SERVICE AND TIMEFRAME DEPENDENT UNLINKABLE ONE-TIME PSEUDONYMS

Kristof Verslype and Bart De Decker
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Heverlee, Belgium

Keywords: Anonymity, Privacy, Security, Pseudonym, Unlinkability, Anonymous credentials.

Abstract: A solution is presented to allow a service provider to limit the number of times per timeframe that a user can access each single service, while maintaining complete unlinkability of different visits by that user. Since the solution is built upon existing building blocks such as anonymous credentials, it is extremely flexible.

1 INTRODUCTION

Privacy on the Internet is gaining importance since the user does not have control over the released personal data and since digital data can be processed and spread very easily. Moreover, the certificates that are used in practice contain several attributes, such as a user identifier. Combined with collected transactional data, this allows the service provider to compose an extended profile linked to the user's identity.

Anonymous credentials try to overcome these shortcomings. They allow for the selective disclosure of properties of credential attributes. For instance "$age > 20$" could be proven, while the user's zip code and date of birth remain hidden although they are included in the credential. Secondly, anonymous credential systems can offer unlinkability of different credential usages, making different visits of a user to a service provider unlinkable.

During issuance, a global show limit can be set on these credentials, which limits the number of times the user can use his credential and this limit is independent of the service that is used.

More versatile limits have been proposed, allowing a service provider to set an access limit towards a specific service or allowing for a global show limit per timeframe. This paper combines both approaches and guarantees unlinkability of accesses by the same user. The service provider can dynamically set the number of times the owner of a valid credential can access a service. This limit can optionally apply to a single timeframe. Both the access limit and the timeframe duration can be chosen and dynamically adjusted by the service provider. Additional restrictions might apply depending on the user's properties, thus further limiting the user's access rights.

For instance, users with a 'golden membership' credential can access a specific service an unlimited number of times, and 'regular members' only 50 times a month. People with a credential without membership privileges can access the service only a very limited number of times, depending on the personal properties they disclose. If they prove being younger than 18, they can access the service once a month, but only in weekends and if they are older than 18 and disclose their exact age, they can access it twice a week.

Section 2 gives an overview of the related work. Section 3 describes the requirements, used notations and assumptions. The required building blocks are presented in section 4. A new building block, the extended provable nym generator, is presented in section 5. Those blocks are combined in our solution in section 6. We conclude in section 7.

2 RELATED WORK

Solutions to the problem of restricting the number of times a user can access a service provided by a service provider while the different accesses by the same user are unlinkable have been presented. Two of these solutions (Teranishi et al., 2004), (Nguyen and Safavinaini, 2005) are based on group signatures. As a consequence, the user can only prove being a member of the group. Another system (Camenisch et al., 2006b) was specifically developed for e-Cash and therefore, did not need personal attribute disclosure properties.

Verslype K. and De Decker B. (2009).
SERVICE AND TIMEFRAME DEPENDENT UNLINKABLE ONE-TIME PSEUDONYMS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 13-20
DOI: 10.5220/0002231100130020

Those systems lack the flexibility provided by anonymous credential systems. Solutions to limit the number of times a token can be used in a single time interval were proposed (Damgard et al., 2006; Camenisch et al., 2006a).

This paper combines both approaches in a flexible way as explained in the introduction and builds upon anonymous credential systems. Most of the previous schemes allow for deanonymization in case of abuse. This is omitted in the presented solution since the underlying anonymous credential system already provides this functionality.

3 REQUIREMENTS, NOTATION AND ASSUMPTIONS

The requirements are now summed up.

Dynamic Service Access Restriction. The service provider can limit the number of times a user can access each service per timeframe. The limits and timeframe size can dynamically change and can depend upon the disclosed data during the credential show.
User Anonymity. A service access by a user cannot be linked to (1) an access to another service, (2) an access to the same service in the same or (3) a different timeframe or (4) to the user's identity.
Flexibility. The flexibility of anonymous credentials must be preserved.
Dishonest entities are represented with a tilde (e.g. $\tilde{P}$). All operations are either in a subgroup of G_p, where G_p has order p, or in $\mathbb{Z}_q$. p and q are prime. The notation $P_1 \leftrightarrows P_2 : (y_1; y_2) \leftarrow \text{protocol}\,(x_0; x_1; x_2)$ is used throughout the paper and represents a protocol run between P_1 and P_2. x_1 and x_2 are the inputs only known by P_1 and P_2 respectively. x_0 is the input known by both P_1 and P_2 and provided by P_1, P_2 or both. y_1 defines the output for P_1, y_2 for P_2.

Two number theoretic assumptions are relevant:
Discrete log (DL) Assumption. Let g be a generator for a finite cyclic group G, let $x \in \mathbb{Z}_q$ and $y \leftarrow g^x$. Finding x when only g and y are known is intractable.
Representation (R) Assumption (Brands, 2000). Finding a representation w.r.t $(g_1, g_2, \ldots, g_m)$ in group G for a given value y is as difficult as solving the DL problem.

4 BUILDING BLOCKS

This section elaborates on existing cryptographic building blocks.

4.1 Proofs of Knowledge

A proof of knowledge (PK) (Bellare and Goldreich, 1992) is a protocol in which a prover P convinces a verifier V that it possesses a secret. More formally, the following relation is defined; $R = \{(y,x) : y \in L, x \in W(y)\}$, where L is a language in NP, x the secret, $W(y)$ the set of possible witnesses for a public value y that should be accepted in the proof and a PK must have the following properties:
Completeness. If $(y,x) \in R$ then $Pr[P(y,x) \leftrightarrows V(y) \rightarrow accept] = 1$.
Validity. There exist a polynomial-time extractor K having oracle access to a potentially cheating prover $\tilde{P}$. K's success probability in extracting x is at least as high as the probability that $\tilde{P}$ succeeds in convincing the verifier V. More formally; $Pr[K(y)^{\tilde{P}(y)} \in R(y)] \geqslant Pr[\tilde{P}(y) \leftrightarrows V(y) \rightarrow accept] - \kappa(y)$, where $\kappa(y)$ denotes the probability that verifier V might accept y, even though prover $\tilde{P}$ does not know a witness x.

A zero-knowledge proof of knowledge (ZKPK) additionally has the following property:
Zero-knowledge. No other data is revealed by P. Formally, there exists a simulator S such that the following two probability ensembles are indistinguishable: $\{P(y,x) \leftrightarrows \tilde{V}(y) \rightarrow .\}_{y \in L}$ and $\{S(y) \rightarrow .\}_{y \in L}$.

The notation of a PK is:

$$P \rightarrow V : (\varnothing, proof) \leftarrow \\ \mathsf{PK}\{(x_1, \ldots, x_m) : properties(x_1, \ldots, x_m)\}.$$

where $x_1, \ldots, x_m$ are the hidden values about which properties are proved. The following properties that can be proven are relevant for this paper: (1) knowledge of a discrete logarithm modulo a prime (Schnorr, 1991), (2) a committed value lies in an integer interval[1] (Boudot, 2000), (3) conjunctions and disjunction of the previous (Cramer et al., 1994). (4) More generally, proving knowledge of representation and equality of secrets is possible, i.e. some secrets used in different operands of a conjunction can be equal. This is denoted as $PK\{(x_1, \ldots, x_u) : \wedge_{i=1}^{n} \Pi_{j \in J_i} g_j^{x_{e_{ij}}} = y_i\}$ where $x_1, \ldots, x_u$ are the different secrets, where n is the number of (conjunctive) operands, $J_i \subseteq \{1, \ldots, l\}$ with l the number of bases and e_{ij} is the index of the secret used in y_i w.r.t base g_j. For such proof of representations and equality of secrets, the following

[1] The scheme was based on composite moduli, while this paper uses DL based commitments. However, the same is possible using the DL commitments. What is actually done to prove that $a < x < b$ is proving that $x - a$ and $b - x$ are positive, which is done by proving that $x - a$ and $b - x$ are both the sum of four squares.

properties hold: (a) They are sigma protocols; they have a commitment[2]-challenge-response structure. (b) For each operand, one multi-base exponentiation by P is required, and one by V. (c) For each operand, one commitment (to a random value) is sent by P. (d) Per proof, one challenge is sent by V. (e) For each secret, one response is sent by P. If a generator g of order q is used, which generates a subgroup in G_p, with p and q prime, a commitment has size $|q|$, a challenge and a response have size $|p|$.

4.2 Commitment Schemes

A commitment scheme (Pedersen, 1992; Damgard et al., 1996) allows an entity to commit to a set of values, while keeping these secret. The commitment hides the values towards the verifier, but allows the creator to prove properties of the committed values. The following protocols are relevant in this paper:

- $P : (C, O) \leftarrow \mathsf{commit}(x_1, \ldots, x_m)$. A commitment to one ore more attributes x_i is created. A secret random value t of sufficient length is chosen in order to make two commitments to the same values unlinkable. The opening info $O = (x_1, \ldots, x_m, t)$ is required in the protocol below.
- $P \rightarrow V : (\varnothing, proof) \leftarrow \mathsf{PK}\{(O) : properties(C)\}$. P proves properties of values committed in commitment C to V in a ZKPK.

A commitment has the following properties:
Hiding. The commitment hides all information about the committed values.
Binding. The values are fixed when the commitment is created and cannot be changed afterwards.
Semantically Secure. Even if the committed values are known and equal, an attacker cannot distinguish between two commitments to the same values.

A commitment to m values $x_1, \ldots x_m$ with randomizer t can be implemented as $y \leftarrow g_1^{x_1} g_2^{x_2} \ldots g_m^{x_m} h^t$ with the group description and all bases g_i and h publicly known. Commitment schemes based on the DL assumption or the RSA assumption exist.

4.3 Joint Secure Random Number Generation

The joint secure random number generation (Camenisch and Lysyanskaya, 2001) between U and O results in a value only known to U, but O is reassured that it is random and of sufficient length. The relevant method and protocol for this paper are:

[2]In this subsection, the term 'commitment' is used in a broad sense. In the sequel of this paper, the term commitment refers to what is discussed in the next subsection.

- $CA : S \leftarrow \mathsf{genRandGenParams}()$. Generates the basic input parameters for the protocol below.
- $U \leftrightarrows O : (r, t, C; C) \leftarrow \mathsf{agreeSecureRand}(S; \varnothing; \varnothing)$. Agree on a secure random value r, which will be committed in C. t is the randomizing factor for the commitment. (r, t) is U's opening info.

4.4 Anonymous Credentials

Anonymous credential systems (Chaum, 1985; Camenisch and Lysyanskaya, 2001; Camenisch and Herreweghen, 2002; Brands, 1999) allow for anonymous yet accountable transactions between users and organizations. They allow for *selective disclosure*; the user is able to reveal a limited set of properties about the attributes embedded in the credential. In the sequel of this paper, *Idemix* credentials are assumed. Multiple Idemix credential shows are unlinkable if no uniquely identifying attribute data are revealed.

The relevant simplified protocols that apply to anonymous credentials are:

- $U \leftrightarrows I : (cred; \varnothing) \leftarrow \mathsf{issueCred}(cert_I, coms, atts; opens; SK_I)$. I issues to U a credential with attributes *atts*. Additionally, values committed by U (*coms*) can be included as attributes in the credential. This requires the corresponding opening info (*opens*).
- $U \leftrightarrows V : (\varnothing; proof) \leftarrow \mathsf{showCred}(coms, props; cred, opens; \varnothing)\{msg\}$. U proves to V the possession of a valid credential *cred*. U can selectively disclose credential attributes or properties thereof (described in *props*). These properties can involve a set of committed values (*coms*) with corresponding opening info (*opens*). U may decide to sign a message *msg* with his credential, creating a provable link between the proof and the message.

Additionally, proofs resulting from a showCred() protocol can be deanonymizable by a predetermined trusted third party. Anonymous credentials can further be issued to be shown only once, a predetermined number of times or an unlimited number of times in total, independent of the services that are contacted. However, these features are not used in this paper.

5 AN EXTENDED PROVABLE NYM GENERATOR (EPNG)

5.1 Definition

An extended provable nym generator (EPNG) is a function $nym \leftarrow f(id, i, tf)$ fulfilling the five proper-

ties listed below. In the next section, each service has its own EPNG, $id \in \mathbb{Z}_q$ is a secret user identifier, $tf \in \mathbb{Z}_q$ the current timeframe and $i \in \mathbb{Z}_q$ is a value, between one and the access limit, that has not yet been used during timeframe tf by that user. Output nym is a one-time pseudonym computed by the user and sent to the service provider.

One-wayness. It is infeasible to derive from a specific nym and EPNG f the corresponding id, i or tf.

Deterministic. Applying the same EPNG multiple times on the same (id,i,tf) tuple results in the same output nym.

Semi-collision Free. (1) For the same id and f, it is infeasible to find an (i,tf) and (i',tf') with $(i,tf) \neq (i',tf')$ such that $f(id,i,tf) = f(id,i',tf')$. (2) For two randomly chosen id and id' and for k and l small compared to id and id', it is infeasible to find $i,i' \in [1,k]$, $tf,tf' \in [1,l]$ such that $f(id,i,tf) = f(id',i',tf)$.

Unlinkability. (1) $nym_1 = f(id,i,tf)$ and $nym_2 = f(id,i',tf')$ with $(i,tf) \neq (i',tf')$ are unlinkable. (2) $nym_1 = f_1(id,i,tf)$ and $nym_2 = f_2(id,i,tf)$ where $f_1 \neq f_2$ are unlinkable.

Provability. If $nym = f(id,i,tf)$, id, i and tf are known, one can prove (1) that nym is well formed (i.e, the result of applying the EPNG on id, i and tf) and (2) properties about id, i and tf.

Two functions/protocols are important.

- $f \leftarrow \mathsf{genEPNG}()$. The generation of an EPNG.
- $P \leftrightarrows V : (com_{id}, com_{i,tf}, open_{id}, open_{i,tf}; com_{id}, com_{i,\text{tf}}, proof) \leftarrow \mathsf{proveEPNG}(f, nym; id, i, tf; \varnothing)$. P generates commitments to id (com_{id}) and i and tf ($com_{i,tf}$), and convinces V that these committed values were used as input for EPNG f to generate nym. This enables P to prove properties about these input values using the corresponding opening info ($open_{id}, open_{i,tf}$). $Proof$ is a proof for V that the commitments contain the proper values w.r.t. nym.

5.2 A Simple Provable Nym Generator

The Provable Nym Generator (PNG) is an EPNG without the timeframe parameter. A concrete PNG can be defined as

$$nym \leftarrow g^{1/(id+i)} \tag{1}$$

in a group G_p of prime order p, where g generates a subgroup of prime order q. This concrete PNG is shown in table 1. Its security is based on the DL assumption. The security of the first four properties can easily be derived from the proofs given in the next subsection, where a concrete EPNG is given. Since the provePNG() implementation approach is different

Table 1: The provePNG() protocol for $nym \leftarrow g^{1/(id+i)}$.

P	$\leftrightarrows$	V
$(com_{id}, com_i, open_{id}, open_i; com_{id}, com_i, proof) \leftarrow \mathsf{provePNG}(f, nym; id, i; \varnothing)$		
$(G_p, g, h, p, q) \leftarrow f$		$(G_p, g, h, p, q) \leftarrow f$
$c_1, c_2 \in_R \mathbb{Z}_q$		**A**
$com_{id} \leftarrow g^{id}.h^{c_1}$		
$com_i \leftarrow g^{i}.h^{c_2}$	$\xrightarrow{com_{id},\ com_i}$	
$t \in_R \mathbb{Z}_q$		**B**
$T \leftarrow nym^t$	$\xrightarrow{T}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_q$
$s \leftarrow c.(id+i)+t$	$\xrightarrow{s}$	
		$g^c.T \stackrel{?}{=} nym^s$
$c_3 \leftarrow c.(1-c_1-c_2)$		**C**
$w \leftarrow g^t.h^{c_3}$	$\xrightarrow{w}$	
$(\varnothing, proof_w) \leftarrow PK\{(t,c_3) : T = nym^t \wedge w = g^t.h^{c_3}\}$		
		$w.com_{id}^c.com_i^c \stackrel{?}{=} g^s.h^c$
$open_{id} \leftarrow (id, c_1)$		$proof \leftarrow (T, c, s, w,$
$open_i \leftarrow (i, c_2)$		$proof_w)$
$\mathsf{return}(com_{id}, com_i$		$\mathsf{return}(com_{id},$
$open_{id}, open_i)$		$com_i, proof)$

(for efficiency reasons) from the proveEPNG(), the provability property is argued below.

After creating the two commitments (A), P proves that he knows an $(id+i)$ such that $nym \leftarrow g^{1/(id+i)}$ (B). The s value can only be calculated if $(id+i)$ is known. The resulting c, s and t values are used to convince V that the committed values are indeed id and i (C). h generates a subgroup of order q in G_p and $log_h(g)$ is unknown. It is indeed a ZKPK:

Completeness. This can easily be seen by expanding the two comparisons verified by V.

Validity. The probability that $\tilde{P}$ passes is $1/(q.k)$ since he has to guess the right values $id \in \mathbb{Z}_q$ and $i \in [1,k]$. If an extractor K runs subprotocols A and B with challenge $c = c_I \in_R \mathbb{Z}_q$, resets the machine and runs A and B again with challenge $c = c_{II} \in_R \mathbb{Z}_q$, he can extract $id+i$, since the same t will be used. Guessing i then results in the correct id. K can thus extract the id and i with a probability of at least $1/k$.

Zero-knowledge. A simulator S could have constructed the proof as follows: (1.1) choose $s \in_R \mathbb{Z}_q$, (1.2) $S \leftarrow nym^s$, (1.3) choose $c \in_R \mathbb{Z}_q$, (1.4) $T \leftarrow S.g^{-c}$, (2.1) choose $com_{id}, com_i \in_R < g,h >$, (2.2) $w \leftarrow g^s.h^c.com_{id}^{-c}.com_i^{-c}$ (3) simulate a proof for $PK\{(t,c_3) : T = nym^t \wedge w = g^t.h^{c_3}\}$ w.r.t. w and nym.

Since the integrated ZK proof is a proof of representation and equality of secrets, the provePNG() protocol requires six modular exponentiations by P, four by V and the communication cost is $6.|p| + 5.|q|$. Computing nym by P requires one multi-base modu-

lar exponentiation by P, sending it to V adds a communication cost of $|p|$. The efficiency of the solution presented in section 6 will also depend on other factors such as the efficiency of proving properties about the committed values. The same holds for the proveEPNG() function.

5.3 A Concrete Extended Provable Nym Generator

The presented EPNG is defined as

$$nym \leftarrow g_1^{1/(id+i)}.g_2^{1/(id+tf)} \quad (2)$$

in a group G_p with known prime order p. g_1 and g_2 each generate a subgroup of prime order q. $log_{g_1}(g_2)$ is unknown. This also implies that $log_{g_2}(g_1)$ is unknown, since q is known. Indeed, finding the inverse of an element in $\mathbb{Z}_q$ is easy using the extended Euclidean algorithm.

The four first properties are fulfilled if the group parameter q and input id are sufficiently long and if for each two EPNGs with respectively (g_1,g_2) and (g_1',g_2') as bases, $log_{g_1}(g_1')$ or $log_{g_2}(g_2')$ is unknown:

One-wayness follows from the R assumption if (id,i,tf) is chosen out of a sufficiently large domain to avoid that all possible inputs are tested for a specific nym. Since i and tf are relatively small (see later), the domain of id needs to be sufficiently large.

Deterministic. Follows from the deterministic underlying algebra.

Semi-collision Free. (1) Since $log_{g_1}(g_2)$ is unknown, for a specific id, finding $i,i',tf,tf' \in \mathbb{Z}_q$ with $(i,tf) \neq (i',tf')$ such that $g_1^{1/(id+i)}.g_2^{1/(id+tf)} = g_1^{1/(id+i')}.g_2^{1/(id+tf')}$ can be considered as two subproblems $g_1^{1/(id+i)} = g_1^{1/(id+i')}$ and $g_2^{1/(id+tf)} = g_2^{1/(id+tf')}$. Since each element has exactly one inverse in $\mathbb{Z}_q$, this is impossible.

(2) If $1/n$ is considered as biggest negligible value, $n \in \mathbb{N}_0$, $i \in [1,k]$, $tf \in [1,l]$ and $k,l \in \mathbb{Z}_q$, then the size of the EPNG's output domain must be at least $k.l.n$. This is now proven. Say, $nymset$ and $nymset'$ are the sets of all possible nyms that can be generated with $id \in \mathbb{Z}_q$ and $id' \in \mathbb{Z}_q$ respectively and say, the output domain of the EPNG f has size $k.l.n$. Both sets thus have size $k.l$. Let $nym \in_R nymset$. Then, $Pr[nym \in nymset'] = 1/(n.k.l)$ and the probability that there is an overlap between $nymset$ and $nymset'$ – i.e. a collision arises – is thus $(k.l)/(n.k.l) = 1/n$. The output size of f is at least q (if $log_{g_1}(g_2) \in \mathbb{Z}_q$). Thus, when q is chosen, it must be at least $n.k.l$, where $k.l$ is an estimated maximum value.

Unlinkable. Due to the one-wayness it is infeasible to derive id from nym.

Let $nym_1 = g_1^{1/(id+i)}.g_2^{1/(id+tf)}$ and $nym_2 = g_1^{1/(id+i')}.g_2^{1/(id+tf)}$ with $i' \in \mathbb{Z}_q$ and $x \leftarrow i'-i$. This means that $nym_2 = nym_1.(g_1^{1/(id+i+x)}/g_1^{1/(id+i)})$. Since id cannot be found, different nyms using the same id and tf are unlinkable. Similarly, different nyms using the same i cannot be linked.

Since $log_{g_1}(g_2)$ is unknown, no relations between $1/(id+i)$ and $1/(id+tf)$ and thus between i and tf can be found.

(2) The bases (g_1,g_2) and (g_1',g_2') of two $EPNG$s must not lead to linkabilities. This can only happen if $log_{g_1}(g_1') = log_{g_2}(g_2')$ and if this value is known (DL assumption). If $nym = g_1^{1/(id+i)}.g_2^{1/(id+tf)}$, $nym' = (g_1')^{1/(id+i)}.(g_2')^{1/(id+tf)}$ and $a = log_{g_1}(g_1') = log_{g_2}(g_2')$, then $nym' = (g_1^a)^{1/(id+i)}.(g_2^a)^{1/(id+tf)} = (g_1^{1/(id+i)}.g_2^{1/(id+tf)})^a = nym^a$.

Provability. The implementation of $P \leftrightarrows V : (com_{id}, com_{i,tf}, open_{id}, open_{i,tf}; com_{id}, com_{i,tf}, proof) \leftarrow proveEPNG(f, nym; id, i, tf; \varnothing)$ is shown in table 2 and returns commitments to id, and to i and tf, such that V is convinced that $nym = g_1^{1/(id+i)}.g_2^{1/(id+tf)}$. An additional base h of order q is required where $log_h(g_1)$ and $log_h(g_2)$ are unknown.

The commitments to id, i and tf ($com_{id}, com_{i,tf}$) are created by P and sent to V. Additionally, P sends to V a 'helper' commitment C to the values $x_1 = \frac{1}{id+i}$ and $y_1 = \frac{1}{id+tf}$. In the first proof of knowledge, P proves to V that $nym = g_1^{x_1}.g_2^{y_1}$. In the second proof of knowledge, P proves that the commitment com_{id} is indeed a commitment of a value id towards basis $g_1.g_2$, or stated otherwise; P proves that the committed value w.r.t. g_1 is the same as the committed value w.r.t. g_2. In the third proof of knowledge, P proofs to V that the sum of the committed values id and i equals the inverse in $\mathbb{Z}_q$ of x_1, which is committed in C. In the final proof of knowledge P proofs to V that the sum of the committed values id and tf equals the inverse in $\mathbb{Z}_q$ of y_1, which is committed in C. The combination of these four proofs of representation and equality of secrets thus guarantees that the committed values id, i and tf correspond to $nym = g_1^{1/(id+i)}.g_2^{1/(id+tf)}$.

5.4 Efficiency

The calculation of nym requires one multi-base modular exponentiation by P, and sending it to V results in a communication cost of $|p|$ bits.

The four proofs of knowledge in the proveEPNG() can be merged into a single one by applying conjunctions. The resulting proof can be computed more efficiently since some operands appear in multiple proofs. Secondly, the second

Table 2: The proveEPNG protocol for $nym \leftarrow g_1^{1/(id+i)} g_2^{1/(id+tf)}$.

$P \leftrightarrows V : (com_{id}, com_{i,tf}, open_{id}, open_{i,tf}; com_{id}, com_{i,tf}, proof) \leftarrow \mathsf{proveEPNG}(f, nym; id, i, tf; \varnothing)$	
$(G_p, p, q, g_1, g_2, h) \leftarrow f$	$(G_p, p, q, g_1, g_2, h) \leftarrow f$
$t_{id}, t_{i,tf}, t_C \in_R \mathbb{Z}_q$ $com_{id} \leftarrow (g_1.g_2)^{id}.h^{t_{id}}$ $com_{i,tf} \leftarrow g_1^i.g_2^{tf}.h^{t_{i,tf}}$ $x_1 \leftarrow \frac{1}{id+i}$, $x_2 \leftarrow id+i$ $y_1 \leftarrow \frac{1}{id+tf}$, $y_2 \leftarrow id+tf$	
$C \leftarrow g_1^{x_1}.g_2^{x_2}.h^{t_C}$	$\xrightarrow{com_{id},\ com_{i,tf},\ C}$
	$(\varnothing; pr_1) \leftarrow PK\{(x_1, y_1, t_C) : C = g_1^{x_1}.g_2^{y_1}.h^{t_C} \wedge nym = g_1^{x_1}.g_2^{y_1}\}$ $[\quad (\varnothing; pr_2) \leftarrow PK\{(id, t_{id}) : com_{id} = g_1^{id}.g_2^{id}.h^{t_{id}}\} \quad]$
$t \leftarrow t_{id} + t_{i,tf}$, $t_g \leftarrow y_1.x_2$, $t_h \leftarrow t_C.x_2$ $(\varnothing; pr_3) \leftarrow PK\{(x_1, x_2, y_1, y_2, t_C, t, t_g, t_h) : C = g_1^{x_1}.g_2^{y_1}.h^{t_C} \wedge com_{id}.com_{i,tf} = g_1^{x_2}.g_2^{y_2}.h^t \wedge g_1 = (\frac{1}{g_2})^{t_g}.(\frac{1}{h})^{t_h}.C^{x_2}\}$ $r_g \leftarrow x_1.y_2$, $r_h \leftarrow t_C.y_2$ $(\varnothing; pr_4) \leftarrow PK\{(x_1, x_2, y_1, y_2, t_C, t, r_g, r_h) : C = g_1^{x_1}.g_2^{y_1}.h^{t_C} \wedge com_{id}.com_{i,tf} = g_1^{x_2}.g_2^{y_2}.h^t \wedge g_2 = (\frac{1}{g_1})^{r_g}.(\frac{1}{h})^{r_h}.C^{y_2}\}$	
$open_{id} \leftarrow (id, t_{id})$, $open_{i,tf} \leftarrow (i, tf, t_{i,tf})$	$proof \leftarrow (pr_1, pr_2, pr_3, pr_4, C)$
return $(com_{id}, com_{i,tf}, open_{id}, open_{i,tf})$	return $(com_{id}, com_{i,tf}, proof)$

proof can be omitted if P will prove something about id after the proveEPNG() protocol, which will always be the case. Indeed, if $com_{id} = g_1^{id}.g_2^{id'}$ with $id \neq id'$, P will be unable to prove anything about the value committed w.r.t. $g_1.g_2$. This leads to a more efficient, slightly relaxed proof which replaces the four previous ones:

$$(\varnothing; pr) \leftarrow PK\{(x_1, x_2, y_1, y_2, t_C, t, t_g, t_h, r_g, r_h) : nym = g_1^{x_1}.g_2^{y_1} \wedge C = g_1^{x_1}.g_2^{y_1}.h^{t_C} \wedge com_{id}.com_{i,tf} = g_1^{x_2}.g_2^{y_2}.h^t \wedge g_1 = (\frac{1}{g_2})^{t_g}.(\frac{1}{h})^{t_h}.C^{x_2} \wedge g_2 = (\frac{1}{g_1})^{r_g}.(\frac{1}{h})^{r_h}.C^{y_2}\}$$

Since there are now three commitments in proveEPNG(), 10 secrets, 1 challenge and 5 operands, 13 modular exponentiations are required, and the total communication cost is $8.|p| + 11.|q|$. It was argued previously that q should be enlarged to $n.k.l$, where $1/n$ is negligible. The next section shows that k and l is used as a maximum for access limits and for timeframe numbers and will typically be small compared to q (E.g. $|k| \leqslant 16$ and $|l| \leqslant 24$ bits). A typical value for $|n|$ is 160 bits (e.g. in hash functions). The $|q|$ for the EPNG must thus be chosen larger than the usual size of 160 bits, but not by more than 25%. The protocol requires four interactions.

Note that the number of multi-base modular exponentiations remains constant, even if the number of input parameters of proveEPNG() is increased.

6 INTEGRATION IN ANONYMOUS CREDENTIAL SYSTEM

This section presents and evaluates a solution for the problem stated in section 3.

6.1 Solution

The main roles are the user U, a service provider SP and the registration authority R. Each user can receive from R a (service independent) anonymous credential $cred_U$, which contains a user-specific secure random number id. R also issues service certificates to SPs - one for each service. Each service certificate contains an EPNG f_S description, which will allow to set access limits per timeframe while guaranteeing unlinkability of accesses.

Setup Registration Authority. A certificate is issued by a certificate authority CA to the registration authority R. It contains a freshly generated parameter set S_{id}, necessary for the joint secure random number generation (see section 4.3). It is possible that $cert_R$ is self-signed ($R = CA$).

Registration of Service. SP registers a new service. Therefore, SP provides R with the necessary service description $desc_s$ (i.e. service provider name, service name, functionality and potentially the access limit policy). R issues to SP a service certificate $cert_s$ containing SP's public key, $desc_s$ and a new extended provable nym generator $f_s \leftarrow \mathsf{genEPNG}()$.

Issue User Credential. U receives a credential that can be used for accessing services with a k-show per timeframe limit. Typically, this phase will be preceded by a mutual authentication phase. First, R checks in a register whether U has already been registered. If not, a new random value (id) to be included in the credential is agreed using the agreeSecureRand() protocol with S_{id} as input. This results in a commitment C to this random value, and opening info O. Finally, R issues a new anonymous credential which contains the committed random number and updates the register.

$U \leftrightarrow R : (cred_U; register') \leftarrow \mathsf{issueUserCred}(cert_R, atts, coms, id_U; opens; SK_R, register)$

(1) U $\leftrightarrows$ R : if ($\mathsf{hasValidCred}(id_U, register)$) abort
(2) U $\leftrightarrows$ R : $(O,C;C) \leftarrow \mathsf{agreeSecureRand}(cert_R.S_{id}; \varnothing; \varnothing)$
(3) U $\leftrightarrows$ **R** : $(cred_U; \varnothing) \leftarrow \mathsf{issueCred}(\{coms \cup C\}, atts, cert_R; \{opens \cup O\}; SK_R)$
(4) U; R : return $(cred_U; \{register \cup (id_U, C)\})$

Accessing a Service with k-limit in Rimeframe *tf*. U wants to access a service. First, SP authenticates towards U using the correct service certificate $cert_s$ and corresponding SK_s. Then, U selects an $i \in [1,\ldots,k]$ that has not yet been used during timeframe tf for that specific service. The single-use nym $nym \leftarrow cert_s.f_s(cred_U.id, i, tf)$ is sent to SP. The latter checks in the service's history whether that nym has already been used. If not, U proves that the nym has been correctly formed (i.e. is the result of applying the EPNG to id, i and tf, $0 < i \leqslant k \wedge id \in cred_U$ and the current timeframe has been used, without revealing id or i. In addition, personal properties $props$ (of attributes embedded in the credential) can be proven or a message can be signed during the credential show. Finally, SP updates the service's history $history_s$ with the new nym and U updates his list of used is for that service during timeframe tf.

$U \leftrightarrow SP : (used'_s; history'_s) \leftarrow \mathsf{accessService}(cert_s, props, k, tf, [msg]; cred_U, used_s; SK_s, history_s)$

1. U $\leftarrow$ SP: $\mathsf{authenticate}(cert_s; \varnothing; SK_s)$
2. U : $i \leftarrow \mathsf{selectFreeIndex}(cert_s.id, tf, used_s)$
3. U $\rightarrow$ SP: $nym \leftarrow cert_s.f_s(cred_U.id, i, tf)$
4. SP : if $((nym, .) \in history_s)$ abort
5. U $\leftrightarrows$ SP: $(c_{id}, c_{i,tf}, o_{id}, o_{i,tf}; c_{id}, c_{i,tf}, pr_f) \leftarrow \mathsf{proveEPNG}(cert_s.f_s, nym; cred_U.id, i; \varnothing)$
6. U $\leftrightarrows$ SP: $(\varnothing; pr_{i,tf}) \leftarrow \mathsf{PK}\{(o_{i,tf}) : 0 < c_{i,tf}.i \leqslant k \wedge c_{i,tf}.tf = tf\}$
7. **U** $\leftrightarrows$ SP: $(\varnothing; pr_{id}) \leftarrow \mathsf{showCred}(c_{id}, props \wedge c_{id}.id = cred_U.id; cred_U, o_{id}; \varnothing)\{[msg]\}$
8. U; SP : return $(\{used_s \cup (tf, i)\}; \{history_s \cup (nym, proof_{(f,i,tf,id)})\})$

The EPNG's semi-collision free property 1 guarantees that the user is able to generate k different nyms for a single timeframe and the determinability guarantees that the user is unable to generate more than k nyms. The probability that two users have the same single-use nym for the same service is negligible due to the EPNG's semi-collision free property 2.

User Anonymity. The EPNG's unlinkable property guarantees that two nyms of the same user for the same or for other services during the same or during different timeframes cannot be linked. Evidently, the anonymity does not hold if identifying data were released as part of the anonymous credential show protocol. The R is trusted not to introduce and disclose relationships between different EPNGs.

Flexibility. All anonymous credentials' functionality is trivially preserved (e.g. conditional deanonymization). Based on the disclosed personal properties, SP can define 1) whether and when users are given access 2) the timeframe duration and access limit for that user type. This combination allows for non-adjectant timeframes. For instance, a timeframe might last one week, but only in weekends non-members can access the service. Timeframe size and limit can change dynamically. The proofs can be made non-interactive (Blum et al., 1988), giving them proof value to parties other than V.

Efficiency. The efficiency of the access service protocol, which is executed most frequently, is now considered. The EPNG in section 5.3 requires 13 multi-base modular exponentiations (which can be computed almost as efficiently as single-base ones). Proving that i lies in the proper interval, is done in 'less than 20 modular exponentiations' (Boudot, 2000) and proving equality of tf with a given timeframe is done in two modular exponentiations. For small k, it might be more efficient to prove that $i = 1 \vee \ldots \vee i = k$.

Instead of a doing a range proof for i, its exact

6.2 Evaluation

Service Access Restriction. The value id is only known to U and both R and SP (which has to trust R) are ensured that it is random, sufficiently long, and embedded in the user's credential as a result of the properties of joint random number generation and the possibility to include committed values in anonymous credentials. If the id-domain is sufficiently large, the probability that two users have the same id is negligible. The verification whether $id \in register$ during the user credential issuance guarantees that each user has no more than one id.

value can be disclosed, since the user can randomly choose an unused i. This is more efficient. However, the service provider then knows that two different nyms in the same timeframe, using the same i cannot originate from the same user. This allows the service provider to get a more exact (higher) lower threshold w.r.t the number of users of the service. Since both i and tf are disclosed and thus no longer need to be hidden by the EPNG function, and since the EPNG's unlinkability property guarantees that id is hidden for the service provider, the one-wayness property can be omitted in this case.

If $k = 1$ or $tf = \infty$, the EPNG is reduced to a PNG ($nym = g_2^{1/(id+tf)}$ or $nym = g_1^{1/(id+i)}$), requiring 10 modular exponentiations. If $k = 1$ and $tf = \infty$, function $nym \leftarrow g_1^{id}$ can be used, of which the correctness can be proven in two modular exponentiations. Finally, the performance depends on the anonymous credential system and the properties that are proved.

7 CONCLUSIONS

This paper presents a solution to dynamically restrict the number of times a user can access a service during a single timeframe. The solution is built on anonymous credential systems, making it extremely flexible. Both the size of the timeframes and the access limit may vary according to the service policy and the user's properties that were disclosed.

Future work includes usage of an EPNG to set sticky policies by the credential issuer to credentials, while hiding the policies for the verifier.

ACKNOWLEDGEMENTS

This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy and the Research Fund K.U.Leuven and the IWT-SBO project (ADAPID) "Advanced Applications for Electronic Identity Cards in Flanders".

REFERENCES

Bellare, M. and Goldreich, O. (1992). On defining proofs of knowledge. pages 390–420. Springer-Verlag.

Blum, M., Feldman, P., and Micali, S. (1988). Non-interactive zero-knowledge and its applications. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112, New York, NY, USA. ACM.

Boudot, F. (2000). Efficient proofs that a committed number lies in an interval. pages 431–444. Springer Verlag.

Brands, S. (1999). A technical overview of digital credentials.

Brands, S. (2000). *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA.

Camenisch, J. and Herreweghen, E. V. (2002). Design and implementation of the idemix anonymous credential system.

Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., and Meyerovich, M. (2006a). How to win the clone wars: Efficient periodic n-times anonymous authentication. Cryptology ePrint Archive, Report 2006/454.

Camenisch, J., Hohenberger, S., and Lysyanskaya, A. (2006b). Balancing accountability and privacy using e-cash (extended abstract. In *In SCN, volume 4116 of LNCS*, pages 141–155. Springer.

Camenisch, J. and Lysyanskaya, A. (2001). An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 93–118, London, UK. Springer-Verlag.

Chaum, D. (1985). Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044.

Cramer, R., Damgård, I., and Schoenmakers, B. (1994). Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, pages 174–187, London, UK. Springer-Verlag.

Damgard, I., Dupont, K., and Pedersen, M. (2006). Unclonable group identification.

Damgard, I., Pedersen, T., and Pfitzmann, B. (1996). Statistical secrecy and multi-bit commitments.

Nguyen, L. and Safavi-naini, R. (2005). Dynamic k-times anonymous authentication. In *In ACNS 2005, number 3531 in LNCS*, pages 318–333. Springer Verlag.

Pedersen, T. (1992). Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK. Springer-Verlag.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. In *Journal of Cryptology*, pages 103–112, New York, NY, USA. Springer.

Teranishi, I., Furukawa, J., and Sako, K. (2004). k-times anonymous authentication (extended abstract. In *In Asiacrypt, volume 3329 of LNCS*, pages 308–322. Springer.