Polymorphic pseudonyms in the education sector

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 22nd November 2015

Abstract We describe a flexible, robust infrastructure giving pupils (and parents) control on personal data exchange in the (public-private) education sector (schools, distributors, publishers). Three generic forms based on homomorphic encryption are used as building blocks. These forms do not form personal numbers, or even personal data from a legal perspective, and have strong, non-traceability properties. Only if required a school provides a party with a party-specific *pseudonym* identifying a pupil. The school is centrally provided the *encrypted pseudonym* based on a *polymorphic pseudonym* formed by the school. Only intended parties, not even schools, have access to pseudonyms. Parties can send pupil test results to a school without being able to assess whether the pupil is the same. The infrastructure can be supplemented with polymorphic attributes and user inspection. The first allows central attribute providers storing personal data in a non-accessible encrypted way. The attribute provider is only able to transform it to a decryptable form for required parties. The second is an implementation of the legal right of individuals to inspect their stored data at organizations and their usage.

Keywords: homomorphic encryption, pseudonyms, privacy enhanced technology

1 Introduction

In this document we cryptographically describe Privacy Enhanced Chain pseudonyms. These are envisioned to be used in an education sector. In this context the end users are pupils of a school. In the education sector, pupils are also known under unique personal number called PN commonly used by all schools. That is, if a pupil would move to another school he would be known there under the same PN. Pupils are able to authenticate themselves to the education portal of the school, e.g. with a user-id / password. The school is then able to retrieve information on the pupil, including his PN, from its administration. The school also uses private parties to provide electronic education and support to its pupils. Prominent examples of private parties are publishers of educational content and distributors of such content. Before a pupil can use the educational content at a publisher its needs to be ordered (and paid for by the school or

someone else). To this end, after authentication at the school portal, a pupil is redirected to a distributor that works with the school. The distributor facilitates the pupil to order educational content at the publishers. This leads to licensing requests at the publishers for the pupil's school. After these requests are successfully handled, the school is able to prepare weblinks for the pupils (typically containing licence information) in the school portal. After the pupil has clicked on such a weblink he is directed to the publisher and he can use the educational content. In many cases the publisher needs to provide (individual) feedback to the school on the results on the pupil.

In such a context it is essential that pupils can be recognized at the various parties in a coherent way. To indicate, the distributor wants to link the earlier orders of the pupil to be able to direct him to publishers with consistent content. The distributor might also be provided some educational information, e.g. the group the pupil resides in or attention points on the pupil. The school needs to be able to identify the pupil from the licence request from the publisher. Likewise, if the publisher provides individual feedback to the school, the school needs to be able to find out to which pupil this relates to. Finally, when a pupil moves from one school to another one, it is convenient that the identity of the pupil at distributors and publishers is unchanged as otherwise valuable historic information on the pupil might be lost.

In the design [1] it is suggested to divide the education sector into subsectors. One might have three subsectors: the primary education sector (based on primary schools), the secondary education sector (based on secondary schools) and the vocational education sector (based on vocational schools). In each of these subsectors Distributors and Publishers operate and some operate in several subsectors. Throughout each of the three subsectors a pupil is known under a *chain pseudonym* based on the PN of the pupil. Although this setup is practical the chain pseudonym introduces a new personal number throughout the whole subsector. Actually in this light, the term *subsector pseudonym* or even *subsector personal number* would be more appropriate. European privacy laws stipulate that for the introduction of such subsector wide used numbers specific and prior approval is necessary from the national data protection authority. This also indicates that the usage of such chain pseudonyms is not good privacy practice. Specifically, this setup is not in line with the data minimisation principle as stipulated in Article 5 in the draft European privacy regulation [8]: "Personal data must be adequate, relevant, and limited to the minimum necessary in relation to the purposes for which they are processed; they shall only be processed if, and as long as, the purposes could not be fulfilled by processing information that does not involve personal data".

The term pseudonym is also reminiscent of the term used by the Dutch Data Protection Authority (DPA) in a ruling [5] on pseudonymization. See also Section 4. This ruling states five conditions under which pseudonymised data are not considered personal data by the DPA, i.e. falling under Dutch privacy laws. These conditions impose that pseudonyms should be generated in accordance with good cryptographic practice and should not be indirectly identifiable. In the suggested

setup, however, this condition is not met as all subsector parties share the *same* pseudonym of a pupil. This facilitates that parties work together to identify a pupil and combine data. More worrisome is when parties (schools, distributors and 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. As more or less random recent related incidents we mention the hack of the US Office of Personnel Management [19] leading to compromise of over 22 million records, the hack of T-Mobile data [4] leading to the compromise of over 15 million records and the hack of Ashley Madison [12] leading to the compromise of over 37 million records. In the latter case, compromised data was published after Ashley Madison did not comply with the blackmail conditions, which allegedly resulted in suicides. See [2].

A division into (three) subsectors is meant as a rudimentary privacy control but in fact also hampers the objective of the design: necessary exchange in the education sector. Indeed, if a pupil moves from one school type to another one, the pupil's data cannot be linked even when it is required, e.g. in the case of continuous testing for dyslexia.

The design document [1] also mentions the possibility of pseudonyms that are unique for each party. In this document we introduce such a variant. It constitutes of a privacy enhanced version of the chain pseudonyms which strongly conforms to the data minimisation principle. Necessary linking of pupils is possible, even between different school types. But unnecessary linking is precluded by default. This forms the basis for a robust data exchange infrastructure with intrinsic resilience against attack of participants. Such privacy enhanced chain pseudonyms (hereafter: PEC pseudonyms) do not introduce new (sub)sector wide personal numbers while still facilitating the functionality described above. Like the original chain pseudonyms introduced above, PEC pseudonyms are also based on the pupil's PN. However a pupil is provided different pseudonyms at the education parties which are cryptographically not relatable. An additional property of PEC pseudonyms is that only the party for which they are meant has access to them: even the pupil's school does not know the pseudonyms of the pupil at other parties. To support this, the PEC infrastructure introduces a notion of Encrypted Pseudonym (EP). All parties in the PEC infrastructure are provided a public / private key pair from a central party called Key Management Authority (KMA). This key pair (of type ElGamal [6]) allows a party to decrypt an encrypted pseudonym leading to the actual pseudonym. EPs also allows a party to refer to a pupil at another party without knowing its pseudonym there. Moreover, EPs can be randomized by anyone, leading to *fresh* copies of EPs that are not relatable to the original. This allows for flexible functionality. To indicate, if a publisher has the pupil's EP at his school, the publisher can provide the school individual feedback by simply including a fresh copy of the pupil's EP to the feedback. Likewise, if the school has pupil's EP at a publisher the school can allow the pupil to login at the publisher in a federated fashion by sending along a fresh copy of this EP.

EPs are formed by another central provider next to the KMA, called *Pseudonymization Facility*. Compare Figure 1 in Section 3.3. This role is quite similar in role to the original Numberfacility (Nummervoorziening in Dutch) introduced in [1]. In [1] a school sends the PN of a pupil to the Numberfacility which then returns the pupil's chain pseudonym. This means that the Numberfacility is processing personal data. In the PEC infrastructure, however, the Pseudonymization Facility is not required to have access to PGNs. Instead the school sends the Pseudonymization Facility an encrypted hash of the PN called *polymorphic pseudonym* accompanied with a reference to a party in the education sector. The Pseudonymization Facility then transforms this into the earlier discussed encrypted pseudonym for this party. As the polymorphic pseudonym can also be randomized, the Pseudonymization Facility cannot even access that an encrypted pseudonym is requested for the same pupil. As the name polymorphic pseudonym indicates, the cryptographic technique that underlies PEC are based on the technique of polymorphic pseudonymization from [14]. In [14] the polymorphic pseudonyms are created by a central party and the transformation to encrypted pseudonyms is performed by the identity providers themselves. In PEC this is just the other way around: the polymorphic pseudonyms are created by the schools (acting as identity providers) and the transformation to encrypted pseudonyms is done by the central party (Pseudonymization Facility). Another difference with [14] is that PEC does not have support for law enforcement agencies (reversing pseudonyms to actual identities). This allows for further simplification of the techniques from [14].

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 the PEC infrastructure, including the system setup, the setup of the KMA, the role of the Pseudonymization Facility and the protocols leading to encrypted pseudonyms, randomization of those and the transformation to pseudonyms. Section 4 discusses security and legal compliance with privacy regulations. In Section 5 we discuss two supplements to the scheme. 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. Section 6 contains conclusions.

2 Notation and preliminaries

Throughout this paper we let $\mathcal{H}(.)$ represent a secure hash function, e.g. the SHA256 hash function as specified in [13]. 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 $x \in \mathrm{GF}(q)$ 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 [16] 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 generator of the group. 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. We assume that all four introduced problems in G are intractable.

For practical implementations one can think of G being a group of points on an elliptic curve such as brainpoolP320r1, including the standard generator from [7]. Here the size of q is 320 bits. Throughout this paper we will let $\mathcal{M}(K, \mathrm{string})$ represent a key derivation function (KDF) that maps a string into secret key in $\mathrm{GF}(q)^*$. One can think of the KDF functions from [9] but also of a HMAC based function modulo q where HMAC is specified in [3]. 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. [11], 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 can be found in [15].

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 [6] 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 $y = y^z$ 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 Theorem 10.20 of [10].

3 PEC Scheme Description

The establishment and operation of the PEC 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.

3.1 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 $\mathcal{M}(.,.)$.

3.2 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, Distributors, 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 and 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 PEC 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.

3.3 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 PEC infrastructure. For convenience we have depicted them in Figure 1 below.

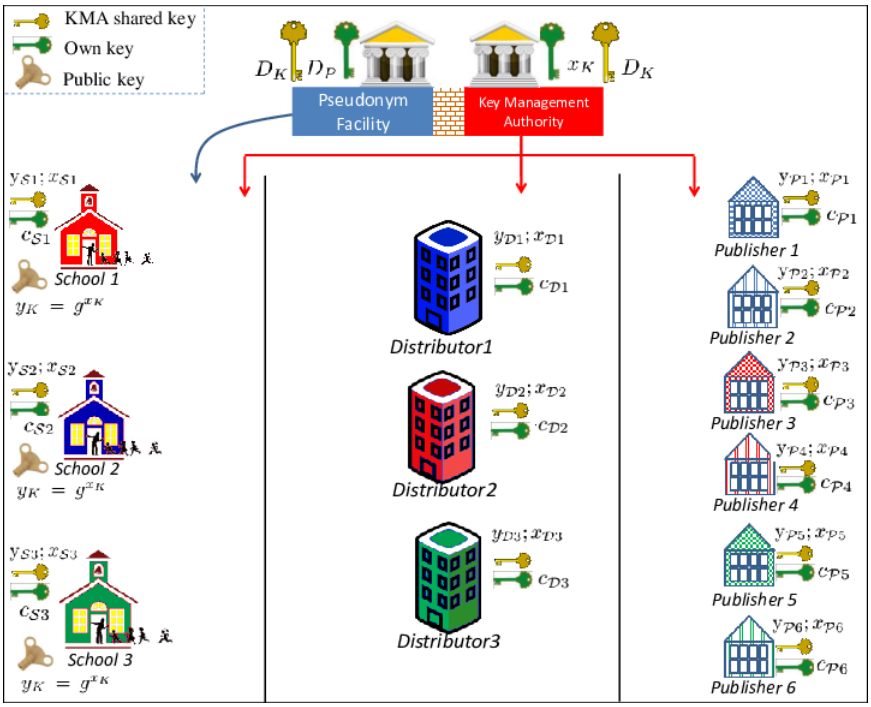

Figure 1. PEC (key) infrastructure

3.4 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.

Note: this means that the hashed PN can be decrypted from the Polymorphic Pseudonym by the Key Management Authority. However the Polymorphic Pseudonym is only sent to the Pseudonymization Facility that does not possess the private key x_K.

3.5 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. □

3.6 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. □

3.7 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

Below we formulate and substantiate the main privacy and security properties of the PEC setup. We start by recalling a ruling [5] of the Dutch data protection authority on pseudonymization consisting of five requirements to be met. If these requirements are met, the pseudonymized personal data is no longer considered to be personal data, i.e. to be within the scope of the Dutch privacy laws. These five requirements are (translated into English):

a. One is deploying pseudonymization using good practice cryptographic techniques whereby the first encryption occurs at the supplier of the data.

b. Technical and organization controls have been taken to prevent repeatability of the encryption (replay back).
c. The processed data are not indirectly identifiable.
d. In an independent assessment (audit) it is determined that conditions a., b. and c. are met prior to the commencement of the processing and periodically after that;
e. The pseudonymization solution shall be documented in a clear and complete way in a public document enabling parties involved to assess the guarantees the chosen solution provides.

We now come to the formulation and substantiation of the main privacy and security properties of the PEC setup.

1. The PEC setup is technically compliant with the CBP ruling on pseudonymization. PEC pseudonyms are not personal data on their own.
The final pseudonyms are formed as keyed hashes of PGNs as indicated in Formula (6). Moreover the supplier of the data (school) provides the data (PN) in a form that is both hashed and encrypted. That is, the central pseudonym facility is not able to retrieve the PGNs by mounting a brute-force attack on the PN. We note that in the current (Dutch) practice surrounding the CBP ruling, the central facility is sent plain hashes of the personal numbers and is thus able to mount a brute-force attack on the PN. This actually is the basis for the second CBP requirement. This is usually adhered to by using organizational controls whereas in PEC we have also adhered to it by cryptographic means which can be considered stronger.

As the Key Management Authority possess the private part x_K of the public key used in the polymorphic pseudonyms, he is technically able to decrypt them and to retrieve the hash of the PN that is inside. Of course, the Key Management Authority is not supposed to be in possession of polymorphic pseudonyms, let alone decrypt them. Compared with current (Dutch) practice the risks related to this vulnerability seem acceptable. Further mitigation of this risk can be achieved by deploying an Hardware Security Module (HSM) at the Key Management Authority restricting the usage of x_K to only secured key distribution to the parties involved. See also the remark at the third claim.

2. The PEC pseudonyms do not form personal numbers in the education domain.
No PEC party is independently able to deduce the PN from a PEC pseudonym. This is due to the form of the PEC pseudonyms indicated in Formula (6). That is, a PEC pseudonym is the result of a keyed hash function where the keys used are shared over three parties, namely the Key Management Authority, the Pseudonymization Facility and the party the pseudonym belongs to. Without access to these keys one cannot mount a brute-force attack on the PN. That is, even the Key Management Authority and the Pseudonymization Facility together cannot mount such an attack.

A PEC pseudonym, cf. Formula (6), is finally formed by the party for which it is meant as a secure hash. This will effectively disable any technique to transform a pseudonym from one party domain to another.

3. Only the PEC party itself has access to the pupil's pseudonym in his domain.
Parties for which the pseudonym is not meant can only have access to encrypted pseudonyms. Such as for instance a publisher that gets hold of an encrypted pseudonym of a pupil in the domain of its school. Without the related private keys such a party is not able to decrypt the pseudonym from the encrypted pseudonym. Also the schools are not able to retrieve the pupil's pseudonym at a PEC party. They are each involved in the creation of an encrypted pseudonym but do not possess the private key of the party involved.

We do not consider the Pseudonymization Facility and the Key Management Authority as regular PEC parties. For the sake of completeness we observe that neither of these entities is independently able to calculate a pseudonym from a PN as they lack a cryptographic key. The Pseudonymization Facility lacks the closing master key and the Key Management Authority lacks the pseudonymization master key. The only vulnerability is that the Key Management Authority can decrypt an intercepted encypted pseudonym. As the Key Management Authority has access to the ElGamal decryption key, this will allow him to retrieve the form indicated in formula (5). However, he lacks the party's closing key to deduce the pseudonym from this. Such a (misbehaving) Key Management Authority is able to assess if two encrypted pseudonyms belong to the same person. The Key Management Authority is also able to see that two polymorphic pseudonyms correspond to the same person.

However, the risks related to these vulnerabilities seem acceptable given the trusted role of the Key Management Authority in the scheme. We also note that in current (and accepted) practice the central pseudonymization provider actually calculates (and thus knows) the pseudonyms at all parties. Further mitigation of this risk can be achieved by deploying an Hardware Security Module (HSM) at the Key Management Authority. In Section 3.2 we suggested to let the PEC parties authenticate themselves to the Key Management Authority using a TLS client certificate during the key distribution protocol. With the HSM one could go one step further: the HSM will only allow the Key Management Authority to export the ElGamal keys of the parties involved under the public key included in a trusted certificate issued to the party.

4. Polymorphic and encrypted pseudonyms are not traceable
This is explained in Section 3.7.

5. The pseudonymization solution shall be documented in a clear and complete way in a public document enabling parties involved to assess the guarantees the chosen solution provides.
This document would be a first implementation of this requirement but would need to further supplemented with documents describing security management.

5 Extensions

In this section we sketch two extensions to the basic polymorphic pseudonym system:

- Polymorphic Attributes,
- 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) on 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 with 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 publisher. The well-known Security Assertion Markup Language (SAML) [17] 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 has attributes of the pupil, encrypts those with under a specific ElGamal public key and sends these to an attribute provider accompanied with 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 3.5,3.6 and 3.7.

That is, if the 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 with encrypted pseudonyms of the pupil at the attribute provider and the publisher
3. the attribute provider decrypts its encrypted pseudonym and looks up the pupil's encrypted attribute,
4. 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 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. For instance, one can reserve some room, say one byte, in the string supporting that the whole encoded string has an x-coordinate and a matching y-coordinate on the curve. By proceeding in 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. We finally remark that ElGamal encryption has very efficient properties with respect to encrypting the same plaintext under different public keys. In [18] is shown that the same ElGamal randomization exponent can be used without security implications.

5.2 Central User Inspection Services

Privacy laws, e.g. [8, Articles 14, 15], give individuals the right to inspect their personal data stored at organizations. Individuals moreover have the right to inspect what organisations had access to this data. In the context of the scheme described in this paper this particulary relates to parties where the pupil is registered (including schools) and the attributes that have been provided to parties. This inspection requirement can be met with a central user inspection service in the described scheme. As part of the pupil registration at a school, the pupil is also registered at the inspection service. To this end the school provides an encrypted pseudonym of the user in the inspection service domain, accompanied with the school name. Moreover, each time the pupil is authenticated to a party though its school, the school sends a record to the registration service and the pupil's encrypted pseudonym. Records would typically only contain the date, time and the identity of the party visited. This setup also includes authentication for attribute providers. However, we can additionally require attribute providers to independently send records to the registration service on each attribute request from a school. For this attribute providers need an encrypted pseudonym of the pupil at the registration 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 without consent.

6 Conclusion

In this paper we have described Privacy Enhanced Chain pseudonyms which are envisioned to be used in an education sector. In the setup each pupil gets a specific pseudonym at each party involved and only that party knows this pseudonym. Moreover, the pseudonyms are not inter linkable and do not form personal numbers in the education domain. In its encrypted forms the pseudonyms have convenient non-traceability properties. For instance two education parties can give individual test results on a pupil to its school without being able to assess that the pupil is actually the same person. We have motivated that the setup is compliant with the technical requirements on pseudonymization imposed by the Dutch data protection authority. The present setup can be further supplemented with polymorphic attributes 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. The concepts developed in this paper can also be applied to other sectors, such as the medical sector. There healthcare facilities such as hospitals and general practitioners would take the role of the schools and (commercial) parties providing paramedical services such as fitness clubs, trainers would take the role of the publishers.

References

1. Project Ontwerpfase Centrale Nummervoorziening, https://www.edustandaard.nl/fileadmin/edustandaard/user_upload/KAT_Edustd_architectuurraad_20150114.pptx.
2. BBC News, Ashley Madison: 'Suicides' over website hack, 24 August 2015. Available (October 10 2015) on: http://www.bbc.com/news/technology-34044506.
3. M. Bellare, R. Canetti, and H. Krawczyk, Keyed Hash Functions and Message Authentication, Proceedings of Crypto'96, LNCS 1109, pp. 1-15.
4. CNBC, Experian data breach hits more than 15M T-Mobile customers, applicants, 1 October 2015 Available (October 10 2015) on: http://www.cnbc.com/2015/10/01/experian-reports-data-breach-involving-info-for-more-than-15m-t-mobile-customers.html.
5. Dutch Data Protection Authority, Pseudonimisering risicoverevening, 6 March 2007. Reference: z2006-1382. Available (October 10 2015) on: http://cbpweb.nl.
6. T. ElGamal, A Public Key Cryptosystem and a Signature scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory 31(4), 1985, pp. 469-472.
7. IETF, Request for Comments 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard, Curves and Curve Generation, March 2010, see http://www.ietf.org.
8. 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.
9. ISO, ISO/IEC 18033-2:2006 Information technology - Security techniques - Encryption algorithms – Part 2: Asymmetric ciphers, 2006.
10. Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography, CRC PRESS, 2008.
11. N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation 48, 1987, pp. 203209.
12. KrebsonSecurity, Online Cheating Site AshleyMadison Hacked , 19 July 2015. Available (October 10 2015) on: http://krebsonsecurity.com/2015/07/online-cheating-site-ashleymadison-hacked/.
13. National Institute of Standards and Technology (NIST), Secure Hash Standard (SHS), FIPS 180-4, March 2012. See http://csrc.nist.gov.
14. Programma eID, Polymorphic Pseudonymization, versie 0.91, 7 juli 2014. Retrievable from http://www.eid-stelsel.nl/documentatie/werkgroepen/.
15. A. Shallue, A., C. van de Woestijne, Construction of rational points on elliptic curves over finite fields, ANTS , Lecture Notes in Computer Science, Volume 4076, Springer, 2006, pp. 510-524.

16. Eric Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, Journal of Cryptology (JOC) 17(4), pp. 277-296, 2004.
17. OASIS, Security Assertion Markup Language, version 2.0. See https://wiki.oasis-open.org.
18. Eric Verheul, Binding ElGamal: A Fraud-Detectable Alternative to Key-Escrow Proposals, Proceedings of Eurocrypt 1996, LNCS 1233, pp. 119-133.
19. Washington Post, Hacks of OPM databases compromised 22.1 million people, federal authorities say, 9 July 2015. Available (October 10 2015) on: http://www.washingtonpost.com/blogs/federal-eye/wp/2015/07/09/hack-of-security-clearance-system-affected-21-5-million-people-federal-authorities-say/.