TYPE 2 STRUCTURE-PRESERVING SIGNATURE SCHEMES REVISITED

SANJIT CHATTERJEE AND ALFRED MENEZES

Abstract. At CRYPTO 2014, Abe, Groth, Ohkubo and Tibouchi presented generic-signer structure-preserving signature schemes using Type 2 pairings. The schemes were claimed to enjoy the smallest number of group elements in signatures and the fastest signature verification. By properly accounting for the concrete structure of the underlying group and 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 in every aspect. We also formally establish that *all* Type 2 structure-preserving signature schemes can be converted to the Type 3 setting without any penalty in security or efficiency, and show that the converse is false.

1. Introduction

The term 'structure-preserving signature scheme' was coined in 2010 by Abe et al. [1] but was first introduced by Groth [17]. 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 have been used in the design of numerous cryptographic protocols; a list of these protocols can be found in [4]. One of the primary reasons for the popularity of structure-preserving signature schemes in protocol design is that they are fully compatible with the breakthrough Groth-Sahai constructions of pairing-based non-interactive witness-indistinguishable (NIWI) and non-interactive zero-knowledge (NIZK) proof systems [18].

In typical applications of structure-preserving signature schemes when used in conjunction with, say, 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[1]. 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[2]. 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 [10, §3.4]). For such reasons, researchers have strived to design structure-preserving signature

Date: August 18, 2014; updated on September 26, 2014.

[1]We use the example of Groth-Sahai because many applications of structure-preserving signature schemes are in conjunction with such non-interactive proof systems. However, structure-preserving signatures find application in other contexts too – see the recent work of Hanser and Slamanig [19].

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

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[3].

At TCC 2014, Abe et al. [3] extended the aforementioned optimality results to the Type 1 setting, thereby unifying the Type 1 and Type 3 settings. They also proposed a selectively randomizable structure-preserving signature scheme which is optimal in terms of signature size and verification complexity in both Type 1 and Type 3 settings.

At CRYPTO 2014, Abe et al. [4] continued their investigation of structure-preserving signature schemes in the Type 2 setting. They 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 smallest signature size and fastest signature verification. Furthermore, they investigated lower bounds on signature size and number of verification equations and showed that their constructions in Type 2 are optimal. 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. According to the authors [4]: "This is significant from a high level pairing-based cryptography perspective, as it provides a concrete example of a property that can be obtained in the Type 2 setting but not in the other settings." This is contrary to the arguments presented in [12] 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 deconstruct the Abe et al. schemes in terms of the underlying elliptic curve group structure in the Type 2 setting. We show that the analysis of the Type 2 generic-signer structure-preserving signature schemes in [4] neglected to account for the concrete group structure and subgroup membership testing of group elements in a signature, leading to erroneous conclusions (see Table 1 of [4]). Incorporating these subgroup membership tests into the signature verification increases the number of group elements in signatures and also increases the number of PPEs in signature verification. We analyze the cost when these structure-preserving signature schemes are composed with Groth-Sahai proofs and show that not all these subgroup membership tests can be dispensed with when the signature scheme is composed with such a proof system.

[3]A *generic signer* has access only to generic group operations in the bilinear pairing setting. This notion was first introduced by Abe et al. in [2] to establish their lower bound results. The same model was used in [4] where it was claimed that all existing structure-preserving signature schemes use generic signing algorithms and "it would be a surprising result in itself to construct a structure-preserving signature with a non-generic signer". Hence, in this paper we focus on the case of generic signers.

Furthermore, since Groth-Sahai proofs in the Type 2 setting are significantly more costly than in the Type 3 setting, the Type 2 schemes are not as efficient as claimed in [4] either in the stand-alone setting or when composed with Groth-Sahai proofs. 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 in all aspects.

Continuing the process of deconstruction, we formally establish that *all* Type 2 generic-signer structure-preserving signature schemes can be converted to Type 3 without any penalty in security and efficiency, but not all Type 3 schemes have a secure Type 2 counterpart. Further, we exhibit the impossibility of having a single pairing-based verification equation in the Type 2 setting even when messages are drawn from $\mathbb{G}_2$ and thereby put the lower bound results of [4] in the correct perspective. Our results demonstrate that any Type 2 structure-preserving signature scheme is merely an inefficient implementation of a corresponding Type 3 scheme.

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 [4] 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 [4], and show that the Type 3 scheme is more efficient. In §5, we present our conversion framework for generic-signer structure-preserving signature schemes from the Type 2 setting to the Type 3 setting, the separation between Type 2 and Type 3, and the impossibility of having a single pairing-based verification equation in the Type 2 setting. We draw our conclusions in §6.

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 [7]. Some prominent families of elliptic curves with even embedding degree include the MNT [22], BN [8], KSS [21], and BLS [6] 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 the order-n subgroup of the multiplicative subgroup of $\mathbb{F}_{q^k}$.

2.1. **Type 3 pairings.** Following [15], we denote by D the CM discriminant of E and set

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

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 $n \mid \#\tilde{E}(\mathbb{F}_{q^d})$ [20]. 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 [15]. There are several Type 3 pairings, of which the most efficient is Vercauteren's optimal pairing [23].

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 [15]. 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 (reduced) 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 [15]. However, as observed in [11], 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 $E[n] \setminus (\mathbb{G}_1 \cup \mathbb{G}_3)$, 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

$$D(Q) = (\psi(Q), \rho(Q)), \tag{5}$$

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 [11] 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 notable 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 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 $e_3(Q_1, I) = e_3(G, Q_2)$ [13]. 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 §3, §4 and §5, 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 [4] and our Type 3 analogue of it. The Type 3 scheme was obtained by following the general recipe given in [12] for converting a protocol from the Type 2 setting to the Type 3 setting.

3.1. Type 2 strongly unforgeable structure-preserving signature scheme [4].

(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 [4, Theorem 2], the Type 2 scheme is proven strongly secure[4] 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.

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 [4, 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 we follow the strategy of [12] in the conversion. The Type 3 scheme is obtained by first replacing all $\mathbb{G}_2$ elements by the corresponding $\mathbb{H}_2$ elements and then discarding the redundant $\mathbb{G}_1$ elements that are not used either in the construction or in security argument in the Type 2 setting.

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.

[4] 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*.

3.3. **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 significantly 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 significantly 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. Note, however, that the high cost of PPE verifications can be mitigated by batching [9, 14].

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 [4] is no longer applicable since the proof makes the assumption that $M, R, S \in \mathbb{G}_2$. Second, and more importantly, 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[5] select a second point $R' \in E[n]$ with $R' \neq R$ and $\psi(R') = \psi(R)$, thereby obtaining a second valid signed message $(M, (R', S))$. Similarly, given $(M, (R, S))$ one can obtain a second valid signed message $(M', (R, S))$ or $(M, (R, S'))$ if membership tests for M or S are omitted.

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 non-interactive proof systems like Groth-Sahai 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 M, R or S. In other words, it needs to convince the verifier that it possesses a solution to the following PPE:

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

In this equation, the group elements G, H, V and W are known to the verifier, whereas the variables are $M, R, S \in \mathbb{G}_2$. However, since Groth-Sahai proofs do not have a mechanism for incorporating the evaluation of $\psi(R)$, the variables in (6) are actually M, $\psi(R)$ and S.

[5]Given $R \in \mathbb{G}_2$, one computes $R_1 = \psi(R)$ and selects arbitrary $R_2' \in \mathbb{G}_3$ with $R_2' \neq R \cdot R_1^{-1}$. Then $R' = R_1 \cdot R_2'$ satisfies $\psi(R') = R_1$ and $R' \neq R$.

In other words, a Groth-Sahai proof for (6) only convinces a verifier that the prover knows $R_1 \in \mathbb{G}_1$ and $M, S \in \mathbb{G}_2$ that satisfy the following PPE:

$$e_2(WR_1, S) = e_2(V, M) \cdot e_2(G, H). \tag{7}$$

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 $R_1 \in \mathbb{G}_1$, $M, R, S \in \mathbb{G}_2$ to the following collection of PPEs:

$$e_2(WR_1, S) = e_2(V, M) \cdot e_2(G, H) \tag{8}$$

$$e_2(R_1, H) = e_2(G, R). \tag{9}$$

When composed with Groth-Sahai proof systems, the verification now has *two* PPEs. This is in contrast to the claim made in [4] 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 $R, S \in \mathbb{G}_2$.

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 $\mathbb{G}_2$ [16], 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 $\mathbb{G}_3$ [18]. 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 a solution (X, Y) to the equation $e_2(A, X) \cdot e_2(B, Y) = t$ in Appendix A.1 is significantly more costly than the DDH-based proof of knowledge of a solution (X, Y) to the equation $e_3(A, X) \cdot e_3(B, Y) = t$ in Appendix A.2; see also the performance estimates given in §3.4 of [10]. 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 [4] and our Type 3 analogue of it. The Type 3 scheme was obtained by following the general recipe given in [12] for converting a protocol from the Type 2 setting to the Type 3 setting.

4.1. **Type 2 randomizable structure-preserving signature scheme** [4]**.**

(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 [4, 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

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

Following the strategy outlined in §3.2, the security proof given in [4, Theorem 1] that the Type 2 scheme is secure against generic forgers can be modified (with minimal changes) for the Type 3 signature scheme.

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 $R'_1 = R_1V^{-1}$, 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 [4], 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. A closer look at Type 2 schemes

In this section we first establish that *all* Type 2 generic-signer structure-preserving signature schemes can be transformed to the Type 3 setting without any penalty in security or efficiency. Next, we demonstrate the impossibility of having signature verification with a single pairing-product equation in the Type 2 setting when messages are drawn from

$\mathbb{G}_2$. Finally, we show a separation between the Type 2 and Type 3 settings by proposing a Type 3 signature scheme that has no secure Type 2 counterpart.

Based on the claimed benefits of their concrete structure-preserving signature schemes in terms of the number of group elements in signatures and verification complexity, Abe et al. [4] asserted that the Type 2 setting "permits the construction of cryptographic schemes with unique properties" and, thereby, settle the open question in [12] in the negative. In contrast, the results of this section formally establish that *all* Type 2 generic-signer structure-preserving signature schemes are merely Type 3 schemes in disguise and cannot beat the established lower bound results even when messages are drawn from $\mathbb{G}_2$.

5.1. **Conversion from Type 2 to Type 3.** Recall the definition of structure-preserving signatures (SPS) from [4, Definition 4]. Based on that definition, any generic-signer structure-preserving signature scheme with message space $\mathbb{G}_2$ can be described as follows.

SPS-T2

(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 contains elements $u_1, u_2, \ldots, v_1, v_2, \ldots \in_R [1, n-1]$. The public key contains elements $U_1, U_2, \ldots \in \mathbb{G}_1$, $V_1, V_2, \ldots \in \mathbb{G}_2$, where $U_i = G^{u_i}$ and $V_j = H^{v_j}$. Note that because the signer is generic, we can assume without loss of generality that the signer knows the discrete logarithm of the U_i and the V_j.

(3) *Signature generation.* The message is $M \in \mathbb{G}_2$. However, unlike the public key, we cannot assume that the signer knows the discrete logarithm of $M = H^m$. Since the signing algorithm can only use generic group operations, a generic signer can only construct signature elements of the form $S_i = \psi(M)^{\alpha_i} G^{\beta_i} \in \mathbb{G}_1$ and $T_j = M^{\gamma_j} H^{\delta_j}$ where $\alpha_i, \beta_i, \gamma_j, \delta_j \in [1, n-1]$ are independent of m. Finally, the algorithm outputs a signature containing elements $(S_1, S_2, \ldots) \in \mathbb{G}_1$ and $(T_1, T_2, \ldots) \in \mathbb{G}_2$.

(4) *Signature verification.* Given a message M and corresponding signature $(S_1, S_2, \ldots, T_1, T_2, \ldots)$, the verifier does the following:
   (a) check that $S_1, S_2, \ldots \in \mathbb{G}_1$;
   (b) check that $M \in G_2$ and $T_1, T_2, \ldots \in \mathbb{G}_2$;
   (c) verify a collection of equations of the following form:

$$\prod_i \prod_j e_2(S_i, T_j)^{a_{qij}} \cdot \prod_i \prod_j e_2(S_i, V_j)^{b_{qij}} \cdot \prod_j e_2(\psi(M), T_j)^{c_{qj}} \cdot \prod_j e_2(\psi(M), V_j)^{d_{qj}}$$
$$\cdot \prod_i e_2(S_i, M)^{e_{qi}} \cdot \prod_i e_2(U_i, M)^{f_{qi}} \cdot \prod_i \prod_j e_2(U_i, T_j)^{g_{qij}} \cdot e_2(\psi(M), M)^{h_q} \quad = \quad 1$$

Note: We use the augmented set $S = \{S_1, S_2, \ldots\} \cup \{\psi(T_1), \psi(T_2), \ldots\}$ in the above verification equation. However, there is no need to consider the elements $\psi(V_j)$ separately because they can, without loss of generality, be included in the public key.

We now propose the following transformation to convert SPS-T2 from the Type 2 to the Type 3 setting. The transformation utilizes the efficiently-computable isomorphism $D : \mathbb{G}_2 \longrightarrow \mathbb{H}_2$ given by $D(Q) = (\psi(Q), \rho(Q))$ where $\mathbb{H}_2 \subseteq \mathbb{G}_1 \times \mathbb{G}_3$ (see §2). Our strategy is very simple. We apply D so that all $\mathbb{G}_2$ elements in SPS-T2 are replaced by their "shorter

representation" as elements of $\mathbb{H}_2$. This strategy, together with the observation that the computation of a Type 2 pairing e_2 is efficiently reduced to the task of computing a Type 3 pairing e_3 (see equation (4)), immediately yields the following Type 3 structure-preserving signature scheme.

SPS-T3

(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.* For each element $V_j = H^{v_j}$ in SPS-T2, compute $V_{j_1} = G^{v_j}$ and $V_{j_2} = I^{v_j}$. The secret key contains elements $u_1, u_2, \ldots, v_1, v_2, \ldots \in_R [1, n-1]$. The public key contains elements $U_1, U_2, \ldots \in \mathbb{G}_1$ (as in SPS-T2) and (V_{1_1}, V_{1_2}), $(V_{2_1}, V_{2_2}), \ldots \in \mathbb{H}_2$.

(3) *Signature generation.* The message $M = H^m$ in SPS-T2 can be written as $(M_1, M_2) = (G^m, I^m) \in \mathbb{H}_2$. Recall that using generic group operations, a generic signer in SPS-T2 can only construct $S_i = M_1^{\alpha_i} G^{\beta_i}$ and $T_j = M^{\gamma_j} H^{\delta_j}$ where $\alpha_i, \beta_i, \gamma_j, \delta_j$ are independent of m. Representing T_j as an element of $\mathbb{H}_2$ we have $T_j = (T_{j_1}, T_{j_2}) = (M_1^{\gamma_j} G^{\delta_j}, M_2^{\gamma_j} I^{\delta_j}) \in \mathbb{H}_2$. It is easy to see that a generic signer can compute the signature element $T_j \in \mathbb{G}_2$ *if and only if* she can compute $M_1^{\gamma_j} G^{\delta_j} \in \mathbb{G}_1$ and $M_2^{\gamma_j} I^{\delta_j} \in \mathbb{G}_3$. Using the above idea we can convert each signature element $T_j \in \mathbb{G}_2$ of SPS-T2 to $(T_{j_1}, T_{j_2}) \in \mathbb{H}_2$ and thereby obtain the corresponding signature elements in SPS-T3. Finally, the algorithm outputs a signature of the form $S_1, S_2, \ldots \in \mathbb{G}_1$ and $(T_{1_1}, T_{1_2}), (T_{2_1}, T_{2_2}), \ldots \in \mathbb{H}_2$.

(4) *Signature verification.* Given a message (M_1, M_2) and corresponding signature $(S_1, S_2, \ldots, (T_{1_1}, T_{1_2}), (T_{2_1}, T_{2_2}), \ldots)$, the verifier does the following:
   (a) check that $S_1, S_2, \ldots \in \mathbb{G}_1$;
   (b) check that $(M_1, M_2), (T_{1_1}, T_{1_2}), (T_{2_1}, T_{2_2}), \ldots \in \mathbb{H}_2$;
   (c) verify a set of equations of the following form:

$$\prod_i \prod_j e_3(S_i, T_{j_2})^{a_{qij}} \cdot \prod_i \prod_j e_3(S_i, V_{j_2})^{b_{qij}} \cdot \prod_j e_3(M_1, T_{j_2})^{c_{qj}} \cdot \prod_j e_3(M_1, V_{j_2})^{d_{qj}}$$
$$\cdot \prod_i e_3(S_i, M_2)^{e_{qi}} \cdot \prod_i e_3(U_i, M_2)^{f_{qi}} \cdot \prod_i \prod_j e_3(U_i, T_{j_2})^{g_{qij}} \cdot e_3(M_1, M_2)^{h_q} \quad = \quad 1$$

Note: We use the augmented set $S = \{S_1, S_2, \ldots\} \cup \{T_{1_1}, T_{2_1}, \ldots\}$ in the above verification equation. As already observed in the context of SPS-T2, there is no need to consider the public key elements $V_{1_1}, V_{2_1}, \ldots$ separately.

Next we show that the derived Type 3 scheme SPS-T3 is as secure as its original Type 2 counterpart SPS-T2 and maintains all the claimed benefits of SPS-T2.

Claim 1. *SPS-T2 is* X*-secure if and only if SPS-T3 is* X*-secure, where* X *stands for any standard security notion for signature schemes such as existential unforgeability under chosen message attack (EUF-CMA).*

Proof. Given an adversary against SPS-T3, we can easily construct an adversary against SPS-T2 and vice versa. In the framework of the conversion described above, we have consistently replaced all $\mathbb{G}_2$ elements in SPS-T2 by the corresponding $\mathbb{H}_2$ elements to derive the corresponding algorithms of SPS-T3. The equivalence between SPS-T2 and SPS-T3

follows from the facts that (i) $D : \mathbb{G}_2 \longrightarrow \mathbb{H}_2$ is an efficiently-computable isomorphism whose inverse is also efficiently computable; and (ii) the task of computing e_2 is efficiently reduced to the task of computing e_3. □

Remark 1. SPS-T3 does not have any efficiency gain (or loss) compared to SPS-T2. Further optimizations for SPS-T3 are usually possible by removing some redundant group elements after a careful scrutiny of the construction and its security argument as suggested in [12]. For example, the Type 3 schemes described in §3 and §4 are optimized versions of their Type 2 counterparts obtained by following the general recipe given above.

Remark 2. The subgroup membership tests described in step 4(b) of SPS-T2 and SPS-T3 involve pairing-based verification equations. We have observed in §3 and §4 that avoiding subgroup membership tests can lead to a random message attack in both the Type 2 and Type 3 settings. Apart from these pairing-based verifications of subgroup membership, signature verification will involve at least one more pairing product equation. See the proof of Theorem 2 below for further details.

5.2. **Impossibility of single PPE in verification.** In Theorem 2 of [2], Abe et al. showed that there is no Type 3 structure-preserving signature scheme with a single pairing-based verification equation that is existentially unforgeable under random message attack. The original argument was for messages in $\mathbb{G}_1$, but can be easily extended when messages are from $\mathbb{G}_3$. In Theorem 3 of [4], Abe et al. showed a similar impossibility result for Type 2 structure-preserving signature schemes with messages in $\mathbb{G}_1$.

We now generalize the above results to show that the impossibility holds even when the messages are drawn from $\mathbb{H}_2$. As a corollary, one concludes that there is no Type 2 structure-preserving signature scheme with a single pairing-based verification equation that is existentially unforgeable under random message attack.

Theorem 2. *No structure-preserving signature scheme with a single pairing-product equation based signature verification is secure in the sense of existential unforgeability under random message attack.*

Proof. The case of messages in $\mathbb{G}_1$ in the Type 3 setting (resp. the Type 2 setting) is proved in [2, Theorem 2] (resp. [4, Theorem 3]). The case of messages in $\mathbb{G}_3$ in the Type 3 setting is analogous to the proof of Theorem 2 in [2]. The case of the Type 1 setting was settled in [3, Theorem 4].

We now show the same impossibility for messages in $\mathbb{G}_2$. For ease of exposition, we will use the structure of SPS-T3, which we have already shown equivalent to SPS-T2, and the message space $\mathbb{H}_2$ (recall that $\mathbb{H}_2$ is isomorphic to $\mathbb{G}_2$, and that an element of $\mathbb{H}_2$ is comprised of a pair in $\mathbb{G}_1 \times \mathbb{G}_3$ both components of which have the same discrete logarithm with respect to the fixed generators G and I). Our argument closely follows the proof of Theorem 2 from [2] but needs to take care of additional complications due to the structure of $\mathbb{H}_2$.

Recall the signature verification for SPS-T3 where in step 4(c) we described the general form of a verification equation. Our claim is that having a *single* verification equation of the form 4(c) and omitting the subgroup membership test in step 4(b) lead to a random message attack. For simplicity, we assume that the signature contains two elements of $\mathbb{H}_2$. Note that Abe et al. claim that two group elements is the optimal signature size in Type 2

– see Table 1 of [4]. However, it is easy to see that our result holds for the more general case.

Consider a structure-preserving signature scheme for messages in $\mathbb{H}_2$ with verification key containing group elements $U_1, U_2, \ldots \in \mathbb{G}_1$, $V_1, V_2, \ldots \in \mathbb{G}_3$, and $Z \in \mathbb{G}_T$.[6] For simplicity, we will assume there are two U_i's and two V_i's. A signature is of the form $(S_1, T_1), (S_2, T_2) \in \mathbb{H}_2$ and is verified by the following PPE:

$$\begin{aligned} e_3(S_1,T_1)^{a_{11}} \cdot e_3(S_1,T_2)^{a_{12}} \cdot e_3(S_2,T_1)^{a_{21}} \cdot e_3(S_2,T_2)^{a_{22}} \\ \cdot e_3(S_1,V_1)^{b_{11}} \cdot e_3(S_1,V_2)^{b_{12}} \cdot e_3(S_2,V_1)^{b_{21}} \cdot e_3(S_2,V_2)^{b_{22}} \\ \cdot e_3(M_1,T_1)^{c_{11}} \cdot e_3(M_1,T_2)^{c_{12}} \cdot e_3(M_1,V_1)^{d_{11}} \cdot e_3(M_1,V_2)^{d_{12}} \\ \cdot e_3(S_1,M_2)^{c_{21}} \cdot e_3(S_2,M_2)^{c_{22}} \cdot e_3(U_1,M_2)^{d_{21}} \cdot e_3(U_2,M_2)^{d_{22}} \\ \cdot e_3(U_1,T_1)^{e_{11}} \cdot e_3(U_1,T_2)^{e_{12}} \cdot e_3(U_2,T_1)^{e_{21}} \cdot e_3(U_2,T_2)^{e_{22}} \\ \cdot e_3(M_1,M_2)^{f} \quad = \quad Z. \end{aligned}$$

Note that terms like $e_3(U_i, V_j)$ can be incorporated in $Z \in \mathbb{G}_T$ without loss of generality.

Given a signature $(S_1, T_1), (S_2, T_2) \in \mathbb{H}_2$ on a random message $(M_1, M_2) \in \mathbb{H}_2$, we isolate S_1, S_2 and M_2 in the verification equation to obtain:

$$\begin{aligned} A_1 &= T_1^{a_{11}} T_2^{a_{12}} V_1^{b_{11}} V_2^{b_{12}} \\ A_2 &= T_1^{a_{21}} T_2^{a_{22}} V_1^{b_{21}} V_2^{b_{22}} \\ B_1 &= M_1^{f} S_2^{c_{22}} U_1^{d_{21}} U_2^{d_{22}} \\ B_2 &= M_1^{f} S_1^{c_{21}} U_1^{d_{21}} U_2^{d_{22}}. \end{aligned}$$

Suppose that $A_1 \neq M_2^{-c_{21}}$. We first rewrite the verification equation as

$$e_3(S_1, M_2)^{c_{21}} \cdot e_3(S_1, A_1) \cdot e_3(B_1, M_2) \cdot \hat{Z} = Z.$$

Note that $\hat{Z}$ does not contain the terms S_1 and M_2. If $c_{21} = 0$, then we set $S_1^{'} = S_1 B_1^{-1}$ and $M_2^{'} = M_2 A_1$. For the message $(M_1, M_2^{'})$ we have a forged signature $(S_1^{'}, T_1), (S_2, T_2)$.[7] If $c_{21} \neq 0$, then we set $S_1^{'} = S_1^{-1} B_1^{-2/c_{21}}$ and $M_2^{'} = M_2^{-1} A_1^{-2/c_{21}}$ and the corresponding forgery is $(S_1^{'}, T_1), (S_2, T_2)$ for message $(M_1, M_2^{'})$.

A similar attack works when $A_2 \neq M_2^{-c_{22}}$.

Suppose now that $A_1 M_2^{c_{21}} = 1$ and $A_2 M_2^{c_{22}} = 1$. So both S_1 and S_2 are cancelled from the verification equation and henceforth we will only consider the signature elements T_1, T_2. Now, the verification equation will be of the form

$$\begin{aligned} e_3(M_1,T_1)^{c_{11}} \cdot e_3(M_1,T_2)^{c_{12}} \cdot e_3(M_1,V_1)^{d_{11}} \cdot e_3(M_1,V_2)^{d_{12}} \\ \cdot e_3(U_1,M_2)^{d_{21}} \cdot e_3(U_2,M_2)^{d_{22}} \\ \cdot e_3(U_1,T_1)^{e_{11}} \cdot e_3(U_1,T_2)^{e_{12}} \cdot e_3(U_2,T_1)^{e_{21}} \cdot e_3(U_2,T_2)^{e_{22}} \\ \cdot e_3(M_1,M_2)^{f} \quad = \quad Z. \end{aligned}$$

[6]Here, as in [2], we have relaxed the original definition of structure-preserving signatures to allow the public verification key to contain an arbitrary element Z from $\mathbb{G}_T$ that appears in the verification equation. As already observed in [2], the relaxation strengthens the impossibility result.

[7]The attack can be prevented by checking that $(M_1, M_2^{'})$ and $(S_1^{'}, T_1)$ are elements of $\mathbb{H}_2$. However that requires *two* additional pairing-product equations in signature verification.

Proceeding as before, we isolate M_1 and M_2 to obtain

$$\begin{aligned} A_3 &= T_1^{c_{11}} T_2^{c_{12}} V_1^{d_{11}} V_2^{d_{12}} \\ B_3 &= U_1^{d_{21}} U_2^{d_{22}}. \end{aligned}$$

Suppose $A_3 \neq M_2^{-f}$. The verification equation can be written as

$$e_3(M_1, M_2)^f \cdot e_3(M_1, A_3) \cdot e_3(B_3, M_2) \cdot Z' = Z.$$

Note that Z' does not contain the elements M_1 and M_2. If $f = 0$, then setting $M_1^{'} = M_1 B_3^{-1}$ and $M_2^{'} = M_2 A_3$ yields the forgery (T_1, T_2) for $(M_1^{'}, M_2^{'})$. If $f \neq 0$, then setting $M_1^{'} = M_1^{-1} B_3^{-2/f}$ and $M_2^{'} = M_2^{-1} A_3^{-2/f}$ yields the forgery (T_1, T_2) for $(M_1^{'}, M_2^{'})$.

Suppose now that $A_3 M_2^f = 1$; so the message element M_1 is also cancelled from the verification equation. Thus the signature verification is reduced to the form:

$$e_3(U_1, M_2)^{d_{21}} \cdot e_3(U_2, M_2)^{d_{22}} \cdot e_3(U_1, T_1)^{e_{11}} \cdot e_3(U_1, T_2)^{e_{12}} \cdot e_3(U_2, T_1)^{e_{21}} \cdot e_3(U_2, T_2)^{e_{22}} = Z.$$

Producing a forgery is now trivial. The adversary obtains signatures (T_1, T_2) and $(T_1^{'}, T_2^{'})$ on random messages (M_1, M_2) and $(M_1^{'}, M_2^{'})$. From these the adversary forms a signature $(T_1^2/T_1^{'}, T_2^2/T_2^{'})$ on a new message $(M_1^2/M_1^{'}, M_2^2/M_2^{'})$. □

5.3. **Separation.** We construct a Type 3 randomizable structure-preserving signature scheme that has no secure counterpart in the Type 2 setting. The Type 3 scheme is a "dual" of the scheme presented in §4.2 in the sense that the former has $V, W \in \mathbb{G}_1$ and $M, S \in \mathbb{G}_3$, whereas the latter has $V, W \in \mathbb{G}_3$ and $M, S \in \mathbb{G}_1$.

(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 $V = I^v$ and $W = I^w$.
(3) *Signature generation.* To sign $M \in \mathbb{G}_1$, select $r \in_R [1, n-1]$ and compute $R_1 = G^r$, $R_2 = I^r$ and $S = M^v G^{r^2+w}$. 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 $S' = S R_1^{2\alpha} G^{\alpha^2}$. 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) $M, R_1, S \in \mathbb{G}_1$ and $R_2 \in \mathbb{G}_3$;
   (b) $e_3(R_1, I) = e_3(G, R_2)$; and
   (c) $e_3(S, I) = e_3(M, V) \cdot e_3(R_1, R_2) \cdot e_3(G, W)$.

Because of the dual nature of the two schemes, the security proof against generic forgers for the Type 3 scheme indicated in §4.2 carries over to the Type 3 scheme described here when we swap the roles of the elements in $\mathbb{G}_1$ and $\mathbb{G}_3$.

However, the above Type 3 scheme does not have a secure and natural counterpart in the Type 2 setting. The natural Type 2 variant has public key $V = H^v$, $W = H^w$, signatures on a message $M \in \mathbb{G}_1$ comprising of $R = H^r$ and $S = M^v G^{r^2+w}$, and verification that checks $M, S \in \mathbb{G}_1$, $R \in \mathbb{G}_2$ and $e_2(S, H) = e_2(M, V) \cdot e_2(\psi(R), R) \cdot e_2(G, W)$. Now, given the public key (V, W) an adversary can mount the following no-message attack. Select arbitrary $m, r \in [1, n-1]$ and compute a forged signature on $M = G^m$ as $R = H^r$ and $S =$

$\psi(V)^m\psi(W)G^{r^2} = M^vG^{r^2+w}$. While the absence of an efficiently-computable isomorphism from $\mathbb{G}_3$ to $\mathbb{G}_1$ allows us to construct the secure Type 3 scheme described above, the availability of ψ in the Type 2 setting provides the adversary with the means to mount the no-message attack.

5.4. **Type 2: A designer's artifact?** In prime-order asymmetric pairing groups, a protocol designer has the choice of using elements from $\mathbb{G}_1$, $\mathbb{G}_3$ and $\mathbb{H}_2 \subseteq \mathbb{G}_1 \times \mathbb{G}_3$. However, the definition of a bilinear group generator in the Type 2 setting recognizes only $\mathbb{G}_1$, $\mathbb{G}_2$ and the isomorphism $\psi : \mathbb{G}_2 \longrightarrow \mathbb{G}_1$; see, for example, the definition of a generic bilinear group generator $\mathcal{G}$ in §2.1 of [4]. The definition fails to take into account the existence of the group $\mathbb{G}_3$ and an efficiently-computable isomorphism $\rho : \mathbb{G}_2 \longrightarrow \mathbb{G}_3$. This incompleteness in the definition of bilinear group generators has a significant bearing on pairing-based cryptographic protocols. As demonstrated in this paper, all the efficiency analysis and the optimality claims for signature size and number of verification equations (see Table 1 of [4]) as well as the main lower bound result of [4][8] suffer from this incompleteness.

More generally, a designer desiring to use the map ψ in a cryptographic protocol or the corresponding security argument unnecessarily restricts herself to $\mathbb{G}_1$ and $\mathbb{G}_2$ (i.e. $\mathbb{H}_2$). This design artifact introduces (costly) redundancy in the cryptographic scheme without any benefit in terms of functionality or security. This observation was first made in [12] based on a careful analysis of existing Type 2 schemes. However, [12] did not attempt a formal proof of the assertion that Type 2 pairings are "merely less efficient implementation of Type 3 pairings". Motivated by the erroneous claim of superiority of Type 2 over Type 3 in [4], in this paper we formally settle the relation between Type 2 and Type 3 settings in the context of generic-signer structure-preserving signature.

6. Concluding remarks

We presented natural Type 3 analogues of the Type 2 strongly unforgeable and randomizable structure-preserving signature schemes that were proposed in [4]. 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. Finally, we show that all generic-signer Type 2 schemes are merely Type 3 schemes in disguise and cannot beat the existing lower bound results. On the other hand, not all Type 3 schemes have a secure Type 2 counterpart. We conclude that the question posed in [12] of the existence of a cryptographic protocol which necessarily has to be restricted to Type 2 for implementation or security reasons is still open.

Acknowledgements

We thank Jens Groth and Francisco Rodríguez-Henríquez for their comments on the paper.

[8] Theorem 4 of [4], as stated, is void because we have established the impossibility of structure-preserving signature schemes with a single verification equation.

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, "Unified, minimal and selectively randomizable structure-preserving signatures", *Theory of Cryptography – TCC 2014*, LNCS 8349 (2014), 688–712.

[4] 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.

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

[6] 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.

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

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

[9] 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.

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

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

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

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

[14] 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.

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

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

[17] J. Groth, "Simulation-sound NIZK proofs for a practical language and constant size group signatures", *Advances in Cryptology – ASIACRYPT 2006*, LNCS 4284 (2006), 444–459.

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

[19] C. Hanser and D. Slamanig, "Structure-preserving signatures on equivalence classes and their application to anonymous credentials", *Advances in Cryptology – ASIACRYPT 2014*, to appear; full version available at http://eprint.iacr.org/2014/705.

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

[21] 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.

[22] 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.

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

Appendix A. Groth-Sahai proofs

In this section, we use additive notation for elements of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$.

A.1. **DLIN-based proofs.** Let $A, B \in \mathbb{G}_1$ and $t \in \mathbb{G}_T$. We present a Groth-Sahai non-interactive witness-indistinguishable proof of knowledge of $X, Y \in \mathbb{G}_2$ such that $e_2(A, X) \cdot e_2(B, Y) = t$. The NIWI proof is derived from the general description in §4.2 of [16]. 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 reference string is (H, U, V, I, J, K).
(3) *Commitment.* Select $s_{11}, s_{12}, s_{13}, s_{21}, s_{22}, s_{23} \in_R [1, n-1]$ and compute $d_{11} = s_{11}U + s_{13}I$, $d_{12} = s_{12}V + s_{13}J$, $d_{13} = X + s_{11}H + s_{12}H + s_{13}K$, $d_{21} = s_{21}U + s_{23}I$, $d_{22} = s_{22}V + s_{23}J$ and $d_{23} = Y + s_{21}H + s_{22}H + s_{23}K$. The commitment is $d = (d_{11}, d_{12}, d_{13}, d_{21}, d_{22}, d_{23})$.
(4) *Proof.* Compute $\theta_1 = s_{11}A + s_{21}B$, $\theta_2 = s_{12}A + s_{22}B$ and $\theta_3 = s_{13}A + s_{23}B$. The proof is $\theta = (\theta_1, \theta_2, \theta_3)$.
(5) *Verification.* Check that $\theta_1, \theta_2, \theta_3 \in \mathbb{G}_1$, $d_{11}, d_{12}, d_{13}, d_{21}, d_{22}, d_{23} \in \mathbb{G}_2$, and

$$\begin{aligned} e_2(A, d_{11}) \cdot e_2(B, d_{21}) &= e_2(\theta_1, U) \cdot e_2(\theta_3, I) \\ e_2(A, d_{12}) \cdot e_2(B, d_{22}) &= e_2(\theta_2, V) \cdot e_2(\theta_3, J) \\ e_2(A, d_{13}) \cdot e_2(B, d_{23}) &= e_2(\theta_1, H) \cdot e_2(\theta_2, H) \cdot e_2(\theta_3, K) \cdot t. \end{aligned}$$

A.2. **DDH-based proofs.** Let $A, B \in \mathbb{G}_1$ and $t \in \mathbb{G}_T$. We present a Groth-Sahai non-interactive witness-indistinguishable proof of knowledge of $X, Y \in \mathbb{G}_3$ such that $e_3(A, X) \cdot e_3(B, Y) = t$. The NIWI proof is derived from the general description in §4.1 of [16]. 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 reference string is (I, U, V, J).
(3) *Commitment.* Select $s_{11}, s_{12}, s_{21}, s_{22} \in_R [1, n-1]$ and compute $d_{11} = s_{11}I + s_{12}V$, $d_{12} = X + s_{11}U + s_{12}J$, $d_{21} = s_{21}I + s_{22}V$ and $d_{22} = Y + s_{21}U + s_{22}J$. The commitment is $d = (d_{11}, d_{12}, d_{21}, d_{22})$.
(4) *Proof.* Compute $\theta_1 = s_{11}A + s_{21}B$ and $\theta_2 = s_{12}A + s_{22}B$. The proof is $\theta = (\theta_1, \theta_2)$.
(5) *Verification.* Check that $\theta_1, \theta_2 \in \mathbb{G}_1$, $d_{11}, d_{12}, d_{21}, d_{22} \in \mathbb{G}_3$, and

$$\begin{aligned} e_3(A, d_{11}) \cdot e_3(B, d_{21}) &= e_3(\theta_1, I) \cdot e_3(\theta_2, V) \\ e_3(A, d_{12}) \cdot e_3(B, d_{22}) &= e_3(\theta_1, U) \cdot e_3(\theta_2, J) \cdot t. \end{aligned}$$

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