Batch Groth–Sahai

Olivier Blazy[1], Georg Fuchsbauer[1], Malika Izabachène[2],
Amandine Jambert[3,4], Hervé Sibert[5], and Damien Vergnaud[1]

[1] École normale supérieure-CNRS-INRIA, 45 rue d'Ulm, 75320 Paris Cedex 05, France.
[2] Université de Versailles, 45 avenue des États-Unis, 78035 Versailles, France.
[3] Orange Labs R&D, 42 rue des Coutures, BP6243, 14066 Caen Cedex, France.
[4] IMB, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence, France
[5] ST-Ericsson, 9-11 rue Pierre-Felix Delarue, 72100 Le Mans Cedex 9, France.

Abstract. In 2008, Groth and Sahai proposed a general methodology for constructing non-interactive zero-knowledge (and witness-indistinguishable) proofs in bilinear groups. While avoiding expensive NP-reductions, these proof systems are still inefficient due to the number of pairing computations required for verification. We apply recent techniques of *batch verification* to the Groth-Sahai proof systems and succeed to improve significantly the complexity of proof verification. We give explicit batch-verification formulas for generic Groth-Sahai equations (whose cost is less than a tenth of the original) as well as for specific popular protocols relying on their methodology (namely Groth's group signatures and the P-signatures by Belenkiy, Chase, Kohlweiss and Lysyanskaya).

Keywords. Pairing-based cryptography, Batch verification, Groth-Sahai proof system.

1 Introduction

In a zero-knowledge proof system, a prover convinces a verifier *via* an interactive protocol that a mathematical statement is true, without revealing anything other than the validity of the assertion. In 1988, Blum, Feldman and Micali [BFM90] showed that the use of a common random string shared between the prover and the verifier permits to design a zero-knowledge proof system for all NP-languages that does not require interaction. These proofs, called non-interactive zero-knowledge (NIZK), turned out to be a particularly useful tool in constructing cryptographic primitives. Unfortunately, their work (as well as subsequent results) does not yield efficient proofs. Until recently, the only way to construct efficient proofs was to rely on the random-oracle model (ROM) [BR93], which has been subject to a series of criticisms starting with [CGH98].

In 2008, Groth and Sahai [GS08] proposed a way to produce efficient and practical NIZK and non-interactive witness-indistinguishable (NIWI) proofs for (algebraic) statements related to groups equipped with a bilinear map. In particular, they give proofs for the simultaneous satisfiability of a set of equations. They proposed three instantiations of their system based on different

(mild) computational assumptions: the subgroup decision problem, the symmetric external Diffie-Hellman problem (SXDH) and the decision linear problem (DLIN). Each one of these has already given rise to many applications such as [BW06,BW07,CGS07,Gro07,GL07,BCKL08,BCC+09,FPV09]. Although it is much more efficient than all previous proposals, their proof system still lacks in practicality compared to the ROM, since the verification of a single equation requires the computation of dozens of bilinear-map evaluations by the verifier.

The aim of this paper is to optimize the verification procedure at the expense of slightly weakening the soundness of the proof system.

Prior Work. In the last twenty years, there has been a lot of work in cryptography in which expensive tasks are processed in batch rather than individually to achieve better efficiency. Batch cryptography was first introduced by Fiat [Fia90], who proposed an algorithm to compute several private RSA key operations (with different exponents) through one full exponentiation and several small exponentiations. Batch cryptography is particularly relevant in settings where many exponentiations need to be verified together: many schemes were proposed to achieve batch verification of digital signatures - e.g. [NMVR94] for DSA signatures, and it seems natural to apply such techniques to the verification of Groth-Sahai proofs, which require expensive evaluations of pairings. In 1998, Bellare, Garay and Rabin [BGR98] took the first systematic look at batch verification and described several techniques for conducting batch verification of exponentiations with high confidence. They proposed three generic methods called the *random-subset test*, the *small-exponents test* and the *bucket test*. More recently, Ferrara, Green, Hohenberger and Pedersen [FGHP09] presented a detailed study on how to securely batch-verify a set of pairing-based equations and some applications on existing signatures schemes.

Our Results. The main result of the paper is a significant reduction of the cost of Groth-Sahai proof systems by using batch-verification techniques. In particular, we give efficient explicit verification procedures for the three[1] instantiations proposed in [GS08]. The essence of our approach is a trade-off between soundness and efficiency: if the verification algorithm returns valid, the verifier is assured that all proved statements are indeed valid with overwhelming probability. The best improvements are for the proofs based on SXDH and DLIN, which are the ones with most practical relevance (see Sections 5 and 6). Table 1 summarizes the number of dominant pairing operations required to verify the different algebraic statements in Groth-Sahai terminology (see Section 3 for details).

In [CHP07], Camenisch *et al.* explicitly mentioned as an "exciting" open problem the development of fast batching schemes for various forms of anonymous authentication, such as group signatures and anonymous credentials. This paper is the first to address this issue in the standard security model by considering two schemes based on the Groth-Sahai methodology.

[1] The results for the (least practical) instantiation based on the subgroup decision problem are deferred to the full version of the paper [BFI+10].

	Naive computation	Batch computation
SXDH		
Pairing-product	$5m + 3n + 16$	$m + 2n + 8$
Multi-scalar multiplication in $\mathbb{G}_1$	$8m + 2n + 14$	$\min(2n + 9, 2m + n + 7)$
Multi-scalar multiplication in $\mathbb{G}_2$	$8n + 2m + 14$	$\min(2m + 9, 2n + m + 7)$
Quadratic	$8m + 8n + 12$	$2\min(m, n) + 8$
DLIN		
Pairing-product	$12n + 27$	$3n + 6$
Multi-scalar multiplication	$9n + 12m + 27$	$3n + 3m + 6$
Quadratic	$18n + 24$	$3n + 6$

Table 1. Number of pairings per proof verification, where n and m stand for the number of different types of variables.

The first scheme we consider was proposed by Groth in 2007 [Gro07]. It is a constant-size group-signature scheme whose security can be proved in the standard model, *i.e.* without relying on the random oracle heuristic. For illustrative purposes, we concentrate on the simpler variant of the scheme that provides CPA anonymity only. Even this variant does not achieve satisfactory efficiency—the verification of a signature requires the computation of 68 expensive pairing operations. In Section 7, we propose an improved verification procedure in which the total number of bilinear-map evaluations drops to 11. In addition, if $n \geq 2$ signatures (for the same group) have to be verified at once, we manage to further decrease this number from $11n$ to $4n + 7$.

In Section 8, we study the *P-signature* scheme[2] proposed by Belenkiy, Chase, Kohlweiss and Lysyanskaya [BCKL08]. Since anonymous credentials are an immediate consequence of P-signatures, we thereby apply our techniques to privacy-preserving authentication mechanisms. Belenkiy *et al.* proposed two instantiations of their protocol (based on SXDH and DLIN). They evaluated that the verification of a proof of possession of a signature would involve respectively 68 and 128 pairing evaluations. We show that this can be reduced to 15 and 12, respectively. Moreover, the number of pairing operations required to verify $n \geq 2$ signatures is reduced to $2n+13$ and $3n+9$, respectively, by using our techniques.

2 Preliminaries

2.1 Bilinear Groups

Since Groth-Sahai proof systems apply to group-dependent languages, we summarize the basics of bilinear groups and pairing-based assumptions. In the sequel,

[2] A *P-signature* scheme is a digital-signature scheme with an additional non-interactive proof of signature possession.

we consider an algorithm $\mathcal{G}$ that, on input a security parameter λ, outputs a tuple $(N, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of order N, g_1 and g_2 generate $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, and e is an admissible bilinear map $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, which means that it is efficiently computable, $e(g_1, g_2)$ generates $\mathbb{G}_T$, and that $e(u^a, v^b) = e(u, v)^{ab}$ for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_N$.

Definition 1. *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ be a bilinear group with p prime. The* Symmetric eXternal Decision Diffie-Hellman (SXDH) assumption [ACHdM05] *states that the decision Diffie-Hellman assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$,* i.e. *the distributions (u, u^x, u^y, u^z) and $(u, u^x, u^y, u^{x \cdot y})$ are computationally indistinguishable for a random group element $u \in \mathbb{G}_i$ and random scalars $x, y, z \in \mathbb{Z}_p$ (for $i \in \{1, 2\}$).*

Definition 2. *Let $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ be a bilinear group where p is prime (and $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$). The* decision linear (DLIN) assumption [BBS04] *states that the two distributions (u, v, w, u^a, v^b, w^c) and $(u, v, w, u^a, v^b, w^{a+b})$ are computationally indistinguishable for random group elements $u, v, w \in \mathbb{G}$ and random $a, b, c \in \mathbb{Z}_p$.*

2.2 Notation

We let "$\cdot$" denote the product of two elements either in $\mathbb{Z}_N$, in $\mathbb{G}$ or in $\mathbb{G}_T$. For equal-dimension vectors or matrices A and B of group elements, $A \odot B$ stands for their entry-wise product (*i.e.* their Hadamard product). For a vector or a matrix $A = (a_{i,j})_{i,j}$ of group elements and $x \in \mathbb{Z}$, we let A^x denote the matrix $(a_{i,j}^x)_{i,j}$. Let $\Gamma = (\gamma_{i,j})_{i,j} \in \mathbb{Z}^{m \times n}$ and $\vec{\mathcal{B}} \in \mathbb{G}^n$. Then $\Gamma \vec{\mathcal{B}} := (\prod_{j=1}^n \mathcal{B}_j^{\gamma_{ij}})_{i=1}^m$. We will use $\langle \cdot, \cdot \rangle$ for bilinear products between vectors of either scalars or group elements. Let $\vec{a}, \vec{b} \in \mathbb{Z}_N^n$ and $\vec{\mathcal{A}}, \vec{\mathcal{B}} \in \mathbb{G}^n$. We define

$$\langle \vec{a}, \vec{b} \rangle := \textstyle\sum_{i=1}^n a_i \cdot b_i \qquad \langle \vec{a}, \vec{\mathcal{B}} \rangle := \prod_{i=1}^n \mathcal{B}_i^{a_i} \qquad \langle \vec{\mathcal{A}}, \vec{\mathcal{B}} \rangle := \prod_{i=1}^n e(\mathcal{A}_i, \mathcal{B}_i)$$

We employ Groth and Sahai's notation of a bilinear product (for $k \in \{2, 3\}$):

$$\bullet\colon \mathbb{G}_1^{n \times k} \times \mathbb{G}_2^{n \times k} \to \mathbb{G}_T^{k \times k}$$

defined as $\vec{\mathbf{c}} \bullet \vec{\mathbf{d}} := (\prod_{\ell=1}^n e(c_{\ell,i}, d_{\ell,j}))_{1 \le i,j \le k}$. For the case $\mathbb{G}_1 = \mathbb{G}_2$ and $k = 3$ we define a symmetric variant[3] $\overset{s}{\bullet}\colon \mathbb{G}^{n \times 3} \times \mathbb{G}^{n \times 3} \to \mathbb{G}_T^{3 \times 3}$ by:

$$\vec{\mathbf{c}} \overset{s}{\bullet} \vec{\mathbf{d}} := \left(\prod_{\ell=1}^n e(c_{\ell,i}, d_{\ell,j})^{\frac{1}{2}} e(c_{\ell,j}, d_{\ell,i})^{\frac{1}{2}} \right)_{1 \le i,j \le 3}$$

[3] Note that in their DLIN instantiation, Groth and Sahai use $\tilde{\bullet}$ for the asymmetric map and $\bullet$ for the symmetric variant.

3 Groth-Sahai Proof Systems

We sketch the results of Groth and Sahai [GS08] on proofs of satisfiability of sets of equations over a bilinear group $(N, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. Due to the complexity of their methodology, we merely give what is needed for our results and refer to the full version of [GS08] for any additional details. The three types of equations are the following:

A *pairing-product equation* over variables $\vec{\mathcal{X}} \in \mathbb{G}_1^m$ and $\vec{\mathcal{Y}} \in \mathbb{G}_2^n$ is of the form

$$\langle \vec{\mathcal{A}}, \vec{\mathcal{Y}} \rangle \cdot \langle \vec{\mathcal{X}}, \vec{\mathcal{B}} \rangle \cdot \langle \vec{\mathcal{X}}, \Gamma \vec{\mathcal{Y}} \rangle = t_T \ , \tag{1}$$

defined by constants $\vec{\mathcal{A}} \in \mathbb{G}_1^n$, $\vec{\mathcal{B}} \in \mathbb{G}_2^m$, $\Gamma \in \mathbb{Z}_N^{m \times n}$ and $t_T \in \mathbb{G}_T$.

A *multi-scalar multiplication equation* over variables $\vec{y} \in \mathbb{Z}_N^n$ and $\vec{\mathcal{X}} \in \mathbb{G}_1^m$ is of the form

$$\langle \vec{y}, \vec{\mathcal{A}} \rangle \cdot \langle \vec{b}, \vec{\mathcal{X}} \rangle \cdot \langle \vec{y}, \Gamma \vec{\mathcal{X}} \rangle = T \ , \tag{2}$$

defined by the constants $\vec{\mathcal{A}} \in \mathbb{G}_1^n$, $\vec{b} \in \mathbb{Z}_N^m$, $\Gamma \in \mathbb{Z}_N^{m \times n}$ and $T \in \mathbb{G}_1$.
A multi-scalar multiplication equation in group $\mathbb{G}_2$ is defined analogously.

A *quadratic equation in* $\mathbb{Z}_N$ over variables $\vec{x} \in \mathbb{Z}_N^m$ and $\vec{y} \in \mathbb{Z}_N^n$ is of the form

$$\langle \vec{a}, \vec{y} \rangle + \langle \vec{x}, \vec{b} \rangle + \langle \vec{x}, \Gamma \vec{y} \rangle = t \ , \tag{3}$$

defined by the constants $\vec{a} \in \mathbb{Z}_N^n$, $\vec{b} \in \mathbb{Z}_N^m$, $\Gamma \in \mathbb{Z}_N^{m \times n}$ and $t \in \mathbb{Z}_N$.

The common reference string for the proof system is a key to make commitments to the variables of the different types. A proof of satisfiability is constructed by first committing to the variables of the respective equation and then constructing a "proof" for each equation. The latter asserts that the committed values indeed satisfy the equation. There are three instantiations of the proof system described in [GS08]; we present only those based on the SXDH and the DLIN assumption (the instantiation based on the *subgroup decision* assumption is described in the full version of the paper [BFI+10]).

SXDH. The language is over a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ where p is prime. The commitment key consists of $\mathbf{u}_1 = (u_{1,1}, u_{1,2}), \mathbf{u}_2 = (u_{2,1}, u_{2,2})$ in $\mathbb{G}_1^2$ and $\mathbf{v}_1 = (v_{1,1}, v_{1,2}), \mathbf{v}_2 = (v_{2,1}, v_{2,2})$ in $\mathbb{G}_2^2$.

We write $\vec{\mathbf{u}} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{pmatrix}$ and $\vec{\mathbf{v}} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} = \begin{pmatrix} v_{1,1} & v_{1,2} \\ v_{2,1} & v_{2,2} \end{pmatrix}$.

Let $X \in \mathbb{G}_1$, $Y \in \mathbb{G}_2$ and $x \in \mathbb{Z}_p$. We define $\iota_1(X) := (1, X)$, $\iota_2(Y) := (1, Y)$, $\iota_1'(x) := (u_{2,1}^x, (u_{2,2} g_1)^x)$ and $\iota_2'(x) := (v_{2,1}^x, (v_{2,2} g_2)^x)$. To commit to $X \in \mathbb{G}_1$, one chooses randomness $s_1, s_2 \in \mathbb{Z}_p$ and sets $\mathbf{c}_X := \iota_1(X) \odot \mathbf{u}_1^{s_1} \odot \mathbf{u}_2^{s_2}$, a commitment to $Y \in \mathbb{G}_2$ is defined as $\mathbf{d}_Y := \iota_2(Y) \odot \mathbf{v}_1^{s_1} \odot \mathbf{v}_2^{s_2}$. To make a commitment to $x \in \mathbb{Z}_p$ in $\mathbb{G}_1^2$ one chooses $s \in \mathbb{Z}_p$ and sets $\mathbf{c}_x := \iota_1'(x) \odot \mathbf{u}_1^s$, a commitment in $\mathbb{G}_2^2$ is defined as $\mathbf{d}_x := \iota_2'(x) \odot \mathbf{v}_1^s$.

To show satisfiability of a set of equations of the form (1), (2) or (3), one first makes commitments to a satisfying *witness* (*i.e.* an assignment to the variables of

each equation) and then adds a "proof" per equation. Groth and Sahai describe how to construct these; for Type (1), they are in $\mathbb{G}_2^{2\times 2} \times \mathbb{G}_1^{2\times 2}$, for Type (2) they are in $\mathbb{G}_2^{2\times 2} \times \mathbb{G}_1^2$ and for Type (3) in $\mathbb{G}_2^2 \times \mathbb{G}_1^2$.

The verification relations for the proofs are given in Section 5, where we also discuss how to optimize them. For convenience we define some notations. Let $t \in \mathbb{Z}_p$, $T_1 \in \mathbb{G}_1$, $T_2 \in \mathbb{G}_2$ and $t_T \in \mathbb{G}_T$. Then we let[4]

$$\iota_T(t_T) := \begin{pmatrix} 1 & 1 \\ 1 & t_T \end{pmatrix},\ \hat{\iota}_T(T_1) := \begin{pmatrix} 1 & 1 \\ e(T_1, v_{2,1}) & e(T_1, v_{2,2}g_2) \end{pmatrix},\ \hat{\iota}_T(T_2) := \begin{pmatrix} 1 & e(u_{2,1}, T_2) \\ 1 & e(u_{2,2}g_1, T_2) \end{pmatrix},$$

$$\text{and } \iota'_T(t) := \left[(u_{2,1}, u_{2,2}g_1)\bullet(v_{2,1}, v_{2,2}g_2)\right]^t = \begin{pmatrix} e(u_{2,1}, v_{2,1})^t & e(u_{2,1}, v_{2,2}g_2)^t \\ e(u_{2,2}g_1, v_{2,1})^t & e(u_{2,2}g_1, v_{2,2}g_2)^t \end{pmatrix}.$$

For the sake of consistency with [GS08], for $\mathbf{c} \in \mathbb{G}_1^{1\times 2}$ and $\mathbf{d} \in \mathbb{G}_2^{1\times 2}$ we denote $F(\mathbf{c}, \mathbf{d}) := [\mathbf{c} \bullet \mathbf{d}]$.

DLIN. In this instantiation, the language is over a bilinear (symmetric) group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ with p prime. The commitment key $\vec{\mathbf{u}} \in \mathbb{G}^{3\times 3}$ is of the form $\mathbf{u}_1 = (u_{1,1}, 1, g)$, $\mathbf{u}_2 = (1, u_{2,1}, g)$, $\mathbf{u}_3 = (u_{3,1}, u_{3,2}, u_{3,3})$. Let $X \in \mathbb{G}$ and $x \in \mathbb{Z}_p$. We define $\iota(X) := (1, 1, X)$ and $\iota'(x) := (u_{3,1}^x, u_{3,2}^x, (u_{3,3}g)^x)$. To commit to $X \in \mathbb{G}$, choose randomness $s_1, s_2, s_3 \in \mathbb{Z}_p$ and set $\mathbf{c}_X := \iota(X) \odot \mathbf{u}_1^{s_1} \odot \mathbf{u}_2^{s_2} \odot \mathbf{u}_3^{s_3}$. To commit to $x \in \mathbb{Z}_p$, choose $s_1, s_2 \in \mathbb{Z}_p$ and set $\mathbf{c}_x := \iota'(x) \odot \mathbf{u}_1^{s_1} \odot \mathbf{u}_3^{s_2}$.

Due to the fact that $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ in this setting, the equations (1), (2) and (3) simplify to the following respective equations:

$$\langle \vec{\mathcal{A}}, \vec{\mathcal{Y}} \rangle \cdot \langle \vec{\mathcal{Y}}, \Gamma\vec{\mathcal{Y}} \rangle = t_T \tag{1'}$$

$$\langle \vec{a}, \vec{\mathcal{Y}} \rangle \cdot \langle \vec{x}, \vec{\mathcal{B}} \rangle \cdot \langle \vec{x}, \Gamma\vec{\mathcal{Y}} \rangle = T \tag{2'}$$

$$\langle \vec{x}, \vec{b} \rangle + \langle \vec{x}, \Gamma\vec{x} \rangle = t \tag{3'}$$

Groth and Sahai show how to construct "proofs" for each type of equation, where for Types (1') and (2'), the proof is in $\mathbb{G}^{3\times 3}$, whereas for Type (3') it is in $\mathbb{G}^{2\times 3}$. The verification relations for the proofs are given in Section 6. We define the following notations. Let $t \in \mathbb{Z}_p$, $T \in \mathbb{G}$ and $t_T \in \mathbb{G}_T$. Then we let

$$\iota_T(t_T) := \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & t_T \end{pmatrix} \qquad \hat{\iota}_T(T) := \begin{pmatrix} 1 & 1 & e(u_{3,1}, T)^{\frac{1}{2}} \\ 1 & 1 & e(u_{3,2}, T)^{\frac{1}{2}} \\ e(u_{3,1}, T)^{\frac{1}{2}} & e(u_{3,2}, T)^{\frac{1}{2}} & e(u_{3,3}g, T) \end{pmatrix}$$

and $\iota'_T(t) := \left[(u_{3,1}, u_{3,2}, u_{3,3}g) \overset{s}{\bullet} (u_{3,1}, u_{3,2}, u_{3,3}g)\right]^t$.

4 Batch Verification of Pairing Equations

We address the problem of securely batching the verification of (potentially many) Groth-Sahai proofs. We achieve a trade-off between soundness and efficiency: if the verification algorithm returns valid, the verifier is assured that

[4] Here (and in the DLIN instantiation) we use the rectifications of $\hat{\iota}_T$ and ι'_T by [GSW09].

all proved statements are valid with overwhelming probability. Ferrara, Green, Hohenberger and Pedersen [FGHP09] presented a detailed study on how to securely batch-verify a set of pairing-based equations, which we briefly recall here (see the full version of [FGHP09] for any additional details).

Given a bilinear structure $(N, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$, a *pairing-based verification equation* is a Boolean relation of the form: $\prod_{i=1}^{k} e(f_i, h_i)^{c_i} \stackrel{?}{=} A$ for $k \in \mathbb{N}$, $(f_i, h_i, c_i) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{Z}_N$ for $i \in \{1, \ldots, k\}$ and $A \in \mathbb{G}_T$. A *pairing-based verifier* is an algorithm which given a pairing-based verification equation outputs *yes* if the Boolean relation holds, and *no* otherwise (except with negligible probability).

In order to design a pairing-based verifier for m pairing-based verification equations, one has to find a way to combine all equations. The technique proposed in [FGHP09] consists in using the *small exponents test* proposed by Bellare et al. [BGR98], which here amounts to pick small random exponents $\delta_1, \ldots, \delta_m$ and checking whether $\prod_{j=1}^{m} \prod_{i=1}^{k_j} e(f_{i,j}, h_{i,j})^{c_{i,j}\delta_j} = \prod_{j=1}^{m} A_j^{\delta_j}$ holds. In order to further reduce the computational needs, three main techniques may be used:

1. **Move the exponent into the pairing:** Since, in practice, exponentiation in $\mathbb{G}_T$ is more expensive[5] than in $\mathbb{G}_1$ and $\mathbb{G}_2$, this gives a first speed up. As we are working on pairings, we can also do the opposite if it allows another technique to apply: $e(f_i, h_i)^{\delta_i} \rightarrow e(f_i^{\delta_i}, h_i)$
2. **Move the product into the pairing:** When two pairings have a common element, they can be combined to reduce the number of pairings:
$$\prod_{j=1}^{m} e(f_j^{\delta_j}, h_i) \rightarrow e\left(\prod_{j=1}^{m} f_j^{\delta_j},\ h_i\right)$$
3. **Switch two products:** Sometimes improvements can be made by moving a product from the first to the second component of a pairing (or vice-versa):
$$\prod_{i=1}^{k} e\left(\prod_{j=1}^{m} f_j^{\delta_{i,j}},\ h_i\right) \leftrightarrow \prod_{j=1}^{m} e\left(f_j,\ \prod_{i=1}^{k} h_i^{\delta_{i,j}}\right)$$

The soundness of the pairing-based verifier based on the small exponents test is quantified in the following theorem [FGHP09, Theorem 3.2]:

Theorem 1. *Given m pairing-based verification equations, the small-exponents verifier described above with random exponents $\delta_1, \ldots, \delta_m$ of ℓ bits is a pairing-based batch verifier that accepts an invalid batch with probability at most $2^{-\ell}$.*

Handling Invalid Proofs. In the case of verification of multiple proofs (as in Sections 7 and 8), if there is an invalid proof in the batch, then the verifier will reject the entire batch with high probability. A simple technique for finding invalid proofs in a batch consists in using a recursive *divide-and-conquer* approach [PMPS00]. Recently, more efficient techniques were proposed for pairing-based signatures (see e.g. [Mat09] and references therein) and they apply as well to our setting.

[5] Note that, for Type 2 pairings, exponentiation in $\mathbb{G}_2$ is more expensive than in $\mathbb{G}_T$ (see [GPS08] for details).

5 Instantiation 2: SXDH

5.1 Pairing-Product Equation

A proof $(\vec{\mathbf{c}}, \vec{\mathbf{d}}, \vec{\pi}, \vec{\theta}) \in \mathbb{G}_1^{m\times 2} \times \mathbb{G}_2^{n\times 2} \times \mathbb{G}_2^{2\times 2} \times \mathbb{G}_1^{2\times 2}$ of satisfiability of an equation of Type (1) is verified by checking the following equation [GS08]:

$$\left[\iota_1(\vec{\mathcal{A}}) \bullet \vec{\mathbf{d}}\right] \odot \left[\vec{\mathbf{c}} \bullet \iota_2(\vec{\mathcal{B}})\right] \odot \left[\vec{\mathbf{c}} \bullet \Gamma \vec{\mathbf{d}}\right] = \iota_T(t_T) \odot \left[\vec{\mathbf{u}} \bullet \vec{\pi}\right] \odot \left[\vec{\theta} \bullet \vec{\mathbf{v}}\right] .$$

Let $\vec{\mathbf{c}} = (c_{i,k})_{\substack{1\le i\le m\\ 1\le k\le 2}} \in \mathbb{G}_1^{m\times 2}$, $\vec{\mathbf{d}} = (d_{j,k})_{\substack{1\le j\le n\\ 1\le k\le 2}} \in \mathbb{G}_2^{n\times 2}$, $\Gamma = (\gamma_{i,j})_{\substack{1\le i\le m\\ 1\le j\le n}} \in \mathbb{Z}_p^{m\times n}$, $\vec{\mathcal{A}} = (\mathcal{A}_j)_{1\le j\le n} \in \mathbb{G}_1^{n\times 1}$ and $\vec{\mathcal{B}} = (\mathcal{B}_i)_{1\le i\le m} \in \mathbb{G}_2^{m\times 1}$.

Plugging in the definitions from Section 3, the left hand side is equal to

$$\begin{pmatrix} \prod_{i=1}^{m} e\left(c_{i,1}, \prod_{j=1}^{n} d_{j,1}^{\gamma_{i,j}}\right) & \prod_{i=1}^{m} e\left(c_{i,1}, \mathcal{B}_i \prod_{j=1}^{n} d_{j,2}^{\gamma_{i,j}}\right) \\ \prod_{j=1}^{n} e\left(\mathcal{A}_j \prod_{i=1}^{m} c_{i,2}^{\gamma_{i,j}}, d_{j,1}\right) & \prod_{j=1}^{n} e(\mathcal{A}_j, d_{j,2}) \prod_{i=1}^{m} e\left(c_{i,2}, \mathcal{B}_i \prod_{j=1}^{n} d_{j,2}^{\gamma_{i,j}}\right) \end{pmatrix} .$$

If we denote $\vec{\pi} = \begin{pmatrix} \pi_{1,1} & \pi_{1,2} \\ \pi_{2,1} & \pi_{2,2} \end{pmatrix}$, $\vec{\theta} = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \end{pmatrix}$, the right hand side is equal to

$$\begin{pmatrix} e(u_{1,1},\pi_{1,1})e(u_{2,1},\pi_{2,1}) & e(u_{1,1},\pi_{1,2})e(u_{2,1},\pi_{2,2}) \\ \quad \cdot e(\theta_{1,1},v_{1,1})e(\theta_{2,1},v_{2,1}) & \quad \cdot e(\theta_{1,1},v_{1,2})e(\theta_{2,1},v_{2,2}) \\ e(u_{1,2},\pi_{1,1})e(u_{2,2},\pi_{2,1}) & t_T e(u_{1,2},\pi_{1,2})e(u_{2,2},\pi_{2,2}) \\ \quad \cdot e(\theta_{1,2},v_{1,1})e(\theta_{2,2},v_{2,1}) & \quad \cdot e(\theta_{1,2},v_{1,2})e(\theta_{2,2},v_{2,2}) \end{pmatrix} .$$

By grouping pairings, we reduced the number of pairings on the left-hand side of the equation from $5m + 3n$ to $3m + 2n$, while the right-hand side remains at 16 pairings. Using the techniques explained in Section 4, *i.e.* taking each element $M_{i,j}$ of the equation to a random power $r_{i,j}$, multiplying all the components, and regrouping pairings, we get the following equation:

$$\begin{aligned} &\prod_{k=1}^{2}\prod_{j=1}^{n} e\left(\left(\textstyle\prod_{i=1}^{m} c_{i,1}^{\gamma_{i,j}}\right)^{r_{1,k}} \left(\mathcal{A}_j \prod_{i=1}^{m} c_{i,2}^{\gamma_{i,j}}\right)^{r_{2,k}}, d_{j,k}\right) \cdot \prod_{i=1}^{m} e\left(c_{i,1}^{r_{1,2}} c_{i,2}^{r_{2,2}}, \mathcal{B}_i\right) \\ &= e(u_{1,1}^{r_{1,1}} u_{1,2}^{r_{2,1}}, \pi_{1,1}) e(u_{2,1}^{r_{1,1}} u_{2,2}^{r_{2,1}}, \pi_{2,1}) e(\theta_{1,1}^{r_{1,1}} \theta_{1,2}^{r_{2,1}}, v_{1,1}) e(\theta_{2,1}^{r_{1,1}} \theta_{2,2}^{r_{2,1}}, v_{2,1}) \\ &\qquad \cdot e(u_{1,1}^{r_{1,2}} u_{1,2}^{r_{2,2}}, \pi_{1,2}) e(u_{2,1}^{r_{1,2}} u_{2,2}^{r_{2,2}}, \pi_{2,2}) e(\theta_{1,1}^{r_{1,2}} \theta_{1,2}^{r_{2,2}}, v_{1,2}) e(\theta_{2,1}^{r_{1,2}} \theta_{2,2}^{r_{2,2}}, v_{2,2}) \cdot t_T^{r_{2,2}} \end{aligned}$$

which requires $m + 2n$ pairings and $2mn + 2m + 4n$ exponentiations in $\mathbb{G}_1$ for the left part and 8 pairing computations and 16 exponentiations in $\mathbb{G}_1$ and one exponentiation in $\mathbb{G}_T$ for the right side of the equation. The alternative expression

$$\prod_{j=1}^{n} e\left(\mathcal{A}_j, d_{j,1}^{r_{2,1}} d_{j,2}^{r_{2,2}}\right) \cdot \prod_{k=1}^{2}\prod_{i=1}^{m} e\left(c_{i,k}, \left(\prod_{j=1}^{n} d_{j,1}^{\gamma_{i,j}}\right)^{r_{k,1}} \left(\mathcal{B}_i \prod_{j=1}^{n} d_{j,2}^{\gamma_{i,j}}\right)^{r_{k,2}}\right)$$

for the left side of the equation requires $2m + n$ pairings and $2mn + 4m + 2n$ exponentiations in $\mathbb{G}_2$.

5.2 Multi-Scalar Multiplication Equation in $\mathbb{G}_1$

Here, we consider equations of Type (2) in $\mathbb{G}_1$ (the case of equations in $\mathbb{G}_2$, which work analogously, is treated in the full version of the paper [BFI+10]). The verification of a proof $(\vec{\mathbf{c}}, \vec{\mathbf{d}}', \vec{\pi}, \theta) \in \mathbb{G}_1^{m\times 2} \times \mathbb{G}_2^{n\times 2} \times \mathbb{G}_2^{2\times 2} \times \mathbb{G}_1^{1\times 2}$ consists in checking the following:

$$\left[\iota_1(\vec{\mathcal{A}}) \bullet \vec{\mathbf{d}}'\right] \odot \left[\vec{\mathbf{c}} \bullet \iota_2'(\vec{b})\right] \odot \left[\vec{\mathbf{c}} \bullet \Gamma \vec{\mathbf{d}}'\right] = \hat{\iota_T}(\mathcal{T}_1) \odot \left[\vec{\mathbf{u}} \bullet \vec{\pi}\right] \odot F(\theta, \mathbf{v}_1).$$

Let $\vec{\mathbf{c}} = (c_{i,k})_{\substack{1\le i\le m\\ 1\le k\le 2}} \in \mathbb{G}_1^{m\times 2}$, $\vec{\mathbf{d}}' = (d'_{j,k})_{\substack{1\le j\le n\\ 1\le k\le 2}} \in \mathbb{G}_2^{n\times 2}$, $\Gamma = (\gamma_{i,j})_{\substack{1\le i\le m\\ 1\le j\le n}} \in \mathbb{Z}_p^{m\times n}$, $\vec{\mathcal{A}} = (\mathcal{A}_j)_{1\le j\le n} \in \mathbb{G}_1^{n\times 1}$, $\vec{b} = (b_i)_{1\le i\le m} \in \mathbb{Z}_p^{m\times 1}$. The left hand-side is equal to

$$\begin{pmatrix} \prod_{i=1}^{m} e(c_{i,1}, v_{2,1}^{b_i} \prod_{j=1}^{n} {d'_{j,1}}^{\gamma_{i,j}}) & \prod_{i=1}^{m} e(c_{i,1}, (v_{2,2}g_2)^{b_i} \prod_{j=1}^{n} {d'_{j,2}}^{\gamma_{i,j}}) \\ \prod_{i=1}^{m} e(c_{i,2}, v_{2,1}^{b_i} \prod_{j=1}^{n} {d'_{j,1}}^{\gamma_{i,j}}) & \prod_{i=1}^{m} e(c_{i,2}, (v_{2,2}g_2)^{b_i} \prod_{j=1}^{n} {d'_{j,2}}^{\gamma_{i,j}}) \\ \cdot \prod_{j=1}^{n} e(\mathcal{A}_j, d'_{j,1}) & \cdot \prod_{j=1}^{n} e(\mathcal{A}_j, d'_{j,2}) \end{pmatrix}$$

while the right-hand side is equal to

$$\begin{pmatrix} e(\theta_1, v_{1,1})e(u_{1,1}, \pi_{1,1})e(u_{2,1}, \pi_{2,1}) & e(\theta_1, v_{1,2})e(u_{1,1}, \pi_{1,2})e(u_{2,1}, \pi_{2,2}) \\ e(\theta_2, v_{1,1})e(u_{1,2}, \pi_{1,1})e(u_{2,2}, \pi_{2,1}) & e(\theta_2, v_{1,2})e(u_{1,2}, \pi_{1,2})e(u_{2,2}, \pi_{2,2}) \\ \cdot e(\mathcal{T}_1, v_{2,1}) & \cdot e(\mathcal{T}_1, g_2 v_{2,2}) \end{pmatrix}$$

By grouping the pairings, the number of pairings on the left-hand side of the equation has already been reduced from $8m + 2n$ to $4m + 2n$. Now, by using the batch technique, i.e., multiplying each member by a random value and multiplying all the components, we obtain on the left-hand side

$$\prod_{k=1}^{2}\prod_{j=1}^{n} e\left(\left(\prod_{i=1}^{m} c_{i,1}^{\gamma_{i,j}}\right)^{r_{1,k}} \left(\mathcal{A}_j \prod_{i=1}^{m} c_{i,2}^{\gamma_{i,j}}\right)^{r_{2,k}}, d'_{j,k}\right)$$
$$\cdot e\left(\left(\prod_{i=1}^{m} c_{i,1}^{b_i}\right)^{r_{1,1}} \left(\prod_{i=1}^{m} c_{i,2}^{b_i}\right)^{r_{2,1}}, v_{2,1}\right) \cdot e\left(\left(\prod_{i=1}^{m} c_{i,1}^{b_i}\right)^{r_{1,2}} \left(\prod_{i=1}^{m} c_{i,2}^{b_i}\right)^{r_{2,2}}, v_{2,2}g_2\right)$$

which requires $2mn + 2m + 4n + 4$ exponentiations in $\mathbb{G}_1$ and $2n + 2$ pairing computations. The alternative expression

$$\prod_{j=1}^{n} e\Big(\mathcal{A}_j, \prod_{k=1}^{2} {d'_{j,k}}^{r_{2,k}}\Big) \prod_{k=1}^{2}\prod_{i=1}^{m} e\Big(c_{i,k}, \Big(v_{2,1}^{b_i} \prod_{j=1}^{n} {d'_{j,1}}^{\gamma_{i,j}}\Big)^{r_{k,1}} \Big((v_{2,2}g_2)^{b_i} \prod_{j=1}^{n} {d'_{j,2}}^{\gamma_{i,j}}\Big)^{r_{k,2}}\Big)$$

for the left side of the equation requires $2mn + 6m + 2n$ exponentiations in $\mathbb{G}_2$ and $2m + n$ pairing computations. On the right-hand side, the same technique achieves a reduction from 14 to 7 pairings:

$$e(\theta_1^{r_{1,1}}\theta_2^{r_{2,1}}, v_{1,1})e(\theta_1^{r_{2,1}}\theta_2^{r_{2,2}}, v_{1,2})e(u_{1,1}^{r_{1,1}}u_{1,2}^{r_{2,1}}, \pi_{1,1})e(u_{2,1}^{r_{1,1}}u_{2,2}^{r_{2,1}}, \pi_{2,1})$$
$$\cdot\ e(u_{2,1}^{r_{1,2}}u_{2,2}^{r_{2,2}}, \pi_{1,2})e(u_{2,1}^{r_{2,1}}u_{2,2}^{r_{2,2}}, \pi_{2,2})e(\mathcal{T}_1, v_{2,1}^{r_{2,1}}(g_2v_{2,2})^{r_{2,2}})$$

5.3 Quadratic Equation

The verification of $(\vec{\mathbf{c}}', \vec{\mathbf{d}}', \pi, \theta) \in \mathbb{G}_1^{m\times 2} \times \mathbb{G}_2^{n\times 2} \times \mathbb{G}_2^{1\times 2} \times \mathbb{G}_1^{1\times 2}$ for an equation of Type (3) consists in checking

$$\left[\iota_1'(\vec{a}) \bullet \vec{\mathbf{d}}'\right] \odot \left[\vec{\mathbf{c}}' \bullet \iota_2'(\vec{b})\right] \odot \left[\vec{\mathbf{c}}' \bullet \Gamma\vec{\mathbf{d}}'\right] = \iota_T'(t) \odot F(\mathbf{u}_1, \pi) \odot F(\theta, \mathbf{v}_1) \ .$$

Let $\vec{\mathbf{c}}' = (c'_{i,k})_{\substack{1\le i\le m\\ 1\le k\le 2}} \in \mathbb{G}_1^{n\times 2}$, $\vec{\mathbf{d}}' = (d'_{j,k})_{\substack{1\le j\le n\\ 1\le k\le 2}} \in \mathbb{G}_2^{n\times 2}$, $\Gamma = (\gamma_{i,j})_{\substack{1\le i\le m\\ 1\le j\le n}} \in \mathbb{Z}_p^{m\times n}$, $\vec{a} = (a_j)_{1\le j\le n} \in \mathbb{Z}_p^{n\times 1}$, $\vec{b} = (b_i)_{1\le i\le m} \in \mathbb{Z}_p^{m\times 1}$. The left hand side is equal to

$$\begin{pmatrix} \prod_{i=1}^{m} e(c'_{i,1}, v_{2,1}^{b_i}) & \prod_{i=1}^{m} e(c'_{i,1}, (v_{2,2}g_2)^{b_i}) \\ \quad \cdot \prod_{j=1}^{n} e\left(u_{2,1}^{a_j} \prod_{i=1}^{m} {c'_{i,1}}^{\gamma_{i,j}}, d'_{j,1}\right) & \quad \cdot \prod_{j=1}^{n} e\left(u_{2,1}^{a_j} \prod_{i=1}^{m} {c'_{i,1}}^{\gamma_{i,j}}, d'_{j,2}\right) \\ \prod_{i=1}^{m} e(c'_{i,2}, v_{2,1}^{b_i}) & \prod_{i=1}^{m} e(c'_{i,2}, (v_{2,2}g_2)^{b_i}) \\ \cdot \prod_{j=1}^{n} e\left((u_{2,2}g_1)^{a_j} \prod_{i=1}^{m} {c'_{i,2}}^{\gamma_{i,j}}, d'_{j,1}\right) & \cdot \prod_{j=1}^{n} e\left((u_{2,2}g_1)^{a_j} \prod_{i=1}^{m} {c'_{i,2}}^{\gamma_{i,j}}, d'_{j,2}\right) \end{pmatrix}$$

Denoting $\pi = (\pi_1, \pi_2)$ and $\theta = (\theta_1, \theta_2)$, for the right-hand side we have

$$\begin{pmatrix} e(u_{1,1}, \pi_1)e(\theta_1, v_{1,1})e(u_{2,1}, v_{2,1})^t & e(u_{1,1}, \pi_2)e(\theta_1, v_{1,2})e(u_{2,1}, v_{2,2}g_2)^t \\ e(u_{1,2}, \pi_1)e(\theta_2, v_{1,1})e(u_{2,2}g_1, v_{2,1})^t & e(u_{1,2}, \pi_2)e(\theta_2, v_{1,2})e(u_{2,2}g_1, v_{2,2}g_2)^t \end{pmatrix}$$

By grouping the pairings, the number of pairings on the left-hand side member of the equation has been reduced from $8m+8n$ to $4m+4n$. By using the batch technique, i.e., multiplying each member by a random value and multiplying all the members, we obtain on the left-hand side:

$$e\Big(\Big(\prod_{i=1}^{m} {c'_{i,1}}^{b_i}\Big)^{r_{1,1}}\Big(\prod_{i=1}^{m} {c'_{i,2}}^{b_i}\Big)^{r_{2,1}}, v_{2,1}\Big) \cdot e\Big(\Big(\prod_{i=1}^{m} {c'_{i,1}}^{b_i}\Big)^{r_{1,2}}\Big(\prod_{i=1}^{m} {c'_{i,2}}^{b_i}\Big)^{r_{2,2}}, v_{2,2}g_2\Big)$$
$$\cdot \prod_{k=1}^{2}\prod_{j=1}^{n} e\Big(\left(u_{2,1}^{a_j}\prod_{i=1}^{m} {c'_{i,1}}^{\gamma_{i,j}}\right)^{r_{1,k}}\Big((u_{2,2}g_1)^{a_j}\prod_{i=1}^{m} {c'_{i,2}}^{\gamma_{i,j}}\Big)^{r_{2,k}}, d'_{j,k}\Big)$$

which requires $2mn + 2m + 6n + 4$ exponentiations in $\mathbb{G}_1$ and $2n + 2$ pairing computations. Alternatively, the left-hand side is also equal to

$$e\Big(u_{2,1}, \Big(\prod_{j=1}^{n} {d'_{j,1}}^{a_j}\Big)^{r_{1,1}}\Big(\prod_{j=1}^{n} {d'_{j,2}}^{a_j}\Big)^{r_{1,2}}\Big) \cdot e\Big(u_{2,2}g_2, \Big(\prod_{j=1}^{n} {d'_{j,1}}^{a_j}\Big)^{r_{2,1}}\Big(\prod_{j=1}^{n} {d'_{j,2}}^{a_j}\Big)^{r_{2,2}}\Big)$$
$$\cdot \prod_{k=1}^{2}\prod_{i=1}^{m} e\Big(c'_{i,k}, \Big(v_{2,1}^{b_i}\prod_{j=1}^{n} {d'_{j,1}}^{\gamma_{i,j}}\Big)^{r_{k,1}}\Big((v_{2,2}g_2)^{b_i}\prod_{j=1}^{n} {d'_{j,2}}^{\gamma_{i,j}}\Big)^{r_{k,2}}\Big)$$

which requires $2mn + 6m + 2n + 4$ exponentiations in $\mathbb{G}_2$ and $2m + 2$ pairing computations. On the right-hand side, the same technique achieves a reduction

from 12 to 6 pairings:

$$e(u_{1,1}^{r_{1,1}}u_{1,2}^{r_{2,1}},\pi_1)e(u_{1,1}^{r_{1,2}}u_{1,2}^{r_{2,2}},\pi_2)e(\theta_1^{r_{1,1}}\theta_2^{r_{2,1}},v_{1,1})e(\theta_1^{r_{1,2}}\theta_2^{r_{2,2}},v_{1,2}) \\ \cdot\ e({u_{2,1}^{r_{1,1}}}^t(u_{2,2}g_1)^{r_{2,1}t},v_{2,1})e({u_{2,1}^{r_{1,2}}}^t(u_{2,2}g_1)^{r_{2,2}t},v_{2,2}g_2)$$

6 Instantiation 3: DLIN

6.1 Pairing-Product Equation

The verification relation of a proof $(\vec{\mathbf{d}},\phi)\in\mathbb{G}^{n\times 3}\times\mathbb{G}^{3\times 3}$ for equation Type $(1')$ is the following:

$$\left[\iota(\vec{\mathcal{A}})\overset{s}{\bullet}\vec{\mathbf{d}}\right]\odot\left[\vec{\mathbf{d}}\overset{s}{\bullet}\Gamma\vec{\mathbf{d}}\right]=\iota_T(t_T)\odot\left[\vec{\mathbf{u}}\overset{s}{\bullet}\vec{\phi}\right]$$

For simplicity, we consider the squares of all $\mathbb{G}_T$ elements on both sides of the equation. Writing $\Gamma\vec{\mathbf{d}}$ as $\left(\prod_{k=1}^n d_{k,j}^{\gamma_{i,k}}\right)_{\substack{1\le i\le n\\1\le j\le 3}}$ and replacing the bilinear product $\overset{s}{\bullet}$ by its definition, we get for the left-hand side:

$$\begin{pmatrix}
\prod_{i=1}^n e(d_{i,1},\prod d_{k,1}^{\gamma_{i,k}})^2 & \prod_{i=1}^n e(d_{i,1},\prod d_{k,2}^{\gamma_{i,k}}) & \prod_{i=1}^n e(\mathcal{A}_i,d_{i,1})e(d_{i,1},\prod d_{k,3}^{\gamma_{i,k}}) \\
 & \cdot e(d_{i,2},\prod d_{k,1}^{\gamma_{i,k}}) & \cdot e(d_{i,3},\prod d_{k,1}^{\gamma_{i,k}}) \\
\prod_{i=1}^n e(d_{i,2},\prod d_{k,1}^{\gamma_{i,k}}) & \prod_{i=1}^n e(d_{i,2},\prod d_{k,2}^{\gamma_{i,k}})^2 & \prod_{i=1}^n e(\mathcal{A}_i,d_{i,2})e(d_{i,2},\prod d_{k,3}^{\gamma_{i,k}}) \\
\cdot e(d_{i,1},\prod d_{k,2}^{\gamma_{i,k}}) & & \cdot e(d_{i,3},\prod d_{k,2}^{\gamma_{i,k}}) \\
\prod_{i=1}^n e(\mathcal{A}_i,d_{i,1}) & \prod_{i=1}^n e(\mathcal{A}_i,d_{i,2}) & \prod_{i=1}^n e(\mathcal{A}_i,d_{i,3})^2 \\
\cdot e(d_{i,3},\prod d_{k,1}^{\gamma_{i,k}}) & \cdot e(d_{i,3},\prod d_{k,2}^{\gamma_{i,k}}) & \cdot e(d_{i,3},\prod d_{k,3}^{\gamma_{i,k}})^2 \\
\cdot e(d_{i,1},\prod d_{k,3}^{\gamma_{i,k}}) & \cdot e(d_{i,2},\prod d_{k,3}^{\gamma_{i,k}}) &
\end{pmatrix}$$

For the right-hand side, we get:

$$\begin{pmatrix}
\prod_{i=1}^3 e(u_{i1},\phi_{i1})^2 & \prod_{i=1}^3 e(u_{i1},\phi_{i2})e(u_{i2},\phi_{i1}) & \prod_{i=1}^3 e(u_{i1},\phi_{i3})e(u_{i3},\phi_{i1}) \\
\prod_{i=1}^3 e(u_{i2},\phi_{i1})e(u_{i1},\phi_{i2}) & \prod_{i=1}^3 e(u_{i2},\phi_{i2})^2 & \prod_{i=1}^3 e(u_{i2},\phi_{i3})e(u_{i3},\phi_{i2}) \\
\prod_{i=1}^3 e(u_{i3},\phi_{i1})e(u_{i1},\phi_{i3}) & \prod_{i=1}^3 e(u_{i3},\phi_{i2})e(u_{i2},\phi_{i3}) & t_T^2\prod_{i=1}^3 e(u_{i3},\phi_{i3})^2
\end{pmatrix}$$

Taking each component $M_{i,j}$ of the equation to the power of $r_{i,j}$, multiplying all components, and regrouping pairings, we get the following for the left-hand side:

$$\prod_{i=1}^{n} e\big(d_{i,1},\ \mathcal{A}_i^{r_{1,3}+r_{3,1}} \prod d_{k,1}^{\gamma_{i,k} 2\cdot r_{1,1}} d_{k,2}^{\gamma_{i,k}(r_{1,2}+r_{2,1})} d_{k,3}^{\gamma_{i,k}(r_{1,3}+r_{3,1})}\big) \cdot$$
$$e\big(d_{i,2},\ \mathcal{A}_i^{r_{2,3}+r_{3,2}} \prod d_{k,1}^{\gamma_{i,k}(r_{1,2}+r_{2,1})} d_{k,2}^{\gamma_{i,k} 2\cdot r_{2,2}} d_{k,3}^{\gamma_{i,k}(r_{2,3}+r_{3,2})}\big) \cdot$$
$$e\big(d_{i,3},\ \mathcal{A}_i^{2\cdot r_{3,3}} \prod d_{k,1}^{\gamma_{i,k}(r_{1,3}+r_{3,1})} d_{k,2}^{\gamma_{i,k}(r_{2,3}+r_{3,2})} d_{k,3}^{\gamma_{i,k} 2 r_{3,3}}\big) \quad (4)$$

and for the right-hand side:

$$\prod_{i=1}^{3} e\big(u_{i,1},\ \phi_{i,1}^{2\cdot r_{1,1}} \phi_{i,2}^{r_{1,2}+r_{2,1}} \phi_{i,3}^{r_{1,3}+r_{3,1}}\big) \cdot e\big(u_{i,2},\ \phi_{i,1}^{r_{1,2}+r_{2,1}} \phi_{i,2}^{2\cdot r_{2,2}} \phi_{i,3}^{r_{2,3}+r_{3,2}}\big)$$
$$\cdot e\big(u_{i,3},\ \phi_{i,1}^{r_{1,3}+r_{3,1}} \phi_{i,2}^{r_{2,3}+r_{3,2}} \phi_{i,3}^{2\cdot r_{3,3}}\big) \cdot t_T^{2 r_{3,3}}$$

Due to the fact that $u_{1,2} = u_{2,1} = 1$, and $u_{1,3} = u_{2,3}$ (*cf.* Section 3) this simplifies to:

$$e\big(u_{1,1},\ \phi_{1,1}^{2\cdot r_{1,1}} \phi_{1,2}^{r_{1,2}+r_{2,1}} \phi_{1,3}^{r_{1,3}+r_{3,1}}\big) \cdot e\big(u_{2,2},\ \phi_{2,1}^{r_{1,2}+r_{2,1}} \phi_{2,2}^{2\cdot r_{2,2}} \phi_{2,3}^{r_{2,3}+r_{3,2}}\big)$$
$$\cdot e\big(u_{1,3},\ (\phi_{1,1}\phi_{2,1})^{r_{1,3}+r_{3,1}} (\phi_{1,2}\phi_{2,2})^{r_{2,3}+r_{3,2}} (\phi_{1,3}\phi_{2,3})^{2\cdot r_{3,3}}\big)$$
$$\cdot e\big(u_{3,1},\ \phi_{3,1}^{2\cdot r_{1,1}} \phi_{3,2}^{r_{1,2}+r_{2,1}} \phi_{3,3}^{r_{1,3}+r_{3,1}}\big) \cdot e\big(u_{3,2},\ \phi_{3,1}^{r_{1,2}+r_{2,1}} \phi_{3,2}^{2\cdot r_{2,2}} \phi_{3,3}^{r_{2,3}+r_{3,2}}\big) \cdot$$
$$e\big(u_{3,3},\ \phi_{3,1}^{r_{1,3}+r_{3,1}} \phi_{3,2}^{r_{2,3}+r_{3,2}} \phi_{3,3}^{2\cdot r_{3,3}}\big) \cdot t_T^{2 r_{3,3}}$$

We reduce the number of pairings from $12n+27$ to $3n+6$ pairings at the expense of adding $9n^2 + 3n$ exponentiations in $\mathbb{G}$ and one exponentiation in $\mathbb{G}_T$.

6.2 Multi-Scalar Multiplication and Quadratic Equations

The description of the batch verification of multi-scalar multiplication equation and quadratic equation is similar to the previous one. Due to space constraints it is given in the full version [BFI+10].

7 Application 1: Groth's Group Signatures

7.1 Description

We demonstrate our techniques by applying them to one of the most practical fully-secure group-signature schemes in the standard model to date: Groth's construction [Gro07]. Groth proposed a methodology of transforming *certified signatures* [BFPW07] that respect a certain structure into group signatures using Groth-Sahai NIWI proofs:

- a member picks a key pair for the certified-signature scheme and asks the issuer to certify her verification key;
- to produce a group signature, the member makes a certified signature, encrypts it and makes a NIWI proof that demonstrates that the ciphertext contains a valid certified signature.

Groth proposed an efficient certified-signature scheme based on the so called q-**U** assumption (see [Gro07] for details). In the CPA-anonymous version[6] of the scheme, the issuer's public key is a triple $(f, h, T) \in \mathbb{G}^2 \times \mathbb{G}_T$ (and its private key is $z \in \mathbb{G}$ such that $e(f, z) = T$) and the certificate for a group member with public key $v = g^x \in \mathbb{G}$ is a pair (a, b) satisfying $e(a, vh)\, e(f, b) = T$. To sign a message $m \in \mathbb{Z}_p$, the group member first computes a weak Boneh-Boyen signature [BB08] $\sigma = g^{1/(x+m)}$ using her private key x; then she forms Groth-Sahai commitments $\mathbf{d}_v$, $\mathbf{d}_b$ and $\mathbf{d}_\sigma$ to the group elements v, b and σ, resp., and makes a proof that they satisfy the following:

$$e(a, vh)\, e(f, b) = T \qquad\qquad e(\sigma, g^m v) = e(g, g) \tag{5}$$

The fact that a is given in the clear is not a problem since the certificate is malleable, so the group member can unlinkably re-randomize it each time she signs a message. A group signature is thus of the form $(a, \mathbf{d}_b, \mathbf{d}_v, \mathbf{d}_\sigma, \psi, \phi)$, where ψ and ϕ denote the Groth-Sahai proofs for the two equations in (5), respectively.

We first instantiate our generic batch construction to verify a single signature more efficiently and then show how to verify multiple signatures at once. The first equation is of a particular form that allows for more efficient proofs and verification. We describe the verification relations and the batch verification in the next section.

7.2 Batching Linear Pairing-Product Equations

We consider a special case of pairing-product equations for which $\Gamma = 0$, called *linear equations*, *i.e.* the equation is of the following form: $\langle \vec{\mathcal{A}}, \vec{\mathcal{Y}} \rangle = t_T$, that is $\prod_{i=1}^n e(\mathcal{A}_i, \mathcal{Y}_i) = t_T$. In this case, the proof simplifies to three group elements and is verified as follows (taking into account that $u_{1,2} = u_{2,1} = 1$, and $u_{1,3} = u_{2,3}$):

$$\begin{aligned}
\textstyle\prod_{i=1}^n e(\mathcal{A}_i, d_{i,1}) &= e(u_{11}, \psi_1)\, e(u_{31}, \psi_3) \\
\textstyle\prod_{i=1}^n e(\mathcal{A}_i, d_{i,2}) &= e(u_{22}, \psi_2)\, e(u_{32}, \psi_3) \\
\textstyle\prod_{i=1}^n e(\mathcal{A}_i, d_{i,3}) &= t_T\, e(u_{13}, \psi_1\psi_2)\, e(u_{33}, \psi_3)
\end{aligned}$$

which can be batch-verified by checking[7]

$$\begin{aligned}
&\prod_{i=1}^n e(\mathcal{A}_i, d_{i,1}^{s_1} d_{i,2}^{s_2} d_{i,3}^{s_3}) \\
&= t_T^{s_3}\, e(u_{11}, \psi_1^{s_1})\, e(u_{13}, (\psi_1\psi_2)^{s_3})\, e(u_{22}, \psi_2^{s_2})\, e(u_{31}, \psi_3^{s_1})\, e(u_{32}, \psi_3^{s_2})\, e(u_{33}, \psi_3^{s_3})\ . \qquad (6)
\end{aligned}$$

[6] Groth also proposes group signatures achieving CCA-anonymity [BSZ05]; for illustrative purposes we restrict ourselves to the basic CPA-anonymous scheme here.

[7] If we considered a single set of equations then it would be more efficient to order the right-hand side by the ψ_i's and save 3 pairings. We order by the u_{ij} though, since this enables us to batch with other equations containing pairings of these constants.

7.3 Batching the Equations for One Group Signature

1st Equation. Instantiating (6) for first equation in (5), we get, after some more optimization (shifting $e(a, h^{-1}))^{s_3}$ to the left-hand side of the equation)

$$e(d_{v,1}^{s_1} d_{v,2}^{s_2} (d_{v,3} h)^{s_3},\, a)\, e(d_{b,1}^{s_1} d_{b,2}^{s_2} d_{b,3}^{s_3},\, f) \;=$$
$$T^{s_3}\, e(u_{11}, \psi_1^{s_1})\, e(u_{13}, (\psi_1\psi_2)^{s_3})\, e(u_{22}, \psi_2^{s_2})\, e(u_{31}, \psi_3^{s_1})\, e(u_{32}, \psi_3^{s_2})\, e(u_{33}, \psi_3^{s_3})$$

2nd Equation. Setting $\vec{\mathcal{A}} := \begin{pmatrix} g^m \\ 1 \end{pmatrix}$, $\vec{\mathcal{Y}} := \begin{pmatrix} \sigma \\ v \end{pmatrix}$, $\Gamma := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $t_T := e(g, g)$, $\mathbf{d}_1 := \mathbf{d}_\sigma$ and $\mathbf{d}_2 := \mathbf{d}_v$ and substituting in (4), we get

$$e\big(d_{\sigma 1},\ (g^m d_{v3})^{(r_{13}+r_{31})} d_{v1}^{2\cdot r_{11}} d_{v2}^{(r_{12}+r_{21})}\big)\, e\big(d_{\sigma 2},\ (g^m d_{v3})^{(r_{23}+r_{32})} d_{v1}^{(r_{12}+r_{21})} d_{v2}^{2\cdot r_{22}}\big)$$
$$\cdot\, e\big(d_{\sigma 3},\ (g^m d_{v3})^{2\cdot r_{33}} d_{v1}^{(r_{13}+r_{31})} d_{v2}^{(r_{23}+r_{32})}\big) \;=$$
$$e\big(u_{11},\ \phi_{11}^{2\cdot r_{11}} \phi_{12}^{r_{12}+r_{21}} \phi_{13}^{r_{13}+r_{31}}\big)\, e\big(u_{13},\ (\phi_{11}\phi_{21})^{r_{13}+r_{31}} (\phi_{12}\phi_{22})^{r_{23}+r_{32}} (\phi_{13}\phi_{23})^{2\cdot r_{33}}\big)$$
$$\cdot\, e\big(u_{22},\ \phi_{21}^{r_{12}+r_{21}} \phi_{22}^{2\cdot r_{22}} \phi_{23}^{r_{23}+r_{32}}\big)\, e\big(u_{31},\ \phi_{31}^{2\cdot r_{11}} \phi_{32}^{r_{12}+r_{21}} \phi_{33}^{r_{13}+r_{31}}\big)$$
$$\cdot\, e\big(u_{32},\ \phi_{31}^{r_{12}+r_{21}} \phi_{32}^{2\cdot r_{22}} \phi_{33}^{r_{23}+r_{32}}\big)\, e\big(u_{33},\ \phi_{31}^{r_{13}+r_{31}} \phi_{32}^{r_{23}+r_{32}} \phi_{33}^{2\cdot r_{33}}\big)\, e(g, g^{2 r_{33}})\ .$$

Multiplying the two equations we get a single verification relation of the following form:

$$e(d_{v,1}^{s_1} d_{v,2}^{s_2} (d_{v,3} h)^{s_3}, a)\, e(d_{b,1}^{s_1} d_{b,2}^{s_2} d_{b,3}^{s_3}, f) e\big(d_{\sigma 1},\ (g^m d_{v3})^{(r_{13}+r_{31})} d_{v1}^{2\cdot r_{11}} d_{v2}^{(r_{12}+r_{21})}\big)$$
$$\cdot\, e\big(d_{\sigma 2},\ (g^m d_{v3})^{(r_{23}+r_{32})} d_{v1}^{(r_{12}+r_{21})} d_{v2}^{2\cdot r_{22}}\big)\, e\big(d_{\sigma 3},\ (g^m d_{v3})^{2\cdot r_{33}} d_{v1}^{(r_{13}+r_{31})} d_{v2}^{(r_{23}+r_{32})}\big)$$
$$= \big(T^{s_3} e(g, g^{2 r_{33}})\big) e\big(u_{13},\ (\phi_{11}\phi_{21})^{r_{13}+r_{31}} (\phi_{12}\phi_{22})^{r_{23}+r_{32}} (\phi_{13}\phi_{23})^{2\cdot r_{33}} (\psi_1\psi_2)^{s_3}\big)$$
$$e\big(u_{11},\ \phi_{11}^{2\cdot r_{11}} \phi_{12}^{r_{12}+r_{21}} \phi_{13}^{r_{13}+r_{31}} \psi_1^{s_1}\big) \cdot e\big(u_{22},\ \phi_{21}^{r_{12}+r_{21}} \phi_{22}^{2\cdot r_{22}} \phi_{23}^{r_{23}+r_{32}} \psi_2^{s_2}\big)$$
$$e\big(u_{31},\ \phi_{31}^{2\cdot r_{11}} \phi_{32}^{r_{12}+r_{21}} \phi_{33}^{r_{13}+r_{31}} \psi_3^{s_1}\big) \cdot e\big(u_{32},\ \phi_{31}^{r_{12}+r_{21}} \phi_{32}^{2\cdot r_{22}} \phi_{33}^{r_{23}+r_{32}} \psi_3^{s_2}\big)$$
$$\cdot\, e\big(u_{33},\ \phi_{31}^{r_{13}+r_{31}} \phi_{32}^{r_{23}+r_{32}} \phi_{33}^{2\cdot r_{33}} \psi_3^{s_3}\big)$$

Analysis. With no use of batching techniques, the verification of a single signature takes for the first equation 13 pairings and for the second 20 pairings for the left-hand side and 35 for its right-hand side. This is an overall of 68 pairing evaluations, compared to 11 for the batched verification.

7.4 Batching Several Group Signatures

Consider the situation where we want to verify multiple group signatures at once. That is given a group public key $(f, h, T, u_{11}, u_{13}, u_{22}, u_{31}, u_{32}, u_{33})$ and n group signatures

$$\Big(a^{(k)}, \mathbf{d}_b^{(k)}, \mathbf{d}_v^{(k)}, \mathbf{d}_\sigma^{(k)}, (\psi_i^{(k)})_{1\le i\le 3}, (\phi_{ij}^{(k)})_{1\le i,j\le 3}\Big)\ .$$

Using the same technique of taking each of the (new) equations to the power of some randomness and multiplying them, we can unify the pairings $e(\cdot, f)$ on the left-hand side and all pairings (which are of the form $e(u_{ij}, \cdot)$) on the right-hand side. Instead of $11n$ pairings needed when checking each equation, the batched version only requires $4n + 7$ pairings.

8 Application 2: P-signatures

8.1 Description

Belenkiy *et al.* [BCKL08] formalize digital signature schemes with an additional non-interactive proof of signature possession that they called *P-signature schemes*. They proposed two constructions[8]: the first is based on the weak Boneh-Boyen signature scheme [BB08] while the second one is inspired by its full version.

Since Belenkiy *et al.*'s first scheme relies on a rather strong assumption, we consider only their second proposal: a signature σ on a message $m \in \mathbb{Z}_p$ is a triple $\sigma = (C_1, C_2, C_3) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1$ such that $e(C_1, vh^m C_2) = e(g, h)$ and $e(f, C_2) = e(C_3, w)$, where f and g are (public) generators of $\mathbb{G}_1$, h is a (public) generator of $\mathbb{G}_2$ and $v, w \in \mathbb{G}_2$ are parts of the signer's public key. To prove possession of such a signature, a prover forms the Groth-Sahai commitments $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ for the group elements $C_1, M_1 = f^m, C_3$ in $\mathbb{G}_1$ and $\mathbf{d_1}$ and $\mathbf{d_2}$ for the group elements $M_2 = h^m$ and C_2 in $\mathbb{G}_2$ (respectively) and provides a proof that they satisfy:

$$e(C_1, vM_2C_2) = e(g, h), \quad e(f, C_2) = e(C_3, w) \quad \text{and} \quad e(f, M_2) = e(M_1, h) \quad (7)$$

8.2 SXDH Instantiation

In [BCKL08], the authors evaluated that the verification of the proof in the SXDH instantiation requires the computation of 68 pairings. In the full version of this paper [BFI+10] we show that it can be reduced to 15.

8.3 DLIN Instantiation

As in Section 7, the last two pairing-product equations from (7) are actually linear pairing-product equations. We denote the Groth-Sahai commitments for the group elements $C_1, C_2, C_3, M_1 = f^m, M_2 = h^m$ in $\mathbb{G}$ by $\mathbf{d_1}$,$\mathbf{d_2}$, $\mathbf{d_3}$, $\mathbf{d_4}$ and $\mathbf{d_5}$ (respectively) and by ϕ, ψ and θ the proofs that they satisfy the first, the second and the third equation (respectively). For the first equation, setting and substituting

$$\vec{\mathcal{A}} = \begin{pmatrix} v \\ 1 \\ 1 \end{pmatrix}, \ \vec{\mathbf{d}} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_5 \end{pmatrix}, \ \Gamma = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad t_T = e(g, g)$$

in (4), we get:

$$\begin{aligned} & e\big(d_{1,1},\ (v\, d_{2,3}\, d_{5,3})^{r_{1,3}+r_{3,1}} (d_{2,1}\, d_{5,1})^{2r_{1,1}} (d_{2,2}\, d_{5,2})^{(r_{1,2}+r_{2,1})}\big) \\ & \qquad \cdot e\big(d_{1,2},\ (v\, d_{2,3}\, d_{5,3})^{r_{2,3}+r_{3,2}} (d_{2,1}\, d_{5,1})^{r_{1,2}+r_{2,1}} (d_{2,2}\, d_{5,2})^{2r_{2,2}}\big) \\ & \qquad \cdot e\big(d_{1,3},\ (v\, d_{2,3}\, d_{5,3})^{2r_{3,3}} (d_{2,1}\, d_{5,1})^{r_{1,3}+r_{3,1}} (d_{2,2}\, d_{5,2})^{r_{2,3}+r_{3,2}}\big) \ = \end{aligned}$$

[8] An extended version of their scheme was recently proposed [BCKL09] but here we restrict ourselves to the basic scheme from [BCKL08].

$$
\begin{aligned}
= \; & e\big(u_{1,1},\ \phi_{1,1}^{2\cdot r_{1,1}}\phi_{1,2}^{r_{1,2}+r_{2,1}}\phi_{1,3}^{r_{1,3}+r_{3,1}}\big)\cdot e\big(u_{2,2},\ \phi_{2,1}^{r_{1,2}+r_{2,1}}\phi_{2,2}^{2\cdot r_{2,2}}\phi_{2,3}^{r_{2,3}+r_{3,2}}\big)\\
& \cdot e\big(u_{1,3},\ (\phi_{1,1}\phi_{2,1})^{r_{1,3}+r_{3,1}}(\phi_{1,2}\phi_{2,2})^{r_{2,3}+r_{3,2}}(\phi_{1,3}\phi_{2,3})^{2\cdot r_{3,3}}\big)\\
& \cdot e\big(u_{3,1},\ \phi_{3,1}^{2\cdot r_{1,1}}\phi_{3,2}^{r_{1,2}+r_{2,1}}\phi_{3,3}^{r_{1,3}+r_{3,1}}\big)\cdot e\big(u_{3,2},\ \phi_{3,1}^{r_{1,2}+r_{2,1}}\phi_{3,2}^{2\cdot r_{2,2}}\phi_{3,3}^{r_{2,3}+r_{3,2}}\big)\\
& \cdot e\big(u_{3,3},\ \phi_{3,1}^{r_{1,3}+r_{3,1}}\phi_{3,2}^{r_{2,3}+r_{3,2}}\phi_{3,3}^{2\cdot r_{3,3}}\big)\cdot e(g,g)^{2r_{3,3}}\ .
\end{aligned}
$$

Substituting $\vec{\mathcal{A}} = \begin{pmatrix} f \\ w^{-1} \end{pmatrix}$, $\vec{\mathbf{d}} = \begin{pmatrix} \mathbf{d}_2 \\ \mathbf{d}_3 \end{pmatrix}$, $t_T = 1$, and $\vec{\mathcal{A}} = \begin{pmatrix} f \\ h^{-1} \end{pmatrix}$, $\vec{\mathbf{d}} = \begin{pmatrix} \mathbf{d}_5 \\ \mathbf{d}_4 \end{pmatrix}$, $t_T = 1$ (respectively) in (6), we obtain the second and third equation. Once the three equations multiplied, we obtain:

$$
\begin{aligned}
& e\big(d_{1,1},\ (v\,d_{2,3}\,d_{5,3})^{r_{1,3}+r_{3,1}}(d_{2,1}\,d_{5,1})^{2r_{1,1}}(d_{2,2}\,d_{5,2})^{(r_{1,2}+r_{2,1})}\big)\\
& \qquad e\big(d_{1,2},\ (v\,d_{2,3}\,d_{5,3})^{r_{2,3}+r_{3,2}}(d_{2,1}\,d_{5,1})^{r_{1,2}+r_{2,1}}(d_{2,2}\,d_{5,2})^{2r_{2,2}}\big)\\
& \qquad e\big(d_{1,3},\ (v\,d_{2,3}\,d_{5,3})^{2r_{3,3}}(d_{2,1}\,d_{5,1})^{r_{1,3}+r_{3,1}}(d_{2,2}\,d_{5,2})^{r_{2,3}+r_{3,2}}\big)\\
& \qquad e(f, d_{2,1}^{s_1}d_{2,2}^{s_2}d_{2,3}^{s_3}d_{5,1}^{t_1}d_{5,2}^{t_2}d_{5,3}^{t_3})e(w^{-1}, d_{3,1}^{s_1}d_{3,2}^{s_2}d_{3,3}^{s_3})e(h^{-1}, d_{4,1}^{t_1}d_{4,2}^{t_2}d_{4,3}^{t_3})\\
= \; & e\big(u_{1,3},\ (\phi_{1,1}\phi_{2,1})^{r_{1,3}+r_{3,1}}(\phi_{1,2}\phi_{2,2})^{r_{2,3}+r_{3,2}}(\phi_{1,3}\phi_{2,3})^{2\cdot r_{3,3}}(\psi_1\psi_2)^{s_3}(\theta_1\theta_2)^{t_3}\big)\\
& \cdot e\big(u_{1,1},\ \phi_{1,1}^{2\cdot r_{1,1}}\phi_{1,2}^{r_{1,2}+r_{2,1}}\phi_{1,3}^{r_{1,3}+r_{3,1}}\psi_1^{s_1}\theta_1^{t_1}\big)\cdot e\big(u_{2,2},\ \phi_{2,1}^{r_{1,2}+r_{2,1}}\phi_{2,2}^{2\cdot r_{2,2}}\phi_{2,3}^{r_{2,3}+r_{3,2}}\psi_2^{s_2}\theta_2^{t_2}\big)\\
& \cdot e\big(u_{3,1},\ \phi_{3,1}^{2\cdot r_{1,1}}\phi_{3,2}^{r_{1,2}+r_{2,1}}\phi_{3,3}^{r_{1,3}+r_{3,1}}\psi_3^{s_1}\theta_3^{t_1}\big)\cdot e\big(u_{3,2},\ \phi_{3,1}^{r_{1,2}+r_{2,1}}\phi_{3,2}^{2\cdot r_{2,2}}\phi_{3,3}^{r_{2,3}+r_{3,2}}\psi_3^{s_2}\theta_3^{t_2}\big)\\
& \qquad e\big(u_{3,3},\ \phi_{3,1}^{r_{1,3}+r_{3,1}}\phi_{3,2}^{r_{2,3}+r_{3,2}}\phi_{3,3}^{2\cdot r_{3,3}}\psi_3^{s_3}\theta_3^{t_3}\big)e(g,g)^{2r_{3,3}}
\end{aligned}
$$

In [BCKL08], the authors evaluated that the verification of the proof in the DLIN instantiation requires the computation of 126 pairings. With our result, we prove it can be reduced to 12.

Batching Several P-Signatures. As in the previous section, in case we want to verify multiple P-signatures at once, we can unify the pairings containing f, h and w on the left-hand side and all pairings (which are of the form $e(u_{i,j}, \cdot)$) on the right-hand side. Instead of $15n$ (*resp.* $12n$) pairings needed when checking each equation, the batched version only requires $2n + 13$ (*resp.* $3n + 9$) pairings.

9 Conclusion

In this paper, we presented efficiency improvements for the verification of Groth-Sahai non-interactive zero-knowledge and witness-indistinguishable proofs and two privacy-preserving authentication schemes, saving up to 90% of the (dominant) pairing operations. These results can be combined with known methods to compute the product of many pairing evaluations efficiently [GS06]. Our results notably provide the first algorithm to batch-verify a group signature scheme in the standard model (an open problem raised in [FGHP09]) and, surprisingly, demonstrate that thanks to batch verification techniques, the DLIN instantiation of the Groth-Sahai proof system may be the most efficient implementation for the verification of a single signature.

Acknowledgments

This work was supported by the French ANR-07-TCOM-013-04 PACE Project, by the European Commission through the IST Program under Contract ICT-2007-216646 ECRYPT II and by EADS.

References

[ACHdM05] G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005.

[BB08] D. Boneh and X. Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO 2004*, *LNCS* 3152, pp.41–55. Springer, 2004.

[BCC+09] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *CRYPTO 2009*, *LNCS* 5677, pages 108–125. Springer, 2009.

[BCKL08] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *TCC 2008*, *LNCS* 4948, pp.356–374. Springer, 2008.

[BCKL09] —. Compact e-cash and simulatable VRFs revisited. In *PAIRING 2009*, *LNCS* 5671, pages 114–131. Springer, 2009.

[BFI+10] O. Blazy, G. Fuchsbauer, M. Izabachène, A. Jambert, H. Sibert, and D. Vergnaud. Batch Groth-Sahai. Cryptology ePrint Archive, Report 2010/040, 2010.

[BFM90] M. Blum, P. Feldman, and S. Micali. Proving security against chosen cyphertext attacks. In *CRYPTO'88*, *LNCS* 403, pp.256–268. Springer, 1990.

[BFPW07] A. Boldyreva, M. Fischlin, A. Palacio, and B. Warinschi. A closer look at PKI: Security and efficiency. In *PKC 2007*, *LNCS* 4450, pp.458–475. Springer, 2007.

[BGN05] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005*, *LNCS* 3378, pp.325–341. Springer, 2005.

[BGR98] M. Bellare, J. A. Garay, and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In *EUROCRYPT'98*, *LNCS* 1403, pages 236–250. Springer, 1998.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pp.62–73. ACM Press, 1993.

[BSZ05] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA 2005*, *LNCS* 3376, pages 136–153. Springer, 2005.

[BW06] X. Boyen and B. Waters. Compact group signatures without random oracles. In *EUROCRYPT 2006*, *LNCS* 4004, pages 427–444. Springer, 2006.

[BW07] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC 2007*, *LNCS* 4450, pp.1–15. Springer, 2007.

[CGH98] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pp.209–218. ACM Press, 1998.

[CGS07] N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP 2007*, *LNCS* 4596, pp.423–434. Springer, 2007.

[CHP07] J. Camenisch, S. Hohenberger, and M. Ø. Pedersen. Batch verification of short signatures. In *EUROCRYPT 2007*, *LNCS* 4515, pages 246–263. Springer, 2007.

[FGHP09] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen. Practical short signature batch verification. In *CT-RSA 2009*, *LNCS* 5473, pages 309–324. Springer, 2009.

[Fia90] A. Fiat. Batch RSA. In *CRYPTO'89*, *LNCS* 435, pp.175–185. Springer, 1990.

[FPV09] Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Transferable constant-size fair e-cash. *CANS 2009*, *LNCS* 5888, pages 226–247. Springer, 2009.

[GL07] J. Groth and S. Lu. A non-interactive shuffle with pairing based verifiability. In *ASIACRYPT 2007*, *LNCS* 4833, pages 51–67. Springer, 2007.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[Gro07] J. Groth. Fully anonymous group signatures without random oracles. In *ASIACRYPT 2007*, *LNCS* 4833, pages 164–180. Springer, 2007.

[GS06] R. Granger and N. P. Smart. On Computing Products of Pairings. Cryptology ePrint Archive, Report 2006/172, 2006.

[GS08] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT 2008*, *LNCS* 4965, pages 415–432. Springer, 2008.

[GSW09] E. Ghadafi, N.P. Smart, and B. Warinschi. Groth–sahai proofs revisited. Cryptology ePrint Archive, Report 2009/599, 2009.

[Mat09] B. J. Matt. Identification of multiple invalid signatures in pairing-based batched signatures. In *PKC 2009*, *LNCS* 5443, pp.337–356. Springer, 2009.

[NMVR94] D. Naccache, D. M'Raïhi, S. Vaudenay, and D. Raphaeli. Can D.S.A. be improved? complexity trade-offs with the digital signature standard. In *EUROCRYPT'94*, *LNCS* 950, pages 77–85. Springer, 1994.

[PMPS00] J. Pastuszak, D. Michatek, J. Pieprzyk, and J. Seberry. Identification of bad signatures in batches. In *PKC 2000*, *LNCS* 1751, pp.28–45. Springer, 2000.