonym of individual id in researcher R_s's pseudonymized database has the form $P(id, R_s) := \text{Enc}_{K_s}(id)$, where $K_s \stackrel{\$}{\leftarrow} \mathscr{K}$ is a random key that the TTP secretly assigns to researcher R_s, but which is never revealed.

We next describe the scheme $\mathscr{P}^{\text{TTP}}$. The TTP is in charge of performing the following operations:

System Setup The TTP also selects a semantically secure symmetric encryption algorithm $\big(\text{Enc}_K(\cdot), \text{Dec}_K(\cdot)\big)$, with $\text{Enc} : \mathscr{K} \times \{0,1\}^M \to \{0,1\}^{M'}$ for some integers

M, M' such that $M \leq M'$. Individuals, suppliers and researchers identifiers are binary strings of length M. The TTP publishes $\langle\, (\mathrm{Enc}, \mathrm{Dec}) \,\rangle$.

Researcher Key generation For each researcher R_j in the system, the TTP generates a secure key as $K_j \xleftarrow{\$} \mathscr{K}$. These keys are *secret* and only known to the TTP.

Supply Keys generation This algorithm is void.

Equijoin Keys generation This algorithm is void.

Researcher Supply The operation $S_l \xrightarrow{\mathrm{TTP}}_P R_d$ is performed as follows.

1. S_l sends to the TTP the list $\mathscr{I}_{S_l}$ of individuals in its database.
2. The TTP computes the list of pseudonyms as $\mathrm{Enc}_{K_d}(\mathscr{I}_{S_l})$. The TTP sends back to S_l the list $\langle\, id, P(id, R_d) \,\rangle$, where $P(id, R_d) = \mathrm{Enc}_{K_d}(id)$.
3. S_l sends to R_d the pseudonymized database $\langle\, P(id, R_d), PD(id, S_l) \,\rangle$.
4. R_d joins the data with already existing pseudonyms, and new rows for new pseudonyms.

Researcher Equijoin The protocol $R_s \xrightarrow{\mathrm{TTP}}_{\bowtie} R_d$ is performed as follows.

1. R_s sends its pseudonymized database $\langle\, P(id, R_s), PD(id, R_s) \,\rangle$ to the TTP.
2. R_d sends its pseudonymized database $\langle\, P(id, R_d), PD(id, R_d) \,\rangle$ to the TTP.
3. The TTP recovers $\mathscr{I}_{R_s}$ by decrypting the pseudonyms in R_s's list. The TTP recovers $\mathscr{I}_{R_d}$ similarly.
4. For every $id \in \mathscr{I}_{R_s} \cap \mathscr{I}_{R_d}$, the TTP computes the pseudonyms $P(id, R_d) := \mathrm{Enc}_{K_d}(id)$ and sends $\langle\, P(id, R_d), PD(id, R_s) \,\rangle$ to R_d.
5. R_d joins the data with already existing pseudonyms, and new rows for new pseudonyms.

Result 1 ($\mathscr{P}^{\mathrm{TTP}}$ is secure) *The pseudonyms' scheme $\mathscr{P}^{\mathrm{TTP}}$ satisfies pseudonymity, unlinkability, secure equijoin and equijoin non-transitivity provided that $\mathscr{E}$ is a semantically secure symmetric encryption scheme.*

Sketch of the proof. These properties are proven in a straightforward manner given the fact that the TTP is invoked in every algorithm, and that suppliers and researchers are given no cryptographic material.

For instance, regarding pseudonymity, one needs to prove that the distributions

$$\begin{pmatrix} id_1 & \dots & id_{n_d} \\ \mathrm{Enc}_{K_d}(id_1) & \dots & \mathrm{Enc}_{K_d}(id_{n_d}) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_1 & \dots & id_t & id_{t+1} & \dots & id_{n_d} \\ \mathrm{Enc}_{K_d}(id_1) & \dots & \mathrm{Enc}_{K_d}(id_t) & Z_{t+1} & \dots & Z_{n_d} \end{pmatrix}$$

are indistinguishable, where $Z_{t+1}, \dots, Z_{n_d} \xleftarrow{\$} \{0,1\}^{M'}$. Given that $\mathscr{E}$ is semantically secure, we know that $\big(id, \mathrm{Enc}_{K_S}(id)\big)$ is indistinguishable from $\big(id, Z\big)$, for any $id \in \{0,1\}^*$ and $Z \xleftarrow{\$} \{0,1\}^{M'}$. The result then follows by applying a standard hybrid argument [17].

Regarding unlinkability, we need to prove that the distributions

$$\begin{pmatrix} id_{t+1} & \dots & id_m \\ \mathrm{Enc}_{K_d}(id_{t+1}) & \dots & \mathrm{Enc}_{K_d}(id_m) \\ \mathrm{Enc}_{K_s}(id_{t+1}) & \dots & \mathrm{Enc}_{K_s}(id_m) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_{t+1} & \dots & id_m \\ \mathrm{Enc}_{K_d}(id_{t+1}) & \dots & \mathrm{Enc}_{K_d}(id_m) \\ Z_{t+1} & \dots & Z_m \end{pmatrix}$$

are indistinguishable. Since $\mathscr{E}$ is semantically secure, we know that

$$\begin{pmatrix} id_{t+1} & \dots & id_m \\ \mathrm{Enc}_{K_d}(id_{t+1}) & \dots & \mathrm{Enc}_{K_d}(id_m) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_{t+1} & \dots & id_m \\ Z_{t+1} & \dots & Z_m \end{pmatrix}$$

are indistinguishable, and that

$$\begin{pmatrix} id_{t+1} & \dots & id_m \\ \mathrm{Enc}_{K_s}(id_{t+1}) & \dots & \mathrm{Enc}_{K_s}(id_m) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_{t+1} & \dots & id_m \\ Z_{t+1} & \dots & Z_m \end{pmatrix}$$

are indistinguishable. These two facts imply that the distributions involved in the unlinkability definition are also indistinguishable.

Finally, secure equijoin and equijoin non-transitivity are implied by the fact that the operation 'equijoin' is performed by the trusted third party. □

The scheme $\mathscr{P}^{\mathrm{TTP}}$ with ubiquitous TTP reaches all the requested security properties at the expense of the TTP being involved in every single transaction in the system. This is something that could be undesirable in certain settings. For instance, if the number of transactions is high, then the TTP becomes a potential bottleneck in the system.

In the next section we propose protocols where the TTP is only required to hand on certain cryptographic keys to the parties. Apart from that, the TTP is not involved in any transaction, be it supplier-to-researcher or researcher-to-researcher.

5 A basic pseudonym scheme with light TTP

In this section we present a basic pseudonym scheme $\mathscr{P}^{\mathrm{basic}}$ and state its security properties. This scheme fulfills all the security properties we have identified, except for equijoin non-transitivity. It is included here as a first step towards a scheme satisfying all four security properties, to be presented in Section 6.

The description of the scheme starts with the system setup and key generation/distribution by the TTP and follows with the two fundamental protocols in the scheme: Researcher Supply and Researcher Equijoin. These protocols distinguish between a sending and a receiving party which are both provided with the necessary cryptographic keys by the TTP. The sending party (supplier or researcher) sends a chunk of data and possibly some temporary cryptographic keys to the receiver (researcher).

In this scheme we use commutative encryption both for creating pseudonyms and implementing the protocols. Researcher R_j's pseudonyms will depend on integers x_j. These quantities are secret and only known to the TTP. Researcher R_j's pseudonyms are elements in $\mathscr{G}$ and take the form $P(id,R_j) = f_{x_j}\big(H(id)\big)$, where $\mathscr{G}$ and f_{x_j} are defined as in Example 1, and $H : \{0,1\}^* \to \mathscr{G}$ is hash function. Our

equijoin protocol heavily relies in the equijoin protocol of Agrawal, Evfimievski and Srikant [4]. To give the basic idea behind our protocol, we illustrate it by extending the intersection algorithm described in Section 3.5 to a pseudonyms intersection algorithm between R_s and R_d.

Recall that in the intersection algorithm from Section 3.5, the inputs of the *Sender* S and *Receiver* R parties are $H(\mathscr{I}_S)$ and $H(\mathscr{I}_R)$ respectively, where $\mathscr{I}_R, \mathscr{I}_S$ are the sets of identities held by each party. In our case, we have that R_s does not hold a set of identities but a set of pseudonyms of the form $P(id,R_s) = H(id)^{x_s}$, and similarly, R_d holds a set of pseudonyms $P(id,R_d) = H(id)^{x_d}$, where $x_s, x_d \in \mathbb{Z}_q$ are unknown to R_s, R_d respectively. The idea is that the TTP feeds R_s (respectively R_d) with a pseudonyms' intersection key $\beta \cdot x_s^{-1}$ (respectively $\beta \cdot x_d^{-1}$), for $\beta \xleftarrow{\$} \mathbb{Z}_q$. This allows researchers R_s, R_d to respectively compute the commutative encryptions $f_{\beta \cdot x_s^{-1}}$ and $f_{\beta \cdot x_d^{-1}}$ of their respective pseudonyms' sets $\mathscr{P}_{R_s}$ and $\mathscr{P}_{R_d}$. That is, in the case of researcher R_s, it computes

$$f_{\beta \cdot x_s^{-1}}\big(P(id,R_s)\big) = f_{\beta \cdot x_s^{-1}}\big(f_{x_s}\big(H(id)\big)\big) = f_\beta\big(H(id)\big),$$

where the last equality is due to the property $f_a \circ f_b = f_{ab}$ of the exponentiation function family $\mathscr{F}$. This property is not used in [4], and this is one of the technical novelties with respect to the protocol in Section 3.5. Therefore, researchers end up with a *common representation* of their pseudonyms, namely, researcher R_s obtains the set $\{H(\mathscr{I}_s)^\beta\}$, while researcher R_d obtains the set $\{H(\mathscr{I}_d)^\beta\}$. At this point, R_s and R_d can run the intersection protocol from Section 3.5, with inputs $\{H(\mathscr{I}_s)^\beta\}$ and $\{H(\mathscr{I}_d)^\beta\}$, respectively. As a result, R_s gets as output $|\mathscr{P}_{\mathscr{I}_s}|$, while R_d gets as output $|\mathscr{P}_{\mathscr{I}_d}|$ and $\mathscr{P}_{\mathscr{I}_s} \cap \mathscr{P}_{\mathscr{I}_d}$.

Here follows the description of the pseudonyms scheme $\mathscr{P}^{\text{basic}}$. The TTP is in charge of performing the following operations:

System Setup The TTP chooses a cyclic group $\langle\, \mathscr{G}, p \,\rangle$ where the Decisional Diffie-Hellman assumption is believed to hold. It next picks a hash function $H : \{0,1\}^\ell \to \mathscr{G}$ for integer ℓ. The TTP also selects a semantically secure symmetric encryption algorithm $\big(\text{Enc}_K(\cdot), \text{Dec}_K(\cdot)\big)$, with $\text{Enc} : \mathscr{G} \times \{0,1\}^M \to \{0,1\}^{M'}$ for integers M, M' with $M \leq M'$. The TTP publishes $\langle\, \mathscr{G}, p, H, (\text{Enc}, \text{Dec}) \,\rangle$.

Researcher Key generation For each researcher R_j in the system, the TTP generates a secure key as $x_j \xleftarrow{\$} \mathbb{Z}_q$. These keys are *never delivered* to the researcher.

Supply Key generation For each pair supplier/researcher (S_l, R_j), the TTP recovers the researcher's assigned key $x_j \in \mathbb{Z}_q$ and hands it to the supplier through a secure channel. These keys are *never delivered* to any researcher.

Equijoin Keys generation For each pair (R_s, R_d) of researchers that is allowed to perform either the protocol $R_s \longrightarrow_{\bowtie} R_d$ (or $R_d \longrightarrow_{\bowtie} R_s$), the TTP generates a random $\beta_{s,d} \in \mathbb{Z}_q$ and sends $\beta_{s,d} \cdot x_s^{-1}$ (respectively $\beta_{s,d} \cdot x_d^{-1}$) to R_s (respectively R_d).

Researcher Supply The operation $S_l \to_P R_d$ is performed as follows:

1. S_l computes $P(id,R_d) = f_{x_j}\big(H(id)\big)$ for $id \in \mathscr{I}_{S_l}$ and sends to R_d the pseudonymized list $\langle\, P(id,R_d),\ PD(id,S_l) \,\rangle$.

2. R_d joins the data with already existing pseudonyms, and new rows for new pseudonyms.

Researcher Equijoin The protocol $R_s \rightarrow_{\bowtie} R_d$ is performed as follows:

1. R_s computes $f_{\beta_{s,d}\cdot x_s^{-1}}\left(\mathscr{P}_{R_s}\right)$ and obtains the set $\{H(\mathscr{I}_s)^{\beta_{s,d}}\}$.
2. R_d computes $f_{\beta_{s,d}\cdot x_d^{-1}}\left(\mathscr{P}_{R_d}\right)$ and obtains the set $\{H(\mathscr{I}_d)^{\beta_{s,d}}\}$.
3. R_s and R_d run the Agrawal, Evfimievski and Srikant's equijoin protocol [4, Section 4] with inputs $\left\{\left(H(id)^{\beta_{s,d}}, PD(id,R_s)\right)\right\}_{\mathscr{I}_s}$ and $\left\{\left(H(id)^{\beta_{s,d}}, PD(id,R_d)\right)\right\}_{\mathscr{I}_d}$ respectively. That is,

   a. R_d generates a random $\kappa_d \xleftarrow{\$} \mathbb{Z}_q$ and sends R_s the list L_0 given by

   $$L_0 = \langle\ f_{\kappa_d}\left(H(id)^{\beta_{s,d}}\right)\ \rangle_{\mathscr{I}_d}.$$

   b. R_s generates random $\kappa_s, \kappa_s' \xleftarrow{\$} \mathbb{Z}_q$ and sends R_d two lists L_0', L_1. First it creates the list L_0' based on L_0 given by

   $$L_0' = \left\langle\ f_{\kappa_d}\left(H(id)^{\beta_{s,d}}\right),\ f_{\kappa_s}\left(f_{\kappa_d}\left(H(id)^{\beta_{s,d}}\right)\right),\ f_{\kappa_s'}\left(f_{\kappa_d}\left(H(id)^{\beta_{s,d}}\right)\right)\ \right\rangle_{\mathscr{I}_d}.$$

   Secondly it creates the list L_1 given by

   $$L_1 = \left\langle\ f_{\kappa_s}\left(H(\overline{id})^{\beta_{s,d}}\right),\ \mathrm{Enc}_{K_s(\overline{id})}\left(PD(\overline{id},R_s)\right)\ \right\rangle_{\mathscr{I}_s},$$

   where $K_s(\overline{id}) = f_{\kappa_s'}\left(H(\overline{id})^{\beta_{s,d}}\right)$ for $\overline{id} \in \mathscr{I}_s$.

   c. Based on L_0' the researcher R_d calculates, by applying $f_{\kappa_d}^{-1}$, the list

   $$L_2 = \left\langle\ H(id)^{\beta_{s,d}},\ f_{\kappa_s}\left(H(id)^{\beta_{s,d}}\right),\ f_{\kappa_s'}\left(H(id)^{\beta_{s,d}}\right)\ \right\rangle_{\mathscr{I}_d}.$$

   d. Then R_d determines the elements in the list L_1, L_2 such that $f_{\kappa_s}\left(H(\overline{id})^{\beta_{s,d}}\right) = f_{\kappa_s}\left(H(id)^{\beta_{s,d}}\right)$. For those elements, which single out the pseudonyms such that $id \in \mathscr{I}_s \cap \mathscr{I}_d$, researcher R_d uses the corresponding values $f_{\kappa_s'}\left(H(id)^{\beta_{s,d}}\right) = K_s(id)$ in L_2 to decrypt

   $$PD(id,R_s) = \mathrm{Dec}_{K_s(\overline{id})}\left(\mathrm{Enc}_{K_s(\overline{id})}\left(\ PD(id,R_s)\ \right)\right).$$

   e. Finally, R_d joins the new data $PD(id,R_s)$ with its already existing data on $P(id,R_d)$.

Researcher equijoin correctness It is easy to see that the intersection algorithm is correct, i.e. at the end of the protocol R_d learns $\left(P(id,R_d), PD(id,R_s)\right)$ for $id \in \mathscr{I}_{R_s} \cap \mathscr{I}_{R_d}$, as long as the hash function $H : \{0,1\}^{\ell} \rightarrow \mathscr{G}$ does not present collisions.

Next we show that the scheme $\mathscr{P}^{\mathrm{basic}}$ satisfies pseudonymity, unlinkability and secure equijoin.

Result 2 *The pseudonym scheme* $\mathscr{P}^{\text{basic}}$ *has* pseudonymity *provided that* $\mathscr{F}$ *is a commutative encryption family.*

Proof. We want to show that the distributions

$$\begin{pmatrix} id_1 & \dots & id_{n_d} \\ f_{x_d}\big(H(id_1)\big) & \dots & f_{x_d}\big(H(id_{n_d})\big) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_1 & \dots & id_t & id_{t+1} & \dots & id_{n_d} \\ f_{x_d}\big(H(id_1)\big) & \dots & f_{x_d}\big(H(id_t)\big) & Z_{t+1} & \dots & Z_{n_d} \end{pmatrix}$$

are indistinguishable, where $Z_{t+1},\dots,Z_{n_d} \xleftarrow{\$} \mathscr{G}$. Given that $\mathscr{F}$ is a commutative encryption family, we know that the distributions

$$\big(H(id), f_{x_d}\big(H(id)\big), H(id'), f_{x_d}\big(H(id')\big)\big) \qquad \text{and} \qquad \big(H(id), f_{x_d}\big(H(id)\big), H(id'), Z\big),$$

where $id, id' \in \{0,1\}^l$, $Z \xleftarrow{\$} \mathscr{G}$ and $x_d \xleftarrow{\$} \text{Dom}\,\mathscr{F}$, are indistinguishable, since for a random oracle H the values $H(id), H(id')$ follow a uniformly random distribution. The result then follows by applying a standard hybrid argument [17, 4]. □

Result 3 *The pseudonym scheme* $\mathscr{P}^{\text{basic}}$ *has* unlinkability *provided that* $\mathscr{F}$ *is a commutative encryption family.*

Proof. We want to show that the distributions

$$\begin{pmatrix} id_t & \dots & id_m \\ f_{x_d}\big(H(id_t)\big) & \dots & f_{x_d}\big(H(id_m)\big) \\ f_{x_s}\big(H(id_t)\big) & \dots & f_{x_s}\big(H(id_m)\big) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} id_{t+1} & \dots & id_m \\ f_{x_d}\big(H(id_t)\big) & \dots & f_{x_d}\big(H(id_m)\big) \\ Z_{t+1} & \dots & Z_m \end{pmatrix}$$

are indistinguishable, where $Z_{t+1},\dots,Z_m \xleftarrow{\$} \mathscr{G}$, $x_d, x_s \xleftarrow{\$} \text{Keys}\,\mathscr{F}$. Given that $\mathscr{F}$ is a commutative encryption family, we use the following lemma.

Lemma 1 ([4]). *For any integer n, the distributions of the tuples*

$$\begin{pmatrix} s_1 & \dots & s_n \\ f_{x_e}(s_1) & \dots & f_{x_e}(s_n) \\ f_{x_d}(s_1) & \dots & f_{x_d}(s_n) \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} s_1 & \dots & s_n \\ y_1 & \dots & y_n \\ z_1 & \dots & z_n \end{pmatrix}$$

are computationally indistinguishable, where $0 \le n$, $\forall i:\ s_i, y_i, z_i \xleftarrow{\$} \textit{Dom}\,\mathscr{F}$, *and* $x_d, x_s \xleftarrow{\$} \text{Keys}\,\mathscr{F}$, *provided that* $\mathscr{F}$ *is a commutative encryption family.*

Now, if we identify $s_i := H(id_i)$, the result holds due to the fact that H is a random oracle. □

Result 4 *The pseudonym scheme* $\mathscr{P}^{\text{basic}}$ *provides* secure equijoin *provided that* $\mathscr{F}$ *is a commutative encryption family.*

Proof. This result follows from the fact that our equijoin protocol is an extension of the equijoin protocol by Agrawal *et al.*. The only change is on the inputs to the protocol. Let us see that. In [4, Section 4] the parties R and S input the sets

$\{(H(id),D(id))\}_{id\in\mathscr{I}_R}$ and $\{(H(id),D(id))\}_{id\in\mathscr{I}_S}$ respectively. The output they obtain is $|\mathscr{I}_R|$, $|\mathscr{I}_S|$ and the equijoin of their databases.

In our protocol, the parties R_s and R_d input the sets $\{(P(id,R_s),PD(id,R_s))\}_{id\in\mathscr{I}_{R_s}}$ and $\{(P(id,R_d),PD(id,R_d))\}_{id\in\mathscr{I}_{R_d}}$ respectively. The output they obtain is $|\mathscr{I}_{R_s}|$, $|\mathscr{I}_{R_d}|$ and the equijoin of their pseudonymized databases.

Now, from the discussion in Section 3.5, we know (and this can also be straightforwardly checked by looking at the protocol in [4, Section 4.3]) that our equijoin protocol is obtained by changing the inputs to the Agrawal *et al.* equijoin protocol. More precisely, we replace $\{\ (H(id),D(id))\ \}_{id\in\mathscr{I}_R}$ and $\{\ (H(id),D(id))\ \}_{id\in\mathscr{I}_S}$ by $\{\ (H(id)^{\beta_{s,d}},PD(id,R_s))\ \}_{id\in\mathscr{I}_{R_s}}$ and $\{\ (H(id)^{\beta_{s,d}},PD(id,R_d))\ \}_{id\in\mathscr{I}_{R_d}}$ respectively. As a consequence, the output is as expected, and the secure equijoin property is preserved, since both $H(id)$ and $H(id)^{\beta_{s,d}}$ follow the uniform distribution for any id. □

Alas, the scheme $\mathscr{P}^{\text{basic}}$ does not satisfy the equijoin non-transitivity property. Indeed,

Result 5 (Security breach with colluding researchers) *The scheme $\mathscr{P}^{\text{basic}}$ does not provide* equijoin non-transitivity.

Proof. Let us assume the pairs of researchers R_s,R_d and R_d,R_o are allowed to compute the equijoin of the corresponding databases. If these researchers collude, they can compute cryptographic keys enabling translation of $P(id,R_s)$ into $P(id,R_d)$, and the translation of pseudonyms $P(id,R_d)$ into $P(id,R_o)$ as follows. Remember the intersection key for R_s is $\beta_{s,d}\cdot x_s^{-1}$, while for R_d is $\beta_{s,d}\cdot x_d^{-1}$. Then $\beta_{s,d}\cdot x_s^{-1}/(\beta_{s,d}\cdot x_d^{-1}) = x_s^{-1}x_d$, and $f_{x_s^{-1}x_d}(P(id,R_s)) = P(id,R_d)$. And similarly for researchers R_d,R_o. Therefore a transformation $P(id,R_s)\mapsto P(id,R_d)\mapsto P(id,R_o)$ can be computed for any pseudonym $P(id,R_s)$ in possession of R_s, which allows R_s,R_d,R_o to compute the equijoin $\mathscr{P}_{R_s}\cap\mathscr{P}_{R_o}$. As a result, equijoin non-transitivity is broken and the scheme is not fully secure. □

In the next section we propose a fully secure protocol using pairings.

6 A fully secure pseudonym scheme with light TTP

The problem with the previous solution lied on the fact that malicious researchers could translate pseudonyms from researcher R_s to researcher R_o by operating with the equijoin keys. This was possible because the equijoin keys were elements in $\mathbb{Z}_q$ that could be manipulated to produce keys enabling translation of pseudonyms. The proposal in this section seeks to solve the problem by giving out keys as elements in a finite group instead of integers in a modular ring. We accomplish that by using pairing groups and by making equijoin keys elements in $\mathbb{G}_2 = \langle h\rangle$. In short, the equijoin protocol remains essentially the same, but the researchers' equijoin keys

will be $h^{\beta_{s,d}\cdot x_s^{-1}}$ for R_s (in contrast to $\beta_{s,d}\cdot x_s^{-1}$ for the basic protocol) and $h^{\beta_{s,d}\cdot x_d^{-1}}$ for R_d (in contrast to $\beta_{s,d}\cdot x_d^{-1}$ for the basic protocol). Informally, when R_s and R_d collude to break equijoin non-transitivity in this new situation, they will need to compute $h^{x_s^{-1}x_d}$ from the elements $h^{\beta_{s,d}\cdot x_s^{-1}}$ and $h^{\beta_{s,d}\cdot x_d^{-1}}$, which would amount to solving the Computational Diffie-Hellman problem in $\mathbb{G}_2$. However, the latter is assumed to be infeasible.

An important change to be noticed is that now R_s, R_d do not input $\{H(\mathscr{I}_s)^{\beta_{s,d}}\}$ and $\{H(\mathscr{I}_d)^{\beta_{s,d}}\}$ respectively to the equijoin protocol, but $\{e\big(H(\mathscr{I}_s),h\big)^{\beta_{s,d}}\}$ and $\{e\big(H(\mathscr{I}_d),h\big)^{\beta_{s,d}}\}$. In the case of R_s, the new set is computed as

$$e\big(P(id,R_s),h^{\beta_{s,d}\cdot x_s^{-1}}\big)=e\big(H(id)^{x_s},h^{\beta_{s,d}\cdot x_s^{-1}}\big)=e\big(H(id)^{x_s\cdot x_s^{-1}},h\big)^{\beta_{s,d}}=e\big(H(id),h\big)^{\beta_{s,d}}$$

thanks to bilinearity of the pairing $e(\cdot,\cdot)$. To be more precise, the new equijoin protocol works with elements $U\in\mathbb{G}_3$ and uses the commutative encryption function family $\mathscr{F}' := \{F_a \,:\, U\mapsto U^a\}_{a\in\mathbb{Z}_q}$.

We proceed to describe the pseudonyms scheme $\mathscr{P}^{\text{advanced}}$ that provides equijoin non-transitivity against colluding researchers.

System Setup The TTP chooses a pairing group $\langle\, \mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,e,g,q\,\rangle$. It next picks a hash function $H:\{0,1\}^\ell\to\mathbb{G}_1$. Individuals' identifiers are binary strings of length ℓ. The TTP also selects a semantically secure symmetric encryption algorithm $\big(\mathrm{Enc}_K(\cdot),\mathrm{Dec}_K(\cdot)\big)$, where K denotes the encryption key. The TTP publishes $\langle\, \mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_3,e,\phi,g,q,H,(\mathrm{Enc},\mathrm{Dec})\,\rangle$.

All operations are as in the basic scheme with the only exception of the equijoin keys generation and equijoin algorithms. For the comprehension of the new protocols, let us note the following equivalences

$$F_a\big(e(u,h)\big)=e(u,h)^a=e(u,h^a)=e(u^a,h)=e(f_a(u),h) \tag{1}$$

for any $a\in\mathbb{Z}_q,\ u\in\mathbb{G}_1,\ h\in\mathbb{G}_2$.

Equijoin Keys generation For each pair (R_s,R_d) of researchers that is allowed to perform the protocol $R_s\longrightarrow\!\!\!\!\!\bowtie R_d$ the TTP generates a random $\beta_{s,d}\in\mathbb{Z}_q$ and sends $h^{\beta_{s,d}\cdot x_s^{-1}}$ to R_s and $h^{\beta_{s,d}\cdot x_d^{-1}}$ to R_d through a secure channel.

Researchers Equijoin The protocol $R_s\rightarrow\!\!\!\!\bowtie R_d$ is performed as follows.

1. R_s computes $e\big(\mathscr{P}_{R_s},h^{\beta_{s,d}\cdot x_s^{-1}}\big)$ and obtains the set $\{e\big(H(\mathscr{I}_s)^{\beta_{s,d}},h\big)\}$.
2. R_d computes $e\big(\mathscr{P}_{R_d},h^{\beta_{s,d}\cdot x_d^{-1}}\big)$ and obtains the set $\{e\big(H(\mathscr{I}_d)^{\beta_{s,d}},h\big)\}$.
3. R_s and R_d run the Agrawal, Evfimievski and Srikant's equijoin protocol [4, Section 4], but with the commutative encryption family $\{F_a \,:\, U\mapsto U^a\}_{a\in\mathbb{Z}_q}$ and inputs $\Big\{\ \Big(e\big(H(id),h\big)^{\beta_{s,d}},PD(id,R_s)\Big)\ \Big\}_{\mathscr{I}_{R_s}}$ and $\Big\{\ \Big(e\big(H(id),h\big)^{\beta_{s,d}},PD(id,R_d)\Big)\ \Big\}_{\mathscr{I}_{R_d}}$ respectively. That is,

   a. R_d generates a random $\kappa_d\xleftarrow{\$}\mathbb{Z}_q$ and sends R_s the list L_0 given by

$$L_0 = \langle\ F_{\kappa_d}\big(e(H(id),h)^{\beta_{s,d}}\big)\ \rangle_{\mathscr{I}_d}.$$

b. R_s generates random $\kappa_s, \kappa_s' \xleftarrow{\$} \mathbb{Z}_q$ and sends R_d two lists L_0', L_1. First it creates the list L_0' based on L_0 given by

$$L_0' = \Big\langle\ F_{\kappa_d}\big(e(H(id),h)^{\beta_{s,d}}\big),\quad F_{\kappa_s}\Big(F_{\kappa_d}\big(e(H(id),h)^{\beta_{s,d}}\big)\Big),\\ F_{\kappa_s'}\Big(F_{\kappa_d}\big(e(H(id),h)^{\beta_{s,d}}\big)\Big)\ \Big\rangle_{\mathscr{I}_d}.$$

Secondly it creates the list L_1 given by

$$L_1 = \Big\langle\ F_{\kappa_s}\big(e(H(\overline{id}),h)^{\beta_{s,d}}\big),\ \mathrm{Enc}_{K_s(\overline{id})}\big(PD(\overline{id},R_s)\big)\ \Big\rangle_{\mathscr{I}_s},$$

where $K_s(\overline{id}) = F_{\kappa_s'}\big(e(H(\overline{id}),h)^{\beta_{s,d}}\big)$ for $\overline{id} \in \mathscr{I}_s$.

c. Based on L_0' the researcher R_d calculates, by applying $F_{\kappa_d}^{-1}$, the list

$$L_2 = \Big\langle\ e(H(id),h)^{\beta_{s,d}},\ F_{\kappa_s}\big(e(H(id),h)^{\beta_{s,d}}\big),\ F_{\kappa_s'}\big(e(H(id),h)^{\beta_{s,d}}\big)\ \Big\rangle_{\mathscr{I}_d}.$$

d. Then R_d determines the elements in the list L_1, L_2 such that $F_{\kappa_s}\big(e(H(\overline{id}),h)^{\beta_{s,d}}\big) = F_{\kappa_s}\big(e(H(id),h)^{\beta_{s,d}}\big)$. For those elements, which single out the pseudonyms such that $id \in \mathscr{I}_s \cap \mathscr{I}_d$, researcher R_d uses the corresponding values

$$F_{\kappa_s'}\big(e(H(id),h)^{\beta_{s,d}}\big) = K_s(id)$$

in L_2 to decrypt

$$PD(id,R_s) = \mathrm{Dec}_{K_s(\overline{id})}\Big(\mathrm{Enc}_{K_s(\overline{id})}\big(\ PD(id,R_s)\ \big)\Big).$$

e. Finally, R_d joins the new data $PD(id,R_s)$ with its already existing data on $P(id,R_d)$.

Researcher equijoin correctness It is easy to see that the equijoin algorithm is correct, i.e. at the end of the protocol R_d learns $\big(P(id,R_d), PD(id,R_s)\big)$ for $id \in \mathscr{I}_{R_s} \cap \mathscr{I}_{R_d}$, as long as the hash function $H : \{0,1\}^\ell \to \mathbb{G}_1$ does not present collisions.

The following results follow directly by extending the corresponding results from Section 5.

Result 6 *The pseudonym scheme* $\mathscr{P}^{\text{advanced}}$ *has* pseudonymity *provided that* $\mathscr{F}'$ *is a commutative encryption family.*

Result 7 *The pseudonym scheme* $\mathscr{P}^{\text{advanced}}$ *has* unlinkability *provided that* $\mathscr{F}'$ *is a commutative encryption family.*

Result 8 *The pseudonym scheme* $\mathscr{P}^{\text{advanced}}$ *has* secure equijoin *provided that* $\mathscr{F}'$ *is a commutative encryption family.*

Next, we show that $\mathscr{P}^{\text{advanced}}$ satisfies the equijoin non-transitivity property.

Result 9 *The pseudonym scheme* $\mathscr{P}^{\text{advanced}}$ *has* secure equijoin transitivity *provided that the Asymmetric DDH assumption holds.*

Proof. Let us first recall that the Asymmetric DDH assumption states the indistinguishability of the probability distributions $(g, g^a, g^b, h^b, g^{ab})$ and (g, g^a, g^b, h^b, g^r) where $g = \phi(h), h$ generate $\mathbb{G}_1, \mathbb{G}_2$ respectively, and $a, b, r \xleftarrow{\$} \mathbb{Z}_q$.

To convey this proof we use a different proof technique from those previously deployed both in this chapter and in [4]. We need to *program* the random oracle. This means that H being a random oracle, the values $H(id)$ are simulated to the adversary. The simulator answers $H(id_i)$ as $(g^a)^{\lambda_i}$ for $id_i \notin \mathscr{I}_{R_d}$, where $\lambda_i \xleftarrow{\$} \mathbb{Z}_q$, and answers $H(id_i)$ as g^{λ_i} for $id_i \in \mathscr{I}_{R_d}$, where $\lambda_i \xleftarrow{\$} \mathbb{Z}_q$. It takes $x_s, x_d, \alpha, \beta \xleftarrow{\$} \mathbb{Z}_q$. Additionally it picks $\overline{h} \xleftarrow{\$} \mathbb{G}_2$ and defines $\phi(\overline{h}) = \overline{g} \in \mathbb{G}_1$. Next, for the pair (R_s, R_d), it sets as R_s's equijoin key the quantity $(\overline{h})^{\alpha \cdot x_s^{-1}}$; while for R_d it is set to $(\overline{h})^{\alpha \cdot x_d^{-1}}$. Next, for the pair (R_d, R_o), it sets as R_d's equijoin key the quantity $(h^b)^{\beta \cdot x_d^{-1}}$; while for R_o it is set to $(\overline{h})^{\beta}$. The pseudonyms are simulated as follows:

- $P(id_i, R_s) = \overline{g}^{\lambda_i x_s}$ for $id_i \in \mathscr{I}_{R_d} \cap \mathscr{I}_{R_s}$
- $P(id_i, R_s) = (g^a)^{\lambda_i x_s}$ for $(id_i \in \mathscr{I}_{R_s}) \wedge (id_i \notin \mathscr{I}_{R_d})$
- $P(id_i, R_d) = \overline{g}^{\lambda_i x_d}$ for $id_i \in \mathscr{I}_{R_d}$
- $P(id_i, R_o) = (g^b)^{\lambda_i}$ for $id_i \in \mathscr{I}_{R_d} \cap \mathscr{I}_{R_o}$
- $P(id_i, R_o) = (g^{ab})^{\lambda_i}$ for $(id_i \in \mathscr{I}_{R_o}) \wedge (id_i \notin \mathscr{I}_{R_d})$

The above simulation is consistent with the adversarial's view. For instance, we have that for any $id \in \mathscr{I}_{R_d} \cap \mathscr{I}_{R_s}$,

$$e\big(P(id, R_s), (\overline{h})^{\alpha \cdot x_s^{-1}}\big) = e\big(\overline{g}, h\big)^{\alpha \lambda_i} = e\big(P(id, R_d), (\overline{h})^{\alpha \cdot x_d^{-1}}\big)$$

and for any $id \in \mathscr{I}_{R_d} \cap \mathscr{I}_{R_o}$,

$$e\big(P(id, R_d), (h^b)^{\beta \cdot x_d^{-1}}\big) = e\big(\overline{g}, h\big)^{\beta b \lambda_i} = e\big(g, \overline{h}\big)^{\beta b \lambda_i} = e\big(P(id, R_o), (\overline{h})^{\beta}\big) \quad (2)$$

where Equation 2 holds because any $h, \overline{h} \in \mathbb{G}_2$ satisfy that $e(\phi(h), \overline{h}) = e(\phi(\overline{h}), h)$, where $\phi : \mathbb{G}_2 \to \mathbb{G}_1$ is an efficiently computable homomorphism.

Finally, since g^{ab} cannot be distinguished from random g^r, and thus they can be swapped in the above expressions, it follows that pseudonyms $P(id, R_o)$ outside the intersection $\mathscr{I}_{R_d} \cap \mathscr{I}_{R_s} \cap \mathscr{I}_{R_o}$ are indistinguishable from random pseudonyms. Thus, the adversary can not match pseudonyms $P(id, R_s)$ to pseudonyms $P(id, R_o)$ for $id \in (\mathscr{I}_{R_s} \cap \mathscr{I}_{R_o}) - (\mathscr{I}_{R_d} \cap \mathscr{I}_{R_s} \cap \mathscr{I}_{R_o})$. □

7 Conclusion

This chapter describes pseudonymisation and equijoin protocols aimed at building pseudonymized data sharing systems. Our pseudonymization algorithm produces unlinkable pseudonyms sets, yet allows secure intersection between them, provided that certain cryptographic keys are available. We have presented three schemes: a first scheme uses a mighty TTP which is invoked in every algorithm; the last two schemes use a "light" TTP which act as a key distribution center. Our last two schemes are proven secure in the Random Oracle Model. One problem that is left open is to provide the light-TTP functionality without resorting to the random oracle idealization.

References

1. *Clef: Clinical e-science framework*, `http://www.clinical-escience.org/`.
2. *Zorg TTP: Privacy & vertrouwen*, `https://www.zorgttp.nl`.
3. ACGT, *Advancing clinico-genomic clinical trials on cancer: Open grid services for improving medical knowledge discovery*, `http://eu-acgt.org/`.
4. Rakesh Agrawal, Alexandre V. Evfimievski, and Ramakrishnan Srikant, *Information sharing across private databases.*, SIGMOD Conference, ACM, 2003, pp. 86–97.
5. Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros, *Untraceable RFID tags via insubvertible encryption*, ACM Conference on Computer and Communications Security, 2005, pp. 92–101.
6. Dutch Data Protection Authority, *Pseudonimisering persoonsgegevens bij risicoverevening*, 2007, `http://www.cbpweb.nl/documenten/uit_z2006-1328.shtml?refer=true&theme=purple`.
7. _____, *Landelijke zorgregistraties (national healthcare registrations)*, www.dutchdpa.nl, 2005.
8. Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose, *Correlation-resistant storage via keyword-searchable encryption*, Cryptology ePrint Archive, Report 2005/417, 2005, `http://eprint.iacr.org/`.
9. Mihir Bellare and Philip Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols*, Proceedings of the 1st ACM CCS, ACM Press, 1993, pp. 62–73.
10. Dan Boneh, Xavier Boyen, and Hovav Shacham, *Short group signatures*, CRYPTO, Lecture Notes in Computer Science, vol. 3152, Springer, 2004, pp. 41–55.
11. Dan Boneh, Ben Lynn, and Hovav Shacham, *Short signatures from the weil pairing*, J. Cryptology **17** (2004), no. 4, 297–319.
12. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya, *Compact e-cash*, EUROCRYPT, Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 302–321.
13. Josep Domingo-Ferrer (ed.), *Inference control in statistical databases, from theory to practice*, Lecture Notes in Computer Science, vol. 2316, Springer, 2002.
14. Josep Domingo-Ferrer and Luisa Franconi (eds.), *Privacy in statistical databases, cenex-sdc project international conference, PSD 2006*, Lecture Notes in Computer Science, vol. 4302, Springer, 2006.
15. International Organization for Standardization, *ISO/TS 25237:2008, Health informatics – Pseudonymization*, 2008.
16. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart, *Pairings for cryptographers*, Discrete Applied Mathematics **156** (2008), no. 16, 3113–3121.

17. Oded Goldreich, *Foundations of cryptography II - Basic applications*, 1st ed., Cambridge University Press, 2004.
18. Petra Knaupa, Sebastian Gardeb, Angela Merzweilerc, Norbert Graf, Freimut Schillin, Ralf Weberf, and Reinhold Hauxg, *Towards shared patient records: An architecture for using routine data for nationwide research*, International Journal of Medical Informatics **75** (2004), 191200.
19. Bradley Malin, *Why pseudonyms dont anonymize: A computational re-identification analysis of genomic data privacy protection systems*, Laboratory for International Data Privacy at Carnegie Mellon University, `http://privacy.cs.cmu.edu/dataprivacy/projects/linkage/lidap-wp19.pdf`.
20. James L. Massey, *An introduction to contemporary cryptology*, Proceedings of the IEEE, vol. 76, IEEE, 2008.
21. Bernhard Riedl, Veronika Grascher, Stefan Fenz, and Thomas Neubauer, *Pseudonymization for improving the privacy in e-health applications*, HICSS, IEEE Computer Society, 2008, p. 255.
22. Adi Shamir, *On the power of commutativity in cryptography*, ICALP, Lecture Notes in Computer Science, vol. 85, Springer, 1980, pp. 582–595.
23. Douglas R. Stinson, *Cryptography: Theory and practice,*, 3rd ed., CRC Press, 2005.