TYPE 2 STRUCTURE-PRESERVING SIGNATURE SCHEMES REVISITED

SANJIT CHATTERJEE AND ALFRED MENEZES

Abstract. Abe, Groth, Ohkubo and Tibouchi recently presented structure-preserving signature schemes using Type 2 pairings. The schemes are claimed to enjoy the fastest signature verification. By properly accounting for subgroup membership testing of group elements in signatures, we show that the schemes are not as efficient as claimed. We present natural Type 3 analogues of the Type 2 schemes, and show that the Type 3 schemes are superior to their Type 2 counterparts.

1. Introduction

The term ‘structure-preserving signature scheme’ was coined in 2010 by Abe et al. [1]. These pairing-based signature schemes have the property that verification keys, messages, and signatures are all group elements. Moreover, signatures are verified by testing the equality of products of pairings of group elements; each such equality is called a product-of-pairings equation (PPE). Structure-preserving signature schemes are fully compatible with Groth-Sahai non-interactive witness-indistinguishable (NIWI) and non-interactive zero-knowledge (NIZK) proof systems [16] and have been used in the design of numerous cryptographic protocols; a list of these protocols can be found in [3].

In typical applications of structure-preserving signature schemes when used in conjunction with Groth-Sahai proofs, a party has a signed message and wishes to convince a second party (the verifier) that it possesses the (valid) signed message without revealing the message or the signature. Groth-Sahai NIWI and NIZK proofs allow a party (the prover) to convince a second party (the verifier) that it possesses a solution to a collection of PPEs[1]. The complexity of verifying a Groth-Sahai proof is heavily dependent on the number of group elements in the signature and the number of PPEs in signature verification (see [9, §3.4]). For this reason, researchers have strived to design structure-preserving signature schemes with the smallest possible number of group elements in a signature and with the smallest possible number of PPEs in signature verification.

At CRYPTO 2011, Abe et al. [2] presented a structure-preserving signature scheme using Type 3 pairings. Verification has two PPEs, which was proven to be optimal in the sense that any Type 3 structure-preserving signature scheme with verification having a single PPE was shown to succumb to a random message attack. Moreover, signatures are comprised of three group elements, which was also shown to be optimal. The scheme was proven to be strongly secure against generic signers.

Date: August 18, 2014.

[1] Two examples of Groth-Sahai NIWI proofs for verifying that the prover possesses the solution Y to an equation $e(A, Y) = t$ where e is a Type 2 or a Type 3 pairing are given in Appendix A.

At CRYPTO 2014, Abe et al. [3] presented a strongly unforgeable structure-preserving signature scheme and a randomizable structure-preserving signature scheme using Type 2 pairings. Both schemes are claimed to have signatures that are comprised of only two group elements, have only one PPE in signature verification, and were proven secure against generic signers. The authors conclude that their schemes enjoy the fastest signature verification. Moreover, in light of the aforementioned lower bounds on the number of group elements in signatures and the number of PPEs in signature verification for Type 3 structure-preserving signature schemes, they conclude that the Type 2 schemes have no analogues in the Type 3 setting. This is contrary to the arguments presented in [11] that any cryptographic protocol that employs Type 2 pairings has a natural counterpart in the Type 3 setting that does not suffer any loss in functionality, security or efficiency.

We observe that the analysis of the Type 2 structure-preserving signature schemes in [3] neglected to account for subgroup membership testing of all group elements in a signature. Incorporating these subgroup membership tests into Groth-Sahai proofs increases the number of group elements in signatures and also increases the number of PPEs in signature verification. Consequently, the Type 2 schemes are not as efficient as claimed in [3]. We present natural Type 3 analogues of the Type 2 schemes, and show that the Type 3 schemes are superior to their Type 2 counterparts.

The remainder of the paper is organized as follows. In §2 we summarize the salient differences between Type 2 and Type 3 pairings derived from elliptic curves having even embedding degrees. In §3, we explain why the strongly unforgeable structure-preserving signature scheme in [3] actually has signatures comprising of three group elements and has two PPEs in signature verification. We present a natural analogue of the scheme in the Type 3 setting, and show that it is more efficient than the Type 2 scheme. In §4, we present our Type 3 analogue of the Type 2 randomizable structure-preserving signature scheme in [3], and show that the Type 3 scheme is more efficient. We draw our conclusions in §5.

2. Asymmetric bilinear pairings

Let $\mathbb{F}_q$ be a finite field of characteristic $p \geq 5$, and let E be an ordinary elliptic curve defined over $\mathbb{F}_q$. Let n be a prime divisor of $\#E(\mathbb{F}_q)$ satisfying $\gcd(n, q) = 1$, and let k (the embedding degree) be the smallest positive integer such that $n \mid q^k - 1$. We will henceforth assume that k is even, since then some important speedups in pairing computations are applicable [6]. Some prominent families of elliptic curves with even embedding degree include the MNT [19], BN [7], KSS [18], and BLS [5] curves.

Since $k > 1$, we have $E[n] \subseteq E(\mathbb{F}_{q^k})$ where $E[n]$ denotes the n-torsion group of E. Let $G \in E(\mathbb{F}_q)[n]$ be an $\mathbb{F}_q$-rational point of order n, and define $\mathbb{G}_1 = \langle G \rangle$. Let $\mathbb{G}_T$ denote order-n subgroup of the multiplicative subgroup of $\mathbb{F}_{q^k}$.

2.1. **Type 3 pairings.** Following [14], we denote by D the CM discriminant of E and set

$$e = \begin{cases} \gcd(k,6), & \text{if } D = -3, \\ \gcd(k,4), & \text{if } D = -4, \\ 2, & \text{if } D < -4, \end{cases} \qquad (1)$$

and $d = k/e$. For example, BN curves have $k = 12$, $e = 6$ and $d = 2$, whereas MNT curves have $k = 6$, $e = 2$ and $d = 3$. Now, E has a unique degree-e twist $\tilde{E}$ defined over $\mathbb{F}_{q^d}$ such that $n \mid \#\tilde{E}(\mathbb{F}_{q^d})$ [17]. Let $\tilde{I} \in \tilde{E}(\mathbb{F}_{q^d})$ be a point of order n, and let $\tilde{\mathbb{G}}_3 = \langle \tilde{I} \rangle$. Then there is a monomorphism $\phi : \tilde{\mathbb{G}}_3 \longrightarrow E(\mathbb{F}_{q^k})$ such that $I = \phi(\tilde{I}) \notin \mathbb{G}_1$. The group $\mathbb{G}_3 = \langle I \rangle$ is the Trace-0 subgroup of $E[n]$, so named because it consists of all points $P \in E[n]$ for which $\mathrm{Tr}(P) = \sum_{i=0}^{k-1} \pi^i(P) = \infty$, where π denotes the q-th power Frobenius. The monomorphism ϕ can be defined so that $\phi : \tilde{\mathbb{G}}_3 \longrightarrow \mathbb{G}_3$ can be efficiently computed in both directions; therefore we can identify $\tilde{\mathbb{G}}_3$ and $\mathbb{G}_3$, and consequently $\mathbb{G}_3$ can be viewed as having coordinates in $\mathbb{F}_{q^d}$ (instead of in the larger field $\mathbb{F}_{q^k}$).

Non-degenerate bilinear pairings $e_3 : \mathbb{G}_1 \times \mathbb{G}_3 \longrightarrow \mathbb{G}_T$ are said to be of Type 3 because no efficiently-computable isomorphisms from $\mathbb{G}_1$ to $\mathbb{G}_3$ or from $\mathbb{G}_3$ to $\mathbb{G}_1$ are known [14]. There are several Type 3 pairings, of which the most efficient is Vercauteren's optimal pairing [20].

2.2. **Type 2 pairings.** Let $H \in E[n]$ with $H \notin \mathbb{G}_1$ and $H \notin \mathbb{G}_3$. Then $\mathbb{G}_2 = \langle H \rangle$ is an order-n subgroup of $E(\mathbb{F}_{q^k})$ with $\mathbb{G}_2 \neq \mathbb{G}_1$ and $\mathbb{G}_2 \neq \mathbb{G}_3$. Non-degenerate bilinear pairings $e_2 : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ are said to be of Type 2 because the map Tr is an efficiently-computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$; note, however, that no efficiently-computable isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$ is known. These pairings have the property that hashing onto $\mathbb{G}_2$ is infeasible (other than by multiplying H by a randomly selected integer).

The computation of e_2 is efficiently reduced to the task of computing Type 3 pairing e_3 [14]. Thus, the costs of computing e_2 and e_3 are approximately equal. To see this, define the maps

$$\psi : E[n] \longrightarrow \mathbb{G}_1, \qquad Q \mapsto \frac{1}{k}\mathrm{Tr}(Q) \tag{2}$$

and

$$\rho : E[n] \longrightarrow \mathbb{G}_3, \qquad Q \mapsto Q - \psi(Q). \tag{3}$$

Recall that e_2 and e_3 are restrictions of the Tate pairing $\hat{e} : E[n] \times E[n] \longrightarrow \mathbb{G}_T$. Hence, for all $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, we have

$$e_2(P,Q) = \hat{e}(P, \psi(Q) + \rho(Q)) = \hat{e}(P, \psi(Q)) \cdot \hat{e}(P, \rho(Q)) = \hat{e}(P, \rho(Q)) = e_3(P, \rho(Q)). \tag{4}$$

2.3. **Comparing the performance of Type 2 and Type 3 pairings.** Since points in $\mathbb{G}_2$ have coordinates in $\mathbb{F}_{q^k}$ whereas points in $\mathbb{G}_3$ have coordinates in the proper subfield $\mathbb{F}_{q^d}$, it would appear that the ratio of the bitlengths of points in $\mathbb{G}_2$ and $\mathbb{G}_3$ is k/d. Similarly, the ratio of the costs of addition in $\mathbb{G}_2$ and $\mathbb{G}_3$ can be expected to be k^2/d^2 bit operations (using naive methods for extension field arithmetic). These ratios are given in Table 3 of [14]. However, as observed in [10], points in $\mathbb{G}_2$ have a shorter representation which we describe next. We emphasize that this representation can be used for *all* order-n subgroups $\mathbb{G}_2$ of $E[n]$ different from $\mathbb{G}_1$ and $\mathbb{G}_3$.

Let H be an arbitrary point from $E[n]\backslash(\mathbb{G}_1 \cup \mathbb{G}_2)$, and set $\mathbb{G}_2 = \langle H \rangle$. Define $G = \frac{1}{k}\mathrm{Tr}(H)$ so that the map ψ restricted to $\mathbb{G}_2$ is an efficiently-computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$ with $\psi(H) = G$. Finally, set $I = H - G$. Then $I \in \mathbb{G}_3$ and the map ρ restricted to $\mathbb{G}_2$ is an efficiently-computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_3$ with $\rho(H) = I$.

Now, given a point $Q \in \mathbb{G}_2$, one can efficiently determine the unique points $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_3$ such that $Q = Q_1 + Q_2$; namely, $Q_1 = \psi(Q)$ and $Q_2 = \rho(Q) = Q - Q_1$. Writing $D(Q) = (\psi(Q), \rho(Q))$, and letting $\mathbb{H}_2 \subseteq \mathbb{G}_1 \times \mathbb{G}_3$ denote the range of D, we have an efficiently-computable isomorphism $D : \mathbb{G}_2 \longrightarrow \mathbb{H}_2$ whose inverse is also efficiently computable. Hence, without loss of generality, points $Q \in \mathbb{G}_2$ can be represented by a pair of points (Q_1, Q_2) with $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_3$. Note that arithmetic in $\mathbb{G}_2$ with this representation is component-wise. Thus the ratio of the bitlengths of points in $\mathbb{G}_2$ and $\mathbb{G}_3$ is in fact $(d+1)/d$, whereas the ratio of the costs of addition in $\mathbb{G}_2$ and $\mathbb{G}_3$ is $(d^2+1)/d^2$.

Table 2 of [10] lists the costs of performing basic operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ for a particular BN curve. The table confirms the expectation that basic operations in $\mathbb{G}_2$ are only marginally more expensive than the operations in $\mathbb{G}_3$. One exception is that testing membership in $\mathbb{G}_2$ is several times more expensive than testing membership in $\mathbb{G}_1$ and $\mathbb{G}_3$. To see this, let us consider the case of BN curves E defined over $\mathbb{F}_q$ where q and $n = \#E(\mathbb{F}_q)$ are prime; recall that these curves have embedding degree $k = 12$ and $d = 2$. Testing membership of a point Q in $\mathbb{G}_1$ is very efficient, and simply entails verifying that Q has coordinates in $\mathbb{F}_q$ and satisfies the equation that defines the curve, i.e., $Q \in E(\mathbb{F}_q)$. Testing membership of a point Q in $\mathbb{G}_3$ involves a fast check that $\phi^{-1}(Q)$ is in $\tilde{E}(\mathbb{F}_{q^2})$, followed by an exponentiation in $\mathbb{G}_3$ to verify that $nQ = \infty$. Testing membership in $\mathbb{G}_2$ is more costly since the known methods require two pairing computations. If the shorter representation (as elements of $\mathbb{G}_1 \times \mathbb{G}_3$) is used for $\mathbb{G}_2$, then membership of (Q_1, Q_2) in $\mathbb{G}_2$ can be determined by first checking that $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_3$, and then verifying that $e_3(Q_1, I) = e_3(G, Q_2)$ [12]. If the longer representation (as elements of $E(\mathbb{F}_{q^{12}})$) is used for $\mathbb{G}_2$, then membership of Q in $\mathbb{G}_2$ can be determined by first checking that $Q \in E(\mathbb{F}_{q^{12}})$ and $nQ = \infty$, and then verifying that $e_2(\psi(Q), H) = e_2(G, Q)$.

In the remainder of the paper, we will use multiplicative notation for elements of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$.

3. Strongly unforgeable structure-preserving signatures

We present the Type 2 strongly unforgeable structure-preserving signature scheme from [3] and our Type 3 analogue of it. The Type 3 scheme was obtained by following the general recipe given in [11] for converting a protocol from the Type 2 setting to the Type 3 setting.

3.1. **Type 2 strongly unforgeable structure-preserving signature scheme** [3].

(1) *Setup.* Let $e_2 : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ be a Type 2 pairing where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ have order n; G, H are fixed generators of $\mathbb{G}_1$, $\mathbb{G}_2$, respectively.

(2) *Key generation.* The secret key is $v, w \in_R [1, n-1]$. The public key is (V, W) where $V = G^v$ and $W = G^w$.

(3) *Signature generation.* To sign $M \in \mathbb{G}_2$, select $t \in_R [1, n-1]$ and compute $R = H^{t-w}$ and $S = M^{v/t} H^{1/t}$. The signature on M is (R, S).

(4) *Signature verification.* To verify a signed message $(M, (R, S))$, check that
   (a) $M, R, S \in \mathbb{G}_2$; and
   (b) $e_2(W\psi(R), S) = e_2(V, M) \cdot e_2(G, H)$.

In [3, Theorem 2], the Type 2 scheme is proven strongly secure[2] against generic forgers. Signatures are comprised of two $\mathbb{G}_2$ elements. Signature verification requires three $\mathbb{G}_2$ membership tests and one PPE verification.

[2] A signature scheme is said to be *secure* if it is existentially unforgeable under chosen-message attack. If, in addition, it is infeasible to find a new signature for a message that has already been signed, then the signature scheme is said to be *strongly secure*.

3.2. **Type 3 strongly unforgeable structure-preserving signature scheme.**

(1) *Setup.* Let $e_3 : \mathbb{G}_1 \times \mathbb{G}_3 \longrightarrow \mathbb{G}_T$ be a Type 3 pairing where $\mathbb{G}_1$, $\mathbb{G}_3$ and $\mathbb{G}_T$ have order n; G, I are fixed generators of $\mathbb{G}_1$, $\mathbb{G}_3$, respectively.

(2) *Key generation.* The secret key is $v, w \in_R [1, n-1]$. The public key is (V, W) where $V = G^v$ and $W = G^w$.

(3) *Signature generation.* To sign $M \in \mathbb{G}_3$, select $t \in_R [1, n-1]$ and compute $R_1 = G^{t-w}$, $R_2 = I^{t-w}$, and $S = M^{v/t} I^{1/t}$. The signature on M is (R_1, R_2, S).

(4) *Signature verification.* To verify a signed message $(M, (R_1, R_2, S))$, check that
   (a) $R_1 \in \mathbb{G}_1$ and $M, R_2, S \in \mathbb{G}_3$;
   (b) $e_3(R_1, I) = e_3(G, R_2)$; and
   (c) $e_3(WR_1, S) = e_3(V, M) \cdot e_3(G, I)$.

Correctness of the Type 3 signature scheme is easily verified since

$$\begin{aligned} e_3(WR_1, S) &= e_3(G^w \cdot G^{t-w}, M^{v/t} I^{1/t}) \\ &= e_3(G^t, M^{v/t} I^{1/t}) \\ &= e_3(G, M^v \cdot I) \\ &= e_3(G, M^v) \cdot e_3(G, I) \\ &= e_3(V, M) \cdot e_3(G, I). \end{aligned}$$

The security proof given in [3, Theorem 2] that the Type 2 scheme is strongly secure against generic forgers also applies (with minimal changes) to the Type 3 signature scheme. The reason that the proof carries over with minimal changes is that it does not employ the isomorphism ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$.

Signatures for the Type 3 scheme are comprised of one $\mathbb{G}_1$ element and two $\mathbb{G}_3$ elements. Signature verification requires one $\mathbb{G}_1$ membership test, three $\mathbb{G}_3$ membership tests, and two PPE verifications.

We note that the verification step 4(b) of the Type 3 scheme cannot be omitted. Indeed, if this step is omitted then the scheme succumbs to the following key-only attack: $(1, (W^{-1}G, 1, I))$ is a valid forgery. Moreover, even if the message $M = 1$ is disallowed, the scheme succumbs to the following random message attack. The forger first obtains a signed message $(M, (R_1, R_2, S))$. It then computes $M' = MS^{-1}$ and $R_1' = R_1 V^{-1}$, thereby obtaining a valid forgery $(M', (R_1', R_2, S))$. We note that this attack is anticipated by the proof of Theorem 2 in [2] which establishes that any Type 3 structure-preserving signature scheme with a single verification equation is existentially forgeable under random message attack.

3.3. **Comparisons.**

3.3.1. *Signature size.* Signatures in the Type 2 scheme are comprised of two $\mathbb{G}_2$ elements or, equivalently, two $\mathbb{G}_1$ and two $\mathbb{G}_3$ elements. Thus, signatures in the Type 3 scheme are smaller than signatures in the Type 2 scheme.

3.3.2. *Signature generation cost.* In signature generation, computing $R = H^{t-w}$ for the Type 2 scheme has exactly the same cost as computing $R_1 = G^{t-w}$ and $R_2 = I^{t-w}$ for the Type 3 scheme. However, the computation of $S = M^{v/t} H^{1/t}$ in the Type 2 scheme is slower than in the Type 3 scheme since the computation takes place in $\mathbb{G}_2$ in the former and in $\mathbb{G}_3$

in the latter. Thus, signature generation is slower in the Type 2 scheme than in the Type 3 scheme.

3.3.3. *Signature verification cost.* Signature verification in the Type 2 scheme is slower than in the Type 3 scheme. This is because, as explained in the last paragraph of §2.3, the subgroup membership tests $M, R, S \in \mathbb{G}_2$ required in the Type 2 scheme each requires the verification of a PPE, whereas the subgroup memberships tests $R_1 \in \mathbb{G}_1$ and $M, R_2, S \in \mathbb{G}_3$ in the Type 3 scheme are relatively inexpensive. Thus, signature verification in the Type 2 scheme requires *four* PPE verifications, whereas only *two* are needed in the Type 3 scheme. The high cost of PPE verifications can be mitigated by batching [8, 13].

The costly subgroup membership tests in step 4(a) of the Type 2 scheme cannot be omitted for two reasons. First, if these tests are omitted then the security proof given in [3] is no longer applicable since the proof makes the assumption that $M, R, S \in \mathbb{G}_2$. Second, there are attacks on the scheme if the membership tests are omitted. For example, given a valid signed message $(M, (R, S))$, one can easily[3] select a second point $R' \in E[n]$ with $R' \neq R$ and $\psi(R') = \psi(R)$, thereby obtaining a second valid signed message $(M, (R', S))$.

3.3.4. *Cost of signature verification with Groth-Sahai proofs.* Structure-preserving signature schemes were not designed to be used as stand-alone signature schemes, but rather in conjunction with Groth-Sahai NIWI and NIZK proofs as explained in §1.

Consider first the Type 2 signature scheme in §3.1 when used in conjunction with a Groth-Sahai proof. The prover wishes to convince a verifier that it possesses a valid signed message $(M, (R, S))$ without revealing anything else about R or S. In other words, it needs to convince the verifier that it possesses a solution to the following PPE:

$$e_2(W\psi(R), S) = e_2(V, M) \cdot e_2(G, H). \tag{5}$$

In this equation, the group elements W, V, M, G and H are known to the verifier, whereas the variables are R, $S \in \mathbb{G}_2$. However, since Groth-Sahai proofs do not have a mechanism for incorporating the evaluation of $\psi(R)$, the variables in (5) are actually $\psi(R)$ and S. In other words, a Groth-Sahai proof for (5) only convinces a verifier that the prover knows $R_1 \in \mathbb{G}_1$ and $S \in \mathbb{G}_2$ that satisfy the following PPE:

$$e_2(W \cdot R_1, S) = e_2(V, M) \cdot e_2(G, H). \tag{6}$$

In particular, the proof does *not* establish that the prover knows $R \in \mathbb{G}_2$ such that $R_1 = \psi(R)$, i.e., the subgroup membership test $R \in \mathbb{G}_2$ is not performed. As we have shown in §3.3.3, if the subgroup membership test $R \in \mathbb{G}_2$ is omitted then the signature scheme is insecure, i.e., not strongly unforgeable. Thus, the prover needs to convince the verifier that it possesses a solution $R_1 \in \mathbb{G}_1$, $R, S \in \mathbb{G}_2$ to the following collection of PPEs:

$$\begin{aligned} e_2(W \cdot R_1, S) &= e_2(V, M) \cdot e_2(G, H) && (7)\\ e_2(R_1, H) \cdot e_2(G, R)^{-1} &= 1. && (8) \end{aligned}$$

The verification now has *two* PPEs. This is in contrast to the claim made in [3] that the Type 2 signature scheme of §3.1 has only *one* PPE. Moreover, signatures are comprised of *three* group elements, namely $R_1 \in \mathbb{G}_1$ and $R, S \in \mathbb{G}_2$.

[3] Given $R \in \mathbb{G}_2$, one computes $R_1 = \psi(R)$ and selects arbitrary $R_2' \in \mathbb{G}_3$ with $R_2' \neq R - R_1$. Then $R' = R_1 + R_2'$ satisfies $\psi(R') = R_1$ and $R' \neq R$.

Recall that the Type 3 signature scheme in §3.2 also has two PPEs and signatures that are comprised of three group elements. Thus, it might appear at first glance that signature verification for the Type 2 and Type 3 schemes costs roughly the same when used in conjunction with Groth-Sahai proofs. However, the Groth-Sahai proofs for the Type 2 setting are based on hardness of the decisional linear (DLIN) problem in $\mathbb{G}_2$ [15], whereas Groth-Sahai proofs for the Type 3 setting can be based on hardness of the decisional Diffie-Hellman (DDH) problem in $\mathbb{G}_1$ and $\mathbb{G}_3$ [16]. Now, DLIN-based Groth-Sahai proofs are significantly more costly than DDH-based Groth-Sahai proofs in terms of commitment size, proof size, and the total number of pairing computations in proof verification. For example, one can see that the DLIN-based proof of knowledge of the solution Y to the equation $e_2(A, Y) = t$ in Appendix A.1 is significantly more costly than the DDH-based proof of knowledge of the solution Y to the equation $e_3(A, Y) =$ in Appendix A.2; see also the performance estimates given in §3.4 of [9]. Thus, the Type 2 structure-preserving signature scheme will be significantly slower than its Type 3 counterpart when combined with Groth-Sahai proofs.

3.3.5. *Conclusions.* The Type 3 strongly unforgeable structure-preserving signature scheme is superior to its Type 2 counterpart with respect to signature size, signature generation cost, and signature verification cost when the schemes are used as stand-alone signature schemes and when used in conjunction with Groth-Sahai proofs. Moreover, the schemes have similar security proofs against generic forgers. Thus, the Type 2 scheme offers no advantages over the Type 3 scheme.

4. Randomizable structure-preserving signatures

We present the Type 2 randomizable structure-preserving signature scheme from [3] and our Type 3 analogue of it. The Type 3 scheme was obtained by following the general recipe given in [11] for converting a protocol from the Type 2 setting to the Type 3 setting.

4.1. Type 2 randomizable structure-preserving signature scheme [3].

(1) *Setup.* Let $e_2 : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ be a Type 2 pairing where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ have order n; G, H are fixed generators of $\mathbb{G}_1$, $\mathbb{G}_2$, respectively.
(2) *Key generation.* The secret key is $v, w \in_R [1, n-1]$. The public key is (V, W) where $V = G^v$ and $W = G^w$.
(3) *Signature generation.* To sign $M \in \mathbb{G}_2$, select $r \in_R [1, n-1]$ and compute $R = H^r$ and $S = M^v H^{r^2+w}$. The signature on M is (R, S).
(4) *Randomization.* To randomize $(M, (R, S))$, select $\alpha \in_R [1, n-1]$ and compute $R' = RH^\alpha$ and $S' = SR^{2\alpha}H^{\alpha^2}$. The randomized signature on M is (R', S').
(5) *Signature verification.* To verify a signed message $(M, (R, S))$, check that
    (a) $M, R, S \in \mathbb{G}_2$; and
    (b) $e_2(G, S) = e_2(V, M) \cdot e_2(\psi(R), R) \cdot e_2(W, H)$.

In [3, Theorem 1], the Type 2 scheme is proven secure against generic forgers. Signatures are comprised of two $\mathbb{G}_2$ elements. Signature verification requires three $\mathbb{G}_2$ membership tests and one PPE verification.

4.2. Type 3 randomizable structure-preserving signature scheme.

(1) *Setup.* Let $e_3 : \mathbb{G}_1 \times \mathbb{G}_3 \longrightarrow \mathbb{G}_T$ be a Type 3 pairing, where $\mathbb{G}_1$, $\mathbb{G}_3$ and $\mathbb{G}_T$ have order n; G, I are fixed generators of $\mathbb{G}_1$, $\mathbb{G}_3$, respectively.

(2) *Key generation.* The secret key is $v, w \in_R [1, n-1]$. The public key is (V, W) where $V = G^v$ and $W = G^w$.

(3) *Signature generation.* To sign $M \in \mathbb{G}_3$, select $r \in_R [1, n-1]$ and compute $R_1 = G^r$, $R_2 = I^r$ and $S = M^v I^{r^2+w}$. The signature on M is (R_1, R_2, S).

(4) *Randomization.* To randomize $(M, (R_1, R_2, S))$, select $\alpha \in_R [1, n-1]$ and compute $R_1' = R_1 G^\alpha$, $R_2' = R_2 I^\alpha$, and $S' = S R_2^{2\alpha} I^{\alpha^2}$. The randomized signature on M is (R_1', R_2', S').

(5) *Signature verification.* To verify a signed message $(M, (R_1, R_2, S))$, check that
   (a) $R_1 \in \mathbb{G}_1$ and $M, R_2, S \in \mathbb{G}_3$;
   (b) $e_3(R_1, I) = e_3(G, R_2)$; and
   (c) $e_3(G, S) = e_3(V, M) \cdot e_3(R_1, R_2) \cdot e_3(W, I)$.

Correctness of the Type 3 signature scheme is easily verified since

$$\begin{aligned} e_3(G,S) &= e_3(G, M^v I^{r^2+w}) \\ &= e_3(G, M^v) \cdot e_3(G, I^{r^2}) \cdot e_3(G, I^w) \\ &= e_3(G^v, M) \cdot e_3(G^r, I^r) \cdot e_3(G^w, I) \\ &= e_3(V, M) \cdot e_3(R_1, R_2) \cdot e_3(W, I). \end{aligned}$$

The security proof given in [3, Theorem 1] that the Type 2 scheme is secure against generic forgers also applies (with minimal changes) to the Type 3 signature scheme. We note that the security proof in [3] does not use the isomorphism ψ.

Signatures for the Type 3 scheme are comprised of one $\mathbb{G}_1$ element and two $\mathbb{G}_3$ elements. Signature verification requires one $\mathbb{G}_1$ membership test, three $\mathbb{G}_3$ membership tests, and two PPE verifications.

We note that the verification equation in step 5(b) of the Type 3 scheme cannot be omitted. Indeed, if this step is omitted then the scheme succumbs to the following random message attack. The forger first obtains a signed message $(M, (R_1, R_2, S))$. It then computes $M' = MR_2$ and $R_1' = R_1 V^{-1}$, thereby obtaining a valid forgery $(M', (R_1', R_2, S))$. Indeed, this attack is anticipated by the proof of Theorem 2 of [2].

4.3. **Comparisons.** The subgroup membership tests in step 5(a) of the Type 2 randomizable structure-preserving signature scheme cannot be omitted. If they are, then an attacker can proceed as follows. Having obtained a valid signature pair $(M, (R, S))$, she computes $M' = MR$ and $R' = RV^{-1}$. Note that $\rho(R') = \rho(R)$. Then $(M', (R', S))$ is a valid signed message since the term $e_2(V, M) \cdot e_2(\psi(R), R)$ in step 5(b) of signature verification remains unchanged:

$$\begin{aligned} e_2(V, M') \cdot e_2(\psi(R'), R') &= e_2(V, MR) \cdot e_2(\psi(R) \cdot \psi(V^{-1}), R') \\ &= e_2(V, M) \cdot e_2(V, R) \cdot e_2(\psi(R), R') \cdot e_2(\psi(V), R')^{-1} \\ &= e_2(V, M) \cdot e_3(V, \rho(R)) \cdot e_3(\psi(R), \rho(R)) \cdot e_3(V, \rho(R))^{-1} \\ &= e_2(V, M) \cdot e_2(\psi(R), R). \end{aligned}$$

The comparisons made between the Type 2 and Type 3 strongly unforgeable structure-preserving signature schemes in §3.3 are also valid for the Type 2 and Type 3 randomizable structure-preserving signature schemes in §4.1 and §4.2. Namely, the Type 3 scheme has smaller signatures, faster signature generation, faster signature verification in stand-alone applications (since it requires the verification of two PPEs instead of four PPEs for the Type 2 scheme), and faster signature verification when used with Groth-Sahai proofs (since both schemes have two PPEs and three group elements in signatures, but the Type 3 proofs are DDH-based instead of DLIN-based).

As mentioned in [3], randomizable structure-preserving signature schemes are useful in building anonymization protocols because the signature component that is uniformly distributed and independent of the message can be revealed without leaking any information about the message or the original signature from which the randomized signature was derived. In the Type 2 randomizable signature scheme of §4.1, the signature component R can be made public. In that case, only the single PPE in step 5(b) of signature verification needs to be transformed when used in conjunction with Groth-Sahai proofs (and the PPE is of the form described in §A.1). Similarly, in the Type 3 randomizable signature scheme of §4.2, the signature components R_1 and R_2 can be made public. In that case, only the single PPE in step 5(c) of signature verification needs to be transformed when used in conjunction with Groth-Sahai proofs (and the PPE is of the form described in §A.2).

In both situations, i.e., whether the message-independent signature components are made public or not, the Type 3 scheme is superior in all respects to its Type 2 counterpart.

5. Concluding remarks

We presented natural Type 3 analogues of the Type 2 strongly unforgeable and randomizable structure-preserving signature schemes that were proposed in [3]. By properly accounting for subgroup membership testing of group elements in signatures, we have shown that the Type 3 schemes are superior to their Type 2 counterparts when the signature schemes are used in a stand-alone setting, and when used in conjunction with Groth-Sahai proofs. The Type 3 schemes are also the most efficient among all structure-preserving signature schemes. We conclude that the question posed in [11] of the existence of a cryptographic protocol which necessarily has to be restricted to Type 2 for implementation or security reasons is still open.

References

[1] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev and M. Ohkubo, “Structure-preserving signatures and commitments to group elements”, *Advances in Cryptology – CRYPTO 2010*, LNCS 6223 (2010), 209–236.

[2] M. Abe, J. Groth, K. Haralambiev and M. Ohkubo, “Optimal structure-preserving signatures in asymmetric bilinear groups”, *Advances in Cryptology – CRYPTO 2011*, LNCS 6841 (2011), 649–666.

[3] M. Abe, J. Groth, M. Ohkubo and M. Tibouchi, “Structure-preserving signatures from Type II pairings”, *Advances in Cryptology – CRYPTO 2014*, LNCS 8616 (2014), 390–407.

[4] M. Abe, J. Groth, M. Ohkubo and M. Tibouchi, “Structure-preserving signatures from Type II pairings”, full version of [3]. Available at http://eprint.iacr.org/2014/312.

[5] P. Barreto, B. Lynn and M. Scott, “Constructing elliptic curves with prescribed embedding degrees”, *Security in Communication Networks – SCN 2002*, LNCS 2576 (2003), 257–267.

[6] P. Barreto, B. Lynn and M. Scott, "Efficient implementation of pairing-based cryptosystems", *Journal of Cryptology*, 17 (2004), 321–334.

[7] P. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order", *Selected Areas in Cryptography – SAC 2005*, LNCS 3897 (2006), 319–331.

[8] M. Bellare, J. Garay and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures" *Advances in Cryptology – EUROCRYPT '98*, LNCS 1403 (1998), 236–250.

[9] M. Chase, "Efficient non-interactive zero-knowledge proofs for privacy applications", Ph.D. thesis, Brown University, 2008.

[10] S. Chatterjee, D. Hankerson, E. Knapp and A. Menezes, "Comparing two pairing-based aggregate signature schemes", *Designs, Codes and Cryptography*, 55 (2010), 141–167.

[11] S. Chatterjee and A. Menezes, "On cryptographic protocols employing asymmetric pairings – The role of ψ revisited", *Discrete Applied Mathematics*, 159 (2011), 1311–1322.

[12] L. Chen, Z. Cheng and N. Smart, "Identity-based key agreement protocols from pairings", *International Journal of Information Security*. 6 (2007), 213–241.

[13] A. Ferrara, M. Green, S. Hohenberger and M. Pedersen, "Practical short signature batch verification", *Topics in Cryptology – CT-RSA 2009*, LNCS 5473 (2009), 309–324.

[14] S. Galbraith, K. Paterson and N. Smart, "Pairings for cryptographers", *Discrete Applied Mathematics*, 156 (2008), 3113–3121.

[15] E. Ghadafi, N. Smart and B. Warinschi, "Groth-Sahai proofs revisited", *Public-Key Cryptography – PKC 2010*, LNCS 6056 (2010), 177–192.

[16] J. Groth and A. Sahai, "Efficient noninteractive proof systems for bilinear groups", *SIAM Journal on Computing* 41 (2012), 1193-1232.

[17] F. Hess, N. Smart and F. Vercauteren, "The eta pairing revisited", *IEEE Transactions on Information Theory*, 52 (2006), 4595–4602.

[18] E. Kachisa, E. Schaefer and M. Scott, "Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field", *Pairing-Based Cryptography – Pairing 2008*, LNCS 5209 (2008), 126–135.

[19] A. Miyaji, M. Nakabayashi and S. Tanako, "New explicit condition of elliptic curve trace for FR-reduction", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E84-A (2001), 1234–1243.

[20] F. Vercauteren, "Optimal pairings", *IEEE Transactions on Information Theory*, 56 (2010), 455–461.

Appendix A. Groth-Sahai proofs

A.1. **DLIN-based proofs.** Let $A \in \mathbb{G}_1$ and $t \in \mathbb{G}_T$. We present a Groth-Sahai non-interactive witness-indistinguishable proof of knowledge of $Y \in \mathbb{G}_2$ such that $e_2(A, Y) = t$. The NIWI proof is derived from the general description in §4.2 of [15]. It can also be used with Type 3 pairings. Security is based on the decisional linear (DLIN) assumption.

(1) *Setup.* Let $e_2 : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ be a Type 2 pairing.

(2) *Common reference string.* Let H be a generator of $\mathbb{G}_2$. Let $a, t, i, j \in_R [1, n-1]$, and define $U = aH$, $V = tH$, $I = iU$, $J = jV$, $K = (i + j)H$. The common string is (H, U, V, I, J, K).

(3) *Commitment.* Select $s_1, s_2, s_3 \in_R [1, n-1]$ and compute $d_1 = s_1 U + s_3 I$, $d_2 = s_2 V + s_3 J$, $d_3 = Y + s_1 H + s_2 H + s_3 K$. The commitment is $d = (d_1, d_2, d_3)$.

(4) *Proof.* Compute $\theta_1 = s_1 A$, $\theta_2 = s_2 A$ and $\theta_3 = s_3 A$. The proof is $\theta = (\theta_1, \theta_2, \theta_3)$.

(5) *Verification.* Check that $\theta_1, \theta_2, \theta_3 \in \mathbb{G}_1$, $d_1, d_2, d_3 \in \mathbb{G}_2$, and

$$\begin{aligned} e_2(A, d_1) &= e_2(\theta_1, U) \cdot e_2(\theta_3, I) \\ e_2(A, d_2) &= e_2(\theta_2, V) \cdot e_2(\theta_3, J) \\ e_2(A, d_3) &= e_2(\theta_1, H) \cdot e_2(\theta_2, H) \cdot e_2(\theta_3, K) \cdot t. \end{aligned}$$

A.2. **DDH-based proofs.** Let $A \in \mathbb{G}_1$ and $t \in \mathbb{G}_T$. We present a Groth-Sahai non-interactive witness-indistinguishable proof of knowledge of $Y \in \mathbb{G}_3$ such that $e_3(A, Y) = t$. The NIWI proof is derived from the general description in §4.1 of [15]. Security is based on the decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}_3$. Since the decisional Diffie-Hellman problem is easy in $\mathbb{G}_2$, the NIWI proof has no counterpart with Type 2 pairings.

(1) *Setup.* Let $e_3 : \mathbb{G}_1 \times \mathbb{G}_3 \longrightarrow \mathbb{G}_T$ be a Type 3 pairing.
(2) *Common reference string.* Let I be a generator of $\mathbb{G}_3$. Let $a, t \in_R [1, n-1]$, and define $U = aI$, $V = tI$, $J = tU$. The common string is (I, U, V, J).
(3) *Commitment.* Select $s_1, s_2 \in_R [1, n-1]$ and compute $d_1 = s_1 I + s_2 V$ and $d_2 = Y + s_1 U + s_2 J$. The commitment is $d = (d_1, d_2)$.
(4) *Proof.* Compute $\theta_1 = s_1 A$ and $\theta_2 = s_2 A$. The proof is $\theta = (\theta_1, \theta_2)$.
(5) *Verification.* Check that $\theta_1, \theta_2 \in \mathbb{G}_1$, $d_1, d_2 \in \mathbb{G}_3$, and

$$\begin{aligned} e_3(A, d_1) &= e_3(\theta_1, I) \cdot e_3(\theta_2, V) \\ e_3(A, d_2) &= e_3(\theta_1, U) \cdot e_3(\theta_2, J) \cdot t. \end{aligned}$$

Department of Computer Science and Automation, Indian Institute of Science
E-mail address: sanjit@csa.iisc.ernet.in

Department of Combinatorics & Optimization, University of Waterloo
E-mail address: ajmeneze@uwaterloo.ca