VLR Group Signatures
How to Achieve Both Backward Unlinkability and Efficient Revocation Checks

Julien Bringer[1] and Alain Patey[1,2]
[1]*Morpho, Safran Group, Paris, France*
[2]*Télécom ParisTech, Paris, France*
Identity and Security Alliance (The Morpho and Télécom ParisTech Research Center)

Keywords: Group Signatures, Verifier-Local Revocation, Backward Unlinkability, Efficiency, Revocation Check.

Abstract: Verifier-Local Revocation (VLR) group signatures are a particular case of dynamic group signature schemes where the revocation process does not influence the activity of the signers. The verifiers use a Revocation List and in all known schemes, checking a signature requires a computational time linear in the number of revoked members. Usually, it requires one pairing per revoked user. Recently, Chen and Li proposed a scheme where Revocation Check uses exponentiations instead of pairings. In this paper, we first propose a correction of their scheme to enable a full proof of the traceability property and we succeed with a constant additional cost only to extend this tweaked scheme to ensure Backward Unlinkability (BU). This important property prevents the loss of anonymity of past signatures when a user is revoked. We thus obtain the scheme with the most efficient Revocation Check among VLR schemes enabling BU.

1 INTRODUCTION

Group signatures, introduced by (Chaum and van Heyst, 1991), enable a registered member to sign anonymously on behalf of a group. The identity of a signer can only be revealed by a Group Manager who knows all the secret parameters of the group. Actual group signature schemes are dynamic: members can join and leave (voluntarily or not) the group at any time. To enable this, a revocation process is established. We focus in this paper on schemes with *Verifier-Local Revocation* (VLR), a particular way to deal with revocation where we do not want additional interactions with the signers. A Revocation List (RL) is built by the group manager and sent only to the verifiers. The signers do not take it into account when they sign. Verifying a signature is divided into two parts: a *Signature Check* to verify if the signer is a registered member and a *Revocation Check* to verify, using RL, whether the signer is revoked. This type of group signature schemes is useful for applications where signers are often offline or are computationally weak devices (TPMs, smartcards…). Several proposals for VLR group signatures have been made, cf. for instance (Boneh and Shacham, 2004; Nakanishi and Funabiki, 2006; Nakanishi et al., 2009; Chen and Li, 2010; Libert and Vergnaud, 2009). A similar concept is introduced in (Kiayias et al., 2004) for traceable signatures with an implicit tracing mechanism. Applications of VLR schemes are, for instance, Direct Anonymous Attestation (DAA) in the context of Trusted Computing (Brickell et al., 2004; Brickell and Li, 2010), Vehicular Ad-hoc NETworks (VANETs) (Studer et al., 2008) or anonymous authentication (Bringer et al., 2008).

One downside of VLR schemes is the lack of efficiency of the Revocation Check during the verification of a signature. Indeed, in the original (Boneh and Shacham, 2004) scheme and many other propositions, this part requires at least one pairing operation per each revoked user. (Nakanishi et al., 2009) proposed a slight variant where products of pairings are used instead of separate pairings. In (Chen and Li, 2010), a VLR scheme using exponentiations in the Revocation Check is proposed. As exponentiations require less computation time, this is a substantial improvement concerning efficiency. However, in the (Chen and Li, 2010) scheme, proofs of security are not detailed and it is unclear how to obtain an extractor for the proof of knowledge included in the signature. Having an extractor is necessary for the proof of *traceability*, one of the essential security properties required from a group signature scheme. This is why we propose a patch to the original (Chen and Li, 2010) scheme and explain explicitly how to build an extractor for the thus modified algorithm. This part

Bringer J. and Patey A..
VLR Group Signatures - How to Achieve Both Backward Unlinkability and Efficient Revocation Checks.
DOI: 10.5220/0004017502150220
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 215-220
ISBN: 978-989-8565-24-2

of our work is a basis for our full scheme and can be seen as a useful tool for our proofs of security.

Another issue in most VLR schemes is the following: once a user has been revoked, all his previous signatures lose their anonymity. The property that prevents this loss is called *Backward Unlinkability* (BU). It also allows a user to come back into the group after having been revoked and use the same keys as before while remaining anonymous. This property was first introduced in (Song, 2001). There have been several proposals to enable BU in schemes using pairings in the Revocation Check, *e.g.* (Nakanishi and Funabiki, 2006; Libert and Vergnaud, 2009). This does not change the type of operations to use in the Revocation Check. The other parts of the signing and verifying algorithms are slightly modified but the difference is constant and small. The same techniques cannot be applied to schemes based on exponentiations. A first proposal for such schemes has been suggested without specific proofs in (Ateniese et al., 2002) in the context of quadratic residues. We present in this paper an improvement, inspired by the technique from (Ateniese et al., 2002) that we adapt to the context of bilinear groups, to the efficient (Chen and Li, 2010) scheme in order to add the BU property. Moreover we obtain full proofs of our security results including the BU functionality and we also patch the (Chen and Li, 2010) scheme in order to ensure traceability. To achieve BU, we use zero-knowledge proofs of knowledge involving double discrete logarithms. This technique requires a number of computations that is a function of a security parameter, but that is independent of the total number of users and of the number of revoked members. Moreover, this technique is generic and can be applied to other exponentiation-based VLR schemes.

Our scheme satisfies *Backward Unlinkability*, *Traceability* and *Exculpability* in the random oracle model. Security is based on the strong Diffie-Hellman (SDH) assumption, a slight adaptation of the Decisional Diffie-Hellman (adapted DDH) assumption and the Discrete Logarithm (DL). Contrary to the various previous constructions of VLR group signature schemes with BU, our contribution succeeds in eliminating pairings in the revocation checks, and thus greatly increases the efficiency when verifying a signature. We increase, by a constant overhead, the size of our signatures and the time required for signing but: 1/the overhead can be pre-computed offline such that the message-depending part of the signature is as efficient as in other VLR schemes; 2/the saving in computation time for the online verification (including revocation check) is very important as soon as the number of revoked members is large (from a few dozens of members).

An extended version of this work, containing in particular full security definitions and proofs, is available in (Bringer and Patey, 2012).

2 VLR GROUP SIGNATURES MODEL

We here extend the VLR group signature model from (Chen and Li, 2010) by including BU following the model of (Nakanishi and Funabiki, 2006). The security properties for generic dynamic group signatures are from (Bellare et al., 2005; Boneh and Shacham, 2004) for Verifier-Local Revocation and from (Nakanishi and Funabiki, 2006) to enable BU.

There are three types of entities in our model: a Group Manager GM, a set of members and a set of verifiers. A VLR Group Signature Scheme with BU and Exculpability consists of the following algorithms:

KeyGen(k,T)**.** On input a security parameter k and a number T of periods, this algorithm, run by GM outputs the group public parameters gpk and the issuing key ik. It also sets an empty Revocation List RL_j, for each period j. These lists will be filled later with the revocation tokens of the revoked users.

Join$(gpk,ik;gpk)$**.** This algorithm is an interactive protocol between GM and a member M_i. GM takes as input the public parameters gpk and the issuing key ik, M_i takes only gpk. In the end, M_i outputs an identity id_i, a secret key sk_i, a credential cre_i and a tracing key tk_i (included in cre_i). GM gets id_i and tk_i and outputs also a list of revocation tokens for M_i: $\mathbf{rt_i} = \{rt_{ij} | j \in \{1,\ldots,T\}\}$.

Revoke(gpk,rt_{ij},j,RL_j)**.** GM runs this algorithm to prevent a member M_i from making valid signatures at period j. It outputs an updated revocation list RL_j for period j, where rt_{ij} has been added.

Sign(gpk,j,sk_i,cre_i,m)**.** This algorithm, run by a member M_i, takes as input a message m, M_i's keys sk_i and cre_i and a message m to sign at period j. It outputs a signature σ.

Verify(gpk,j,RL_j,m,σ)**.** This algorithm, run by a verifier takes as input a message m, its signature σ, a period j, the corresponding Revocation List RL_j and the public parameters gpk. It checks if the message has been signed by an unrevoked group member, without revealing the signer's identity. The possible outputs are **valid** and **invalid**.

The scheme is *correct* if every signature created

by an unrevoked member is verified as valid. The model of *Backward Unlinkability* enables a user that has been revoked at a given time period to remain anonymous in any other time periods. The aim of the *Exculpability* property is to offer protection against the Group Manager. In the *Exculpability* game, roles are inverted: the adversary is the GM and, consequently, knows the group's secret key and all the players' credentials. The goal of the adversary is to forge a valid signature that will be attributed to an honest (*i.e.* not corrupted) member. This signature must be such that it cannot be denied by the signer.

3 PRELIMINARIES

3.1 Bilinear Groups, Pairings and Complexity Assumptions

Let G_1 be a cyclic group of prime order p, G_2 be a group of order a power of p, G_T be a cyclic group of prime order p, ψ be an homomorphism from G_2 to G_1, g_2 be an order-p element of G_2, g_1 a generator of G_1 such that $\psi(g_2) = g_1$ and $e : G_1 \times G_2 \to G_T$ a *pairing*.

Discrete Logarithm (DL) Problem. Given G a multiplicative finite cyclic group, with generator g, and g^n (with $n \in_R \mathbb{Z}$), the problem is to find n.

q-Strong Diffie-Hellman (q-SDH) Problem (Boneh and Boyen, 2004). Given bilinear groups G_1, G_2, G_T, g_1, g_2 and a pairing e, as in Section 3.1, and a q-tuple $(g_2^{\gamma}, .., g_2^{(\gamma^q)})$ ($\gamma \in_R \mathbb{Z}_p^*$). The problem is to compute a pair $(g_1^{1/(\gamma+x)}, x)$, with $x \in \mathbb{Z}_p^*$.

Adapted DDH Problem. Given G a multiplicative finite cyclic group of order a safe prime p, with generator g, g^a, g^b ($a, b \in_R \mathbb{Z}_p^*$), u a generator of a subgroup of $\mathbb{Z}_q^*$ (q prime) of order $(p-1)/2$ and u^a. The problem is to distinguish g^{ab} from a random element $z \in G$.

3.2 Proofs of Knowledge for Double Discrete Logarithms

Let G be a cylic group of safe prime order p, $g \in G$, h a generator of a subgroup of order $(p-1)/2$ of $\mathbb{Z}_q^*$, where q is prime, and $x \in \mathbb{Z}_p^*$. Let $K = g^x$ and $L = g^{h^x}$. We want to build a Non-Interactive Zero-Knowledge Proof of Knowledge of x.

We must use binary challenges instead of a modular integer in the classical NIZK PK *à la* (Schnorr, 1989). This technique is due to (Stadler, 1996; Camenisch and Stadler, 1997) and has been suggested for group signatures by (Ateniese et al., 2002). We describe in Table 1 how such a proof works for a security parameter λ (we keep the same notations for g, h, x, K and L). In (Stadler, 1996), the authors state that an attacker can cheat successfully only with probability $2^{-\lambda}$.

Table 1: Signature-proof of knowledge of the equality of a logarithm and a double logarithm.

Proof Generation:

1. For $l = 1 \ldots \lambda$, pick $r_l \in_R \mathbb{Z}_p$ and compute $V_l = g^{r_l}$, $W_l = g^{h^{r_l}}$.
2. Compute $c = H(m||K||L||(V_l, W_l)_{l=1,\ldots,\lambda})$.
3. For $l = 1 \ldots \lambda$, let b_l denote the l^{th} bit of c. Set $s_l = r_l - b_l x$.
4. Return $g, K, L, c, s_1, \ldots, s_\lambda$.

Proof Verification:

1. For $l = 1 \ldots \lambda$, let b_l denote the l^{th} bit of c. Compute $V_l' = g^{s_l} K^{b_l}$ and $W_l' = (g^{1-b_l} L^{b_l})^{h^{s_l}}$
2. Compute $c' = H(m||K||L||(V_l', W_l')_{l=1,\ldots,\lambda})$
3. If $c = c'$, accept the proof, else reject it

4 PROPOSED SCHEME

In this section we describe our extension of (Chen and Li, 2010) to prove traceability and to achieve Backward Unlinkability.

The *KeyGen* algorithm is described in Algorithm 1: we use bilinear groups and the notations of Section 3.1 (G_1, G_2, G_T, e, g_1, g_2). The issuing key is $\gamma \in_R \mathbb{Z}_p$, its public counterpart is $w = g_2^{\gamma}$. Notice that p must be a safe prime so that adapted DDH holds in the group containing the revocation tokens.

Algorithm 1: *KeyGen*(k, T).

1: Choose bilinear groups G_1, G_2, G_T of order a k-bit prime number p that is safe (i.e. $(p-1)/2$ prime number), a prime number q and a pairing $e : G_1 \times G_2 \to G_T$. Let g_1, g_2 be generators of G_1 and G_2.
2: Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_p$ and a security parameter λ for the proofs of knowledge involving double logarithms.
3: Choose $\tilde{g}_1, \hat{g}_1 \in_R G_1, \gamma \in_R \mathbb{Z}_p^*, h_1, \ldots, h_T \in_R \mathbb{Z}_q^*$, ($\gamma$ and the h_j's of order $(p-1)/2$) and compute $w = g_2^{\gamma}$.
4: Compute $T_1 = e(g_1, g_2)$, $T_2 = e(\tilde{g}_1, g_2)$, $T_3 = e(\hat{g}_1, g_2)$ and $T_4 = e(\hat{g}_1, w)$.
5: Output: gpk = (G_1, G_2, G_T, e, p, g_1, g_2, $\tilde{g}_1$, $\hat{g}_1$, w, H, T_1, T_2, T_3, T_4, λ, $h_1, \ldots, h_T$) and $ik = \gamma$.

The *Join* algorithm is explained in Algorithm 2. Each member M_i chooses a secret key $sk_i = f_i \in_R \mathbb{Z}_p$, not known by GM. M_i gives to GM an identity $id_i = \tilde{g}_1^{f_i}$ and proves the knowledge of f_i. GM sends him, over a secure channel, a credential $cre_i = (A_i, x_i)$. To enable BU, we divide the time into T periods. For

each period j, there is a public token h_j. The revocation token for a member M_i at period j is $rt_{ij} = h_j^{x_i}$ and $tk_i = x_i$ is the tracing key. Revocation lists are different at each period, they are denoted RL_j.

Algorithm 2: *Join*(gpk, ik ; gpk).

1: GM sends a nonce $n_i \in \{0,1\}^k$ to M_i.
2: M_i chooses $f_i \in_R \mathbb{Z}_p$ and computes $F_i = \tilde{g}_1^{f_i}$. He sets $sk_i = f_i$ and $id_i = F_i$. He chooses $r_f \in_R \mathbb{Z}_p$ and computes $R = \tilde{g}_1^{r_f}$. He computes $c = H(gpk||F_i||R||n_i)$ then $s_f = r_f + c f_i$.
3: M_i sends $comm = (F_i, c, s_f)$ to GM.
4: GM computes $R' = \tilde{g}_1^{s_f} F_i^{-c}$ and checks that $s_f \in \mathbb{Z}_p$ and $c = H(gpk||F||R'||n_i)$. He chooses $x_i \in_R \mathbb{Z}_p$ and computes $A_i = (g_1 F_i)^{1/(x_i+\gamma)}$. He sets $cre_i = (A_i, x_i)$, $tk_i = x_i$ and $id_i = F_i$.
5: GM sends cre_i to M_i, using a secure channel.
6: M_i checks that $e(A_i, wg_2^{x_i}) = e(g_1 \tilde{g}_1^{f_i}, g_2)$ and outputs (id_i, sk_i, cre_i).
7: The revocation token for M_i at period j is $rt_{ij} = h_j^{x_i}$.

The *Sign* algorithm is described in Algorithm 3. When a member M_i creates a signature, he first chooses a random $B \in_R G_1$ and computes $J = B^{f_i}$, $K = B^{x_i}$ and $L = B^{h_j^{x_i}}$. He picks a random $a \in_R \mathbb{Z}_p$, computes $b = ax_i$ and $T = A_i \hat{g}_1^a$. He then does a NIZK PK of (f_i, A_i, x_i) satisfying $J = B^{f_i}$, $K = B^{x_i}$ and $e(A_i, wg_2^{x_i}) = e(g_1 \tilde{g}_1^{f_i}, g_2)$. He also provides evidence that $b = ax_i$ as in (Boneh and Shacham, 2004) to ensure traceability based on an extractor. He finally computes a Proof of Knowledge $(c, (V_l, W_l)_{l=1\ldots\lambda})$, as described in Section 3.2, of the equality: $log_B K = log_{h_j}(log_B L) \quad (= x_i)$.

Algorithm 3: *Sign*(gpk, sk_i, cre_i, m, j).

1: Choose $B \in_R G_1$ and compute $J = B^{f_i}$, $K = B^{x_i}$ and $L = B^{h_j^{x_i}}$.
2: Choose $a \in_R \mathbb{Z}_p$, compute $b = ax_i$ and $T = A_i \hat{g}_1^a$.
3: Choose $r_f, r_x, r_a, r_b, r_1, \ldots, r_\lambda \in_R \mathbb{Z}_p$.
4: Compute $R_1 = B^{r_f}$, $R_2 = B^{r_x}$, $R_4 = K^{r_a} B^{-r_b}$, $R_3 = e(T, g_2)^{-r_x} T_2^{r_f} T_3^{r_b} T_4^{r_a}$, $V_l = B^{r_l}$ and $W_l = B^{h_j^{r_l}}$, $\forall l = 1\ldots\lambda$.
5: Compute c=$H(gpk||B||J||K||L||T||R_1||R_2||R_3||\ R_4||j||m)$.
6: Compute $d = H(c||(V_l, W_l)_{l=1\ldots\lambda})$.
7: Compute $s_f = r_f + c f_i$, $s_x = r_x + cx_i$, $s_a = r_a + ca$ and $s_b = r_b + cb$.
8: $\forall l = 1\ldots\lambda$, let b_l be the l^{th} bit of d. Set $s_l = r_l - b_l x$.
9: Output: $\sigma = (B, J, K, L, T, c, d, s_f, s_x, s_a, s_b, s_1, \ldots, s_\lambda)$.

Remark 1. *Note that the steps 1 to 4 in Algorithm 3 can be fully pre-computed in advance. Particularly, it includes the costly proof of knowledge of the equality of a logarithm and a double logarithm that we introduce here to enable BU. This leads to a message-depending part of signature generation almost as efficient as in the other VLR schemes.*

The *Verify* algorithm is described in Algorithm 4.

Algorithm 4: *Verify*(gpk, m, σ, RL_j, j).

1: **Signature Check:**
2: Check that $B, J, K, L, T \in G_1$ and $s_f, s_x, s_a, s_b, s_1, \ldots, s_\lambda \in \mathbb{Z}_p$.
3: Compute $R'_1 = B^{s_f} J^{-c}$, $R'_2 = B^{s_x} K^{-c}$, $R'_3 = e(T, g_2)^{-s_x} T_2^{s_f} T_3^{s_b} T_4^{s_a} T_1^c e(T, w)^{-c}$ and $R'_4 = K^{s_a} B^{-s_b}$.
4: Check that c=$H(gpk||B||J||K||L||T||R'_1||R'_2|\ |R'_3||R'_4||j||m)$.
5: $\forall l = 1\ldots\lambda$, let b_l be the l^{th} bit of d. Compute $V'_l = B^{s_l} K^{b_l}$ and $W'_l = (B^{1-b_l} L^{b_l})^{h_j^{s_l}}$.
6: Check that $d = H(c'||(V'_l, W'_l)_{l=1\ldots\lambda})$.
7: **Revocation Check:**
8: Check that $\forall rt_{ij} \in RL_j, L \neq B^{rt_{ij}}$.
9: Output **valid** if all checks succeed. Otherwise output **invalid**.

5 SECURITY

In (Chen and Li, 2010), the *Sign* algorithm was slightly different. Of course, there was no proof of knowledge of a double logarithm. What is important to notice is that there moreover was no R_4 value. We think there is something missing in the proof of traceability in (Chen and Li, 2010), that does not explicitly give an extractor. This is why we add the R_4 part in our algorithms. Notice that adding R_4 does not change the signature size but only adds one multi-exponentiation in both *Sign* and *Verify* algorithms. In (Bringer and Patey, 2012) we prove that one can actually obtain an extractor when using the *Sign* procedure from Algorithm 3. Note that as a group signature is essentially a proof of knowledge (POK) of a member key, the notion of extractor is here the same as in the context of POKs.

Theorem 1. *There exists an extractor for the group signature scheme as defined in Section 4, that extracts a valid key (x, f, A) from a convincing signer.*

The correctness of our scheme is straightforward. We also prove (cf. (Bringer and Patey, 2012)) that it achieves the other expected security properties as stated below.

Theorem 2. *Under the ROM and the hardness of the adapted DDH problem in G_1, the scheme described in Section 4 achieves BU. Under the ROM and the SDH assumption in (G_1, G_2, G_T), the scheme achieves Traceability. Under the ROM and the DL assumption, the scheme achieves Exculpability.*

6 EFFICIENCY

6.1 Analysis of the Proposed Scheme

We compare here our proposal with the patched (Chen and Li, 2010) scheme. Notice that we add in the signature λ elements $s_1, \ldots, s_\lambda$ of $\mathbb{Z}_p$ and one element L of G_1. Chen and Li proposed to use 256-bit Barreto-Naehrig curves(Barreto and Naehrig, 2005). In this

context, each element of G_1 can be represented using 257 bits. A (Chen and Li, 2010) signature using these parameters is 2308-bit long. A signature from our scheme using a security parameter $\lambda = 80$ is 23301-bit long, *i.e.* about ten times bigger. Concerning computation times, our modification requires $2\lambda + 1$ additional exponentiations in G_1 and λ exponentiations over $\mathbb{Z}_q$ for the signing part. Nevertheless, all these additional exponentiations are independent of the message and can be pre-computed offline by the signer.

It also requires 2λ additional exponentiations in G_1 and λ exponentiations over $\mathbb{Z}_q$ for the verifying part. Note that, despite this overhead, one important property of our solution is that revocation check remains as fast as in the original scheme. Consequently the cost of this overhead will be amortized with large revocation lists (cf. next section).

This is summed up in Table 2. (**me** stands for multi-exponentiations in G_1, *me* stands for multi-exponentiations in $\mathbb{Z}_q^*$, ME for multi-exponentiations in G_T and *P* for pairings. The "patched CL" scheme is the (Chen and Li, 2010) scheme modified to obtain an extractor, *i.e.* our scheme without the proof of knowledge of a double logarithm. And we denote by CL-BU$_\lambda$ our scheme with a security parameter λ.)

Table 2: Computational costs for (Chen and Li, 2010) and our scheme.

Scheme	Cost of *Sign* (offline)	Cost of *Sign* (online)	Cost of *Verify*
patched CL	6 **me** + 1 ME	negligible (1 hash)	$(4 + \lvert RL\rvert)$ **me** + 1 ME + 1 P
our scheme (CL-BU$_\lambda$)	$(7 + 2\lambda)$ **me** + λ *me* + 1 ME	negligible (2 hash)	$(4 + \lvert RL_j\rvert + 2\,\lambda)$ **me** + λ *me* + 1 ME + 1 P

6.2 Comparison with Existing Works

We now compare the additional cost for BU for the (Chen and Li, 2010) scheme and for a pairing-based scheme. We use as an example the (Boneh and Shacham, 2004) and the (Nakanishi and Funabiki, 2006) schemes (denoted respectively by BS and NF in the subsequent tables), the latter being a modification of the BS scheme enabling BU. We implemented all these algorithms on a PC with a 2.93 GHz Intel®Core^{TM}2 Duo processor. The implementation uses the C++ programming language and the NTL library (Shoup,). The order p of the groups used in the implementation is a 160-bit integer, q is a 1248-bit integer. We compute pairings using an optimization technique from (Stogbauer, 2004).

Remark 2. *The size of q is chosen so that the DL problem is hard over the subgroups of $\mathbb{Z}_q^*$ of order $(p-1)/2$. Note that the impact on the performances is very limited as only some of the additional exponentiations for the proof of knowledge of the double logarithm / logarithm equality are made on $\mathbb{Z}_q^*$ and, in particular, the exponentiations made during revocation check remain in G_1 (whose order is the smaller prime p).*

One can find in Table 3 our results for the schemes (Boneh and Shacham, 2004) (BS) and our correction of (Chen and Li, 2010) without Backward Unlinkability. We give the computation times for the *Sign* algorithm, for the constant part of the *Verify* algorithm and, finally, the time needed for each additional revoked user. Our results imply that computing a pairing is about four times longer than computing an exponentiation, which confirms the improvement brought by exponentiations in terms of efficiency.

In Table 4, we describe the additional time needed to add Backward Unlinkability to these schemes. The operations in the *Revocation Check* part have the same cost, that is why they are not mentioned here. We can see that, for pairing based schemes ((Boneh and Shacham, 2004; Nakanishi and Funabiki, 2006)), BU is essentially for free. In our scheme, it requires about 100 more milliseconds per security level (that can be handled offline for the signing part), which is coherent with the theoretical costs of Table 2. We also show why, despite the additional cost due to the security parameter of the proof of knowledge, our scheme becomes quickly more efficient than a pairing-based scheme with BU. In Table 5, we show the time needed by the *Verify* algorithm for the NF scheme (Nakanishi and Funabiki, 2006) and for our scheme, using different security parameters. We can see that there is a threshold value for the number of revoked members from which our scheme is more efficient. Our scheme is the most adapted for large groups (from a few dozens of users).

For instance, we remark that the overall time for signing and verifying when there are 1000 revoked members is divided by 3 for our scheme CL-BU$_{80}$ compared to the (Nakanishi and Funabiki, 2006) scheme.

Table 3: Computation times for the schemes without BU.

Scheme	BS	patched CL
Signature	1000 ms	450 ms
Verification	1170 ms	400 ms
Rev. Check (/rev.)	180 ms	45 ms

Table 4: Additional time for Backward Unlinkability.

Scheme with/ without BU	NF/BS	CL-BU$_{80}$/CL	CL-BU$_{128}$/CL
Signature	80 ms	8 s (offline)	13 s (offline)
Verification	40 ms	8 s	13 s

Table 5: Overall computational time for the *Verify* algorithm, depending on the number of revoked members.

Revoked members	NF	CL-BU$_{80}$	CL-BU$_{128}$
10	3 s	9 s	14 s
100	19 s	13 s	18 s
1000	3 min	53 s	58 s

7 CONCLUSIONS

We present the first VLR Group Signature scheme that enables BU where the revocation check (which is the costliest part) requires $|RL|$ (number of revoked users) exponentiations instead of $|RL|$ pairings. Our technique can be applied for adding BU to other VLR schemes that rely on exponentiations in the Revocation Check. By applying our technique to (Chen and Li, 2010), that we moreover modified to give a full security proof for traceability, we obtain the most efficient VLR scheme enabling Backward Unlinkability.

ACKNOWLEDGEMENTS

This work is partially funded under the European FP7 FIDELITY project (SEC-2011-284862).

REFERENCES

Ateniese, G., Song, D. X., and Tsudik, G. (2002). Quasi-efficient revocation in group signatures. In Blaze, M., editor, *Financial Cryptography*, volume 2357 of *LNCS*, pages 183–197. Springer.

Barreto, P. S. L. M. and Naehrig, M. (2005). Pairing-friendly elliptic curves of prime order. In Preneel, B. and Tavares, S. E., editors, *Selected Areas in Cryptography*, volume 3897 of *LNCS*, pages 319–331. Springer.

Bellare, M., Shi, H., and Zhang, C. (2005). Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, pages 136–153.

Boneh, D. and Boyen, X. (2004). Short signatures without random oracles. In Cachin, C. and Camenisch, J., editors, *EUROCRYPT*, volume 3027 of *LNCS*, pages 56–73. Springer.

Boneh, D. and Shacham, H. (2004). Group signatures with verifier-local revocation. In Atluri, V., Pfitzmann, B., and McDaniel, P. D., editors, *ACM Conference on Computer and Communications Security*, pages 168–177. ACM.

Brickell, E. and Li, J. (2010). A pairing-based daa scheme further reducing tpm resources. In Acquisti, A., Smith, S. W., and Sadeghi, A.-R., editors, *TRUST*, volume 6101 of *LNCS*, pages 181–195. Springer.

Brickell, E. F., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In Atluri, V., Pfitzmann, B., and McDaniel, P. D., editors, *ACM Conference on Computer and Communications Security*, pages 132–145. ACM.

Bringer, J., Chabanne, H., Pointcheval, D., and Zimmer, S. (2008). An application of the Boneh and Shacham group signature scheme to biometric authentication. In Matsuura, K. and Fujisaki, E., editors, *IWSEC*, volume 5312 of *LNCS*, pages 219–230. Springer.

Bringer, J. and Patey, A. (2012). Backward unlinkability for a VLR group signature scheme with efficient revocation check. IACR Cryptology ePrint Archive, Report 2011/376. http://eprint.i,acr.org/.

Camenisch, J. and Stadler, M. (1997). Efficient group signature schemes for large groups (extended abstract). In Jr., B. S. K., editor, *CRYPTO*, volume 1294 of *LNCS*, pages 410–424. Springer.

Chaum, D. and van Heyst, E. (1991). Group signatures. In *EUROCRYPT*, pages 257–265.

Chen, L. and Li, J. (2010). VLR group signatures with indisputable exculpability and efficient revocation. In *PASSAT*.

Kiayias, A., Tsiounis, Y., and Yung, M. (2004). Traceable signatures. In Cachin, C. and Camenisch, J., editors, *EUROCRYPT*, volume 3027 of *LNCS*, pages 571–589. Springer.

Libert, B. and Vergnaud, D. (2009). Group signatures with verifier-local revocation and backward unlinkability in the standard model. In Garay, J. A., Miyaji, A., and Otsuka, A., editors, *CANS*, volume 5888 of *LNCS*, pages 498–517. Springer.

Nakanishi, T. and Funabiki, N. (2006). A short verifier-local revocation group signature scheme with backward unlinkability. In Yoshiura, H., Sakurai, K., Rannenberg, K., Murayama, Y., and ichi Kawamura, S., editors, *IWSEC*, volume 4266 of *LNCS*, pages 17–32. Springer.

Nakanishi, T., Sudarsono, A., Sakemi, Y., Nogami, Y., and Funabiki, N. (2009). A group signature scheme with efficient verifier-local revocation check. In *SCIS*.

Schnorr, C.-P. (1989). Efficient identification and signatures for smart cards. In Brassard, G., editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer.

Shoup, V. Number theory library. http://www.shoup.net/ntl.

Song, D. X. (2001). Practical forward secure group signature schemes. In *ACM Conference on Computer and Communications Security*, pages 225–234.

Stadler, M. (1996). Publicly verifiable secret sharing. In *EUROCRYPT*, pages 190–199.

Stogbauer, M. (2004). Efficient algorithms for pairing-based cryptosystems. Master's thesis, Darmstadt University of Technology.

Studer, A., Shi, E., Bai, F., and Perrig, A. (2008). Tacking together efficient authentication, revocation, and privacy in vanets. Technical report, Carnegie Mellon CyLab.