Privacy protection in electronic education based on polymorphic pseudonymization

Eric R. Verheul

Radboud University Nijmegen, KeyControls
P.O. Box 9010, NL-6500 GL Nijmegen, The Netherlands.
eric.verheul@[cs.ru.nl,keycontrols.nl]
Draft 28th December 2015

Abstract In [13] Dutch government proposes an identity scheme supporting personal data exchange of pupils with private e-textbook publishers. This design propagates sharing personal numbers of pupils among private parties violating the data minimisation principle in privacy laws. We describe a privacy friendly alternative, giving pupils (and parents) control on exchange of their personal data. Three generic forms based on homomorphic encryption are used as building blocks. These forms do not yield personal numbers, or even personal data from a legal perspective, and have strong, unlinkability properties. Only if required a school provides a party with a party-specific *pseudonym* identifying a pupil. For this the school is provided an *encrypted pseudonym* by a central party based on a *polymorphic pseudonym* formed by the school. Only intended parties, not even schools, have access to pseudonyms. Different publishers can send pupil test results to a school without being able to assess whether pupils are identical. We also describe support for privacy friendly attributes and user inspection as required by privacy laws.

Keywords: e-textbooks, homomorphic encryption, pseudonyms, privacy enhancing technology

1 Introduction

Schools[1] are replacing conventional books with their electronic analogues, electronic textbooks, or e-textbooks for short. See [4]. The term e-textbook is somewhat misleading as it encompasses much more functionality than a conventional textbook. For instance, an e-textbook allows for richer forms of contents such as audio and video. It can also provide for bookmarks, allows for interaction with the pupil and can support the teacher with feedback on its pupils progress. The latter also allows for further tailoring of the e-textbook, e.g. the teacher giving specific tasks to the pupil. Typically an e-textbook is hosted by an educational

[1] For simplicity we talk about schools, but the same discussion applies to other educational institutions, e.g. universities, too.

publisher as a website. To access its e-textbook the pupil needs to login (authenticate) to a portal of the school and is then redirected to the e-textbook site at the publisher. At first usage of the e-textbook the publisher sends a request to the school to pay for a software license for the pupil's e-textbook. This first use-case illustrates the importance that a pupil identity in the publisher environment is linked to its identity in the school environment. A second use-case is that the school directs the pupil from the school portal to the pupil specific e-textbook instance at the publisher. The third use-case is that the publisher provides feedback on the pupil, e.g. test results, to the school. Finally, when a pupil moves to another school it is important that the pupil identities at publishers are unchanged as otherwise historic information on the pupil gets lost.

A basic electronic identity (eID) scheme supporting the above functionality is to let schools share the full identity of pupils with publishers, e.g. the first and last name. As these are typically not unique on a national scale it would be functionally better to supply publishers a *unique personal number*, called PN hereafter, e.g. a social security number. Although this solution provides all required functionality it is not compliant with data protection laws in many countries, including countries in the European Union. Specifically, this setup is not in line with the data minimisation principle as stipulated in Article 5 of the draft European privacy regulation [7]: "Personal data must be adequate, relevant, and limited to the minimum necessary in relation to the purposes for which they are processed". We also note that the educational data itself, such as the textbook nature and test results, can also be privacy sensitive. Indeed, such educational data can be commercially valuable to profile pupils. The textbook could also refer to a medical disorder such as dyslexia or to a specific thing that happened to the pupil, e.g. abuse. These kinds of personal data are identified as sensitive in regulation [7] and are required to receive specific protection. This further motivates why publishers should not be provided full pupil identities.

In [13] a simple variant of the basic eID scheme is outlined for Dutch education. Here education is divided into three sectors: primary, secondary and vocational schools. Throughout each sector a pupil is known under a *sector pseudonym* where the social security number of the pupil is used as PN. To this end, the school sends the pupil's PN in hashed form to a central party, called Numbering Facilty. This transforms it into a pseudonym using a secret operation and provides it to the school. All communication on the pupil in the sector is then accompanied with its pseudonym. We remark that hashing the PN is not adding much security as the Numbering Facilty can brute-force the PN based on its hash value. We also remark that [13] euphemistically uses the term "chain pseudonym" and on [13, p.20] it is suggested that the pseudonym could depend on a "chain" of parties in the sector. However it is explicitly stated on [13, p.11] that the pseudonym is the same for all publishers in the sector. This can also be deduced from the property on [13, p.6] that the pseudonym stays the same when a pupils moves to another school (that might use different publishers).

Although practical, sector pseudonyms introduce a new personal number throughout the whole sector. Actually in this light, the term *sector personal*

number would be more appropriate. We argue that like the basic scheme, setup [13] is still in conflict with the data minimisation principle of the EU regulation. Indeed, with the sector pseudonyms publishers can link their databases which is not only unnecessary but which should be avoided. Indeed, the sector pseudonyms facilitate that parties work together to identify a pupil and combine data. More worrisome is when parties (schools, publishers) get hacked. Then the attackers can perform this identification and the resulting personal data can be abused, sold, or even published as part of blackmailing. Three recent related incidents [2,11,18] demonstrate the relevancy of such hacks. The division in [13] into three sectors is meant as a rudimentary privacy control but in fact also hampers the objective of the design: necessary exchange in the education domain. Indeed, if a pupil moves to another school type, the pupil's data cannot be linked even when required, e.g. in case of continuous dyslexia testing.

In this paper we introduce polymorphic pseudonymization (PP) and describe how this can be used in a privacy enhanced eID scheme in education . Our scheme provides the required functionality (and even more) but pupil privacy is better protected than in [13] and is in line with the data minimisation principle.

Outline of the paper
In Section 2 we introduce the cryptographic building blocks for the scheme based on the ElGamal encryption scheme. Section 3 describes polymorphic pseudonymization and the eID scheme based upon it. Section 4 discusses security and legal compliance with privacy regulations of our scheme. In Section 5 we discuss two supplements to our scheme consisting of privacy friendly attributes and user inspection which is a legal right of individuals. Section 6 contains conclusions and future work.

2 Notation and preliminaries

In this section we develop the cryptographic tools used in polymorphic pseudonymization. We suggest that the reader first reads the general idea from Section 3.1. Throughout this paper we let $\mathcal{H}(.)$ represent a secure hash function, e.g. the SHA256 hash function as specified in [12]. In this paper we also let $G = \langle g \rangle$ be a multiplicative group of prime order q generated by a generator element g. By GF(q) we denote the Galois field of the integers modulo q. The cryptographic security of G can be formulated in four problems in the context of the Diffie-Hellman key agreement protocol with respect to g. The first one is the *Diffie-Hellman problem*, which consists of computing the values of the function $DH_g(g^x, g^y) = g^{xy}$. Two other problems are related to the Diffie-Hellman problem. The first one is the *Decision Diffie-Hellman* (DDH) problem with respect to g: given $\alpha, \beta, \delta \in G$ decide whether $\delta = DH_g(\alpha, \beta)$ or not. The DH problem with respect to g is at least as difficult as the DDH problem with respect to g. The second related problem is the *discrete logarithm* (DL) problem in G with respect to g: given $\alpha = g^x \in G$, with $x \in$ GF(q) then find $x = DL_g(\alpha)$. The DL problem with respect to g is at least as difficult as the DH problem with respect to g.

One can easily show that if one can solve the discrete logarithms with respect to one generator, one can solve it with respect to any generator of G. That is, the hardness of the discrete logarithm problem is independent of the generator of the group. In [15] a similar property is shown for the Diffie-Hellman problem. It seems very unlikely that the hardness of the Decision Diffie-Hellman problem is dependent of the group generator. However, as far as we know such a result is not known to be provable. To this end, we say that one can solve the Decision Diffie-Hellman problem with respect to the group G if one can solve the Decision Diffie-Hellman problem with respect to any generator of the group. An equivalent definition is as follows. Any quadruple of points in G can be written as (h, j, h^x, j^y) for some (unknown) $x, y \in \mathrm{GF}(q)$. Then the general Decision Diffie-Hellman problem amounts to deciding whether a random quadruple of points in G is a *DDH quadruple*, i.e. if $x = y$. We assume that all four introduced problems in G are intractable.

For practical implementations one can use a group of points G on an elliptic curve, cf. [6]. The size of the group q in bits should be at least 256. Throughout this paper we will let $\mathcal{M}(K, \mathrm{string})$ represent a key derivation function (KDF) that maps a string into a secret key in $\mathrm{GF}(q)^*$. One can think of the KDF functions from [8]. For easy reference we simply refer to such keys as *KDF keys*.

We will also distinguish a secure hash function $\mathcal{I}(.) : \{0,1\}^* \to G$ that maps a string into the group G. In the context of an elliptic curve group $E(\mathrm{GF}(p))$ over a finite field $\mathrm{GF}(p)$ two approaches exist for such an embedding. A straightforward approach, cf. [10], is probabilistic. Here one uses a standard secure hash function to map the string to an element $x \in \mathrm{GF}(p)$ and verifies there exists a curve point with this x-coordinate. If this is not the case one varies the string in a deterministic fashion, e.g. by concatenating a string corresponding to an incrementing counter that starts with 1 and tries again. Each try has a fifty percent of success so eventually one will find a point on the curve. A deterministic polynomial-time algorithm to embed strings in elliptic curves is in [14].

For $S \in G$, $x, k \in \mathrm{GF}(q)$ and $y = g^x$ we let $\mathcal{EG}(S, y, k)$ denote the ElGamal encryption [5] of *plaintext* $S \in G$ with respect to the *public key* y and *private* key x. Technically, an ElGamal encryption consists of a pair of points in G of the form $(g^k, S \cdot y^k)$. The number k is called the *randomization exponent*. As can be easily verified, the decryption of an ElGamal encryption (A, B) is given by B/A^x. Throughout the paper we consider the generator g as the basis for all ElGamal encryptions which is why we do not explicitly include g as a parameter in $\mathcal{EG}(.)$. We consider g and in fact the specifications of the group G to be implicitly defined in the scheme specifications.

We remark that strictly speaking the public key y does not need to be included in the ElGamal encryption $\mathcal{EG}$ specification. Indeed, the party for which the encryption is intended does not require it as he already possesses it (or can calculate it from the private key x). There are two reasons why we let the public key be part of the ElGamal encryption. The first, and most important, reason is that it allows for easy randomization of ElGamal encryptions (see the third part of Proposition 2.1 below) which is a convenient tool to avoid linkability based

on cryptograms in the e-ID infrastructure. The second reason is that including the public key facilitates easy look up of the required private key of the intended party. For these reasons we let the ElGamal encryption $\mathcal{EG}(S, y, k)$ have the form of the triple $(g^k, S \cdot y^k, y)$. Below we have outlined the homomorphic properties of ElGamal encryption that are the building blocks of our scheme.

Proposition 2.1 *Let $\mathcal{EG}(S, y, k) = (A, B, C)$ be an ElGamal encryption of plaintext S under public key $y = g^x$ and let z be an element of $\mathrm{GF}(q)^*$. Then the following equalities hold:*

1. $(A^z, B^z, C) = \mathcal{EG}(S^z, y, k{\cdot}z)$,
2. $(A^z, B, C^{(z^{-1})}) = \mathcal{EG}(S, y^{(z^{-1})}, k{\cdot}z)$,
3. $(A \cdot g^z, B \cdot C^z, C) = \mathcal{EG}(S, y, k + z)$.

Proof: Easy verification. □

From the first part of Proposition 2.1 it follows that anyone can perform an exponentiation on the plaintext S without knowing the value itself. Moreover, from the second part of Proposition 2.1 it follows that anyone can transform an ElGamal encryption under a public key y to another one of the form y^z with related private key $x{\cdot}z$. Finally, the transformation in the last part of Proposition 2.1 is called the *randomization* of an ElGamal encryption. With this transformation anyone can transform an existing ElGamal encryption, only using the public g and y, into a fresh one holding the same plaintext S but which is not linkable to the original one. This is due to the assumption that the Decision Diffie-Hellman problem is hard in G. This is a commonly known result, compare for instance [9, Theorem 10.20].

3 An eID scheme based on polymorphic pseudonyms

3.1 Idea

Before specifying the PP scheme in detail we describe the idea and relate it to the three use-cases from Section 1. Compare Figure 1 below. A central Key Management Authority (KMA) generates a system-wide public key y_K and distributes this to the schools. Each *participant* (schools, publishers) is also provided with their own public/private key pairs by the KMA. To introduce a pupil in the scheme, its school encrypts the hash of the pupil's PN with y_K. This is called the *polymorphic pseudonym* (PP) and is stored in the pupil administration of the school. Next the school sends the PP of the pupil to another central party called Pseudonym Facility (PF) and requests the *encrypted pseudonyms* (EP) of the pupil at the school itself and for all publishers the pupil needs to interact with. Only the intended party can decrypt the EP to derive the actual pseudonym which is party specific too. In the first use-case from Section 1 the school directs the pupil to the publisher accompanied by the EP of the school itself and that of the publisher. The first EP enables the publisher to send a licence request back to the school linked to the pupil. Indeed the school can decrypt

this and recognize the pupil. The second EP can be decrypted by the publisher leading to the pupil's pseudonym at the publisher. This enables the publisher to recognize the pupil from then on. That is, in further visits of the pupil to the publisher, the school sends along the EP of the publisher. The EP of the pupil at the school also allows the publisher to send test results of the pupil back to the school, similar as how the licence request was sent. This is the third use-case.

The PP and EP forms are "randomizable" meaning that anyone can transform them to equivalent forms that have the same contents inside but are different from the outside and in fact cryptographically unlinkable with the original. See the last part of Proposition 4. This property allows that repeated usage of PPs and EPs cannot be linked by outsiders.

It follows that the nature of the KMA is rather static, i.e. the KMA can be implemented fairly unconnected to the internet. The nature of the PF is more internet connected but does have particularly high availability requirements. For the most part the EPs required for a school can be gathered somewhere at the start of the school year.

3.2 Establishment and operation of the PP eID scheme

As indicated in the previous section, two central parties exist in the PP eID scheme: a Key Management Authority (KMA) and a Pseudonymization Facility (PF). We assume that the KMA and PF do not share (cryptographic key) information, i.e. that there is a Chinese wall between these organisations. The establishment and operation of the PP eID scheme consists of the following steps:

- System setup
- Key Management Authority setup and key distribution
- Setup of the Pseudonymization Facility
- Polymorphic Pseudonym generation by schools
- Transformation of Polymorphic Pseudonyms to Encrypted Pseudonyms by the Pseudonymization Facility
- Decryption of Encrypted Pseudonyms
- Randomisation of Polymorphic and Encrypted Pseudonyms

We will describe these steps in details in the following sections.

3.3 System setup

The parties involved first agree on a security parameter t for the scheme where 2^t operations form the security threshold of the scheme. Then they agree on the specific choices for all primitives explained in Section 1 in line with the security parameter t. That is, they agree on a multiplicative group G, a generating element g for it, a secure hash $\mathcal{I}(.) : \{0,1\}^* \to G$ and a key derivation function $\mathcal{M}(.,.)$.

3.4 Key Management Authority setup and key distribution

The Key Management Authority generates an ElGamal public key $y_K = g^{x_K}$ where $x_K \in_R \mathrm{GF}(q)$ is the associated private key. The public key y_K is provided to all schools in a reliable fashion, e.g. wrapped in a digital certificate associated with the Key Management Authority. Next the Key Management Authority chooses a random KDF key D_K, called the *ElGamal master key*. The ElGamal master key D_K is securely distributed to the Pseudonymization Facility.

Each registered party (Schools, Publishers) is securely associated with a name string $\mathcal{N}$, e.g. through an URL that is included in TLS client certificate. Next, each party is provided an ElGamal public key $y_\mathcal{N} = g^{x_\mathcal{N}}$ where $x_\mathcal{N}$ is the associated private key which is formed as

$$x_\mathcal{N} = \frac{x_K}{\mathcal{M}(D_K, \mathcal{N})}.$$

Note that by this construction the following relation holds between the public ElGamal key $y_\mathcal{N}$ of this party and that of the Key Management Authority:

$$y_\mathcal{N} = y_K^{(\mathcal{M}(D_K,\mathcal{N})^{-1})} \tag{1}$$

The ElGamal public and private key pair are securely distributed to them. For instance, the party involved could collect it by establishing a TLS connection to the Key Management Authority where the party authenticates with a TLS client certificate issued on the name $\mathcal{N}$. To conclude the registration process, each party is required to choose a random *pseudonymization closing key* $c_\mathcal{N} \in \mathrm{GF}(q)$. That is, each party has two secret keys: $x_\mathcal{N}$ shared with the Key Management Authority and $c_\mathcal{N} \in \mathrm{GF}(q)$ that is under sole control of the party.

3.5 Setup of the Pseudonymization Facility

The Pseudonymization Facility chooses a random KDF key D_P, called the *pseudonymization master key*.

With the specification of the pseudonymization master key we have concluded the specification of the cryptographic keys in the PP infrastructure. For convenience we have depicted them in Figure 1 below.

3.6 Polymorphic Pseudonym generation by schools

Let p be the PN of a pupil of a school. The school calculates a Polymorphic Pseudonym for this pupil by first calculating the embedding $\mathcal{I}(p) \in G$ and then encrypting this with the public key y_K of the Key Management Authority. That is, the school picks a $k \in_R \mathrm{GF}(q)$ and forms

$$(g^k, \mathcal{I}(p) \cdot y_K^k, y_K)$$

as the Polymorphic Pseudonym for the pupil.

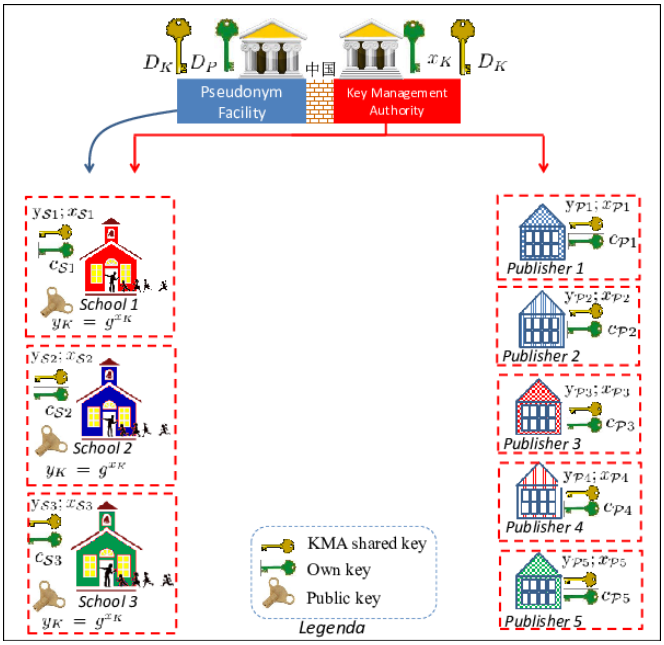

Figure 1. The PP (key) infrastructure

3.7 Transformation of Polymorphic Pseudonyms to Encrypted Pseudonyms by the Pseudonymization Facility

In this context a Polymorphic Pseudonym and a name $\mathcal{N}$ for a party involved is securely sent to the Pseudonymization Facility. The latter is then requested to form an Encrypted Pseudonym for that party. If we denote the Polymorphic Pseudonym by (E_1, E_2, E_3) then the Pseudonymization Facility performs the following three operations. It first forms (F_1, F_2, F_3) by

$$(F_1, F_2, F_3) = (E_1^{\mathcal{M}(D_P,\mathcal{N})}, E_2^{\mathcal{M}(D_P,\mathcal{N})}, E_3). \tag{2}$$

Next the Pseudonymization Facility forms (G_1, G_2, G_3) by

$$(G_1, G_2, G_3) = (F_1^{\mathcal{M}(D_K,\mathcal{N})}, F_2, F_3^{(\mathcal{M}(D_K,\mathcal{N})^{-1})}). \tag{3}$$

Finally, the Pseudonymization Facility chooses $l \in_R \mathrm{GF}(q)$ and transforms (G_1, G_2, G_3) into

$$(I_1, I_2, I_3) = (G_1 \cdot g^l, G_2 \cdot G_3^l, G_3), \tag{4}$$

which is the Encrypted Pseudonym for the party associated with name $\mathcal{N}$. One can easily show that the result of the three operations is equal to

$$(E_1^{\mathcal{M}(D_P,\mathcal{N})\cdot\mathcal{M}(D_K,\mathcal{N})} \cdot g^l, E_2^{\mathcal{M}(D_P,\mathcal{N})} \cdot E_3^{(l\cdot\mathcal{M}(D_K,\mathcal{N})^{-1})}, E_3^{(\mathcal{M}(D_K,\mathcal{N})^{-1})})$$

Proposition 3.1 *In the context above the expression* (I_1, I_2, I_3) *is a random ElGamal encryption under the public key* $y_{\mathcal{N}}$ *of the party associated with name* $\mathcal{N}$ *containing*

$$\mathcal{I}(p)^{\mathcal{M}(D_P,\mathcal{N})}$$

Proof: We first note that the polymorphic pseudonym is formed as an ElGamal encryption of $\mathcal{I}(p)$ for the Key Management Authority, where p is the PN number of the pupil involved. According to the first part of Proposition 2.1, the step in expression (2) changes the plaintext of this ElGamal encryption to $\mathcal{I}(p)^{\mathcal{M}(D_K,\mathcal{N})}$. According to the second part of Proposition 2.1, the step in expression (3) changes the encryption to one under the public key

$$F_3^{(\mathcal{M}(D_K,\mathcal{N})^{-1})} = E_3^{(\mathcal{M}(D_K,\mathcal{N})^{-1})}.$$

By expression (1) this is equal to the public key of the party associated with name $\mathcal{N}$. Finally, it follows from the third part of Proposition 2.1 that the step in expression (4) transforms the ElGamal encryption into a random one. □

3.8 Decryption of Encrypted Pseudonyms

In this context an Encrypted Pseudonym (I_1, I_2, I_3) is received by a party with name $\mathcal{N}$. This party wants to retrieve the pseudonym of the associated pupil in the domain of the party. To this end, the party performs the following operations. First it uses its private ElGamal key $x_{\mathcal{N}}$ to decrypt the Elgamal encryption, i.e. to form

$$J = I_2/I_1^{x_{\mathcal{N}}}. \tag{5}$$

Next it uses its pseudonymization closing key $c_{\mathcal{N}}$ to form

$$K = J^{c_{\mathcal{N}}}.$$

Finally, it takes the secure hash of the latter result, i.e. it forms $\mathcal{H}(K)$. This is the pseudonym of the pupil associated with the original encrypted pseudonym.

Proposition 3.2 *In the context above the pseudonym* $P_{p,\mathcal{N}}$ *of a pupil with PN* p *at a party with the name* $\mathcal{N}$ *is equal to*

$$P_{p,\mathcal{N}} = \mathcal{H}(\mathcal{I}(p)^{\mathcal{M}(D_P,\mathcal{N})\cdot c_{\mathcal{N}}}). \tag{6}$$

Proof: This easily follows from Proposition 3.1. □

3.9 Randomisation of Polymorphic and Encrypted Pseudonyms

In this context a party possesses a polymorphic or encrypted pseudonym and wants to randomize this, i.e. make a fresh copy of it as introduced in the Introduction (Section 1). If we let the polymorphic or encrypted pseudonym be represented by (C_1, C_2, C_3) the party chooses a random $l \in \mathrm{GF}(q)$ and forms

$$(C_1 \cdot g^l, C_2 \cdot C_3^l, C_3).$$

According to the third part of Proposition 2.1 this step results in a random polymorphic or encrypted pseudonym containing the same plaintext. Note that we already used this technique in expression (4).

4 Security and legal compliance

In the proposition below we state and prove the main security properties of the PP setup, all based on well-known security properties of ElGamal encryption. If properly implemented the Dutch government scheme in [13] will only have the first and possibly the last property.

Proposition 4.1 *We assume that group G has the cryptographic properties specified in Section 2, that the Key Management Authority (KMA), Pseudonymization Facility (PF) and schools do not deviate from the protocols and that parties always use fresh copies of PPs and EPs (cf. Section 3.9). Then the polymorphic pseudonymization scheme has the following cryptographic properties.*

1. *Publishers are not able to link their pupil pseudonyms with personal numbers.*
2. *Cooperating publishers are not able to link their pupil pseudonyms.*
3. *Schools are not able to calculate pseudonyms of pupils at other participants, to link them with personal numbers or to link them over participants.*
4. *The PF gets no information on pupil activities and is not able to link pupil pseudonyms with personal numbers or to link them over participants.*
5. *The KMA gets no information on pupil activities and is not able to link pupil pseudonyms with personal numbers or to link them over participants.*
6. *Eavesdropping parties are not able to link pupil information exchanges based on polymorphic or encrypted pseudonyms.*

Proof: We only provide sketches which can be further formalized in the so-called random oracle model [1]. Consider two publishers with names $\mathcal{P}_1, \mathcal{P}_2$. Also compare the form of the pupil pseudonyms in Formula (6). The publishers apply their closing keys $c_{\mathcal{P}_1}, c_{\mathcal{P}_2}$ and the hash calculation in that expression. So the publishers know these values without those operations applied, i.e. the values in Formula (5). For the first property, suppose $\mathcal{P}_1$ knows the pseudonym of a pupil with personal number p_1. Also suppose he has another pseudonym of a pupil with unknown personal number p. For $\mathcal{P}_1$ to decide if this pseudonym belongs to a pupil with personal number p' amounts to decide if

$$(\mathcal{I}(p_1), \mathcal{I}(p'), \mathcal{I}(p_1)^{\mathcal{M}(D_P,\mathcal{P}_1)}, \mathcal{I}(p)^{\mathcal{M}(D_P,\mathcal{P}_1)}) \tag{7}$$

is a DDH quadruple (cf. Section 2). We assumed this problem was intractable. For the second property, suppose $\mathcal{P}_1, \mathcal{P}_2$ know that two of their pseudonyms belong to the same pupil, with (unknown) personal number p_1. To link a $\mathcal{P}_1$ pseudonym of a pupil with (unknown) personal numbers p to a $\mathcal{P}_2$ pseudonym of a pupil with (unknown) personal numbers p' amounts to decide if the quadruple

$$(\mathcal{I}(p_1)^{\mathcal{M}(D_P,\mathcal{P}_1)}, \mathcal{I}(p)^{\mathcal{M}(D_P,\mathcal{P}_1)}, \mathcal{I}(p_1)^{\mathcal{M}(D_P,\mathcal{P}_2)}, \mathcal{I}(p')^{\mathcal{M}(D_P,\mathcal{P}_2)}) \tag{8}$$

is a DDH quadruple (cf. Section 2). We assumed this problem was intractable.

The two unlinkability parts of the third, fourth and fifth property can be shown under the weaker condition that no final hash operation is applied in Formula (6). One can then proceed as in the proofs of the first two properties. In the proof of the fourth property the role of the diversified keys $\mathcal{M}(D_P, \mathcal{N})$, i.e. in Formulae (7) and (8), is then taken by the closing keys $c_{\mathcal{N}}$. In the proof of the third and fifth property this role is taken by the product of the diversified keys and the closing keys. It follows that the two unlinkability parts of the third, fourth and fifth property also follow from the intractability of the DDH problem in G. With respect to the first part of the third property: schools only get access to the encrypted pseudonyms of pupils at a publisher $\mathcal{P}$. The corresponding private key is only known by the publisher. That is, this property follows from the security of the ElGamal encryption scheme, cf. [9, Section 10.5]. With respect to the first part of the fourth property: the PF only sees fresh PPs, i.e. one-time used ElGamal encryptions under the system-wide public key y_K of Personal Number hashes of pupils. As the PF has no access to the corresponding private key, it cannot link the PPs in the various EP requests. This is just the semantic security property of the ElGamal scheme mentioned earlier. With respect to the first part of the fifth property: the KMA has access to the private key corresponding to y_K and is able to decrypt and link the PPs in the various EP requests to the PF. However, we assume that the KMA and the PF are separate entities and that the PF does not share these requests with the KMA. The sixth property follows from the semantic security of ElGamal encryption. □

It follows that the final hash operation in Formula (6) is not necessary for the proof of Proposition 4. This hash operation is provided for extra robustness in the scheme as it effectively removes all G-group structures from pseudonyms. The Dutch data protection authority has issued a ruling [3] related to pseudonymization by a trusted third party. This consists of five requirements. When these are met the resulting pseudonyms are not considered personal data, i.e. are not subject to Dutch privacy laws. Applied to our context the requirements stipulate that one should deploy good practice cryptographic techniques and that the supplier of the data, i.e. the school, should only send a secure hash of the personal number to the PF. The latter part is satisfied a fortiori in our scheme as we send the personal number hash encrypted with the system-wide ElGamal public key. Proposition 4 indicates the first part is met. We have motivated that pseudonyms in our scheme are not considered personal data in the Netherlands.

5 Extensions

In this section we sketch two extensions to the basic polymorphic pseudonym system: Polymorphic Attributes and Central User Inspection Services.

5.1 Polymorphic Attributes

The basic scheme described simply provides for *attribute providers*. These are central parties that possess information (attributes) of a pupil, e.g. date of birth,

address, qualifications etcetera. In a straightforward implementation attributes are associated with the pseudonym of the pupil in the domain of the attribute provider. If a party, e.g a publisher, would require access to some attributes, a school would send an attribute request to the attribute provider accompanied by an encrypted pseudonym of the pupil. The attribute provider then decrypts the pseudonym, looks up the attributes and sends them to the publisher. Typically the request of the school would contain the name of the publisher and an encrypted pseudonym of the pupil at the publisher. The well-known Security Assertion Markup Language (SAML) [16] facilitates such exchange of attributes and also supports attribute encryption under a public key of the publisher.

A compromise of an attribute provider in this setup would result in the loss of large amounts of personal data, cf. the incidents we mentioned in Section 1. Moreover, through attributes that (in)directly identify the pupil the attribute provider can follow the movements of pupils. To remedy this, we can also apply the polymorphism idea to attributes. A party that possesses attributes of the pupil, encrypts those with under a specific ElGamal public key and sends these to an attribute provider accompanied by the pseudonym of the pupil in the attribute provider domain. Similar to the Pseudonymization Facility, the attribute provider does not have access to the private key related to the ElGamal key. However, the attribute provider is able to transform encrypted attributes to a form decryptable by parties in the scheme. For this one can apply the techniques from Sections 3.7,3.8 and 3.9. For robustness it is best to use different groups for polymorphic pseudonyms and attributes. If a pupil authenticates through a school to visit a party requesting an attribute, e.g. a publisher, then:

1. the school requests or validates consent of the pupil for the attribute,
2. the schools sends the attribute provider the request accompanied by encrypted pseudonyms of the pupil at the attribute provider and the publisher
3. the attribute provider decrypts its own encrypted pseudonym and looks up the pupil and its encrypted attribute,
4. the attribute provider transforms the attribute in a form decryptable by the publisher and sends that together with the publisher's encrypted pseudonym to the publisher,
5. the publisher can decrypt both the encrypted pseudonym and the attribute.

Provided attributes are not too long, they can be efficiently bijectively embedded in elliptic curves by using a standard encoding of a string as a number. By proceeding this way, one could in fact perform ElGamal encryptions directly on the encoded attributes. In this way the encrypted attributes could be randomized (cf. the remarks following Proposition 2.1) further improving the privacy properties. For attributes that cannot be coded into a group element one can use hybrid encryption [9, Section 10.4]. That is, one encrypts the attribute A using a symmetric encryption scheme using a random session key. Next one encrypts the session key with the ElGamal public key of the intended party. These two structures then form the cryptogram. By repeating this procedure on the cryptogram one can still randomize the resulting cryptogram to avoid linkability.

However this is less convenient as each randomization will increase the length of the cryptogram with the length of an ElGamal encrypted session key.

We finally remark that ElGamal encryption has very efficient properties with respect to encrypting the same plaintext under different public keys. In [17] it is shown that the same ElGamal randomization exponent can be used without security implications. This allows for efficient sending of attributes to various parties simultaneously.

5.2 Central User Inspection Services

Privacy laws, e.g. [7, Articles 14, 15], give individuals the right to inspect their personal data stored at organizations. Individuals also have the right to inspect what organisations had access to their data. In the context of the scheme in this paper this relates to the parties where the pupil is registered (including schools) and the attributes that have been provided to parties. We show how this inspection requirement can be met with a central user inspection service.

During pupil registration in the scheme, the pupil is also registered at the inspection service. For this the Pseudonymization Facility provides the inspection service with an encrypted pseudonym of the pupil in its domain. This is accompanied with the name of the school and the names of all parties for which an EP was provided to the school. The school is also provided with an EP of the pupil in the inspection service domain. Each time the pupil is authenticated to a party through its school, the school sends a record of this to the inspection service using the pupil's encrypted pseudonym. Records would typically only contain the date, time and the identity of the party visited. This setup can also include the usage of attribute providers. We can additionally require attribute providers to independently send records to the inspection service on each attribute request from a school. For this attribute providers need an encrypted pseudonym of the pupil at the inspection service. In the described setup, the pupil (or its parents) can then logon to the inspection service and review the records. This will for instance allow to discover that a pupil has been registered at schools it does not know about or that attributes were shared with parties without consent.

6 Conclusion

We have introduced an electronic identity system in education based on polymorphic pseudonymization. Here a pupil gets a specific pseudonym at each party required and only that party knows this pseudonym. Moreover, pseudonyms do not form personal numbers are not linkable. In its encrypted form pseudonyms have convenient unlinkability properties. For instance two publishers can send individual test results on a pupil to its school without being able to assess that the pupil is actually the same. We have demonstrated security and motivated that the setup is compliant with the requirements on pseudonymization of the Dutch data protection authority. Our scheme can be further supplemented with privacy friendly attribute and user inspection services. The first allows central

(cloud) parties to store personal data in an encrypted way such that the party itself is not able to access it, but is able to transform it to a form only decryptable for parties having legitimate purposes. The second provides an implementation of the legal right of individuals to inspect their stored data at organizations and their usage. Future work includes testing the practical application of our scheme. Preliminary tests indicate our scheme easily implements in standard federative environments such as Microsoft's ADFS, SimpleSAMLPHP and Shibboleth. Future work also includes application of polymorphic pseudonymization to other sectors, such as the medical sector. There healthcare facilities such as hospitals and doctors take the role of schools and private parties providing paramedical services such as health APPs and fitness clubs take the role of publishers.

7 Acknowledgement

We thank Hans Harmannij, Bart Jacobs and Carlo Meijer for helpful comments and discussions.

References

1. M. Bellare, Mihir, Rogaway, Phillip, Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, ACM Conference on Computer and Communications Security, 1995, pp.6273.
2. CNBC, Experian data breach hits more than 15M T-Mobile customers, applicants, 1 October 2015 Available (December 21 2015) on http://www.cnbc.com.
3. Dutch Data Protection Authority, Pseudonimisering risicoverevening, 6 March 2007. Reference: z2006-1382. Available (December 21 2015) on http://cbpweb.nl.
4. Educause, 7 Things you should know about e-books, 2006. Available (December 21 2015) on http://www.educause.edu/eli
5. T. ElGamal, A Public Key Cryptosystem and a Signature scheme Based on Discrete Logarithms, IEEE Trans. on Information Theory 31(4), 1985, pp. 469-472.
6. IETF, Request for Comments 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard, Curves and Curve Generation, March 2010, see http://www.ietf.org.
7. EUROPEAN COMMISSION, Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), 2012/0011 (COD), 25.1.2012.
8. ISO, ISO/IEC 18033-2:2006 Information technology - Security techniques - Encryption algorithms – Part 2: Asymmetric ciphers, 2006.
9. Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, CRC PRESS, 2008.
10. N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48, 1987, pp. 203209.
11. KrebsonSecurity, Online Cheating Site AshleyMadison Hacked , 19 July 2015. Available (December 21 2015) on http://krebsonsecurity.com.
12. National Institute of Standards and Technology (NIST), Secure Hash Standard (SHS), FIPS 180-4, March 2012. See http://csrc.nist.gov.

13. PBLQ HEC, Privacy Impact Assessment Nummervoorziening in de Leermiddelenketen, version 1.0, 27 May 2015. Available from: https://www.rijksoverheid.nl/.
14. A. Shallue, A., C. van de Woestijne, Construction of rational points on elliptic curves over finite fields, ANTS , LNCS, Volume 4076, Springer, 2006, pp. 510-524.
15. Eric Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, Journal of Cryptology (JOC) 17(4), pp. 277-296, 2004.
16. OASIS, Security Assertion Markup Language, version 2.0. See https://wiki.oasis-open.org.
17. Eric Verheul, Binding ElGamal: A Fraud-Detectable Alternative to Key-Escrow Proposals, Proceedings of Eurocrypt 1996, LNCS 1233, pp. 119-133.
18. Washington Post, Hacks of OPM databases compromised 22.1 million people, federal authorities say, 9 July 2015. Available (December 21 2015) on http://www.washingtonpost.com.