Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures

Georg Fuchsbauer

École normale supérieure, CNRS - INRIA, Paris, France
http://www.di.ens.fr/~fuchsbau

Abstract

We introduce the notion of *automorphic signatures*, which satisfy the following properties: the verification keys lie in the message space, messages and signatures consist of elements of a bilinear group, and verification is done by evaluating a set of pairing-product equations. These signatures make a perfect counterpart to the powerful proof system by Groth and Sahai (Eurocrypt 2008). We provide practical instantiations of automorphic signatures under appropriate assumptions and use them to construct the first efficient round-optimal blind signatures. By combining them with Groth-Sahai proofs, we moreover give practical instantiations of various other cryptographic primitives, such as fully-secure group signatures, non-interactive anonymous credentials and anonymous proxy signatures. To do so, we show how to transform signature schemes whose message space is a group to a scheme that signs arbitrarily many messages at once.

1 Introduction

One of the main goals of modern cryptography is anonymity. A classical primitive ensuring user anonymity is group signatures [Cv91]: they allow members that were enrolled by a group manager to sign on behalf of a group while not revealing their identity. To prevent misuse, anonymity can be revoked by an authority. Another example is anonymous credentials [Cha85], by which a user can prove that she holds a certain credential, and at the same time remain anonymous. Blind signatures [Cha82] were introduced for electronic cash to prevent the linking of a coin to its spender, and are also used in electronic voting systems, where anonymity is indispensable.

Security of such primitives is addressed by defining a security model, which is typically first proved to be satisfiable in theory under general assumptions. Let us consider the example of *dynamic group signatures* by Bellare et al. [BSZ05]. To show feasibility of their strong model, they give the following generic construction: Assume the existence of a signature scheme, an encryption scheme and general zero-knowledge proofs. The group manager publishes a signature verification key and uses the corresponding signing key to issue certificates on the group members' personal verification keys. A member produces a group signature by first signing the message with her personal signing key, and then encrypting her certificate, her verification key, and the signature on the message. The group signature consists of these ciphertexts completed by a non-interactive zero-knowledge (NIZK) proof that the certificate and the signature in the plaintext are valid. The fact that a signature is a ciphertext and a NIZK proof that leaks no information guarantees user anonymity.

For a long time the only efficient ways to instantiate such primitives was to either rely on the random-oracle heuristic [BR93] for NIZK—or to directly use *interactive* assumptions (like the LRSW assumption [LRSW00] and its variants, or "one-more" assumptions [BNPS03]). Due to a series of criticisms starting with [CGH98], more and more practical schemes are being proposed and proved secure in the *standard model* (i.e., without random oracles) and under *falsifiable* (and thus non-interactive) assumptions [Nao03]. In particular, groups with a bilinear map (pairing) turned out to be an attractive tool to achieve efficiency. Many of the practical instantiations use ad hoc constructions, since the generic ones—in particular zero-knowledge proofs—are by far too inefficient.

The Groth-Sahai Proof System. In [GS08], Groth and Sahai propose *efficient* zero-knowledge proofs for a large class of statements over bilinear groups, which already found use in many implementations [CGS07, Gro07, GL07, BCKL08, CCS09, BCKL09, BCC$^+$09, FPV09]. They start by constructing witness-indistinguishable (WI) proofs of satisfiability of various types of equations: given a witness of satisfiability, one makes *commitments* to its values and then constructs proofs which assert that the committed values satisfy the equations. As already observed by [Gro06], the most interesting and widely used type is the following: pairing-product equations (PPE) whose variables are elements of the bilinear group (cf. Sect. 2.2). A PPE consists of products of pairings applied to the variables and constants from the group. Since the employed commitments to group elements are extractable, the resulting proofs actually constitute *proofs of knowledge* as well.

To efficiently implement the generic construction of group signatures from [BSZ05], Groth [Gro07] instantiates encryption and proofs of plaintext validity with the Groth-Sahai WI proof system. Extractability of the commitments serves two purposes: first, it lets the opener extract the user's verification key and thereby trace the signer (the commitments are thus used as encryptions that can be decrypted with the extraction key); second, it makes it possible to reduce unforgeability of group signatures directly to unforgeability of the underlying signatures. For the Groth-Sahai methodology to be applicable, Groth gives certification and signing schemes such that certificates, signature verification keys and signatures (i.e., the components that need to be hidden) are group elements whose validity is verified by evaluating PPEs.[1] (cf. Sect. 3.3).

Signatures and the Groth-Sahai Proof System. The first practical schemes to use Groth-Sahai-like proofs were the group signatures by Boyen and Waters [BW06, BW07], who independently developed their proofs using techniques from [GOS06]. They require weakly secure[2] signatures whose components and messages can be encrypted (committed to) and proved to be valid. To produce certificates lying the bilinear group, they modify the weak Boneh-Boyen signatures [BB04], which consist of one group element and whose messages are scalars: instead of giving the scalar directly, they give it as an exponentiation of two different group generators. The security of their construction holds under a variant of the *strong Diffie-Hellman assumption* (SDH) [BB04] called *hidden SDH* (HSDH).

Belenkiy et al. [BCKL08] apply the Boneh-Boyen [BB04] transformation "from weak to strong security" to the Boyen-Waters scheme. They thereby obtain fully secure signatures, at the price of introducing a "very strong assumption" (according to [BCC$^+$09]) they call *triple Diffie-Hellman*. Their signatures consist of group elements, yet the messages are scalars. To construct anonymous credentials, they make commitments to a message and a signature on it and prove that their content is valid using Groth-Sahai proofs. Since from the employed commitments only group elements can be extracted efficiently, they are obliged to define f*-extractability*, meaning that only a function of the committed value can be extracted. This entails stronger security notions ("F-unforgeability") for the signature scheme in order to prove security of their construction.

In the abovementioned group signatures from [Gro07] this drawback is avoided by designing the key-certification scheme so that all committed values are group elements. The key certification is thus different from the signature scheme whose keys are certified. Moreover, the certificate-verification key is an element of the *target* group. As opposed to standard group signatures, in hierarchical group signatures [TW05] or anonymous proxy signatures [FP08], or more generally, to instantiate certification *chains*, verification keys are not only certified once, but must also serve to certify other keys. The message space must thus contain the verification keys. If we want to apply the Groth-Sahai methodology to "anonymize" such schemes and prove unforgeability by reducing it to the security of the underlying signatures, everything has to be in the bilinear group.

We identify the all-purpose building block to efficiently instantiate privacy-related primitives as the following: a practical signature scheme secure against adaptive chosen-message attacks that can sign its own verification keys; and which at the same time respects the pairing-product paradigm, that is, keys, messages and signatures consist of group elements and the signature-verification relations are PPEs. We call such a scheme an *automorphic signature*, as it is able to sign its *own* keys and verification preserves the *structure* of keys and messages, which makes it perfectly suitable to be combined with Groth-Sahai proofs. We believe that working with group elements enables a modular approach of combining signatures with Groth-Sahai proofs, and automorphic signatures are the

[1] The certified signatures defined by Ateniese et al. [ACHM05] satisfy these properties as well (and they can be completely randomized). The certificates are (a variant of) CL signatures [CL04] on the user's secret key; certification is thus an interactive protocol. Moreover, their construction strongly relies on *interactive* (thus non-falsifiable) assumptions, such as the strong LRSW [ACM05] assumption.

[2] Throughout the paper we call a signature scheme *weakly secure* if an adversary getting signatures on *random* messages cannot produce a signature on a new message.

building block tailored to do so. As demonstrated in Sect. 3, they yield straightforward efficient implementations of generic constructions of a variety of primitives, by simply plugging in concrete schemes for generic ones.

We note that a scheme in [Gro06] based on the *decision linear assumption* [BBS04] can be considered automorphic, but should rather be regarded as a proof of concept due to its inefficiency (a signature consists of hundreds of thousands of group elements), whereas we give practical-level efficiency under reasonable assumptions.

Blind signatures. Blind signatures, introduced by Chaum [Cha82], allow a user to obtain a signature on a message such that the signer cannot relate the resulting message/signature pair to the execution of the signing protocol. They were formalized by [JLO97, PS00] and practical schemes without random oracles have been constructed in e.g. [CKW04, KZ06, Oka06, KZ08]. However, all these schemes require more than one round (i.e., two moves) of communication between the user and the signer to issue a blind signature. This is even the case for most instantiations in the random-oracle model, an exception being Chaum's scheme proved secure in [BNPS03] under an interactive assumption.

In [Fis06], Fischlin gives a generic construction of *round-optimal* blind signatures in the common-reference string (CRS) model: the signing protocol consists of one message from the user to the signer and one response by the signer. This immediately implies *concurrent* security, an explicit goal in other works such as [HKKL07]. Up to now, a practical instantiation of round-optimal blind signatures in the standard model has been an open problem.

Anonymous Proxy Signatures. Proxy signatures allow the delegation of signing rights; they were introduced by [MUO96] and later formalized by [BPW03, SMP08]. Anonymous proxy signatures [FP08] unify (multi-level) proxy signatures and group signatures by guaranteeing anonymity to the proxy signer and intermediate delegators.

They enable users ("*original delegators*") to delegate others to sign on their behalf; the latter can furthermore re-delegate the received rights to other users. Anonymity ensures that proxy signatures do not reveal who signed and who re-delegated; however, they guarantee that there exists a delegation chain from the original delegator to the proxy signer. As for group signatures, an algorithm to revoke anonymity is provided to deter from misuse. Due to consecutiveness of delegation, this primitive also models hierarchical group signatures satisfying a security model generalizing the one of [BSZ05]. The only concrete instantiation of anonymous proxy signatures was given in [FP09] using Groth-Sahai-like proofs; it is however fairly impractical and relies on a new type of assumption.

Our Contribution

We define automorphic signatures and start with giving illustrative applications. The first is an efficient instantiation of round-optimal (and thus concurrently secure) blind signatures in the common-reference-string model [Fis06], which solves an open problem. A concrete round-optimal scheme that is more efficient than the instantiation of the generic construction is given in Sect. 5.2.

In Sect. 3.2 and 3.3, we use automorphic signatures to build CCA-secure group signatures and revisit the construction of non-interactive anonymous credentials of [BCKL08]; in particular, we achieve actual message extractability and give an efficient credential-issuing protocol. We then present the first efficient instantiation of anonymous proxy signatures (APS) in the standard model. We use automorphic signatures to certify public keys, so delegation is done by simply signing the delegatee's public key. An anonymous proxy signature is a Groth-Sahai (GS) proof of knowledge of a certification chain that starts at the original delegator and ends at the message.

We then strengthen the model for APS by enhancing the anonymity guarantees (Sect. 3.5). We first revise delegation so that intermediate delegators remain anonymous to the delegatee whereas the generic construction in [FP09] only provides anonymity w.r.t. the verifier. Moreover, we give a protocol for *blind delegation*: a user can be delegated to without revealing her identity. These enhancements do not affect the signature size, which grows linearly in the number of delegations (which is optimal, since the signature must contain opening information.)

Recently, Belenkiy et al. [BCC$^+$09] introduced *delegatable anonymous credentials* (DAC). They also provide mechanisms enabling users to prove possession of certain rights while remaining anonymous; and they consider re-delegation of received rights. Similarly to the construction of APS, a delegatable credential consists of a chain of certificates that is encrypted and proved valid. The core protocol of DAC lets a user obtain a proof of knowledge of a signature on her *secret* key, without revealing the identity of neither the signer nor the user. This imposes interactivity of the delegation process, while (non-blind) delegations for APS are non-interactive, even when delegators remain anonymous. (We show how to achieve delegatee anonymity at the expense of non-interactivity). Besides, DAC only deal with authentication rather than signing, and do not provide tracing mechanisms.

We believe that APS is a conceptually simpler primitive than DAC and provides a similar functionality. Moreover, automorphic signatures combined with GS proofs yield efficient instantiations in a straightforward fashion, whereas this is not the case for DAC (see below).

Instantiations. We give two concrete instantiations of automorphic signatures and show them to be strongly unforgeable under chosen-message attack (Sect. 5.1). The first one relies on an assumption recently introduced by Fuchsbauer et al. [FPV09]: the *double hidden SDH* assumption (DHSDH) is a variant of SDH in the flavor of HSDH in symmetric bilinear groups ("Type-1" in the terminology of [GPS08]). As also pointed out by [GSW09], the most efficient instantiation of Groth-Sahai proofs is the one in *asymmetric* bilinear groups ("Type-3") based on SXDH (cf. Sect. 2.1). In order to construct automorphic signatures over these groups, we define a variant of DHSDH in asymmetric groups, called ***A***DHSDH and prove it secure in the generic group model [Sho97]. Lastly, we give a new type of flexible CDH assumption, which is weaker than all previous versions such as [LV08]. Together with ADHSDH it implies strong unforgeability of our second automorphic-signature instantiation in asymmetric bilinear groups. The scheme can be combined with the SXDH-instantiation of GS proofs and its signatures consist of only 5 group elements. We insist that all our assumptions are non-interactive and falsifiable [Nao03], and hold in the generic group model.

In Sect. 5.2, we use our schemes to give the first efficient instantiation of round-optimal blind signatures. The blind signature and the user message are of order 30 group elements (depending on the instantiation of the employed GS proofs) and the signer message consists of 5 elements. They can be based on either DHSDH or ADHSDH, the latter leading to a scheme that is automorphic itself, which makes it especially suitable for our applications. In the last section, we give a generic transformation of a signature scheme whose message space is an algebraic group and contains the verification keys to one that signs vectors of arbitrary length. Our transformation preserves the structure of verification; thus applied to an automorphic scheme the resulting scheme is automorphic.

Comparison of our APS instantiation with the DAC instantiation of Belenkiy et al. In [BCC$^+$09], the underlying signature (called *authenticator*) on a user private key is in $\mathbb{G}_1^5 \times \mathbb{G}_2^2$, thus of size similar to that of our automorphic signature. In the delegation protocol, the issuer first sends a GS proof of knowledge of the first 6 signature components. The issuer and the user then run a two-party protocol to jointly compute the last component, using a homomorphic encryption scheme and interactive ZK proofs that blinding values are in the correct ranges. The authors suggest using Paillier encryption [Pai99] based on an RSA modulus of size at least $2^{3k}p^2$.

Using the NIST recommendations from 2007 [NIS07] for $k = 128$ bits of security, the RSA modulus N must be at least 2^{3072}; Paillier ciphertexts are thus of size $N^2 \geq 2^{6144}$. Since the interactive proofs of knowledge of plaintexts and values lying in certain intervals are not given explicitly, it is not clear how many rounds of interaction the protocol requires and how many elements are sent in each of them. Our blind delegation protocol for APS is the issuing protocol for our blind signatures (see above), which is round-optimal; moreover, in the SXDH instantiation, the user message and the signer message together consist of 20 elements from $\mathbb{G}_1$ and 18 elements from $\mathbb{G}_2$. For 128-bit security, using e.g. the groups suggested by Barreto and Naehrig [BN05], elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ are represented by 256 and 512 bits, respectively. The total of communication bandwidth for a blind delegation in our scheme is thus roughly of the size of only 2 Paillier ciphertexts for comparable security parameters.

Concerning the assumptions on which the employed signature schemes are based, they are very similar and both fall in Boyen's [Boy08] generalized "Uber-Assumption" family and have the same generic security bound (see Appendix C.1).

2 Preliminaries

2.1 Primitives

We recall some standard concepts from the literature.

Commitments. A non-interactive commitment scheme $\mathbf{Com}$ is composed of an algorithm $\mathsf{Setup}_{\mathsf{Com}}$, outputting a *commitment key* ck, and an algorithm Com with arguments ck, a message M and randomness $\rho \in \mathcal{R}$. We require that (1) the scheme is *perfectly binding*, i.e., for a commitment $\mathbf{c}$ there exists only one M s.t.: $\mathbf{c} = \mathsf{Com}(ck, M, \rho)$ for some ρ; (2) the scheme is *computationally hiding*, in particular, there exists $\mathsf{SmSetup}_{\mathrm{Com}}$ outputting keys that are computationally indistinguishable from those output by $\mathsf{Setup}_{\mathrm{Com}}$, and which generate perfectly hiding commitments. A commitment scheme is *extractable* if there exists an algorithm $\mathsf{ExSetup}_{\mathrm{Com}}$ outputting (ck, ek), where *ck* is distributed as the output of $\mathsf{Setup}_{\mathrm{Com}}$, and an algorithm Extr that on input the *extraction key ek* and a commitment extracts the committed value from it. Note that a commitment scheme with all the above properties can be viewed as a *lossy encryption scheme* [BHY09].

Digital Signatures. A digital signature scheme **Sig** consists of the following algorithms: $\mathsf{Setup}_{\mathrm{Sig}}$ outputs public parameters *pp*. $\mathsf{KeyGen}_{\mathrm{Sig}}$ outputs a pair (vk, sk) of verification and signing key. $\mathsf{Sign}(sk, M)$ outputs a signature σ, which is verified by $\mathsf{Verify}_{\mathrm{Sig}}(vk, M, \sigma)$. Signatures are *existentially unforgeable under chosen-message attack* (EUF-CMA) [GMR88] if no adversary, given *vk* and a signing oracle for messages of its choice, can output a pair (M, σ) s.t. M was never queried and $\mathsf{Verify}(vk, M, \sigma) = 1$. Signatures are *strongly* UEF-CMA if no adversary can output a valid pair (M, σ) such that $(M, \sigma) \neq (M_i, \sigma_i)$ for all i, with M_i being the i-th oracle query and σ_i the response.

Blind Signatures. Blind signatures extend digital signatures by an interactive protocol $\mathsf{Issue} \leftrightarrow \mathsf{Obtain}$ between the signer and a user allowing the latter to obtain a signature on a message hidden from the signer. Security is defined by the following requirements [Oka06, HKKL07]: *Blindness:* An adversary impersonating the signer interacting with Obtain twice for messages of its choice cannot relate the resulting signatures to their issuings. *Unforgeability:* No adversary interacting $q - 1$ times with Issue can output q different messages and valid signatures on them.

Bilinear Groups. A *bilinear group* is a tuple $\mathcal{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of prime order p, G_1 and G_2 generate $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficient non-degenerate bilinear map, i.e., $\forall X \in \mathbb{G}_1\ \forall Y \in \mathbb{G}_2\ \forall a, b \in \mathbb{Z} : e(X^a, Y^b) = e(X, Y)^{ab}$, and $e(G_1, G_2)$ generates $\mathbb{G}_T$. We will denote group elements by capital letters and integers by lower-case letters. $\mathcal{BG}$ is called symmetric if $\mathbb{G}_1 = \mathbb{G}_2$ and $G_1 = G_2$.

The *Symmetric External Diffie-Hellman (**SXDH**) Assumption* [ACHM05] states that given (G_1^r, G_1^s, G_1^t) for random $r, s \in \mathbb{Z}_p$, it is hard to decide whether $t = rs$ or t is random; and analogously for G_2.

The *Decision Linear (**DLIN**) Assumption* [BBS04] in a symmetric group $(p, \mathbb{G}, \mathbb{G}_T, e, G)$ states that given $(G^\alpha, G^\beta, G^{r\alpha}, G^{s\beta}, G^t)$ for random $\alpha, \beta, r, s \in \mathbb{Z}_p$, it is hard to decide whether $t = r + s$ or t is random.

Throughout the paper, we will assume two fixed generators G and H of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively (with $G \neq H$ when $\mathbb{G}_1 = \mathbb{G}_2$). We call a pair $(A, B) \in \mathbb{G}_1 \times \mathbb{G}_2$ a *Diffie-Hellman pair* (w.r.t. (G, H)), if there exists $a \in \mathbb{Z}_p$ such that $A = G^a$ and $B = H^a$. Using the bilinear map e, such pairs are efficiently decidable by checking $e(A, H) = e(G, B)$. We let $\mathcal{DH}$ denote the set of DH pairs and implicitly assume them to be w.r.t. G and H.

2.2 Groth-Sahai Proofs for Pairing-Product Equations

We start with presenting perfectly binding extractable commitments, which are computationally hiding under either SXDH or DLIN, and then give an overview of Groth-Sahai proofs introduced in [GS08].

SXDH Commitments. Let $\mathcal{BG}$ be a bilinear group in which SXDH holds; we define $\mathbf{Com}_{\mathrm{X}}$. $\mathsf{Setup}_{\mathrm{X}}(\mathcal{BG})$ chooses $\alpha_1, \alpha_2, t_1, t_2 \leftarrow \mathbb{Z}_p$ and returns $ck = \big(\mathbf{u}_1 = (G_1, G_1^{\alpha_1}), \mathbf{v}_1 = (G_1^{t_1}, G_1^{\alpha_1 t_1}), \mathbf{u}_2 = (G_2, G_2^{\alpha_2}), \mathbf{v}_2 = (G_2^{t_2}, G_2^{\alpha_2 t_2})\big)$. $\mathsf{ExSetup}_{\mathrm{X}}$ additionally outputs the extraction key $ek := (\alpha_1, \alpha_2)$. Let k be 1 or 2. A commitment to a group element $X \in \mathbb{G}_k$ using randomness $\rho = (\rho_1, \rho_2) \in \mathcal{R}_{\mathrm{X}} := \mathbb{Z}_p^2$ is defined as $\mathsf{Com}_{\mathrm{X}}(ck, X, \rho) := (\mathbf{u}_{k,1}^{\rho_1} \cdot \mathbf{v}_{k,1}^{\rho_2},\ X \cdot \mathbf{u}_{k,2}^{\rho_1} \cdot \mathbf{v}_{k,2}^{\rho_2})$. Extraction from (c_1, c_2) in $\mathbb{G}_k$ done by computing $\mathsf{Extr}_{\mathrm{X}}\big((\alpha_1, \alpha_2), (c_1, c_2)\big) := c_2 \cdot c_1^{-\alpha_k}$. $\mathsf{SmSetup}_{\mathrm{X}}$ replaces $\mathbf{v}_{k,2}$ in *ck* by $G_k^{\alpha_k t_k - 1}$ for $k = 1, 2$ (which is indistinguishable by SXDH), resulting in perfectly hiding commitments.

Linear Commitments. For $\mathbf{Com}_{\mathrm{L}}$, let $\mathcal{BG}$ be a symmetric bilinear group in which DLIN holds. $\mathsf{Setup}_{\mathrm{L}}(\mathcal{BG})$ chooses $\alpha, \beta, r_1, r_2 \leftarrow \mathbb{Z}_p$ and outputs $ck = \big(\mathbf{u}_1 = (G^\alpha, 1, G), \mathbf{u}_2 = (1, G^\beta, G), \mathbf{u}_3 = (G^{r_1\alpha}, G^{r_2\beta}, G^{r_1+r_2})\big)$. $\mathsf{ExSetup}_{\mathrm{L}}$ also outputs the extraction key $ek := (\alpha, \beta)$. A commitment to $X \in \mathbb{G}$ with randomness $\rho \in \mathcal{R}_{\mathrm{L}} := \mathbb{Z}_p^3$ is defined as $\mathsf{Com}_{\mathrm{L}}\big(ck, X, \rho\big) := \big(\prod \mathbf{u}_{i,1}^{\rho_i},\ \prod \mathbf{u}_{i,2}^{\rho_i},\ X \cdot \prod \mathbf{u}_{i,3}^{\rho_i}\big)$. $\mathsf{Extr}_{\mathrm{L}}\big((\alpha, \beta), (c_1, c_2, c_3)\big)$ outputs $c_3 \cdot c_1^{-1/\alpha} \cdot c_2^{-1/\beta}$. $\mathsf{SmSetup}_{\mathrm{L}}$ replaces $\mathbf{u}_{3,3}$ in *ck* with $G^{r_1+r_2-1}$, which is indistinguishable by DLIN.

Groth-Sahai WI Proofs. We use Groth-Sahai witness-indistinguishable (WI) proofs of *satisfiability of pairing-product equations*. A *pairing-product equation* (PPE) over variables $\mathcal{X}_1, \ldots, \mathcal{X}_m \in \mathbb{G}_1$, $\mathcal{Y}_1, \ldots, \mathcal{Y}_n \in \mathbb{G}_2$ is an equation of the form

$$\prod_{i=1}^{n} e(A_i, \mathcal{Y}_i) \prod_{i=1}^{m} e(\mathcal{X}_i, B_i) \prod_{i=1}^{m}\prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = \mathbf{t}_T \ , \tag{E}$$

determined by $A_j \in \mathbb{G}_1$, $B_i \in \mathbb{G}_2$, $\gamma_{i,j} \in \mathbb{Z}_p$, for $1 \le i \le m$ and $1 \le j \le n$, and $\mathbf{t}_T \in \mathbb{G}_T$.

Depending on the instantiation, the proof system makes use of one of the above commitment schemes; let $\mathbf{Com} = (\mathsf{Setup}, \mathsf{Com}, \mathsf{ExSetup}, \mathsf{Extr}, \mathsf{SmSetup})$ denote thus either $\mathbf{Com}_{\mathrm{X}}$ or $\mathbf{Com}_{\mathrm{L}}$. The proof system for a bilinear group $\mathcal{BG}$ is set up by running $\mathsf{Setup}(\mathcal{BG})$ which produces a perfectly binding commitment key *ck*. Given an assignment $\mathcal{X}_i \leftarrow X_i$ and $\mathcal{Y}_j \leftarrow Y_j$, for $X_i \in \mathbb{G}_1$ and $Y_j \in \mathbb{G}_2$, satisfying E, one first *commits* to the values X_i, Y_j by choosing randomness $\rho_i, \tau_j \leftarrow \mathcal{R}$ and setting $\mathbf{c}_{X_i} := \mathsf{Com}(ck, X_i, \rho_i)$ and $\mathbf{c}_{Y_j} := \mathsf{Com}(ck, Y_j, \tau_j)$ for all i, j. Running $\mathsf{Prove}_{\mathrm{GS}}(ck, E, (X_i, \rho_i)_{i=1}^m, (Y_j, \tau_j)_{j=1}^n)$ generates a proof[3] ϕ which asserts that the values committed in $\mathbf{c}_{X_i}$ and $\mathbf{c}_{Y_j}$ satisfy E. A proof ϕ for equation E and commitments $(\mathbf{c}_{X_i})_{i=1}^m$ and $(\mathbf{c}_{Y_j})_{j=1}^n$ under *ck* is verified by $\mathsf{Verify}_{\mathrm{GS}}(ck, E, (\mathbf{c}_{X_i})_{i=1}^m, (\mathbf{c}_{Y_j})_{j=1}^n, \phi)$. An honestly computed proof for commitments to values satisfying E is always accepted by $\mathsf{Verify}_{\mathrm{GS}}$.

Security. *Soundness.* Given commitments $(\mathbf{c}_{X_i})_{i=1}^m, (\mathbf{c}_{Y_j})_{j=1}^n$ s.t. $\mathsf{Verify}_{\mathrm{GS}}(ck, E, (\mathbf{c}_{X_i})_{i=1}^m, (\mathbf{c}_{Y_j})_{j=1}^n, \phi) = 1$ for some ϕ and the extraction key *ek* output by $\mathsf{ExSetup}$, algorithm Extr applied to $\mathbf{c}_{X_i}$ and $\mathbf{c}_{Y_j}$ for all i, j yields vectors $(X_i)_{i=1}^m, (Y_j)_{j=1}^n$ satisfying E.

Witness Indistinguishability (WI). If the commitment key is replaced by ck^* output by $\mathsf{SmSetup}$ (which is indistinguishable) then a commitment $\mathbf{c} := \mathsf{Com}(ck^*, X, \rho)$ is perfectly hiding; i.e., given $\mathbf{c}$, then for any X there exists $\rho \in \mathcal{R}$ s.t. $\mathbf{c} = \mathsf{Com}(ck^*, X, \rho)$. Moreover, given values $((X_1, \rho_1), \ldots, (X_m, \rho_m), (Y_1, \tau_1), \ldots, (Y_n, \tau_n))$ and $((X'_1, \rho'_1), \ldots, (X'_m, \rho'_m), (Y'_1, \tau'_1), \ldots, (Y'_n, \tau'_n))$ such that for all i, j: $\mathsf{Com}(ck^*, X_i, \rho_i) = \mathsf{Com}(ck^*, X'_i, \rho'_i)$ and $\mathsf{Com}(ck^*, Y_j, \tau_j) = \mathsf{Com}(ck^*, Y'_j, \tau'_j)$, and $(X_1, \ldots, X_m, Y_1, \ldots, Y_n)$ and $(X'_1, \ldots, X'_m, Y'_1, \ldots, Y'_n)$ both satisfy E, then $\mathsf{Prove}_{\mathrm{GS}}(ck^*, E, (X_i, \rho_i)_{i=1}^m, (Y_j, \tau_j)_{j=1}^n)$ and $\mathsf{Prove}_{\mathrm{GS}}(ck^*, E, (X'_i, \rho'_i)_{i=1}^n, (Y'_j, \tau'_j)_{j=1}^n)$ generate the same distribution of proofs.

Examples. (1) *Proof of Two Commitments Containing the Same Value.* Let $E_{\mathrm{equal}}(X_1, X_2)$ denote the equation $e(X_1, H)\, e(X_2, H^{-1}) = 1$. Given two commitments $\mathbf{c}_M = \mathsf{Com}(ck, M, \rho)$ and $\mathbf{c}_N = \mathsf{Com}(ck, N, \sigma)$, $\mathsf{Prove}(ck, E_{\mathrm{equal}}, (M, \rho), (N, \sigma))$ proves that $\mathbf{c}_M$ and $\mathbf{c}_N$ commit to the same value.

(2) *Proof of Commitments to a $\mathcal{DH}$-Pair.* Define $E_{\mathcal{DH}}(X, Y)$ as $e(X, H)\, e(G^{-1}, Y) = 1$. A proof for Equation $E_{\mathcal{DH}}$ proves that a pair of committed values is in $\mathcal{DH}$. Under $\mathbf{Com}_{\mathrm{L}}$, the proof is in $\mathbb{G}^3$.

Randomizing Groth-Sahai Proofs. As observed by [FP09] and [BCC+09] and formalized by the latter, Groth-Sahai WI proofs of knowledge can be *randomized*. This means that there exists an algorithm $\mathsf{RdCom}_{\mathrm{GS}}$ that on input *ck*, a commitment $\mathbf{c}$ and fresh randomness ρ' outputs a *randomization* of $\mathbf{c}$ under ρ'.

Moreover, a proof ϕ for an equation E and vectors of commitments $(\mathbf{c}_{X_i})_{i=1}^m$ and $(\mathbf{c}_{Y_j})_{j=1}^n$ can be *adapted* (and randomized itself) to the randomizations $\mathbf{c}'_{X_i} = \mathsf{RdCom}_{\mathrm{GS}}(ck, \mathbf{c}_{X_i}, \rho'_i)$ and $\mathbf{c}'_{Y_j} = \mathsf{RdCom}_{\mathrm{GS}}(ck, \mathbf{c}_{Y_j}, \tau'_j)$: running $\mathsf{RdProof}_{\mathrm{GS}}(ck, E, (\mathbf{c}_{X_i}, \rho'_i)_{i=1}^m, (\mathbf{c}_{Y_j}, \tau'_j)_{j=1}^n, \phi)$ computes ϕ' such that $((\mathbf{c}'_{X_i})_{i=1}^m, (\mathbf{c}'_{Y_j})_{j=1}^n, \phi')$ is distributed as

$$\Big(\big(\mathsf{Com}_{\mathrm{GS}}(ck, X_i, \hat\rho_i)\big)_{i=1}^m,\ \big(\mathsf{Com}_{\mathrm{GS}}(ck, Y_j, \hat\tau_j)\big)_{j=1}^n,\ \mathsf{Prove}_{\mathrm{GS}}\big(ck, E, (X_i, \hat\rho_i)_{i=1}^m, (Y_j, \hat\tau_j)_{j=1}^n\big)\Big)$$

for $\hat\rho_i$ and $\hat\tau_j$ uniformly distributed in $\mathcal{R}$ (and therefore $\mathsf{Verify}_{\mathrm{GS}}(ck, E, (\mathbf{c}'_{X_i})_{i=1}^m, (\mathbf{c}'_{Y_j})_{j=1}^n, \phi') = 1$). Basically, if (for all i, j) ρ_i, τ_j are the randomness of the original commitments then $\mathbf{c}'_{X_i} = \mathsf{Com}_{\mathrm{GS}}(ck, X_i, \rho_i + \rho'_i)$ and $\mathbf{c}'_{Y_j} = \mathsf{Com}_{\mathrm{GS}}(ck, Y_j, \tau_j + \tau'_j)$, and ϕ' is distributed as proofs output by $\mathsf{Prove}_{\mathrm{GS}}(ck, E, (X_i, \rho_i + \rho'_i)_{i=1}^m, (Y_j, \tau_j + \tau'_j)_{j=1}^n)$ (see [FPV09] for the DLIN instantiation).

[3] In the SXDH instantiation, a proof for a PPE is in $\mathbb{G}_1^4 \times \mathbb{G}_2^4$. In the DLIN instantiation, the proof is in $\mathbb{G}^9$; however, if E is a *linear equation* (i.e., $\gamma_{i,j} = 0$ for all i, j), then the proof reduces to 3 group elements. Note that in this context the word *proof* can either denominate "proof of satisfiability" (or language-membership)—which thus includes the commitments—or mean a proof *that the content of some given commitments satisfies a given equation*. We adopt the latter diction, and say *proof of knowledge* when we include the commitments.

3 Automorphic Signatures and Their Applications

Definition 1. *An* automorphic signature *over a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ is an EUF-CMA secure signature whose verification keys are contained in the message space. Moreover, the messages and signatures consist of elements of $\mathbb{G}_1$ and $\mathbb{G}_2$, and the verification predicate is a conjunction of pairing-product equations over the verification key, the message and the signature.*

Before giving concrete instantiations in Sect. 5, we highlight the multitude of applications of automorphic signatures. As going into details would be beyond the scope of this paper, we merely sketch the application areas.

3.1 Round-Optimal Blind Signatures

In [Fis06], Fischlin gives a generic construction for concurrently executable blind-signature schemes with optimal round complexity in the common reference string (CRS) model. The construction relies on commitment, encryption and signature schemes and generic NIZK proofs for NP-languages. In the signature-issuing protocol, the user first sends a commitment to the message to the signer (issuer), who responds with a signature on the commitment. The user then constructs the blind signature as follows: she encrypts the commitment and the signature and adds a NIZK proof that the signature is valid on the commitment and that the committed value is the message.

Following [HKKL07], Abe and Ohkubo [AO09] replace the NIZK proof in Fischlin's construction by a witness-indistinguishable proof and concretely suggest Groth-Sahai (GS) proofs. (Note that GS commitments on group elements can be "decrypted" using the extraction key.) To be compatible, the signature scheme must have messages and signatures consisting of group elements and verification must amount to evaluating pairing-product equations. However, they only mention the highly inefficient scheme from [Gro06] as a feasibility result and leave open the problem of an efficient construction. Automorphic signatures satisfy all the compatibility requirements and enable thus an efficient instantiation of round-optimal blind signatures; it suffices to construct a commitment scheme such that commitments lie in the message space of the signature (or are vectors of messages) and correct opening is verifiable by PPEs.

We directly construct a scheme in Sect. 5.2 which has smaller blind signatures than an instantiation of the generic construction: in the end of our issuing protocol, the user holds a signature on the *message* rather than on a commitment to it. To make this possible, the user sends a *randomization* of the message to the issuer in addition to the commitment. From this, the issuer makes a "pre-signature" and sends it to the user, who turns it into an actual signature on the message by adapting the randomness. The blind signature is then a GS proof of knowledge of a signature on the message (rather than a commitment), which avoids a proof that the commitment opens to the message. The size of our signature is around 30 group elements (depending on the GS instantiation) and the two messages sent during issuing are even smaller.

3.2 P-Signatures and Anonymous Credentials

In order to realize *non-interactive anonymous credentials*, Belenkiy et al. [BCKL08] introduce a new primitive called *P-signature*. It extends a signature and a commitment scheme by the following functionalities: a protocol $\mathsf{Issue} \leftrightarrow \mathsf{Obtain}$ between a signer and a user allows the latter to obtain a signature on a value the signer only knows a commitment to; and the holder of a message and a signature on it can produce a commitment to the message and a proof of knowledge of the signature. The commitments and proofs are instantiated with the Groth-Sahai methodology; the compatible signature scheme is the one discussed in Sect. 1. Using an automorphic signature instead has the following advantages: the signatures and messages being group elements, they can be extracted in the security reduction, which avoids notions like F-unforgeability. Moreover, a small modification of the signature-issuing protocol of our blind signatures (cf. Remark 6) yields an efficient $\mathsf{Issue} \leftrightarrow \mathsf{Obtain}$ protocol (whereas the one in [BCKL08] resorts to generic secure multiparty computation).

3.3 Fully-Secure Group Signatures

In order to implement the model for group signatures by [BSZ05], Groth [Gro07] uses the following ingredients to achieve CCA-anonymity: the tag-based encryption scheme [MRY04] $\mathbf{Enc}_{\text{tb}}$ by Kiltz [Kil06] and a strong one-time signature scheme[4] $\mathbf{Sig}_{\mathrm{ot}}$. A tag-based encryption scheme is a public-key encryption scheme whose encryption and decryption algorithms take as additional argument a *tag*. Kiltz' scheme is *selective-tag weakly CCA-secure*, i.e., an adversary outputting a tag t^* (before receiving the public key) and two messages and getting an encryption of one of them under t^* cannot decide which one was encrypted—even when provided with an oracle decrypting any ciphertext with tag $t \neq t^*$

In Groth's scheme a user produces a signature key pair (vk, sk) and is enrolled by the issuer who gives her a certificate *cert* on *vk*. Now to make a group signature on a message M, the user holding $(cert, vk, sk)$ generates a key pair $(vk_{\mathrm{ot}}, sk_{\mathrm{ot}})$ for $\mathbf{Sig}_{\mathrm{ot}}$, makes a signature *sig* on vk_{ot} under *vk* and produces a Groth-Sahai WI proof of knowledge π of $(cert, vk, sig)$ s.t. *cert* is a valid certificate on *vk* and *sig* is a signature on vk_{ot} valid under *vk*. She produces an $\mathbf{Enc}_{\mathrm{tb}}$-ciphertext C encrypting *sig* under tag vk_{ot} and adds a Groth-Sahai NIZK proof ζ that the encrypted value *sig* is the same as in the commitment contained in π. Using sk_{ot}, she finally makes a signature sig_{ot} on $(M, vk_{\mathrm{ot}}, \pi, C, \zeta)$ and outputs the group signature $\sigma = (vk_{\mathrm{ot}}, \pi, C, \zeta, sig_{\mathrm{ot}})$. To verify σ, check whether sig_{ot}, the proofs π and ζ, and the ciphertext C are valid. The opener holds a key enabling her to extract $(cert, vk, sig)$ from π. The key *vk* allows to determine the signer and *sig* acts as a non-frameable proof of correct tracing.

Using automorphic signatures to instantiate the schemes for *cert* and *sig* immediately yields a group signature scheme secure in the BSZ-model. More concretely, [FPV09] suggest to substitute the certified-signature scheme used by Groth, which is based on the "q-U Assumption", by one based on the more natural DHSDH (cf. Sect. 4). Their replacement however uses Waters signatures [Wat05] which entail a dramatic increase of the public-key size. This can be avoided by using instead the certified-signature scheme given in Remark 3 (based on DHSDH as well).

3.4 Anonymous Delegation of Signing Rights

Anonymous Proxy Signatures. Anonymous proxy signatures (APS) generalize group signatures in that everyone can become a group manager by delegating his signing rights to other users who can then anonymously sign in his name; moreover, received rights can be *re-delegated*. We give a brief overview of the model defined in [FP08].

Algorithm Setup establishes the public parameters. *Users* generate key pairs using KeyGen and run a protocol Reg with the *issuer* and their *opener* when joining the system. (This is essential to achieve traceability; see below.) To delegate to Bob, Alice runs Delgt on Bob's public key, which produces a *warrant* she gives to Bob. With this warrant, Bob can either sign or *re-delegate* to Carol, in which case Carol can again re-delegate or produce an *proxy signature* with PSign on behalf of Alice, which is verifiable by Verify on Alice's verification key.

Anonymity ensures that from a proxy signature one cannot tell who actually signed (or re-delegated), thus Bob and Carol remain anonymous. To prevent misuse, Alice's opener can revoke the anonymity of the intermediate delegators and the proxy signer. *Traceability* asserts that every valid signature can be opened to registered users and *non-frameability* guarantees that no adversary, even when colluding with the issuer, openers and other users, can produce a signature that opens to an honest user for a delegation or a signing she did not perform.

A Generic Construction. The generic construction by [FP08] proving feasibility of the model is as follows. Assume an EUF-CMA-secure signature scheme. The issuer and the users choose a signing/verification key pair each. When enrolling, a user U_i obtains a signature $cert_i$ on her verification key vk_i from the issuer. A warrant $warr_{1\to 2}$ from user U_1 to user U_2 is a signature on (vk_1, vk_2) valid under vk_1. U_2 re-delegates to U_3 by sending $warr_{1\to 2}$ and $warr_{2\to 3}$, a signature on (vk_1, vk_2, vk_3) under vk_2. Additionally, in each delegation step, the delegators' certificates are also passed on. Given a warrant $(warr_{1\to 2}, warr_{2\to 3})$, U_3 proxy-signs a message M on behalf of U_1 as follows: first produce a signature *sig* on (vk_1, vk_2, vk_3, M) using sk_3; then define the *plain* proxy signature as $(warr_{1\to 2}, vk_2, cert_2, warr_{2\to 3}, vk_3, cert_3, sig)$. In general we say that a plain proxy signature $\Sigma = (warr_{1\to 2}, \ldots, vk_k, cert_k, sig)$ on message M under vk_1 is valid if:

- $\forall i$: $cert_i$ is a signature on vk_i valid under the issuer's verification key;
- $\forall i$: $warr_{i\to i+1}$ is a signature on $(vk_1, \ldots, vk_{i+1})$ valid under vk_i; $\qquad$ ($\mathrm{Ver}_{\mathrm{PPS}}$)
- *sig* is a signature on $(vk_1, \ldots, vk_k, M)$ valid under vk_k.

[4] A signature scheme is *strongly one-time* if an adversary making a single chosen-message query *before receiving the public key* can neither output a new signed message nor a new signature on the queried message. Groth uses weak Boneh-Boyen signatures [BB04].

Now to transform this into an *anonymous* proxy signature, the signer encrypts Σ under the public key of U_1's opener (contained in vk_1) and adds a NIZK proof that the plaintext satisfies the relations in (Ver_{PPS}). Due to her decryption key, the opener can retrieve the plain signature and thus trace the delegators and the signer. The warrants and *sig* are non-frameable proofs of correct tracing.

Concrete Instantiations. Restricting the model to CPA-anonymity, the building blocks can be instantiated as follows: define encryption to be Groth-Sahai commitments (which can be "decrypted" due to extractability) and use Groth-Sahai proofs to show that the verification relations are satisfied by the committed values. For this to work however, the plain proxy signatures must fit the Groth-Sahai framework; meaning that the EUF-CMA signature scheme's verification keys, messages and signatures must be group elements satisfying pairing-product equations; in short, they must be automorphic signatures. We note that Fuchsbauer and Pointcheval [FP09] gave a CPA-anonymous instantiation of APS which is however fairly inefficient due to the used signature scheme (its public keys contain several commitments to each *bit* of the corresponding secret key). Moreover, they only consider one general opener and there is a maximum number of consecutive re-delegations. These limitations are easily overcome by using automorphic signatures.

In Appendix A.1, we show how to make the above scheme CCA-anonymous and thus fully satisfy the security model defined by [FP08]. In Appendix A.2 we discuss how to sign one message on behalf of several delegators. In all our constructions, public *attributes* can be easily included as messages for the signatures in delegation. The delegators can thus specify for which tasks they delegate signing rights.

3.5 Anonymous Proxy Signatures with Enhanced Anonymity Guarantees

We briefly sketch how to instantiate the extended model of APS discussed at the end of Sect. 1. A formal description can be found in Appendix E.

Blind Delegation. Using our blind automorphic signatures from Sect. 5.2, we can define *blind delegation*: instead of directly signing the delegatee's public key, the delegator runs a blind issuing protocol with the delegatee. In the end, the latter holds an actual warrant (cf. Sect. 3.1) and continues as in the scheme above.

Delegator Anonymity. Due to the modularity of Groth-Sahai proofs (for each equation its proof only depends on the commitments to the variables appearing in it), the "anonymization" of a signature need not be delayed until the proxy signing: warrants can be anonymized by the delegators already and randomized in each delegation step (which prevents linkability of signatures). However, we need to revise the way warrants are defined, since the present scheme requires knowledge of the identities of all previous delegators to construct them. We follow the general approach by [BCC$^+$09], who associate an identifier *id* to each original delegation. A warrant from the user at level i in the delegation chain to the next one is then a signature on $(\mathsf{Hash}(id \,\|\, i), vk_{i+1})$ under vk_i, where $\mathsf{Hash}\colon \{0,1\}^* \to \mathbb{G}$ is a collision-resistant hash function.[5] The hash value prevents combining different warrants and reordering the same warrant.

Consider the following situation (we simplify our exposition by assuming the certificate from the issuer is contained in the user public key, and by omitting the hash values): Oliver (the **o**riginal delegator), owning vk_O, delegated to Alice by giving her a signature $warr_{O\to A}$ on her key vk_A. Alice re-delegates to Bob sending him $(warr_{O\to A}, vk_A, warr_{A\to B})$. Bob can now delegate to Carol *without revealing Alice's identity*: He makes commitments $\mathbf{c}_{O\to A}$, $\mathbf{c}_A$ and $\mathbf{c}_{A\to B}$ to $warr_{O\to A}$, vk_A and $warr_{A\to B}$, respectively. He makes a *trivial* commitment $\mathbf{c}_B = \mathsf{Com}_{\mathrm{GS}}(ck, vk_B, 0)$ to his own key, and the following proofs: $\phi_{O\to A}$ for $\mathbf{c}_{O\to A}$ containing a valid warrant from vk_O to the content of $\mathbf{c}_A$, and $\phi_{A\to B}$ for $\mathbf{c}_{A\to B}$ containing a valid warrant from the content of $\mathbf{c}_A$ to the content of $\mathbf{c}_B$. He sends $\widetilde{warr} := (vk_O, \mathbf{c}_A, \mathbf{c}_{O\to A}, \phi_{O\to A}, \mathbf{c}_B, \mathbf{c}_{A\to B}, \phi_{A\to B})$ and a warrant $warr_{B\to C}$ to Carol.

Now, Carol produces a signature on behalf of Oliver on M as follows (re-delegation works analogously): make a signature *sig* on M valid under vk_C; *randomize* the commitments and adapt the proofs in $\widetilde{warr}$, in particular, set $\mathbf{c}'_B := \mathsf{RdCom}_{\mathrm{GS}}(ck, \mathbf{c}_B, \rho_B)$; make commitments to $warr_{B\to C}$, vk_C and *sig*, and proofs of validity of $warr_{B\to C}$ and *sig*. Note that for the first proof the randomness of the related commitments—in particular $\mathbf{c}'_B$—is required. Since $\mathbf{c}_B$ was a trivial commitment, the randomness of $\mathbf{c}'_B$ is ρ_B which was chosen by Carol (cf. end of Sect. 2.2).

[5] Since *id* and i are publicly known, $\mathsf{Hash}(id \,\|\, i) \in \mathbb{G}$ will be considered a constant in the Groth-Sahai proofs.

Remark 1. (1) Note that delegator-anonymous delegation is compatible with blind delegation: instead of simply sending $warr_{B\to C}$, Bob runs the interactive blind-issuing protocol with Carol, upon which she obtains $warr_{B\to C}$ and continues as above.

(2) Bob could also hide *his own identity* to Carol as follows: he sends (hiding) commitments to his own key and to $warr_{B\to C}$, and in addition a trivial commitment to Carol's key and proof of validity of $warr_{B\to C}$. Carol randomizes what Bob sent her, commits to a signature on the message and proves validity. In Appendix E, we formally describe an instantiation of anonymous proxy signatures with delegator anonymity.

4 Assumptions

We first restate the assumption from [FPV09], present a variant for asymmetric groups and introduce another mild assumption.

Assumption 1 (q-DHSDH)**.** *In a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, G)$, given $(H, K, X = G^x) \in \mathbb{G}^3$ and $q-1$ tuples*

$$\big(A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}},\ C_i = G^{c_i},\ D_i = H^{c_i},\ V_i = G^{v_i},\ W_i = H^{v_i}\big)_{i=1}^{q-1}\ , \quad \text{for } c_i, v_i \leftarrow \mathbb{Z}_p,$$

it is hard to output a new tuple $(A^*, C^*, D^*, V^*, W^*) \in \mathbb{G}^5$ of this form, i.e., a tuple that satisfies

$$e(A^*, X \cdot C^*) = e(K \cdot V^*, G) \qquad e(C^*, H) = e(G, D^*) \qquad e(V^*, H) = e(G, W^*) \tag{1}$$

Argument. As pointed out by its inventors, under the *Knowledge-of-Exponent Assumption* (KEA) [Dam92, BP04], hardness of q-DHSDH follows from hardness of the following problem:

q**-SDH-III:** Given $\big(G, K, X = G^x, (A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}}, c_i, v_i)_{i=1}^{q-1}\big)$, produce a new tuple (A^*, c^*, v^*) satisfying $e(A^*, X \cdot G^{c^*}) = e(K \cdot G^{v^*}, G)$.

(KEA asserts that given (G, H), then from an adversary returning (G^{c^*}, H^{c^*}) and (G^{v^*}, H^{v^*}) one can extract c^* and v^*.) They then prove that hardness of q-SDH-III is implied by hardness of q-SDH [BB04], a well-established assumption by now. DHSDH is thus similar to HSDH [BW07] which is also KEA-equivalent to a problem (forging a weak Boneh-Boyen signature) whose hardness is implied by SDH.[6] □

We introduce a variant of DHSDH for asymmetric bilinear groups (*A*DHSDH) to allow for more flexibility and in addition a more efficient instantiation of automorphic signatures. The element H will now be in $\mathbb{G}_2$ and the other generators in $\mathbb{G}_1$; we add an additional generator $F \in \mathbb{G}_1$ and give the elements C_i as $C_i = F^{c_i}$. This makes it possible to include $Y = H^x$ needed for verification (if we also gave G^{c_i}, we arrive at an easy problem; see Appendix C.1). Due to asymmetry, the first verification equation for a tuple changes.

Assumption 2 (q-ADHSDH)**.** *Given $(G, F, K, X = G^x;\ H, Y = H^x) \in \mathbb{G}_1^4 \times \mathbb{G}_2^2$ and $q-1$ tuples*

$$\big(A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}},\ C_i = F^{c_i},\ V_i = G^{v_i},\ D_i = H^{c_i},\ W_i = H^{v_i}\big)_{i=1}^{q-1}\ , \quad \text{for } c_i, v_i \leftarrow \mathbb{Z}_p,$$

it is hard to output a new tuple $(A^*, C^*, V^*, D^*, W^*) \in \mathbb{G}_1^3 \times \mathbb{G}_2^2$ of this form, i.e., a tuple that satisfies

$$e(A^*, Y \cdot D^*) = e(K \cdot V^*, H) \qquad e(C^*, H) = e(F, D^*) \qquad e(V^*, H) = e(G, W^*) \tag{2}$$

Argument. Due to the fact that we give $Y = H^x$, the KEA-reduction to SDH does not apply here. Instead, we directly prove that the assumption holds in the generic group model [Sho97] in Appendix C.2. □

[6] The q-HSDH assumption states that given G, H, G^x and $q-1$ triples $(G^{\frac{1}{x+c_i}}, G^{c_i}, H^{c_i})$ for random $c_i \in \mathbb{Z}_p$, it is hard to produce a new triple $(G^{\frac{1}{x+c^*}}, G^{c^*}, H^{c^*})$ with $c^* \neq c_i$ for all i. HSDH is incomparable to SDH: an HSDH instance can be computed from an SDH instance and an HSDH solution can be computed from an SDH solution, but not the other way round. Note that BB-HSDH [BCC+09], which states that given G^x, H^x and tuples $(G^{\frac{1}{x+c_i}}, c_i)$ it is hard to compute $(G^{\frac{1}{x+c^*}}, (G')^{c^*}, H^{c^*})$ for a new c^*, is easily shown to be stronger than SDH (cf. Appendix C.1).

Remark 2. Assumption 2 is also valid in generic *symmetric* bilinear groups; in particular, in Appendix C.2 we prove generic security of ADHSDH in the symmetric setting (thus, a fortiori it holds when $\mathbb{G}_1 \neq \mathbb{G}_2$).

The next assumption is a weaker variant of the *1-flexible CDH* assumption [LV08], which is itself a weakening of the *2-out-of-3 CDH* assumption [KP06]. The latter states that given (G, G^a, G^b), it is hard to output (R, R^{ab}) for an arbitrary R; to solve 1-flexible CDH, one must additionally compute R^a. We weaken the assumption further by defining a solution as (R, R^a, R^b, R^{ab}), and call it the *weak flexible CDH* assumption.

Assumption 3 (WFCDH)**.** *Given $(G, G^a, G^b) \in \mathbb{G}^3$ for random $a, b \leftarrow \mathbb{Z}_p$, it is hard to output a non-trivial tuple (R, R^a, R^b, R^{ab}), i.e., with $R \in \mathbb{G}^*$.*

We define a generalization to asymmetric groups of the above assumption. G will be the generator of $\mathbb{G}_1$ and instead of G^b, we give a random generator H of $\mathbb{G}_2$; so a solution $(G^r, G^{ra}, G^{rb}, G^{rab})$ becomes $(G^r, G^{ra}, H^r, H^{rb})$ and can be efficiently verified due to the pairing.

Assumption 4 (AWFCDH)**.** *Given random generators $G \in \mathbb{G}_1$ and $H \in \mathbb{G}_2$, and $A = G^a$ for $a \leftarrow \mathbb{Z}_p$, it is hard to output $(G^r, G^{ra}, H^r, H^{ra}) \in (\mathbb{G}_1^*)^2 \times (\mathbb{G}_2^*)^2$, i.e., a tuple (R, M, S, N) that satisfies*

$$e(A, S) = e(M, H) \qquad e(M, H) = e(G, N) \qquad e(R, H) = e(G, S) \tag{3}$$

Argument. Under KEA, Assumption 4 is equivalent to the discrete-logarithm (DL) assumption, thus a fortiori it holds in the generic group model. Let $(G, A) \in \mathbb{G}_1$ be a DL instance, i.e., we have to compute $a := \log_G A$. Let $H \in \mathbb{G}_2$ be the group element for KEA. Give the adversary (G, A, H). From a successful output, by KEA, we can extract $m := \log_G M = \log_H N$ and $r := \log_G R = \log_H S$. From (3), we have $ar = m$, and since $r \neq 0$, a solution $a = \frac{m}{r}$. □

5 Instantiations

5.1 Automorphic Signatures

DHSDH immediately yields a weakly secure signature scheme if we consider X as the public key, (V, W) as a message in $\mathcal{DH} = \{(G^v, H^v) \mid v \in \mathbb{Z}_p\}$ and (A, C, D) as the signature.[7] We show how to transform this into a CMA-secure signature scheme by assuming WFCDH. We add some more randomness to the signature that lets us map a query for a message chosen by the adversary to a given tuple $(A_i, C_i, D_i, V_i, W_i)$ from a DHSDH instance. WFCDH then asserts that the adversary cannot produce a signed new message $\big((A^*, C^*, D^*, R^*, S^*), (M^*, N^*)\big)$ that maps back to a tuple from the instance (see the proof of Theorem 2).

Scheme 1 (**Sig**$_{\text{FPV}}$)**.** *Given a bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, G)$,* $\mathsf{Setup}_{\mathsf{FPV}}$ *chooses parameters $(H, K, T) \leftarrow \mathbb{G}^3$, which define the message space as $\mathcal{DH} := \{(G^m, H^m) \mid m \in \mathbb{Z}_p\}$.* $\mathsf{KeyGen}_{\mathsf{FPV}}$ *chooses a secret key $x \leftarrow \mathbb{Z}_p$ and sets $vk := G^x$. A message $(M, N) \in \mathcal{DH}$ is signed by* $\mathsf{Sign}_{\mathsf{FPV}}(x, (M, N))$ *which chooses $c, r \leftarrow \mathbb{Z}_p$ and outputs*

$$\big(A := (K \cdot T^r \cdot M)^{\frac{1}{x+c}},\ C := G^c,\ D := H^c,\ R := G^r,\ S := H^r\big)\ .$$

$\mathsf{Verify}_{\mathsf{FPV}}$ *accepts a signature (A, C, D, R, S) on a message $(M, N) \in \mathcal{DH}$ for public key X if it satisfies*

$$e(A, X \cdot C) = e(K \cdot M, G)\, e(T, R) \qquad e(C, H) = e(G, D) \qquad e(R, H) = e(G, S) \tag{4}$$

Theorem 1. *Under q-DHSDH and WFCDH,* **Sig**$_{\text{FPV}}$ *is strongly existentially unforgeable against adversaries making at most $q - 1$ adaptive chosen-message queries.*

The proof is analogous to that of Theorem 2.

[7] Note that this is not the case for the q-HSDH assumption (cf. Footnote 6): we cannot regard (G^c, H^c) as the message, since the signer must know c in order to produce $G^{\frac{1}{x+c}}$. If the message is a public key then the exponent cannot be given to the signer, which is precisely the reason for the complex protocol in [BCC+09].

Remark 3. (1) The above scheme can be easily extended to a *certified signature*[8] [BFPW07]: consider two instances of $\mathbf{Sig}_{\text{FPV}}$ (one for certification, one for signatures) that share parameters G, K and T but use a different H_i each. The certification authority's key is G^x, user public keys are of the form (G^v, H_1^v) and messages of the form (G^m, H_2^m). Security follows analogously to the next construction:
(2) From the certified signature we can construct an automorphic scheme $\mathbf{Sig}_{\text{2FPV}}$ as follows.[9] The public key is a certification-authority key extended to (G^x, H_2^x). An automorphic signature on a message (G^m, H_2^m) is produced by generating a random user key (G^v, H_1^v) and making a certified signature on the message under that key.

Public keys of $\mathbf{Sig}_{\text{2FPV}}$ are thus contained in the message space. Security follows from the following hybrid argument. Forgeries using a new one-time key (G^v, H_1^v) are reduced to forgeries for the 1st-level scheme (the simulator chooses $h \leftarrow \mathbb{Z}_p$, sets $H_2 := G^h$ and can thus produce a $\mathbf{Sig}_{\text{2FPV}}$ key from a $\mathbf{Sig}_{\text{FPV}}$ key). Forgeries recycling a key from a signing query are reduced security of the 2nd-level scheme (the simulator sets $H_1 := G^h$, guesses the recycled key (G^v, H_1^v) and sets it to (X, X^h) with X a challenge public key of the 2nd-level scheme). A signature consists of 12 group elements satisfying 7 PPEs (of which 5 are linear).

In the asymmetric setting (or assuming ADHSDH rather than DHSDH in symmetric groups), we get the following more efficient construction, whose signatures are in $\mathbb{G}_1^3 \times \mathbb{G}_2^2$.

Scheme 2 ($\mathbf{Sig}_{\text{A}}$). $\mathsf{Setup}_{\text{A}}$ *Given $\mathcal{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$, choose additional generators $F, K, T \in \mathbb{G}_1$. The message space containing the public key space is $\mathcal{DH} := \{(G^m, H^m) \mid m \in \mathbb{Z}_p\}$.*

$\mathsf{KeyGen}_{\text{A}}$ *Choose $sk = x \leftarrow \mathbb{Z}_p$ and set $vk = (G^x, H^x)$.*

Sign_{A} *A signature on $(M, N) \in \mathcal{DH}$, valid under public key (G^x, H^x), is defined as*

$$\big(A := (K \cdot T^r \cdot M)^{\frac{1}{x+c}},\, C := F^c,\, D := H^c,\, R := G^r,\, S := H^r\big)\,, \quad \textit{for random } c, r \leftarrow \mathbb{Z}_p$$

$\mathsf{Verify}_{\text{A}}$ *(A, C, D, R, S) is valid on a message $(M, N) \in \mathcal{DH}$ under a public key $vk = (X, Y) \in \mathcal{DH}$ iff*

$$e(A, Y \cdot D) = e(K \cdot M, H)\, e(T, S) \qquad e(C, H) = e(F, D) \qquad e(R, H) = e(G, S) \tag{5}$$

Theorem 2. *Assuming q-ADHSDH and AWFCDH, $\mathbf{Sig}_{\text{A}}$ is strongly existentially unforgeable against adversaries making at most $q - 1$ adaptive chosen-message queries.*

A proof can be found in Appendix D.1. Note that the scheme can also be instantiated for $\mathbb{G}_1 = \mathbb{G}_2$.

Remark 4. $\mathbf{Sig}_{\text{A}}$ can also sign bit strings (matching thus the standard definition of signatures) if we assume a collision-resistant hash function $\mathsf{Hash}\colon \{0,1\}^* \to \mathbb{Z}_p$. Define $\mathbf{Sig}_{\text{A}}^* := (\mathsf{Setup}_{\text{A}}, \mathsf{KeyGen}_{\text{A}}, \mathsf{Sign}_{\text{A}}^*, \mathsf{Verify}_{\text{A}}^*)$ with $\mathsf{Sign}_{\text{A}}^*(sk, m) := \mathsf{Sign}_{\text{A}}(sk, (G^{\mathsf{Hash}(m)}, H^{\mathsf{Hash}(m)}))$ and $\mathsf{Verify}_{\text{A}}^*(vk, \Sigma, m) := \mathsf{Verify}_{\text{A}}(vk, \Sigma, (G^{\mathsf{Hash}(m)}, H^{\mathsf{Hash}(m)}))$. Security against chosen-message attacks follows by a straightforward reduction to security of $\mathbf{Sig}_{\text{A}}$ and collision resistance of Hash.

5.2 Blind Automorphic Signatures

Before defining it formally, we give some intuition for the scheme discussed in Sect. 3.1. To obtain a blind signature on (M, N), the user chooses a random $\rho \leftarrow \mathbb{Z}_p$, and *blinds* M by the factor T^ρ. In addition to $U := T^\rho \cdot M$, she sends Groth-Sahai commitments to (M, N) and (G^ρ, H^ρ) and proofs of consistency. The signer now formally produces a "signature" on U,[10] for which we have $A = (K \cdot T^r \cdot U)^{1/(x+c)} = (K \cdot T^{r+\rho} \cdot M)^{1/(x+c)}$; thus A is the first component of a signature on (M, N) with randomness $r + \rho$. The user can complete the signature by adapting randomness r to $r + \rho$ in the other components. The blind signature is a Groth-Sahai proof of knowledge of this signature.

[8] A certified signature consists of the user public key, a certificate on it and a signature on the message under the user public key. Given certified signatures for various public keys, it must be hard to produce a new certified signature (either with a new or a given user key).

[9] Basically, we transform a certified-signature scheme whose authority keys lie in the message space to an automorphic-signature scheme.

[10] Note that the user does *not* obtain a signature on U (unless $U = M$), since it is not an element of the message space. To produce $(U, H^{\log_G U}) \in \mathcal{DH}$ in addition to M, N, P and Q, the user would have to break weak flexible CDH.

Obtain$(ck, pk, (M,N))$ Choose $\rho \leftarrow \mathbb{Z}_p$, $\tau_M, \tau_N, \tau_P, \tau_Q \leftarrow \mathcal{R}$, set $P := G^\rho$, $Q := H^\rho$, and send the following:

$$\mathbf{c}_M := \mathsf{Com}_{\mathrm{GS}}(ck, M, \tau_M) \quad \mathbf{c}_N := \mathsf{Com}_{\mathrm{GS}}(ck, N, \tau_N) \quad \phi_M := \mathsf{Prove}_{\mathrm{GS}}\big(ck, E_{\mathcal{DH}},\ (M,\tau_M), (N,\tau_N)\big)$$
$$\mathbf{c}_P := \mathsf{Com}_{\mathrm{GS}}(ck, P, \tau_P) \quad \mathbf{c}_Q := \mathsf{Com}_{\mathrm{GS}}(ck, Q, \tau_Q) \quad \phi_P := \mathsf{Prove}_{\mathrm{GS}}\big(ck, E_{\mathcal{DH}},\ (P,\tau_P), (Q,\tau_Q)\big)$$
$$U := T^\rho \cdot M \qquad \phi_U := \mathsf{Prove}_{\mathrm{GS}}\big(ck, E_U,\ (M,\tau_M), (Q,\tau_Q)\big)$$

with Equation $E_U(M,Q)$ defined as $e(T,Q)\,e(M,H) = e(U,H)$.

Issue(ck, x) If all proofs are valid, choose $c, r \leftarrow \mathbb{Z}_p$ and send:

$$A := (K \cdot T^r \cdot U)^{\frac{1}{x+c}} \qquad C := F^c \qquad D := H^c \qquad R' := G^r \qquad S' := H^r$$

The user sets $R := R' \cdot P$, $S := S' \cdot Q$, and checks whether (A, C, D, R, S) is valid on (M,N) under *pk*. The blind signature is a Groth-Sahai proof of knowledge $(\vec{\mathbf{c}}, \vec{\pi})$ of (A, C, D, R, S) satisfying (5) for *ck*.

Figure 1: Two-move blind signing protocol.

Scheme 3 (**BSig**)**.** $\mathsf{Setup}_{\mathrm{B}}(\mathcal{BG})$ *runs* $(G, F, K, T, H) \leftarrow \mathsf{Setup}_{\mathrm{A}}(\mathcal{BG})$ *and ck* $\leftarrow \mathsf{Setup}_{\mathrm{GS}}(\mathcal{BG})$ *and returns these outputs as common parameters pp. As for* $\mathbf{Sig}_{\mathrm{A}}$*, the message space is* $\mathcal{DH}$.

$\mathsf{KeyGen}_{\mathrm{B}}$ *is defined as* $\mathsf{KeyGen}_{\mathrm{A}}$.

$\mathsf{Issue} \leftrightarrow \mathsf{Obtain}$ *The blind signing protocol is given in Fig. 1.*

$\mathsf{Verify}_{\mathrm{B}}(pp, (X,Y), (M,N), (\vec{c}, \vec{\pi}))$ *For* $(X,Y), (M,N) \in \mathcal{DH}$*,* $\mathsf{Verify}_{\mathrm{B}}$ *runs* $\mathsf{Verify}_{\mathrm{GS}}(ck, E_{\mathsf{Ver}_{\mathrm{A}}}, \vec{\mathbf{c}}, \vec{\pi})$*, with* $E_{\mathsf{Ver}_{\mathrm{A}}}$ *being the equations in* (5).

Theorem 3. *Under Assumptions ADHSDH and AWFCDH, and SXDH or DLIN (depending on the instantiation), scheme* $\mathbf{BSig}$ *is an unforgeable blind-signature scheme.*

Unforgeability is shown by reduction to unforgeability of $\mathbf{Sig}_{\mathrm{A}}$. In the WI setting, two GS proofs of knowledge of different signatures on the same message are indistinguishable; moreover, the issuer gets no information on the message during the issuing protocol. Together this implies blindness. See Appendix D.2 for a proof.

The round complexity of the scheme is optimal [Fis06]. In the DLIN instantiation, the user sends 22 group elements (GE), since all proofs are for linear equations (cf. Footnote 3), and the signer sends 5 GE. Blind signatures consist of 30 GE ($\vec{\mathbf{c}}$ is in $\mathbb{G}^{5\times 3}$ and $\vec{\pi}$ consists of $9 + 2\cdot 3$ GE). In the SXDH instantiation, the user message is in $\mathbb{G}_1^{17} \times \mathbb{G}_2^{16}$, the signer message in $\mathbb{G}_1^{3} \times \mathbb{G}_2^{2}$ and a blind signature is in $\mathbb{G}_1^{18} \times \mathbb{G}_2^{16}$. Note that the scheme remains automorphic, since commitments and proofs are composed of group elements and are verified by checking PPEs.

Remark 5 (Weaker Assumptions)**.** If we base $\mathbf{BSig}$ on a symmetric bilinear group and the scheme $\mathbf{Sig}_{\mathrm{FPV}}$ rather than $\mathbf{Sig}_{\mathrm{A}}$, we obtain a round-optimal blind signature scheme which is not automorphic but which is secure under DHSDH and WFCDH (and SXDH or DLIN).

Remark 6 (Signing Committed Values)**.** The core building block for P-signatures [BCKL08] is an interactive protocol allowing a user that published a commitment to obtain a signature on the committed value. If the user publishes $(\mathbf{c}_M, \mathbf{c}_N, \phi_M)$ before running the blind-signature protocol we get exactly this.

5.3 Automorphic Signatures on Message Vectors

In order to sign vectors of messages of arbitrary length, we proceed as follows. We first show how to transform a scheme whose message space forms a *group* (and contains the public-key space) to one that signs 2 messages at once—if we exclude the neutral element from the message space of the transform. A signature on a message pair will contain 3 signatures (of the original scheme) of different *products* of the messages. In Appendix B we show that 3 are indeed necessary. Note that $\mathcal{DH}$, the message space for the schemes $\mathbf{Sig}_{\mathrm{FPV}}$ and $\mathbf{Sig}_{\mathrm{A}}$, is a group when the group operation is defined as component-wise multiplication.

We then give a straightforward generic transformation from any scheme signing 2 messages (and whose verification keys lie in the message space) to one signing message vectors of arbitrary length (Def. 3). Both transformations do not modify setup and key generation and they are invariant w.r.t. the structure of verification; in particular, if the verification predicate of the original scheme is a conjunction of PPEs then so is that of the transform.

Definition 2. *Let* $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be a signature scheme whose message space* $(\mathcal{M}, \cdot)$ *is a group and contains the verification keys. The* pair transform *of* $\mathbf{Sig}$ *with message space* $(\mathcal{M} \setminus \{1\})^2$ *is defined as* $\mathbf{Sig}' = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}', \mathsf{Verify}')$ *with*

$\mathsf{Sign}'_{sk}(M_1, M_2)$
- $(vk_0, sk_0) \leftarrow \mathsf{KeyGen}$
- $\sigma := \big(vk_0,\ \mathsf{Sign}_{sk}(vk_0),\ \mathsf{Sign}_{sk_0}(M_1),\ \mathsf{Sign}_{sk_0}(M_1 \cdot M_2),\ \mathsf{Sign}_{sk_0}(M_1 \cdot M_2^3)\big).$

$$\mathsf{Verify}'_{vk}\big((M_1, M_2),\ (vk_0, \tau, \sigma_1, \sigma_2, \sigma_3)\big) := \mathsf{Verify}_{vk}(vk_0, \tau) \wedge \mathsf{Verify}_{vk_0}(M_1, \sigma_1) \wedge \mathsf{Verify}_{vk_0}(M_1 \cdot M_2, \sigma_2) \wedge \mathsf{Verify}_{vk_0}(M_1 \cdot M_2^3, \sigma_3).$$

Theorem 4. *If* $\mathbf{Sig}$ *is secure against EUF-CMA then so is* $\mathbf{Sig}'$.

Definition 3. *Let* $\mathbf{Sig} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be a signature scheme whose message space contains its public-key space and which signs* pairs *of messages. Assume an efficiently computable injection* Inj *from* $\{1, \ldots, n_{\max}\}$ *to the message space, where* $n_{\max}$ *is the maximum length of a message vector. The* vector transform *of* $\mathbf{Sig}$ *is defined as* $\mathbf{Sig}'' = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}'', \mathsf{Verify}'')$ *with*

$\mathsf{Sign}''_{sk}(M_1, \ldots, M_n)$: $(vk_0, sk_0) \leftarrow \mathsf{KeyGen}$; *return* $\sigma := \big(vk_0, \mathsf{Sign}_{sk}(vk_0, \mathsf{Inj}(n)), \big(\mathsf{Sign}_{sk_0}(M_i, \mathsf{Inj}(i))\big)_{i=1}^{n}\big).$

$\mathsf{Verify}''_{vk}\big((M_1, \ldots, M_n),\ (vk_0, \sigma_0, \sigma_1, \ldots, \sigma_n)\big) := \mathsf{Verify}_{vk}\big((vk_0, \mathsf{Inj}(n)), \sigma_0\big) \wedge \bigwedge_{i=1}^{n} \mathsf{Verify}_{vk_0}\big((M_i, \mathsf{Inj}(i)), \sigma_i\big).$

Theorem 5. *If* $\mathbf{Sig}$ *is secure against EUF-CMA then so is* $\mathbf{Sig}''$.

Proofs of Theorems 4 and 5 can be found in Appendices D.3 and D.4, resp. In Appendix B, we discuss why the construction in Def. 2 is somewhat optimal and why it seems hard to construct a vector transform directly.

6 Conclusions

We introduced the concept of automorphic signatures and gave two instantiations; the first is based on known assumptions while the second is more efficient and can be instantiated in asymmetric bilinear groups. It relies on a new assumption, which we prove to hold in the generic group model. We used our scheme to give the first efficient instantiation of Fischlin's round-optimal blind signatures. Furthermore, we illustrated the numerous benefits of automorphic signatures by constructing fully-secure group signatures and anonymous credentials, and by giving the first efficient instantiation of anonymous proxy signatures, providing additional anonymity guarantees that have not been considered so far.

We leave as an open problem the construction of a practical automorphic signature whose messages are single group elements. It would also be interesting to see if the techniques used in Def. 2 can be generalized to vectors of arbitrary (but fixed) length; that is, to define a direct transformation from a signature scheme whose message space is a group to one signing an arbitrarily fixed number of messages.

Acknowledgments

The author would like to thank David Pointcheval and Damien Vergnaud for many invaluable discussions that led to the present paper. The author is also grateful to Masayuki Abe for a discussion on round-optimal blind signatures and for pointing out a flaw in an earlier construction of the vector transform. This work was supported by EADS, the French ANR-07-SESU-008-01 PAMPA Project and the European Commission through the ICT Program under Contract ICT-2007-216646 ECRYPT II.

References

[ACHM05] Giuseppe Ateniese, Jan Camenisch, Susan Hohenberger, and Breno de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org/`.

[ACM05] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In Vijayalakshmi Atluri, Catherine Meadows, and Ari Juels, editors, *ACM CCS 05*, pages 92–101. ACM Press, November 2005.

[AO09] Masayuki Abe and Miyako Ohkubo. A framework for universally composable non-committing blind signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 435–450. Springer, December 2009.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, May 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, August 2004.

[BCC+09] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125. Springer, August 2009.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, March 2008.

[BCKL09] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Compact e-cash and simulatable VRFs revisited. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 114–131. Springer, August 2009.

[BFPW07] Alexandra Boldyreva, Marc Fischlin, Adriana Palacio, and Bogdan Warinschi. A closer look at PKI: Security and efficiency. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 458–475. Springer, April 2007.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, April 2009.

[BN05] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In Bart Preneel and Stafford Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, August 2005.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003.

[Boy08] Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008*, volume 5209 of *LNCS*, pages 39–56. Springer, September 2008.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, August 2004.

[BPW03] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096, 2003. `http://eprint.iacr.org/`.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, February 2005.

[BW06] Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, May / June 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, April 2007.

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, April 2009.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CGS07] Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 423–434. Springer, July 2007.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[CKW04] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 134–148. Springer, September 2004.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 56–72. Springer, August 2004.

[Cv91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer, April 1991.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, August 1992.

[Fis06] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 60–77. Springer, August 2006.

[FP08] Georg Fuchsbauer and David Pointcheval. Anonymous proxy signatures. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN 08*, volume 5229 of *LNCS*, pages 201–217. Springer, September 2008.

[FP09] Georg Fuchsbauer and David Pointcheval. Proofs on encrypted values in bilinear groups and an application to anonymity of signatures. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 132–149. Springer, August 2009. Full version available at `http://eprint.iacr.org/2008/528`.

[FPV09] Georg Fuchsbauer, David Pointcheval, and Damien Vergnaud. Transferable constant-size fair e-cash. In Juan A. Garay, Atsuko Miyaji, and Akira Otsuka, editors, *CANS 09*, volume 5888 of *LNCS*, pages 226–247. Springer, December 2009. Full version available at `http://eprint.iacr.org/2009/146`.

[GL07] Jens Groth and Steve Lu. A non-interactive shuffle with pairing based verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67. Springer, December 2007.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, May / June 2006.

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

[Gro06] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, December 2006.

[Gro07] Jens Groth. Fully anonymous group signatures without random oracles. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, December 2007.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, April 2008.

[GSW09] Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Groth–sahai proofs revisited. Cryptology ePrint Archive, Report 2009/599, 2009. `http://eprint.iacr.org/`.

[HKKL07] Carmit Hazay, Jonathan Katz, Chiu-Yuen Koo, and Yehuda Lindell. Concurrently-secure blind signatures without random oracles or setup assumptions. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 323–341. Springer, February 2007.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 150–164. Springer, August 1997.

[Kil06] Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, March 2006.

[KP06] Sébastien Kunz-Jacques and David Pointcheval. About the security of MTI/C0 and MQV. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 156–172. Springer, September 2006.

[KZ06] Aggelos Kiayias and Hong-Sheng Zhou. Concurrent blind signatures without random oracles. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 49–62. Springer, September 2006.

[KZ08] Aggelos Kiayias and Hong-Sheng Zhou. Equivocal blind signatures and adaptive UC-security. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 340–355. Springer, March 2008.

[LRSW00] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard M. Heys and Carlisle M. Adams, editors, *SAC 1999*, volume 1758 of *LNCS*, pages 184–199. Springer, August 2000.

[LV08] Benoît Libert and Damien Vergnaud. Multi-use unidirectional proxy re-signatures. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 08*, pages 511–520. ACM Press, October 2008.

[MRY04] Philip D. MacKenzie, Michael K. Reiter, and Ke Yang. Alternatives to non-malleability: Definitions, constructions, and applications. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 171–190. Springer, February 2004.

[MUO96] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto. Proxy signatures for delegating signing operation. In *ACM CCS 96*, pages 48–57. ACM Press, March 1996.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, August 2003.

[NIS07] Recommendation for key management, special publication 800-57 part 1, NIST, 03/2007, 2007.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 80–99. Springer, March 2006.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 1999.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, May 1997.

[SMP08] Jacob C. N. Schuldt, Kanta Matsuura, and Kenneth G. Paterson. Proxy signatures secure against proxy key exposure. In Ronald Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 141–161. Springer, March 2008.

[TW05] Mårten Trolin and Douglas Wikström. Hierarchical group signatures. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP 2005*, volume 3580 of *LNCS*, pages 446–458. Springer, July 2005.

[Wat05] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, May 2005.

A Further Extensions of Anonymous Proxy Signatures

A.1 CCA-Anonymous Proxy Signatures.

CCA-anonymity (i.e., anonymity against adversaries provided with an opening oracle) of Groth's group signatures [Gro07] (sketched in Sect. 3.3) is proved as follows: modify the security game by substituting the opener's commitment key by one that results in perfectly hiding commitments and WI proofs; then due to the additional encryptions contained in a group signature, opening queries for all but the challenge signature can still be simulated.

We transform the anonymous proxy signature scheme given in Sect. 3.4 into one satisfying CCA-anonymity analogously. Suppose a proxy signer holds $W := (vk_1, (warr_i, cert_i, vk_i)_{i=2}^k)$ and sk_k. To make a signature, she first chooses keys for a one-time signature $(vk_{\mathrm{ot}}, sk_{\mathrm{ot}}) \leftarrow \mathsf{KeyGen}_{\mathrm{ot}}$ and signs vk_{ot} (instead of M) with her personal key sk_k yielding *sig*. She makes commitments $\vec{\mathbf{c}}$ to the elements of W and *sig*, and adds a WI proof ϕ_j for each equation E_j in ($\mathrm{Ver}_{\mathrm{PPS}}$) in Sect. 3.4, which are satisfied by W and *sig*—as in the original scheme.

In addition, for $2 \leq i \leq k$ she computes an $\mathbf{Enc}_{\mathrm{tb}}$-encryption C_i of $warr_i$ under tag vk_{ot} and, as in [Gro07], she makes a Groth-Sahai NIZK proof ζ_i that the plaintext of C_i is the value committed in $\mathbf{c}_{warr_i}$. She computes $sig_{\mathrm{ot}} := \mathsf{Sign}_{\mathrm{ot}}(sk_{\mathrm{ot}}, (vk_{\mathrm{ot}}, M, \vec{\mathbf{c}}, \vec{\phi}, \vec{C}, \vec{\zeta}))$ and outputs the signature $(vk_{\mathrm{ot}}, \vec{\mathbf{c}}, \vec{\phi}, \vec{C}, \vec{\zeta}, sig_{\mathrm{ot}})$. A signature is *valid* if sig_{ot} is valid under vk_{ot}, the proofs ϕ_j are valid for all j, and the proofs ζ_i and the ciphertexts C_i are valid for all i. Given a valid signature, the opener returns the values $(vk_i, warr_i)_{i=1}^k$ extracted from the commitments $\vec{\mathbf{c}}$ using the extraction key.

The proof for CCA-anonymity is analogous to that for Groth's group signatures. Let Game 0 denote the game for CCA-anonymity defined by [FP08]. The adversary $\mathcal{A}$ controls the issuer and the users and has on opening oracle for an honest opener. After the first phase $\mathcal{A}$ returns a public key *pk* for an original delegator, two user secret keys and two valid warrants of equal length from *pk* to the users, as well as a message. $\mathcal{A}$ receives an anonymous proxy signature produced with one of the secret keys and the corresponding warrant. After a second phase of opening queries, $\mathcal{A}$ has to decide which key/warrant pair was used.

In Game 1, the opening queries are simulated by decrypting $\vec{C}$, checking for which users the warrants are valid and returning their registered keys together with the warrants. Soundness of the proofs $\vec{\zeta}$ guarantees perfect simulation. In Game 2, we replace the opener's commitment key by a witness-indistinguishable one and in Game 3 we simulate the proofs in $\vec{\zeta}$. The unforgeability of the one-time signature $\mathbf{Sig}_{\mathrm{ot}}$ prevents the adversary from querying opening of a proxy signature which is different from the challenge but contains the same vk_{ot}. We can thus use an adversary winning Game 3 to break selective-tag weak CCA security of $\mathbf{Enc}_{\mathrm{tb}}$ since we only have to answer decryption queries for tags $vk'_{\mathrm{ot}} \neq vk_{\mathrm{ot}}$.

A.2 Multiple Original Delegators

If in anonymous proxy signatures, we allow delegation to take the form of a tree (whose leaves represent original delegators, and delegation goes from the leaves to the root) rather than a list, we can define proxy signatures on behalf of several originators. For example, consider three original delegators O, P, Q, the first of which delegates to A who re-delegates to B. User B is also delegated by P and re-delegates the rights for both O and P to C. Moreover Q delegates to C. Now C can produce a signature on behalf of O, P *and* Q.

In general, we define a *multi-originator signature* (MOS) recursively: A (plain) MOS consists of a signature on the message, the signer's verification key and a list of objects *del* for the signer (which represent the delegations to her). A *del* for user U is either a warrant from an originator for U or a warrant from a user U', the verification key of U' and a list of *del*'s for U'. A (plain) signature on behalf of a set of originators is valid if the signature on the message is valid, all warrants are valid and it contains a warrant from each of the originators. As for the single-originator case, a plain signature is anonymized by committing to its components and adding proofs of validity.

In the above example, a signature by C on behalf of O, P and Q has the following form (we let $\psi_{U_1 \to U_2}$ denote $\mathbf{c}_{U_1 \to U_2} \,\|\, \phi_{U_1 \to U_2}$, and ψ_M denote a commitment to *sig* and a proof of validity):

$$\Big\{ \psi_M, \mathbf{c}_C, \big\{ \{ \psi_{B \to C}, \mathbf{c}_B, \{\{\psi_{A \to B}, \mathbf{c}_A, \psi_{O \to A}\}, \psi_{P \to B}\} \big\}, \psi_{Q \to C} \big\} \Big\} .$$

B A Discussion on the Transformations in Section 5.3

Transforming a signature scheme whose verification keys lie in the message space to one that signs vectors of messages of arbitrary length is somewhat hard. An approach that comes to mind is the following: For each signature, the signer first produces a temporary key pair (vk, sk), signs vk with her secret key and uses sk to sign every component of the vector. An easy attack would be to reorder the messages of a queried vector. To prevent this shuffling attack, we let sk sign one transient key per message component, which will sign the message and its *index*. To thwart an attack that returns a truncated message, we also sign the length.

To sign the indices and the length, we need to assume an injection Inj from natural numbers into the message space as in Def. 3. The above construction however succumbs to a series of attacks, which come from the fact that verification keys, images under Inj, and message all have the same form, which is inherent. An adversary could for example query a signature on the message $(\mathsf{Inj}(2), \mathsf{Inj}(1))$ and return a signature on $(\mathsf{Inj}(1), \mathsf{Inj}(2))$ by simply reordering the signature components. If however we start from a signature scheme signing 2 messages, we avoid all these problems as can be seen by the natural construction in Def. 3 and its straightforward proof in Appendix D.4

The crucial step is thus that from 1 to 2 messages. If we assume some structure on the message space (which is the case for our constructions, since messages are elements of an algebraic group), then we could try to sign several messages at once by signing their *product*. Again, we first sign a "one-time" key with the actual key, and use that key to produce the signatures contained in a signature of the transform. This prevents the adversary from combining signatures received from different queries and we thus only have to handle one-time attacks. As it turns out, we have to construct the messages we actually sign very carefully to prevent the adversary from deriving a signature on a new message from a signing-query response. If we only sign *one* product of the components, there are trivial attacks. Signing two products seems more promising, but we show that this do not suffice either:

Concretely, we want to devise a scheme that signs (M_1, M_2) by signing two linear combinations of the messages; i.e., a signature on (M_1, M_2) consists of a signature on $(M_1^{a_1} \cdot M_2^{a_2})$ and one on $(M_1^{b_1} \cdot M_2^{b_2})$, for some fixed $(a_1, a_2, b_1, b_2) \in \mathbb{Z}^4$.

Assume first that (a_1, a_2) and (b_1, b_2) are linearly dependent, i.e., $b_1 = ca_1$ and $b_2 = ca_2$ for some c and that $a_1 \neq 0$ (otherwise signatures would be independent of M_1 and thus easily forgeable). After querying a transform signature on (M_1, M_2) (and thus receiving signatures on $(M_1^{a_1} \cdot M_2^{a_2})$ and $(M_1^{ca_1} \cdot M_2^{ca_2})$), one can produce a forgery as follows: set $M_1^* := M_1 \cdot M_2^{a_2/a_1} (M_2^*)^{-a_2/a_1}$ for an arbitrary $M_2^* \neq M_2$. A signature on this messages consists thus of a signature on $(M_1^*)^{a_1} \cdot (M_2^*)^{a_2} = M_1^{a_1} \cdot M_2^{a_2}$ and $(M_1^*)^{ca_1} \cdot (M_2^*)^{ca_2} = M_1^{ca_1} \cdot M_2^{ca_2}$, thus the precise two messages for which we have signatures from the signing query.

Assume now that (a_1, a_2) and (b_1, b_2) are linearly independent, i.e., $a_1 b_2 - b_1 a_2 \neq 0$; w.l.o.g., assume that $b_2 \neq 0$. Querying (M_1, M_2) yields signatures Σ_1 and Σ_2 on $(M_1^{a_1} \cdot M_2^{a_2})$ and $(M_1^{b_1} \cdot M_2^{b_2})$, respectively. Setting $M_1^* := M_1^{(b_1 b_2 - a_1 a_2)/D} \cdot M_2^{(b_2^2 - a_2^2)/D}$ (with $D := a_1 b_2 - b_1 a_2$) and $M_2^* := M_1^{a_1/b_2} \cdot M_2^{a_2/b_2} \cdot (M_1^*)^{-b_1/b_2}$ makes $(M_1^*)^{a_1} \cdot (M_2^*)^{a_2} = M_1^{b_1} \cdot M_2^{b_2}$ and $(M_1^*)^{b_1} \cdot (M_2^*)^{b_2} = M_1^{a_1} \cdot M_2^{a_2}$, thus we can reuse the signatures, i.e., produce a forgery (Σ_2, Σ_1) on (M_1^*, M_2^*).

Moreover, note that finding *three* linear combinations leading to a valid scheme is not trivial either. E.g., choosing M_1, $M_1 \cdot M_2$ and $M_1 \cdot M_2^2$ succumbs to the following attack: Setting $M_1^* := M_1 \cdot M_2^2$ and $M_2^* := M_2^{-1}$, we can recycle and reorder the signatures from the query.

C The q-ADHSDH Assumption

C.1 A Note on ADHSDH

One could be tempted to transfer the DHSDH assumption to asymmetric groups by adding $Y := (\log_G) H$ to the instance, which would allow to check validity of a tuple (A, C, V, D, W). However, this assumption is wrong, as it succumbs to the following attack: Given an instance $(G, H, K, X, Y, (A_i, C_i, V_i, D_i, W_i)_{i=1}^{q-1})$, set $A^* := A_1^{-1}$, $C^* := X^{-2} \cdot C_1^{-1}$, $D^* := Y^{-2} \cdot D_1^{-1}$, $V^* := V_1$, $W^* := W_1$. Then we have $e(A^*, Y \cdot D^*) = e(A_1^{-1}, (Y \cdot D_1)^{-1}) = e(K \cdot V_1, H) = e(K \cdot V^*, H)$. The attack comes from the fact that we can use X and Y to simultaneously build C^* and D^*. This is what makes it indispensable to use a different basis for the C, leading to a generically secure assumption, as proved in the next section.

The q-ADHSDH assumption is quite similar to the q-BB-HSDH assumption introduced in [BCC+09], which states the following:

Assumption 5 (BB-HSDH). *Let $x, c_1, \ldots, c_{q-1} \leftarrow \mathbb{Z}_p$. Then on input $G, G^x, F \in \mathbb{G}_1$ and $H, H^x \in \mathbb{G}_2$ and tuples $(G^{\frac{1}{x+c_i}}, c_i)_{i=1}^{q-1}$, it is infeasible to output a tuple $(G^{\frac{1}{x+c}}, F^c, H^c)$ with $c \neq c_i$ for all i.*

It is however incomparable to ADHSDH, since while ADHSDH gives the adversary more flexibility in his output, BB-HSDH gives him more information as input, since the c_i are given explicitly. Moreover, BB-HSDH is somehow asymmetric, in that the task is to output a tuple that is easier to construct than a tuple that has the form of the $q-1$ input tuples. Note that if we had $F = G$ (as in the original definition of HSDH in [BW07]), the BB-HSDH problem would become easy as the attack sketched above would work as well.

C.2 Generic Security of the q-ADHSDH Assumption

We prove generic security of ADHSDH in symmetric bilinear groups, as this covers the asymmetric case as well. For convenience we restate the assumption.

(q-ADHSDH) Given $(G, F, H, K, X = G^x, Y = H^x) \in \mathbb{G}^6$ and $q - 1$ tuples

$$\left(A_i = (K \cdot G^{v_i})^{\frac{1}{x+c_i}},\ B_i = F^{c_i},\ D_i = H^{c_i},\ V_i = G^{v_i},\ W_i = H^{v_i}\right),$$

with $c_i, v_i \leftarrow \mathbb{Z}_p^*$ for $i = 1, \ldots, q-1$, it is hard to output a new tuple $(A^*, B^*, D^*, V^*, W^*)$ that satisfies

$$e(A^*, Y \cdot D^*) = e(K \cdot V^*, H) \qquad e(B^*, H) = e(F, D^*) \qquad e(V^*, H) = e(G, W^*) \ . \tag{6}$$

Theorem 6. *The q-ADHSDH assumption holds in generic bilinear groups when q is a polynomial.*

Proof. We assume that the reader is familiar with the methodology of proofs in the generic group model and thus focus on our particular assumption. We work with the "discrete-log" representation of all group elements w.r.t. basis G. A q-ADHSDH instance is thus represented by the following rational fractions (each lower-case letter denotes the logarithm of the group elements denoted by the corresponding upper-case letter):

$$1,\ f,\ h,\ k,\ x,\ y = xh,\ \left\{a_i = \tfrac{k+v_i}{x+c_i},\ b_i = c_i f,\ d_i = c_i h,\ v_i,\ w_i = v_i h\right\}_{i=1}^{q-1} \tag{7}$$

Considering the logarithms of the $\mathbb{G}_T$-elements in (6) w.r.t. the basis $e(G,G)$ yields

$$a^*(xh + d^*) = (k + v^*)h \qquad b^* h = d^* f \qquad v^* h = w^* \tag{8}$$

In a generic group, all the adversary can do is apply the group operation to the elements of its input. We will show that the only linear combinations $(a^*, b^*, d^*, v^*, w^*)$ of elements in (7) satisfying (8) are $(a^* = a_i = \frac{k+v_i}{x+c_i}, b^* = b_i = c_i f, d^* = d_i = c_i h, v^* = v_i, w^* = w_i = v_i h)$ for some i; which means all the adversary can do is return a quintuple from the instance. We make the following ansatz for a^* (and analogously for b^*, d^*, v^* and w^*):

$$a^* = \alpha + \alpha_f f + \alpha_h h + \alpha_k k + \alpha_x x + \alpha_y xh + \sum \alpha_{a_i} \tfrac{k+v_i}{x+c_i} + \sum \alpha_{b,i} c_i f + \sum \alpha_{d,i} c_i h + \sum \alpha_{v,i} v_i + \sum \alpha_{w,i} v_i h$$

Since for any v^* the adversary forms, it has to provide $v^* h$ as well, we can limit the elements used for v^* to those of which their product with h is also given: $1, x$ and v_i (for all i). Similarly, plugging in the ansätze for b^* and d^* in the second equation of (8) and equating coefficients eliminates most of the coefficients. Thus, the last two equations of (8) simplify b^*, d^*, v^* and w^* to

$$\begin{aligned} b^* &= \gamma_f f + \textstyle\sum \gamma_{b,i} c_i f & \qquad v^* &= \mu + \mu_x x + \textstyle\sum \mu_{v,i} v_i \\ d^* &= \gamma_f h + \textstyle\sum \gamma_{b,i} c_i h & \qquad w^* &= \mu h + \mu_x xh + \textstyle\sum \mu_{v,i} v_i h \end{aligned}$$

We substitute a^*, d^*, v^* by their ansätze in the first equation of (8), that is $a^*(xh + d^*) - v^* h = kh$. After some rearranging we get (for convenience, we omit one h per term, i.e., we symbolically "divided" the equation by h):

$$(\alpha\gamma_f - \mu)\, 1 + (\alpha_f\gamma_f)\, f + (\alpha_h\gamma_f)\, h + (\alpha + \alpha_x\gamma_f - \mu_x)\, x + (\alpha_h + \alpha_y\gamma_f)\, xh + \tag{9a}$$

$$\sum (\alpha_{a,i}\gamma_f)\, \tfrac{k+v_i}{x+c_i} + \sum (\alpha_{b,i}\gamma_f + \alpha_f\gamma_{b,i})\, c_i f + \sum (\alpha_{d,i}\gamma_f + \alpha_h\gamma_{b,i})\, c_i h + \sum (\alpha_{w,i}\gamma_f)\, v_i h + \tag{9b}$$

$$(\alpha_f)\, xf + (\alpha_k)\, xk + (\alpha_x)\, x^2 + (\alpha_y)\, x^2 h + \sum (\alpha_{d,i} + \alpha_y\gamma_{b,i})\, c_i xh + \sum (\alpha_{b,i})\, c_i xf + \tag{9c}$$

$$\sum (\alpha_{v,i})\, v_i x + \sum (\alpha_{w,i})\, v_i xh + \sum (\alpha\gamma_{b,i})\, c_i + \sum (\alpha_k\gamma_{b,i})\, c_i k + \sum (\alpha_x\gamma_{b,i})\, xc_i + \tag{9d}$$

$$\sum\sum (\alpha_{b,i}\gamma_{b,j})\, c_i c_j f + \sum\sum (\alpha_{d,i}\gamma_{b,j})\, c_i c_j h + \sum\sum (\alpha_{v,i}\gamma_{b,j})\, v_i c_j + \sum\sum (\alpha_{w,i}\gamma_{b,j})\, v_i c_j h + \tag{9e}$$

$$\underbrace{(\alpha_k\gamma_f)}_{=:\lambda_k}\, k + \sum \underbrace{(\alpha_{v,i}\gamma_f - \mu_{v,i})}_{=:\lambda_{v,i}}\, v_i + \sum \underbrace{(\alpha_{a,i})}_{=:\lambda_{xa,i}}\, \tfrac{x(k+v_i)}{x+c_i} + \sum\sum \underbrace{(\alpha_{a,i}\gamma_{b,j})}_{=:\lambda_{ca,i,j}}\, \tfrac{c_j(k+v_i)}{x+c_i} = k \tag{9f}$$

Comparison of coefficients[11] of the two sides of the equation shows that all coefficients in lines (9a)–(9e) must be 0, whereas for the last line we have a different situation: adding $\frac{x(k+v_i)}{x+c_i}$ and $\frac{c_i(k+v_i)}{x+c_i}$ reduces to $k + v_i$ (but this is the only combination that reduces); we have thus

$$\text{for all } i: \lambda_{xa,i} = \lambda_{ca,i,i} \qquad \text{for all } i \neq j: \lambda_{ca,i,j} = 0 \tag{10}$$

$$\text{coefficient of } k\text{:} \quad \textstyle\sum \lambda_{xa,i} + \lambda_k = 1 \qquad \text{coefficient of } v_i\text{:} \quad \lambda_{xa,i} + \lambda_{v,i} = 0 \tag{11}$$

We now solve the equations "all coefficients in Lines (9a) to (9e) equal 0", and Equations (10) and (11) for the values $\left(\alpha, \alpha_f, \alpha_h, \alpha_k, \alpha_x, \alpha_y, \gamma_f, \mu, \mu_x, \{\alpha_{a,i}, \alpha_{b,i}, \alpha_{d,i}, \alpha_{v,i}, \alpha_{w,i}, \gamma_{b,i}, \mu_{v,i}\}\right)$:

The first four terms and the last term in Line (9c) and the first two terms in Line (9d) immediately yield: $\alpha_f = \alpha_k = \alpha_x = \alpha_y = \alpha_{b,i} = \alpha_{v,i} = \alpha_{w,i} = 0$ for all i. Now $\alpha_y = 0$ implies $\alpha_h = 0$ by the last term in (9a), and

[11] To do straightforward comparison of coefficients, we actually would have to multiply the equation by $\prod_{i=1}^{q-1}(x + c_i)$ first. For the sake of presentation, we keep the fractions and instead introduce new equations for the cases where a linear combination leads to a fraction that cancels down.

$\alpha_y = 0$ implies $\alpha_{d,i} = 0$ for all i by the fifth term in in (9c). Plugging in these values, the only equations different from "$0 = 0$" are the following:

$$\alpha\,\gamma_f - \mu = 0 \qquad\qquad \alpha - \mu_x = 0 \tag{12}$$

$$\alpha_{a,i}\,\gamma_f = 0 \;\; (\forall i) \qquad\qquad \alpha\,\gamma_{b,i} = 0 \;\; (\forall i) \tag{13}$$

$$\alpha_{a,i}(1-\gamma_{b,i}) = 0 \;\; (\forall i) \qquad\qquad \alpha_{a,i}\,\gamma_{b,j} = 0 \;\; (\forall i \neq j) \tag{14}$$

$$\textstyle\sum_{i=1}^{q-1} \alpha_{a,i} = 1 \qquad\qquad \alpha_{a,i} - \mu_{v,i} = 0 \;\; (\forall i) \tag{15}$$

where the second equation in (12) "(12.2)" follows from the fourth term in (9a) and $\alpha_x = 0$. (13.1) and (13.2) follow from the first term in (9b) and the third term in (9d), respectively. Equations (14) are the equations in (10); and those in (15) are the ones from (11) taking into account that $\alpha_k = 0$ and $\alpha_{v,i} = 0$ for all i. The variables not yet proved to be 0 are $\alpha, \gamma_f, \mu, \mu_x, \alpha_{a,i}, \gamma_{b,i}$ and $\mu_{v,i}$ for $1 \leq i \leq q-1$.

We first show that there exists $i^* \in \{1,\ldots,q-1\}$ such that $\alpha_{a,j} = 0$ for all $j \neq i^*$: assume there exist $i \neq j$ such that $\alpha_{a,i} \neq 0$ and $\alpha_{a,j} \neq 0$; then by (14.1) we have $\gamma_{b,i} = \gamma_{b,j} = 1$, which contradicts (14.2).

This result implies the following: by (15.1) we have $\alpha_{a,i^*} = 1$ and by (14.1) we have $\gamma_{b,i^*} = 1$, whereas for all $j \neq i^*$: $\gamma_{b,j} = 0$ by (14.2). We have thus shown that $\alpha_{a,i^*} = \gamma_{b,i^*} = 1$ and $\alpha_{a,j} = \gamma_{b,j} = 0$ for all $j \neq i^*$.

This now implies $\alpha = 0$ (by (13.2)) and thus $\mu = \mu_x = 0$ by ((12.1) and (12.2), respectively). Moreover $\gamma_f = 0$ (by (13.1)) and for all i: $\alpha_{a,i} = \mu_{v,i}$ (by (15.2)). The only non-zero variables are thus $\alpha_{a,i^*} = \gamma_{b,i^*} = \mu_{v,i^*} = 1$.

Plugging in our results in the ansätze for a^*, b^*, d^*, v^* and w^*, we proved that there exists $i^* \in \{1,\ldots,q-1\}$ such that $a^* = \frac{k+v_{i^*}}{x+c_{i^*}}$, $b^* = c_{i^*}f$, $d^* = c_{i^*}h$, $v^* = v_{i^*}$ and $w^* = v_{i^*}h$. This means that the only tuples $(A^*, B^*, D^*, V^*, W^*)$ satisfying (6) and being generically constructable from a ADHSDH instance are the tuples from that instance, which concludes our proof of generic security of ADHSDH. □

D Proofs

D.1 Proof of Theorem 2

Consider an adversary that after receiving parameters (G, F, K, T, H) and public key (X, Y) is allowed to ask for $q-1$ signatures $(A_i, C_i, D_i, R_i, S_i)$ on messages $(M_i, N_i) \in \mathcal{DH}$ of its choice and outputs $(M, N) \in \mathcal{DH}$ and a valid signature (A, C, D, R, S) on it, such that either (M, N) was never queried, or $(M, N) = (M_i, N_i)$ and $(A, C, D, R, S) \neq (A_i, C_i, D_i, R_i, S_i)$. We distinguish two kinds of forgers: An adversary is called of Type I if its output satisfies the following

$$\forall\, 1 \leq i \leq q-1 : \left[e(T, S{\cdot}S_i^{-1}) \neq e(M_i{\cdot}M^{-1}, H) \;\vee\; C \neq C_i \right] ; \tag{16}$$

otherwise it is called of Type II. We will use the first type to break q-ADHSDH and the second type to break AWFCDH.

Type I Let $\big(G, F, K, X, H, Y, (A_i, C_i, V_i, D_i, W_i)_{i=1}^{q-1}\big)$ be a q-ADHSDH challenge. It satisfies thus

$$e(A_i, Y{\cdot}D_i) = e(K{\cdot}V_i, H) \qquad e(C_i, H) = e(F, D_i) \qquad e(V_i, H) = e(G, W_i) \tag{17}$$

Let $\mathcal{A}$ be a forger of Type I. Choose $t \leftarrow \mathbb{Z}_p$ and give parameters $(G, F, K, T := G^t, H)$ and the public key (X, Y) to $\mathcal{A}$. The i-th query for $(M_i, N_i) \in \mathcal{DH}$ is answered as

$$\big(A_i, C_i, D_i, R_i := (V_i{\cdot}M_i^{-1})^{\frac{1}{t}}, S_i = (W_i{\cdot}N_i^{-1})^{\frac{1}{t}}\big).$$

It is easily verified that it satisfies (5); and it is correctly distributed since v_i is uniformly random in the ADHSDH instance. If the adverseray produces a valid signature/message pair $((A, C, D, R, S), (M, N))$ then by the last 2 equations of (5), there exist c, r s.t. $C = F^c, D = H^c, R = G^r, S = H^r$, and

$$e(A, Y{\cdot}D) = e(K{\cdot}M, H)\, e(T, S)\ . \tag{18}$$

The tuple $(A, C, D, V := R^t \cdot M, W := S^t \cdot N)$ satisfies (2), since (C, D) and (V, W) are Diffie-Hellman pairs and $e(K \cdot V, H) = e(K \cdot (G^r)^t \cdot M, H) = e(K \cdot M, H)\, e(T, S) \overset{(18)}{=} e(A, Y \cdot D)$. Moreover, it is a solution for the ADHSDH instance, since it is a *new* tuple: assume that for some i we have $C = C_i$ and $W = W_i$, that is $S^t \cdot N = S_i^t \cdot N_i$. Since $(M, N), (M_i, N_i) \in \mathcal{DH}$, we have $e(T, S)\, e(M, H) = e(T, S)\, e(G, N) = e(G, S^t \cdot N) = e(G, S_i^t \cdot N_i) = e(T, S_i)\, e(G, N_i) = e(T, S_i)\, e(M_i, H)$. We have thus $e(T, S \cdot S_i^{-1}) = e(M_i \cdot M^{-1}, H)$ and $C = C_i$ which contradicts (16) and thus the fact that $\mathcal{A}$ is of Type I.

Type II Let $(G, H, T = G^t)$ be an AWFCDH instance; let $\mathcal{A}$ be a forger of Type II. Pick $F, K \leftarrow \mathbb{G}_1$ and $x \leftarrow \mathbb{Z}_p$, set $X := G^x$, $Y := H^x$ and give the adversary parameters (G, F, K, T, H) and public key (X, Y). Answer a signing query on $(M_i, N_i) \in \mathcal{DH}$ by returning a signature $(A_i, C_i, D_i, R_i, S_i)$ produced by $\mathsf{Sign_A}(x, \cdot)$. Suppose $\mathcal{A}$ returns $((A, C, D, R, S), (M, N))$ satisfying (5) s.t. $e(T, S \cdot S_i^{-1}) = e(M_i \cdot M^{-1}, H)$ and $C = C_i$ for some i. Then $(M^* := M_i \cdot M^{-1}, N^* := N_i \cdot N^{-1}, R^* := R \cdot R_i^{-1}, S^* := S \cdot S_i^{-1})$ is a AWFCDH solution: (S^*, M^*), (M^*, N^*) and (R^*, S^*) satisfy the respective equations in (3), and (M^*, N^*, R^*, S^*) is non-trivial: if $M^* = 1 = R^*$ then $M = M_i$ and $R = R_i$; since moreover $C = C_i$ and since the values M, C and R completely determine a message/signature pair, this means that $\mathcal{A}$ returned a message and a signature that it obtained from a query for this message, which means that $\mathcal{A}$ did not break strong unforgeability.

□

D.2 Proof of Theorem 3

The protocol is correct: The signer sends $A = (K \cdot T^r \cdot U)^{\frac{1}{x+c}} = (K \cdot T^{r+\rho} \cdot M)^{\frac{1}{x+c}}$, $C = F^c$, $D = H^c$, $R' = G^r$, $S' = H^r$ and the user sets $R := R' \cdot P = G^{r+\rho}$ and $S := S' \cdot Q = H^{r+\rho}$, which makes it a valid signature on (M, N).

Blindness: *If we are given two messages from the adversary and run* Obtain *twice for these messages (in random order) with the adversary, and then give the two resulting signature/message pairs, then the adversary cannot relate them to their issuings.*

We modify the security game by setting setting *ck* $\leftarrow$ $\mathsf{SmSetup}$ (leading to perfectly WI commitments and proofs). This modification is indistinguishable by DLIN or SXDH (depending on the used Groth-Sahai instantiation). A signature/message pair $\big((\vec{\mathbf{c}}, \pi), (M, N)\big)$ that the adversary gets in the end now perfectly hides the signature, since the commitments are under *ck*. Moreover, for every pair $(M', N') \in \mathcal{DH}$, there exists $\rho' \in \mathbb{Z}_p$ s.t. $U = T^{\rho'} \cdot M'$. By witness indistinguishability of Groth-Sahai proofs, every such tuple $(M', N', P' := G^{\rho'}, Q' := H^{\rho'})$ leads to the same distribution of $(\mathbf{c}_M, \mathbf{c}_N, \mathbf{c}_P, \mathbf{c}_Q, \phi_M, \phi_P, \phi_U)$. The adversary's view after the first round of the protocol is thus independent of (M, N).

Unforgeability: *After running the protocol $q - 1$ times with an honest signer, no adversary can output q different messages and valid blind signatures on them.*

We reduce unforgeability to the security of the signature scheme $\mathbf{Sig}_\mathrm{A}$, which follows from ADHSDH and AWFCDH by Theorem 2. Given parameters $pp_A = (G, F, K, T, H)$ and a public key (X, Y) for $\mathbf{Sig}_\mathrm{A}$, we first run $(ck, ek) \leftarrow \mathsf{ExSetup}$ and give the adversary $pp = (pp_A, ck)$. We then run the protocol (simulating the signer) with adversary $\mathcal{A}$ as follows. Whenever $\mathcal{A}$ sends $(\mathbf{c}_M, \mathbf{c}_N, \phi_M, \mathbf{c}_P, \mathbf{c}_Q, \phi_P, U, \phi_U)$, we use *ek* to extract (M, N, P, Q). Soundness of the proofs ϕ_M, ϕ_P, ϕ_U ensures that there exist $m, \rho \in \mathbb{Z}_p$ s.t. $M = G^m, N = H^m, P = G^\rho, Q = H^\rho$ and $U = T^\rho \cdot M$. We query our $\mathsf{Sign_A}$ oracle for a signature on (M, N). On receiving (A, C, D, R, S), we give the adversary $(A, C, D, R' := R \cdot P^{-1}, S' := S \cdot Q^{-1})$. This perfectly simulates Issue: let c and $\hat{r}$ be such that $C = F^c$ and $R = G^{\hat{r}}$; then $A = (K \cdot T^{\hat{r}} \cdot M)^{\frac{1}{x+c}} = (K \cdot T^{\hat{r}-\rho} \cdot U)^{\frac{1}{x+c}}$, $R' = G^{\hat{r}-\rho}$ and $S' = H^{\hat{r}-\rho}$, which corresponds to a real Issue reply using randomness c and $r := \hat{r} - \rho$.

The adversary wins the game if after $q - 1$ issuings, it outputs q blind signatures on different messages. We extract the $\mathbf{Sig}_\mathrm{A}$ signature on a message which we did not query to our own oracle. By soundness of GS proofs, this is a valid signature and can thus be returned as a forgery. □

D.3 Proof of Theorem 4

Consider an adversary $\mathcal{A}$ making q queries on messages $(M_1^{(i)}, M_2^{(i)})$ for $1 \leq i \leq q$ and outputting a new message (M_1^*, M_2^*) and a valid signature $\sigma^* = (vk_0^*, \tau^*, \sigma_1^*, \sigma_2^*, \sigma_3^*)$ on it. Let vk be a challenge for $\mathbf{Sig}$. We call adversaries Type 1 if $vk_0^* \neq vk_0^{(i)}$ for all $1 \leq i \leq q$. Type 1 forgeries are reduced by giving vk to the adversary as the challenge key and answering signing queries by choosing $(vk_0, sk_0) \leftarrow \mathsf{KeyGen}$, querying vk_0 to the signing oracle and using sk_0 to complete a $\mathbf{Sig}'$ signature. From the adversary's output we can return (vk_0^*, τ^*) as a forgery under vk.

Forgeries of Type 2, i.e., for some i we have $vk_0^* = vk_0^{(i)}$, are handled as follows. Let vk be a $\mathbf{Sig}$ challenge key. We choose $(vk', sk') \leftarrow \mathsf{KeyGen}$ and $i^* \leftarrow \{1, \ldots, q\}$ and give vk' to the adversary. Knowing sk', we answer the signing queries by running $\mathsf{Sign}'_{sk'}$ —except for the i^*-th query: being queried message (M_1, M_2), we set $vk_0^{(i^*)} := vk$, and use our signing oracle on messages M_1, $M_1 \cdot M_2$ and $M_1 \cdot M_2^3$ to simulate a $\mathbf{Sig}'$ signature. We show that if we guessed correctly ($i^* = i$) then from $\mathcal{A}$'s output we can extract a forgery under vk.

In particular, we show that any valid forgery σ^* with $vk_0^* = vk$ on (M_1^*, M_2^*) must contain a signature on a message we have not queried to our oracle. We proceed by case distinction: if σ_1^*, the signature on M_1^*, is a signature on a message we have queried our oracle then M_1^* is either M_1, $M_1 \cdot M_2$ or $M_1 \cdot M_3^3$.

- $M_1^* = M_1$. In this case, if the message of σ_2^* (i.e., $M_1^* \cdot M_2^*$) has also been queried, then either
  - $M_1^* \cdot M_2^* = M_1$, thus $M_2^* = 1$ which is not in the message space and thus the adversary did not win, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2$, thus $M_2^* = M_2$, thus the adversary did not return a valid forgery since $(M_1^*, M_2^*) = (M_1, M_2)$, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2^3$, thus $M_2^* = M_2^3$, thus σ_3^* is a valid signature on $M_1^* \cdot (M_2^*)^3 = M_1 \cdot M_2^9$, which we have not queried to our oracle, since $M_2 \neq 1$ (see below).
- $M_1^* = M_1 \cdot M_2$. Again, if we queried $M_1^* \cdot M_2^*$, then either
  - $M_1^* \cdot M_2^* = M_1$, thus $M_2^* = M_2^{-1}$, thus σ_3^* is a valid signature on $M_1^* \cdot (M_2^*)^3 = M_1 \cdot M_2^{-2}$, which we have not queried to our oracle, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2$, thus $M_2^* = 1$, which is not a valid message, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2^3$, thus $M_2^* = M_2^2$, thus σ_3^* is a valid signature on $M_1^* \cdot (M_2^*)^3 = M_1 \cdot M_2^7$, which we have not queried to our oracle.
- $M_1^* = M_1 \cdot M_2^3$. Again, if we queried $M_1^* \cdot M_2^*$, then either
  - $M_1^* \cdot M_2^* = M_1$, thus $M_2^* = M_2^{-3}$, thus σ_3^* is a valid signature on $M_1^* \cdot (M_2^*)^3 = M_1 \cdot M_2^{-6}$, which we have not queried to our oracle, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2$, thus $M_2^* = M_2^{-2}$, thus σ_3^* is a valid signature on $M_1^* \cdot (M_2^*)^3 = M_1 \cdot M_2^{-3}$, which we have not queried to our oracle, or
  - $M_1^* \cdot M_2^* = M_1 \cdot M_2^3$, thus $M_2^* = 1$, which is not a valid message.

Note that all the above messages were indeed not queried to the oracle: they are all of the form $M_1 \cdot M_2^i$ with $i \notin \{0, 1, 3\}$, whereas the messages queried to the $\mathbf{Sig}$ oracle are of the form $M_1 \cdot M_2^j$ with $j \in \{0, 1, 3\}$. If we had $M_1 \cdot M_2^i = M_1 \cdot M_2^j$ for any of the above values of i and j, we would have $M_2^{i-j} = 1$ for $i \neq j$ and thus $M_2 = 1$, which would not have been accepted in a signing request.

We thus showed that any valid message/signature pair the adversary returns contains a forgery. $\square$

D.4 Proof of Theorem 5

Let q be the maximal number of the adversary's signing queries. Let $\vec{M}^{(i)} := (M_1^{(i)}, \ldots, M_{n_i}^{(i)})$ denote the adversary's i-th signing query, let $\sigma^{(i)} := (vk_0^{(i)}, \sigma_0^{(i)}, \ldots, \sigma_{n_i}^{(i)})$ denote the replies, and let the adversary's final output be $\big((M_1^*, \ldots, M_{n^*}^*),\ (vk_0^*, \sigma_0^*, \ldots, \sigma_{n^*}^*)\big)$. Let vk be a challenge for $\mathbf{Sig}$. We distinguish two types of forgers and show how to reduce them to EUF-CMA of $\mathbf{Sig}$.

1. $\forall i : \left(vk_0^* \neq vk_0^{(i)} \vee n^* \neq n_i\right)$. Give vk to the adversary and answer the i-th signing query by choosing $(vk_0^{(i)}, sk_0^{(i)})$, querying $(vk_0^{(i)}, \mathsf{Inj}(n_i))$ to the Sign-oracle and using $sk_0^{(i)}$ to sign $(M_j^{(i)}, \mathsf{Inj}(j))$ for all j. If σ^* is of Type 1, then $((vk_0^*, \mathsf{Inj}(n^*)), \sigma_0^*)$ is a forgery under vk.
2. $\exists i : \left(vk_0^* = vk_0^{(i)} \wedge n^* = n_i\right)$. Choose $i^* \leftarrow \{1, \ldots, q\}$, produce $(vk', sk') \leftarrow \mathsf{KeyGen}(1^k)$ and give the adversary vk' as challenge. Answer all queries as in the protocol, except for the i^*-th query: set $vk_0^{(i^*)} := vk$ and query signatures on $(M_j^{(i^*)}, \mathsf{Inj}(j))$ for all j to the Sign-oracle and complete the signature using sk'. Suppose σ^* is of Type 2 and we guessed correctly ($i^* = i$). Since $(M_1^*, \ldots, M_{n_i}^*)$ is a valid forgery, for some $1 \leq j \leq n_i$ we have $M_j^* \neq M_j^{(i)}$. Thus $\left((M_j^*, \mathsf{Inj}(j)), \sigma_j^*\right)$ is a valid forgery under vk for a message we did not query.

□

E An Anonymous Proxy Signature Scheme with Delegator Anonymity

We formally describe an instantiation of anonymous proxy signatures with delegator anonymity as discussed in Remark 1 (2).

E.1 Building Blocks

To instantiate APS with delegator anonymity, we will use the following building blocks that were introduced in Sections 2.2 and 5.1, respectively. We can instantiate them over asymmetric bilinear groups in which SXDH holds, or over symmetric groups in which DLIN is hard.

- Commitments: $\mathsf{ExSetup}(\cdot)$ takes as input the asymmetric (or symmetric) bilinear group and outputs a commitment key $ck \in \mathbb{G}_1^3 \times \mathbb{G}_2^3$ (or $ck \in \mathbb{G}^5$) and an extraction key $ek \in \mathbb{Z}_p^2$. On inputs a commitment key, a group element, and randomness from $\mathcal{R} := \mathbb{Z}_p^2$ (or $\mathcal{R} := \mathbb{Z}_p^3$), $\mathsf{Com}(\cdot,\cdot,\cdot)$ outputs a commitment in consisting of 2 (or 3) group elements. $\mathsf{RdCom}(\cdot,\cdot,\cdot)$ takes a commitment key, a commitment and fresh randomness, and outputs a randomized commitment to the same value; $\mathsf{Extr}(\cdot,\cdot)$ outputs the committed value on input ek and a commitment.
- Groth-Sahai proofs: $\mathsf{Prove}(\cdot,\cdot,\cdot)$ produces a proof in $\mathbb{G}_1^4 \times \mathbb{G}_2^4$ (for the DLIN instantiation, the proofs are in $\mathbb{G}^3$ for linear equations, and in $\mathbb{G}^9$ for general equations) on inputs a commitment key, the description of a PPE and a vector of pairs of committed values / randomness. On inputs the commitment key, the equation description, a vector of commitments and a proof, $\mathsf{Verify}(\cdot,\cdot,\cdot)$ outputs a value in $\{0,1\}$. The algorithm $\mathsf{RdProof}(\cdot,\cdot,\cdot)$ takes as inputs a commitment key, an equation description, a vector of pairs of commitments and fresh randomness, and a proof; and outputs a new proof adapted to the randomizations of the commitments.
- Automorphic signatures: let **Sig** $= (\mathsf{Setup}_{\mathrm{sig}}, \mathsf{KeyGen}_{\mathrm{sig}}, \mathsf{Sign}_{\mathrm{sig}}, \mathsf{Verify}_{\mathrm{sig}})$ denote Scheme 2 in Sect. 5.1. For $vk = (X, Y)$, $m = (M, N)$ and $\sigma = (A, C, D, R, S)$, let $E_{\mathrm{sig}}(vk, m, \sigma)$ denote the equations in (5) and the following two equations: $e(X, H) = e(G, Y)$ and $e(M, H) = e(G, N)$. (We implicitly assume fixed parameters (G, F, H, K, T).) Analogously, let $E'_{\mathrm{sig}}(vk, (m_1, m_2), \sigma)$ be the verification relations for a signature on a message consisting of 2 $\mathcal{DH}$-pairs from Definition 2.

E.2 Instantiation

$\mathsf{Setup}_{\mathrm{aps}}(1^\lambda)$

- Generate a bilinear group $\mathcal{BG}$ for security parameter λ.
- Run $\mathsf{Setup}_{\mathrm{sig}}(\mathcal{BG})$ to get parameters pp_{sig}.
- Run $\mathsf{KeyGen}_{\mathrm{sig}}(pp_{\mathrm{sig}})$ to produce a key pair (ipk, ik). Return the public parameters $pp := (pp_{\mathrm{sig}}, ipk)$ and the issuer's key ik.

$\mathsf{Reg}_{\text{aps}}$ is a protocol between a new user, the issuer and the user's opener.

- The user runs $(vk, sk) \leftarrow \mathsf{KeyGen}_{\text{sig}}(pp_{\text{sig}})$ and produces a signature (possibly via an external PKI[12]) σ_{pki} on vk. She sends $(vk, \sigma_{\text{pki}})$ to the issuer and vk to the opener.
- The issuer checks σ_{pki}, produces $cert \leftarrow \mathsf{Sign}_{\text{sig}}(ik, vk)$, sends $cert$ to the user, and writes $(vk, \sigma_{\text{pki}})$ to its register.
- The opener runs $(ck, ek) \leftarrow \mathsf{ExSetup}(\mathcal{BG})$ and sends ck to the user. It sets the opening key as $ok := (vk, ck, ek)$.
- The user sets his public key $upk = (vk, ck)$ and his secret key $usk = (upk, sk, cert)$.

$\mathsf{Delgt}_{\text{aps}}(usk, [\boldsymbol{warr}], upk)$

- Set $k = 0$ if this is an original delegation (i.e., there is no optional argument $\boldsymbol{warr}$), otherwise let k be s.t. this is the k-th intermediate delegation. Parse usk as $\big((vk_k, ck_k), sk_k, cert_k\big)$ and upk as (vk_{k+1}, ck_{k+1}).
- If $k = 0$ then choose an identifier id, compute $warr_{0\to1} \leftarrow \mathsf{Sign}_{\text{sig}}\big(sk_0, (\mathsf{Hash}(id \,\|\, 1), vk_1)\big)$ and return $(ck_0, id, vk_0, warr_{0\to1})$.
- If $k = 1$ then do the following:
  - Parse $\boldsymbol{warr}$ as $(ck, id, vk_0, warr_{0\to1})$.
  - Compute $warr_{1\to2} \leftarrow \mathsf{Sign}_{\text{sig}}\big(sk_1, (\mathsf{Hash}(id \,\|\, 2), vk_2)\big)$.
  - Choose $\rho^{(v)}, \rho^{(c)}, \rho_1^{(w)}, \rho_2^{(w)} \leftarrow \mathcal{R}$ and compute the following commitments and proofs:
    $\mathbf{c}_{warr_{0\to1}} \leftarrow \mathsf{Com}(ck, warr_{0\to1}, \rho_1^{(w)})$, $\mathbf{c}_{vk_1} \leftarrow \mathsf{Com}(ck, vk_1, \rho^{(v)})$, $\mathbf{c}_{cert_1} \leftarrow \mathsf{Com}(ck, cert_1, \rho^{(c)})$,
    $\mathbf{c}_{warr_{1\to2}} \leftarrow \mathsf{Com}(ck, warr_{1\to2}, \rho_2^{(w)})$, $\mathbf{c}_{vk_2} \leftarrow \mathsf{Com}(ck, vk_2, 0)$, (Footnote[13])
    $\phi_{cert_1} \leftarrow \mathsf{Prove}\big(ck, E_{\text{sig}}(ipk, \cdot, \cdot), ((vk_1, \rho_1^{(v)}), (cert_1, \rho^{(c)})\big)$,
    $\phi_{warr_{0\to1}} \leftarrow \mathsf{Prove}\big(ck, E'_{\text{sig}}\big(vk_0, (\mathsf{Hash}(id \,\|\, 1), \cdot), \cdot\big), \big((vk_1, \rho^{(v)}), (warr_{0\to1}, \rho_1^{(w)})\big)\big)$,
    $\phi_{warr_{1\to2}} \leftarrow \mathsf{Prove}\big(ck, E'_{\text{sig}}\big(\cdot, (\mathsf{Hash}(id \,\|\, 2), \cdot), \cdot\big), \big((vk_1, \rho^{(v)}), (vk_2, 0), (warr_{1\to2}, \rho_2^{(w)})\big)\big)$.
  - Return $\boldsymbol{warr}' := \big(ck, id, vk_0, (\mathbf{c}_{warr_{0\to1}}, \phi_{warr_{0\to1}}, \mathbf{c}_{vk_1}, \mathbf{c}_{cert_1}, \phi_{cert_1}), \mathbf{c}_{warr_{1\to2}}, \phi_{warr_{1\to2}}, \mathbf{c}_{vk_2}\big)$.

- Otherwise, do the following:
  - Parse $\boldsymbol{warr}$ as $\big(ck, id, vk_0, (\mathbf{c}_{warr_{(i-1)\to i}}, \phi_{warr_{(i-1)\to i}}, \mathbf{c}_{vk_i}, \mathbf{c}_{cert_i}, \phi_{cert_i})_{i=1}^{k-1},$
    $\mathbf{c}_{warr_{(k-1)\to k}}, \phi_{warr_{(k-1)\to k}}, \mathbf{c}_{vk_k}\big)$.
  - Compute $warr_{k\to(k+1)} \leftarrow \mathsf{Sign}_{\text{sig}}\big(sk_k, (\mathsf{Hash}(id \,\|\, k+1), vk_{k+1})\big)$.
  - Choose randomness for commitments and randomization: Pick $\rho_i^{(v)}, \rho_i^{(c)}, \rho_i^{(w)} \leftarrow \mathcal{R}$ for $1 \le i \le k$ and $\rho_{k+1}^{(w)} \leftarrow \mathcal{R}$.
  - Randomize the commitments and adapt the proofs in $\boldsymbol{warr}$:
    For $1 \le i \le k$: $\mathbf{c}'_{warr_{(i-1)\to i}} \leftarrow \mathsf{RdCom}(ck, \mathbf{c}_{warr_{(i-1)\to i}}, \rho_i^{(w)})$, $\mathbf{c}'_{vk_i} \leftarrow \mathsf{RdCom}(ck, \mathbf{c}_{vk_i}, \rho_i^{(v)})$,
    $\phi'_{warr_{(i-1)\to i}} \leftarrow \mathsf{RdProof}\big(ck, E'_{\text{sig}}\big(\cdot, (\mathsf{Hash}(id \,\|\, i), \cdot), \cdot\big),$
    $\big((\mathbf{c}_{vk_{i-1}}, \rho_{i-1}^{(v)}), (\mathbf{c}_{vk_i}, \rho_i^{(v)}), (\mathbf{c}_{warr_{(i-1)\to i}}, \rho_i^{(w)})\big), \phi_{warr_{(i-1)\to i}}\big)$.
    For $1 \le i \le k-1$: $\mathbf{c}'_{cert_i} \leftarrow \mathsf{RdCom}(ck, \mathbf{c}_{cert_i}, \rho_i^{(c)})$,
    $\phi'_{cert_i} \leftarrow \mathsf{RdProof}\big(ck, E_{\text{sig}}(ipk, \cdot, \cdot), \big((\mathbf{c}_{vk_i}, \rho_i^{(v)}), (\mathbf{c}_{cert_i}, \rho_i^{(c)})\big), \phi_{cert_i}\big)$.
  - Compute the following commitments and proofs:
    $\mathbf{c}_{cert_k} \leftarrow \mathsf{Com}(ck, cert_k, \rho_k^{(c)})$, $\mathbf{c}_{warr_{k\to(k+1)}} \leftarrow \mathsf{Com}(ck, warr_{k\to(k+1)}, \rho_{k+1}^{(w)})$,
    $\mathbf{c}_{vk_{k+1}} \leftarrow \mathsf{Com}(ck, vk_{k+1}, 0)$,
    $\phi_{cert_k} \leftarrow \mathsf{Prove}\big(ck, E_{\text{sig}}(ipk, \cdot, \cdot), ((vk_k, \rho_k^{(v)}), (cert_k, \rho_k^{(c)})\big)$

[12] To achieve strong notions of non-frameability, it is necessary to assume an external PKI infrastructure (cf. [BSZ05])

[13] $\mathbf{c}_{vk_2}$ is thus a *trivial* commitment.

$$\phi_{warr_{k\to(k+1)}} \leftarrow \mathsf{Prove}\big(ck, E'_{\mathrm{sig}}\big(\cdot, (\mathsf{Hash}(id\,\|\,k+1), \cdot), \cdot\big), \\ \big((vk_k, \rho^{(v)}_k), (vk_{k+1}, 0), (warr_{k\to(k+1)}, \rho^{(w)}_{k+1})\big)\big).$$

– Return $\boldsymbol{warr}' = \big(ck, id, vk_0, (\mathbf{c}'_{warr_{(i-1)\to i}}, \phi'_{warr_{(i-1)\to i}}, \mathbf{c}'_{vk_i}, \mathbf{c}'_{cert_i}, \phi'_{cert_i})_{i=1}^{k-1}, \\ (\mathbf{c}'_{warr_{(k-1)\to k}}, \phi'_{warr_{(k-1)\to k}}, \mathbf{c}'_{vk_k}, \mathbf{c}_{cert_k}, \phi_{cert_k}), \mathbf{c}_{warr_{k\to(k+1)}}, \phi_{warr_{k\to(k+1)}}, \mathbf{c}_{vk_{k+1}}\big).$

$\mathsf{PSign}_{\mathrm{aps}}(usk, \boldsymbol{warr}, msg)$ Signing is done similarly to delegation, where the message now plays the rôle of vk_{k+1}. Since the message is public, it is not committed to; moreover, ck and vk_0 are part of the verification key and need thus not be included in the signature (see (19)).

$\mathsf{Verify}_{\mathrm{aps}}(upk, msg, \Sigma)$

- Parse upk as (vk_0, ck) and parse the signature Σ as

$$\big(id, (\mathbf{c}_{warr_{(i-1)\to i}}, \phi_{warr_{(i-1)\to i}}, \mathbf{c}_{vk_i}, \mathbf{c}_{cert_i}, \phi_{cert_i})_{i=1}^{k}, \mathbf{c}_{sig}, \phi_{sig}\big) \ . \tag{19}$$

- Return 1 if all of the following return 1, otherwise return 0.
  – $\mathsf{Verify}\big(ck, E_{\mathrm{sig}}(ipk, \cdot, \cdot), (\mathbf{c}_{vk_i}, \mathbf{c}_{cert_i}), \phi_{cert_i}\big)$, for $1 \le i \le k$;
  – $\mathsf{Verify}\big(ck, E'_{\mathrm{sig}}(vk_0, (\mathsf{Hash}(id\,\|\,i), \cdot), \cdot), (\mathbf{c}_{vk_1}, \mathbf{c}_{warr_{0\to1}}), \phi_{warr_{0\to1}}\big)$;
  – $\mathsf{Verify}\big(ck, E'_{\mathrm{sig}}(\cdot, (\mathsf{Hash}(id\,\|\,i), \cdot), \cdot), (\mathbf{c}_{vk_{i-1}}, \mathbf{c}_{vk_i}, \mathbf{c}_{warr_{(i-1)\to i}}), \phi_{warr_{(i-1)\to i}}\big)$, for $2 \le i \le k$;
  – $\mathsf{Verify}\big(ck, E'_{\mathrm{sig}}(\cdot, (\mathsf{Hash}(id\,\|\,k+1), msg), \cdot), (\mathbf{c}_{vk_k}, \mathbf{c}_{sig}), \phi_{sig}\big)$.

$\mathsf{Open}_{\mathrm{aps}}(ok, msg, \Sigma)$ Parse ok as (vk, ck, ek), parse Σ as (19) and check if it is valid. If so then set $vk_i \leftarrow \mathsf{Extr}(ek, \mathbf{c}_{vk_i})$ and $warr_{(i-1)\to i} \leftarrow \mathsf{Extr}(ek, \mathbf{c}_{warr_{(i-1)\to i}})$ for $1 \le i \le k$, and $sig \leftarrow \mathsf{Extr}(ek, \mathbf{c}_{sig})$. Return $\big((vk_1, \ldots, vk_k), (warr_{0\to1}, \ldots, warr_{(k-1)\to k}, sig)\big)$, where the second component is the proof.