Published in S. Dolev, Ed., *2014 ACM Symposium on Principles of Distributed Computing (PODC 2014)*, pp. 303-312, ACM Press, 2014.

Born and Raised Distributively: Fully Distributed Non-Interactive Adaptively-Secure Threshold Signatures with Short Shares

Benoît Libert*
ENS Lyon, LIP
Lyon, France
benoit.libert@ens-lyon.fr

Marc Joye
Technicolor
Palo Alto, CA, USA
marc.joye@technicolor.com

Moti Yung
Google Inc. and Columbia U.
New York, NY, USA
moti@cs.columbia.edu

ABSTRACT

Threshold cryptography is a fundamental distributed computational paradigm for enhancing the availability and the security of cryptographic public-key schemes. It does it by dividing private keys into n shares handed out to distinct servers. In threshold signature schemes, a set of at least $t + 1 \leq n$ servers is needed to produce a valid digital signature. Availability is assured by the fact that any subset of $t + 1$ servers can produce a signature when authorized. At the same time, the scheme should remain robust (in the fault tolerance sense) and unforgeable (cryptographically) against up to t corrupted servers; *i.e.*, it adds quorum control to traditional cryptographic services and introduces redundancy. Originally, most practical threshold signatures have a number of demerits: They have been analyzed in a static corruption model (where the set of corrupted servers is fixed at the very beginning of the attack), they require interaction, they assume a trusted dealer in the key generation phase (so that the system is not fully distributed), or they suffer from certain overheads in terms of storage (large share sizes). In this paper, we construct practical *fully distributed* (the private key is born distributed), non-interactive schemes —where the servers can compute their partial signatures without communication with other servers— with adaptive security (*i.e.*, the adversary corrupts servers dynamically based on its full view of the history of the system). Our schemes are very efficient in terms of computation, communication, and scalable storage (with private key shares of size $O(1)$, where certain solutions incur $O(n)$ storage costs at each server). Unlike other adaptively secure schemes, our schemes are erasure-free (reliable erasure is a hard to assure and hard to administer property in actual systems). To the best of our knowledge, such a fully distributed highly constrained scheme has been an open problem in the area. In particular, and of special interest, is the fact that Pedersen's traditional distributed key generation (DKG) protocol can be safely employed in the initial key generation phase when the system is born —although it is well-known not to ensure uniformly distributed public keys. An advantage of this is that this protocol only takes one round optimistically (in the absence of faulty player).

Categories and Subject Descriptors

E.3 [**Data**]: Data encryption; H.4 [**Information Systems Applications**]: Miscellaneous; C.2.4 [**Computer Systems Organization**]: Distributed systems

General Terms

Theory, Security

Keywords

Threshold signature schemes, fully distributed systems, non-interactivity, adaptive security, efficiency, availability, fault tolerance, distributed key generation, erasure-free schemes

1. INTRODUCTION

Threshold cryptography [29, 30, 15, 28] is a paradigm where cryptographic keys are divided into $n > 1$ shares to be stored by distinct servers, which increases the system's availability and resilience to failures. In (t, n)-threshold cryptosystems, private-key operations require the cooperation of at least $t+1$ out of n servers (any subset is good). By doing so, the system remains secure against adversaries that break into up to t servers. (The mechanism can be viewed as extending Shamir's secret sharing of one value [67] to sharing of the capability to apply cryptographic function efficiently). The public-key portion of the function (*e.g.*, signature verification key) does not change from its usual format.

Threshold primitives are widely used in distributed protocols. Threshold homomorphic encryption schemes are utilized in voting systems (see, *e.g.*, [22, 23]) and multiparty computation protocols [24]. Threshold signatures enhance the security of highly sensitive private keys, like those of certification authorities (*e.g.*, [18]). They can also serve as tools for distributed storage systems [48, 65]. RSA and Elgamal-type constructions have been at the core of many threshold protocols the last two decades (see, *e.g.*, [28, 39, 40, 47]). A *fully distributed* public-key system is one where the public (and the distributed private) key are jointly generated by the same servers which end up holding the private key's

*This research was done while this author was at Technicolor, France.

PODC'14, July 15–18, 2014, Paris, France.

http://dx.doi.org/10.1145/2611462.2611498.

shares (*e.g.*, via a threshold secret sharing [67]). Efficient distributed key generation (DKG) protocols were put forth for both RSA [12, 34, 33, 26] and discrete-logarithm-based systems [62, 41, 35, 16, 43].

Non-Interactive Threshold Signatures. For a long time, RSA-based threshold signatures have been the only solutions to enable non-interactive distributed signature generation. By "non-interactive", we mean that each server can compute its own partial signature without any online conversation with other servers: each server should send a single message to an entity, called *combiner*, which gathers the signature shares so as to obtain a full signature. Unlike threshold versions of Schnorr and DSA signatures [42, 39], threshold RSA signatures are well-suited to non-interactive signing protocols as they are deterministic. Hence, they do not require the servers to jointly generate a randomized signature component in a first round before starting a second round. Practical robust non-interactive threshold signatures were described by Shoup [68] under the RSA assumption and by Katz and Yung [50] assuming the hardness of factoring. Boldyreva [10] showed a threshold version of Boneh-Lynn-Shacham signatures [14], which provided an alternative non-interactive scheme with robustness and short signatures. The latter construction [10] was subsequently generalized by Wee [69]. These solutions are only known to resist static attacks, where the set of corrupted servers is chosen by the adversary at the very beginning of the attack, before even seeing the public key.

Adaptive Corruptions. More realistically than the static model, the adaptive corruption model allows adversaries to choose whom to corrupt at any time, based on their entire view so far. Adaptive adversaries are known to be strictly (see, *e.g.*, [25]) stronger. The first adaptively secure threshold signatures were independently described in 1999 by Canetti *et al.* [16] and by Frankel *et al.* [35, 36]. These constructions rely on a technique, called "single inconsistent player" (SIP), which inherently requires interaction. The SIP technique basically consists in converting a t-out-of-n secret sharing into an t-out-of-t secret sharing in such a way that, in the latter case, there is only one server whose internal state cannot be consistently revealed to the adversary. Since this player is chosen at random by the simulator among the n severs, it is only corrupted with probability less than $1/2$ and, upon this undesirable event, the simulator can simply rewind the adversary back to one of its previous states. After this backtracking operation, the simulator uses different random coins to simulate the view of the adversary, hoping that the inconsistent player will not be corrupted again (and the expected number of rewinding-s is bounded by 2).

Jarecki and Lysyanskaya [49] extended the SIP technique in order to eliminate the need for servers to reliably erase intermediate computation results. However, their adaptively secure version of the Canetti-Goldwasser threshold cryptosystem [17] requires a substantial amount of interaction at each private-key operation. The same holds for the adaptively secure threshold signatures of Lysyanskaya and Peikert [57] and the universally composable protocols of Abe and Fehr [1].

In 2006, Almansa, Damgård and Nielsen [4] showed a variant of Rabin's threshold RSA signatures [64] and proved them adaptively secure using the SIP technique and ideas from [35, 36]. Similar techniques were used in [70] to construct adaptively secure threshold Waters signatures [71]. While the SIP technique provides adaptively secure threshold signatures based on RSA or the Diffie-Hellman assumption, these fall short of minimizing the amount of interaction. The constructions of [35, 36] proceed by turning a (t, n) polynomial secret sharing into a (t, t) additive secret sharing by first selecting a pool of at least t participants. However, if only one of these fails to provide a valid contribution to the signing process, the whole protocol must be restarted from scratch. The protocol of Almansa *et al.* [4] is slightly different in that; like [64], it proceeds by sharing an RSA private key in an additive (n, n) fashion (*i.e.*, the private RSA exponent d is split into shares $d_1, \ldots, d_n$ such that $d = \sum_{i=1}^{n} d_i$). In turn, each additive share d_i is shared in a (t, n) fashion using a polynomial verifiable secret sharing and each share $d_{i,j}$ of d_i is distributed to another server j. This is done in such a way that, if one participant fails to provide a valid RSA signature share $H(M)^{d_i}$, the missing signature share can be re-constructed by running the reconstruction algorithm of the verifiable secret sharing scheme that was used to share d_i. The first drawback of this approach is that it is only non-interactive when all players are honest as a second round is needed to reconstruct missing multiplicative signature shares $H(M)^{d_i}$. Another disadvantage is that players have to store $\Theta(n)$ values, where n is the number of servers, as each player has to store a polynomial share of other players' additive share. Ideally, we would like a solution where each player only stores $O(1)$ elements, regardless of the number of players.

Recently, Libert and Yung [55, 56] gave several constructions of adaptively secure threshold encryption schemes with chosen-ciphertext security. They also suggested an adaptively secure and non-interactive threshold variant [55] of a signature scheme due to Lewko and Waters [51]. The use of bilinear maps in composite order groups makes the scheme of [55] expensive when it comes to verifying signatures: as discussed by Freeman [38], computing a bilinear map in composite order groups is at least 50 times slower than evaluating the same bilinear map in prime-order groups at the 80-bit security level (things can only get worse at higher security levels). The techniques of Lewko [52] can be used to adapt the construction of [55] to the setting of prime-order groups. In the resulting construction, each signature consists of 6 group elements. The use of asymmetric bilinear maps (see [19]) allows reducing the signature size to 4 group elements. Unfortunately, the techniques of [52, 19] assume a trusted dealer and, if implemented in a distributed manner, their key generation phase is likely to be communication-expensive (resorting to generic multiparty secure computations). In particular, they seem hardly compatible with a round-optimal DKG protocol. The reason is that [19] requires to generate public keys containing pairs of matrices of the form $g^{\mathbf{A}} \in \mathbb{G}^{n \times n}$ and $g^{\mathbf{A}^{-1}} \in \mathbb{G}^{n \times n}$, for some matrix $\mathbf{A} \in \mathbb{Z}_p^{n \times n}$, and it is not clear how these non-linear operations can be achieved in a round-optimal distributed manner (let alone with adaptive security). Finally, the solutions of [55] require reliable erasures due to the use of the dual system encryption technique [72, 51] and the existence of several distributions of partial signatures.

Our Contributions. We consider the problem of devising a fully distributed, non-interactive, robust, and adaptively

secure construction which is as efficient as the centralized schemes obtained from [55, 19] and does not rely on erasures. In particular, we want to retain private-key shares of $O(1)$ size, no matter how many players are involved in the protocol. Here, "fully distributed" implies that the public key is jointly generated by all players —so that no trusted dealer is needed— while guaranteeing the security of the scheme against an adaptive adversary. As mentioned above, we wish to avoid the costly and hard-to-control use of reliable erasures. This means that, whenever the adversary corrupts a player, it learns the entire history of that player.

At the same time, the distributed key generation phase should be as communication-efficient as possible. Ideally, a single communication round should be needed when the players follow the protocol. Finally, we would like to avoid interaction during the distributed signing process. To the best of our knowledge, no existing solution combines all the aforementioned highly constraining properties. We thus provide the first candidates.

Our constructions are derived from linearly homomorphic structure-preserving signatures (LHSPS). As defined by Abe *et al.* [2, 3], structure-preserving signatures (SPS) are signature schemes where messages and public keys live in an abelian group over which a bilinear map is efficiently computable. Recently, Libert *et al.* [54] considered SPS schemes with additive homomorphic properties: given signatures on linearly independent vectors of group elements, anyone can publicly compute a signature on any linear combination of these vectors. In order to sign a message $M \in \{0,1\}^*$ in a distributed manner, our idea is to hash M onto a vector of group elements which is signed using the LHSPS scheme of [54]. In the random oracle model, we prove that the resulting system is a secure digital signature even if the underlying LHSPS scheme satisfies a weak security definition. Since the LHSPS signing algorithm is deterministic and further presents certain homomorphic properties over the key space, the resulting signature is also amenable for non-interactive distributed signature generation. In the threshold setting, we take advantage of specific properties of the LHSPS scheme of [54] to prove that the scheme provides security against adaptive corruptions in the absence of secure erasures.

More surprisingly, we prove that the scheme remains adaptively secure if the public key is generated using Pedersen's DKG protocol [62]. The latter basically consists in having all players verifiably share a random value using Feldman's verifiable secret sharing (VSS) [32] before computing the shared secret as the sum of all well-behaved players' contributions. While very efficient (as only one round is needed in the absence of faulty players), this protocol is known [41] *not* to guarantee the uniformity of the resulting public key. Indeed, even a static adversary can bias the distribution by corrupting only two players. Nonetheless, the adversary does not have much control on the distribution of the public key and Pedersen's protocol can still be safely used in some applications, as noted by Gennaro *et al.* [42, 43]. For example, it was recently utilized by Cortier *et al.* [21] in the context of voting protocols. However, these safe uses of Pedersen's protocol were in the static corruption setting and our scheme turns out to be its first application in an adaptive corruption model. To our knowledge, it is also the first adaptively secure threshold signature where the DKG phase takes only one round when all players follow the specification.

As an extension of our first scheme, we describe a variant supporting signature aggregation: as suggested by Boneh *et al.* [13], a set of n signatures for distinct public keys $PK_1, \ldots, PK_s$ on messages $M_1, \ldots, M_s$ can be aggregated into a single signature σ which convinces a verifier that, for each i, M_i was signing by the private key underlying PK_i. In the threshold setting, this property allows for de-centralized certification authorities while enabling the compression of certification chains.

As a final contribution, we give a non-interactive adaptively secure threshold signature scheme in the standard model that retains all the useful properties (including the erasure-freeness) of our first realization. In particular, Pedersen's protocol can still be used in the key generation phase if a set of uniformly random common parameters —which can be shared by many public keys— is set up beforehand. As is natural for standard-model constructions, this scheme is somewhat less efficient than its random-oracle-based counterpart but it remains sufficiently efficient for practical applications.

Like Gennaro *et al.* [43] and Cortier *et al.* [21], we prove security via a direct reduction from the underlying number theoretic assumption instead of reducing the security of our schemes to that of their centralized version. We emphasize that our proof technique is different from those of [43, 21], where the reduction runs Pedersen's DKG protocol on behalf of honest players and embeds a discrete logarithm instance in the contribution of honest players to the public key, using a proper simulation of Feldman's verifiable secret sharing. In the adaptive corruption setting, this would at least require an adaptively secure variant of Feldman's VSS, such as [1], and thus extra communications. Instead, our reduction always faithfully runs the protocol on behalf of honest players, and thus always knows their internal state so as to perfectly answer corruption queries. Yet, we can use the adversary's forgery to break the underlying hardness assumption by taking advantage of key homomorphic properties of the scheme, which allow us to turn a forgery for the jointly generated public key —of possibly skewed distribution— into a forgery for some uniformly random key.

2. BACKGROUND

2.1 Definitions for Threshold Signatures

A non-interactive (t, n)-threshold signature scheme consists of a tuple $\Sigma = (\mathsf{Dist\text{-}Keygen}, \mathsf{Share\text{-}Sign}, \mathsf{Share\text{-}Verify}, \mathsf{Verify}, \mathsf{Combine})$ of efficient algorithms or protocols such that:

$\mathsf{Dist\text{-}Keygen}(\mathsf{params}, \lambda, t, n)$: This is an interactive protocol involving n players $P_1, \ldots, P_n$, which all take as input common public parameters params, a security parameter $\lambda \in \mathbb{N}$ as well as a pair of integers $t, n \in \mathsf{poly}(\lambda)$ such that $1 \leq t \leq n$. The outcome of the protocol is the generation of a public key PK, a vector of private key shares $\mathbf{SK} = (SK_1, \ldots, SK_n)$ where P_i only obtains SK_i for each $i \in \{1, \ldots, n\}$, and a public vector of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_n)$.

$\mathsf{Share\text{-}Sign}(SK_i, M)$: is a possibly randomized algorithm that takes in a message M and a private key share SK_i. It outputs a signature share σ_i.

$\mathsf{Share\text{-}Verify}(PK, \mathbf{VK}, M, (i, \sigma_i))$: is a deterministic algorithm that takes as input a message M, the public key PK,

the verification key $\mathbf{VK}$ and a pair (i, σ_i) consisting of an index $i \in \{1, \ldots, n\}$ and signature share σ_i. It outputs 1 or 0 depending on whether σ_i is deemed as a valid signature share or not.

$\mathsf{Combine}(PK, \mathbf{VK}, M, \{(i, \sigma_i)\}_{i \in S})$: takes as input a public key PK, a message M and a subset $S \subset \{1, \ldots, n\}$ of size $|S| = t + 1$ with pairs $\{(i, \sigma_i)\}_{i \in S}$ such that $i \in \{1, \ldots, n\}$ and σ_i is a signature share. This algorithm outputs either a full signature σ or $\perp$ if $\{(i, \sigma_i)\}_{i \in S}$ contains ill-formed partial signatures.

$\mathsf{Verify}(PK, M, \sigma)$: is a deterministic algorithm that takes as input a message M, the public key PK and a signature σ. It outputs 1 or 0 depending on whether σ is deemed valid share or not.

We shall use the same communication model as in, *e.g.*, [41, 42, 43], which is partially synchronous. Namely, communications proceed in synchronized rounds and sent messages are always received within some time bound in the same round. All players have access to a public broadcast channel, which the adversary can use as a sender and a receiver. However, the adversary cannot modify messages sent over this channel, nor prevent their delivery. In addition, we assume private and authenticated channels between all pairs of players.

In the adaptive corruption setting, the security of non-interactive threshold signatures can be defined as follows.

DEFINITION 1. *A non-interactive threshold signature scheme Σ is adaptively secure against chosen-message attacks if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the game hereunder. At any time, we denote by $\mathcal{C} \subset \{1, \ldots, n\}$ and $\mathcal{G} := \{1, \ldots, n\} \setminus \mathcal{C}$ the dynamically evolving subsets of corrupted and honest players, respectively. Initially, we set $\mathcal{C} = \emptyset$.*

1. *The game begins with an execution of the DKG protocol* $\mathsf{Dist\text{-}Keygen}(\mathsf{params}, \lambda, t, N)$ *during which the challenger plays the role of honest players P_i and the adversary $\mathcal{A}$ is allowed to corrupt players at any time. When $\mathcal{A}$ chooses to corrupt a player P_i, the challenger sets $\mathcal{G} = \mathcal{G} \setminus \{i\}$, $\mathcal{C} = \mathcal{C} \cup \{i\}$ and returns the internal state of P_i. Moreover, $\mathcal{A}$ is allowed to act on behalf of P_i from this point forward. The protocol ends with the generation of a public key PK, a vector of private key shares $\mathbf{SK} = (SK_1, \ldots, SK_n)$ and the corresponding verification keys $\mathbf{VK} = (VK_1, \ldots, VK_n)$. At the end of this phase, the public key PK and $\{SK_i\}_{i \in \mathcal{C}}$ are available to the adversary $\mathcal{A}$.*

2. *On polynomially many occasions, $\mathcal{A}$ adaptively interleaves two kinds of queries.*

   - *Corruption query: At any time, $\mathcal{A}$ can choose to corrupt a server. To this end, $\mathcal{A}$ chooses $i \in \{1, \ldots, n\}$ and the challenge returns SK_i before setting $\mathcal{G} = \mathcal{G} \setminus \{i\}$ and $\mathcal{C} = \mathcal{C} \cup \{i\}$.*
   - *Signing query: For any $i \in \mathcal{G}$, $\mathcal{A}$ can also submit a pair (i, M) and ask for a signature share on an arbitrary message M on behalf of player P_i. The challenger responds by computing $\sigma_i \leftarrow \mathsf{Share\text{-}Sign}(SK_i, M)$ and returning σ to $\mathcal{A}$.*

3. *$\mathcal{A}$ outputs a message $M^\star$ and a signature $\sigma^\star$. We define $\mathcal{V} = \mathcal{C} \cup \mathcal{S}$, where $\mathcal{S} \subset \{1, \ldots, n\}$ is the subset of players for which $\mathcal{A}$ made a signing query of the form $(i, M^\star)$. The adversary wins if the following conditions hold: (i) $|\mathcal{V}| < t + 1$; (ii)* $\mathsf{Verify}(PK, M^\star, \sigma^\star) = 1$.

$\mathcal{A}$'s advantage is defined as its probability of success, taken over all coin tosses.

Since we focus on non-interactive schemes, Definition 1 allows the adversary to individually query each partial signing oracle whereas usual definitions only provide the adversary with an oracle that runs the distributed signing protocol on behalf of all honest players. We also remark that Definition 1 allows the adversary to obtain some partial signatures on the forgery message $M^\star$ as long as its output remains a non-trivial forgery. In a weaker (but still compelling) definition, partial signing queries for $M^\star$ would be completely disallowed. In the following, we will stick to the stronger definition.

2.2 Hardness Assumptions

We first recall the definition of the Decision Diffie-Hellman problem.

DEFINITION 2. *In a cyclic group $\mathbb{G}$ of prime order p, the* Decision Diffie-Hellman Problem *(DDH) in $\mathbb{G}$, is to distinguish the distributions (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c), with $a, b, c \xleftarrow{R} \mathbb{Z}_p$. The Decision Diffie-Hellman assumption is the intractability of DDH for any PPT distinguisher.*

We use bilinear maps $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ over groups of prime order p. We will work in asymmetric pairings, where we have $\mathbb{G} \neq \hat{\mathbb{G}}$ so as to allow the DDH assumption to hold in $\mathbb{G}$ (see, *e.g.*, [66]). In certain asymmetric pairing configurations, DDH is even believed to hold in both $\mathbb{G}$ and $\hat{\mathbb{G}}$. This assumption is called *Symmetric eXternal Diffie-Hellman* (SXDH) assumption and it implies that no isomorphism between $\hat{\mathbb{G}}$ and $\mathbb{G}$ be efficiently computable.

For convenience, we also use the following problem in asymmetric pairing configurations.

DEFINITION 3 ([3]). *The* Double Pairing problem *(DP) in $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ is, given $(\hat{g_z}, \hat{g_r}) \in_R \hat{\mathbb{G}}^2$, to find a pair $(z, r) \in \mathbb{G}^2 \setminus \{(1_{\mathbb{G}}, 1_{\mathbb{G}})\}$ that satisfies $e(z, \hat{g_z}) \cdot e(r, \hat{g_r}) = 1_{\mathbb{G}_T}$. The Double Pairing assumption asserts that the DP problem is infeasible for any PPT algorithm.*

The DP problem is known [3] to be at least as hard as DDH in $\hat{\mathbb{G}}$. Given $(\hat{g_z}, \hat{g_r}, \hat{g_z}^{\theta_1}, \hat{g_r}^{\theta_2})$, a solution (z, r) allows deciding whether $\theta_1 = \theta_2$ or not by testing if the equality $e(z, \hat{g_z}^{\theta_1}) \cdot e(r, \hat{g_r}^{\theta_2}) = 1_{\mathbb{G}_T}$ holds.

2.3 Linearly Homomorphic Structure-Preserving Signatures

Structure-preserving signatures [2, 3] are signature schemes that allow signing elements of an abelian group while preserving their algebraic structure, without hashing them first. In [54], Libert *et al.* described structure-preserving signatures with linearly homomorphic properties. Given signatures on several vectors $\vec{M}_1, \ldots, \vec{M}_n$ of group elements, anyone can publicly derive a signature on any linear combination of $\vec{M}_1, \ldots, \vec{M}_n$. They suggested the following scheme, which is a one-time LHSPS (namely, it only allows signing one linear subspace) based on the DP assumption.

$\mathsf{Keygen}(\lambda, N)$: Given a security parameter λ and the dimension $N \in \mathbb{N}$ of the subspace to be signed, choose bilinear group $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Then, choose $\hat{g_z}, \hat{g_r} \xleftarrow{R} \hat{\mathbb{G}}$. For $k = 1$ to N, pick $\chi_k, \gamma_k \xleftarrow{R} \mathbb{Z}_p$ and compute $\hat{g_k} = \hat{g_z}^{\chi_k} \hat{g_r}^{\gamma_k}$. The private key is $\mathsf{sk} = \{\chi_k, \gamma_k\}_{i=1}^N$ while the public key consists of $\mathsf{pk} = \big(\hat{g_z},\ \hat{g_r},\ \{\hat{g_k}\}_{k=1}^N\big)$.

$\mathsf{Sign}(\mathsf{sk}, (M_1, \ldots, M_N))$: To sign $(M_1, \ldots, M_N) \in \mathbb{G}^N$ using $\mathsf{sk} = \{\chi_k, \gamma_k\}_{k=1}^N$, compute $z = \prod_{k=1}^N M_k^{-\chi_k}$ and $r = \prod_{k=1}^N M_k^{-\gamma_k}$. Return $\sigma = (z, r) \in \mathbb{G}^2$.

$\mathsf{SignDerive}(\mathsf{pk}, \{(\omega_i, \sigma^{(i)})\}_{i=1}^\ell)$: Given the public key pk and ℓ tuples $(\omega_i, \sigma^{(i)})$, parse $\sigma^{(i)}$ as $\sigma^{(i)} = \big(z_i, r_i\big) \in \mathbb{G}^3$ for $i = 1$ to ℓ. Then, compute and return $\sigma = (z, r)$, where $z = \prod_{i=1}^\ell z_i^{\omega_i}$ and $r = \prod_{i=1}^\ell r_i^{\omega_i}$.

$\mathsf{Verify}(\mathsf{pk}, \sigma, (M_1, \ldots, M_N))$: Given a purported signature $\sigma = (z, r) \in \mathbb{G}^2$ and a vector $(M_1, \ldots, M_N)$, return 1 if and only if $(M_1, \ldots, M_N) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and (z, r) satisfies $1_{\mathbb{G}_T} = e(z, \hat{g_z}) \cdot e(r, \hat{g_r}) \cdot \prod_{k=1}^N e(M_k, \hat{g_k})$.

A useful property of the scheme is that, if the DP assumption holds, it is computationally hard to come up with two distinct signatures on the same vector, *even* if the private key is available.

3. A PRACTICAL ADAPTIVELY SECURE NON-INTERACTIVE THRESHOLD SIGNATURE

The construction notably relies on the observation that, as shown in the full version of the paper [53], any one-time linearly homomorphic SPS can be turned into a fully secure ordinary signature by introducing a random oracle. The public key is simply that of a linearly homomorphic SPS for vectors of dimension $n > 1$. Messages are signed by hashing them to a vector $\vec{H} \in \mathbb{G}^n$ and generating a one-time homomorphic signature on $\vec{H}$. The security reduction programs the random oracle in such a way that all signed messages are hashed into a proper subspace of $\mathbb{G}^n$ whereas, with some probability, the adversary forges a signature on a message which is hashed outside this subspace. Hence, a forgery for this message translates into an attack against the underlying linearly homomorphic SPS.

In the threshold setting, our system can be seen as an adaptively secure variant of Boldyreva's threshold signature [10], which builds on the short signatures of Boneh, Lynn, and Shacham [14].

The DKG phase uses Pedersen's protocol [62] (or, more precisely, a variant with two generators). Each player verifiably shares a random secret using Pedersen's verifiable secret sharing [63] —where verification is enabled by having all parties broadcast commitments to their secret polynomials— and the final secret key is obtained by summing up the shares of non-disqualified players. When all parties follow the protocol, a single communication round is needed. Moreover, we do not need to rely on zero-knowledge proofs or reliable erasures at any time.

In order to sign a message using his private key share, each player first hashes the message M to obtain a vector $(H_1, H_2) \in \mathbb{G}^2$ of two group elements, which can be signed using the linearly homomorphic structure-preserving signature of Section 2.3. We actually build on the observation that any one-time linearly homomorphic SPS implies a fully secure digital signature in the random oracle model. In the threshold setting, we take advantage of two specific properties in the underlying homomorphic signature. First, it is also key homomorphic[1] and thus amenable for non-interactively distributing the signing process. Second, in the security proof of [54], the reduction always knows the private key, which allows consistently answering adaptive corruption queries.

3.1 Description

In the description below, we assume that all players agree on public parameters params consisting of asymmetric bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with generators $\hat{g_z}, \hat{g_r} \in_R \hat{\mathbb{G}}$ and a hash function $H : \{0,1\}^* \to \mathbb{G}^2$ that ranges over $\mathbb{G} \times \mathbb{G}$. This hash function is modeled as a random oracle in the security analysis. While no party should know $\log_{\hat{g_z}}(\hat{g_r})$, we do not need an extra round to generate $\hat{g_r}$ in a distributed manner as it can simply be derived from a random oracle.

$\mathsf{Dist\text{-}Keygen}(\mathsf{params}, \lambda, t, n)$: Given common public parameters $\mathsf{params} = \{(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T), \hat{g_z}, \hat{g_r}, H\}$, a security parameter λ and integers $t, n \in \mathbb{N}$ such that $n \geq 2t + 1$, each player P_i conducts the following steps.

1. Each player P_i shares two pairs $\{(a_{ik0}, b_{ik0})\}_{k=1}^2$. To this end, he does the following:
   (a) For each $k \in \{1, 2\}$, choose random polynomials $A_{ik}[X] = a_{ik0} + a_{ik1}X + \cdots + a_{ikt}X^t$, $B_{ik}[X] = b_{ik0} + b_{ik1}X + \cdots + b_{ikt}X^t \in \mathbb{Z}_p[X]$ of degree t and broadcast
   $$\hat{W}_{ik\ell} = \hat{g_z}^{a_{ik\ell}} \hat{g_r}^{b_{ik\ell}} \qquad \forall \ell \in \{0, \ldots, t\}\ .$$
   (b) For $j = 1$ to n, send $\{(A_{ik}(j), B_{ik}(j))\}_{k=1}^2$ to P_j.
2. For each set of shares $\{(A_{jk}(i), B_{jk}(i))\}_{k=1}^2$ received from another player P_j, P_i verifies that
   $$\hat{g_z}^{A_{jk}(i)} \hat{g_r}^{B_{jk}(i)} = \prod_{\ell=0}^{t} \hat{W}_{jk\ell}^{\ i^\ell} \qquad \text{for } k = 1, 2\ . \quad (1)$$
   If these equalities do not both hold, P_i broadcasts a complaint against the faulty sender P_j.
3. Any player who received strictly more than t complaints from other players is immediately disqualified. Each player P_i who received a complaint from another player P_j responds by returning the correct shares $\{(A_{ik}(j), B_{ik}(j))\}_{k=1}^2$. If any of these new shares does not satisfy (1), P_i is disqualified. Let $\mathcal{Q} \subset \{1, \ldots, n\}$ be the set of non-disqualified players at the end of step 3.
4. The public key is obtained as $PK = \{\hat{g}_k\}_{k=1}^2$, where $\hat{g}_k = \prod_{i \in \mathcal{Q}} \hat{W}_{ik0} = \hat{g_z}^{\sum_{i \in \mathcal{Q}} a_{ik0}} \hat{g_r}^{\sum_{i \in \mathcal{Q}} b_{ik0}}$.

[1] Namely, the private key space forms an additive group such that, for any message M, given any two signatures $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, M)$ and $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, M)$, anyone can compute a valid signature on M for the private key $sk_1 + sk_2$.

Each P_i locally defines his private key share

$$SK_i = \{(A_k(i), B_k(i))\}_{k=1}^2 = \{\big(\sum_{j\in\mathcal{Q}} A_{jk}(i), \sum_{j\in\mathcal{Q}} B_{jk}(i)\big)\}_{k=1}^2$$

and anyone can publicly compute his verification key $VK_i = \big(\hat{V}_{1,i}, \hat{V}_{2,i}\big)$ as

$$VK_i = \big(\hat{g_z}^{A_1(i)}\hat{g_r}^{B_1(i)}, \hat{g_z}^{A_2(i)}\hat{g_r}^{B_2(i)}\big) = \Big(\prod_{j\in\mathcal{Q}}\prod_{\ell=0}^{t}\hat{W}_{j1\ell}^{i^\ell},\ \prod_{j\in\mathcal{Q}}\prod_{\ell=0}^{t}\hat{W}_{j2\ell}^{i^\ell}\Big)\ .$$

For any disqualified player $i \in \{1,\ldots,n\}\setminus\mathcal{Q}$, the i-th private key share is implicitly set as $SK_i = \{(0,0)\}_{k=1}^2$ and the corresponding verification key is $VK_i = (1_{\hat{\mathbb{G}}}, 1_{\hat{\mathbb{G}}})$.

This completes the generation of the private key shares $\mathbf{SK} = (SK_1,\ldots,SK_n)$, the vector of verification keys $\mathbf{VK} = (VK_1,\ldots,VK_n)$ and the public key, which consists of $PK = \Big(\mathsf{params},\ (\hat{g_1},\hat{g_2})\Big)$.

When the protocol ends, the obtained private key shares $\{A_k(i)\}_{k=1}^2$ and $\{B_k(i)\}_{k=1}^2$ all lie on t-degree polynomials $A_k[X] = \sum_{j\in\mathcal{Q}} A_{jk}[X]$ and $B_k[X] = \sum_{j\in\mathcal{Q}} B_{jk}[X]$. Each player also holds an additive share $\{(a_{ik0}, b_{ik0})\}_{k=1}^2$ of the secret key $\{(A_k(0), B_k(0)) = (\sum_{i\in\mathcal{Q}} a_{ik0}, \sum_{i\in\mathcal{Q}} b_{ik0})\}_{k=1}^2$ but these shares will not be used in the scheme.

Share-Sign(i, SK_i, M): To generate a partial signature on a message $M \in \{0,1\}^*$ using his private key share $SK_i = \{(A_k(i), B_k(i))\}_{k=1}^2$, P_i first computes the hash value $(H_1, H_2) = H(M) \in \mathbb{G}\times\mathbb{G}$ and generates the partial signature $\sigma_i = (z_i, r_i) \in \mathbb{G}^2$ as

$$z_i = \prod_{k=1}^{2} H_k^{-A_k(i)}, \qquad r_i = \prod_{k=1}^{2} H_k^{-B_k(i)}\ .$$

Share-Verify$\big(PK, \mathbf{VK}, M, (i,\sigma_i)\big)$: Given a candidate partial signature $\sigma_i = (z_i, r_i) \in \mathbb{G}^2$ and the verification key $VK_i = \big(\hat{V}_{1,i}, \hat{V}_{2,i}\big)$, the partial verification algorithm first computes $(H_1, H_2) = H(M) \in \mathbb{G}^2$. It returns 1 if the equality $e(z_i, \hat{g_z})\cdot e(r_i, \hat{g_r})\cdot\prod_{k=1}^2 e(H_k, \hat{V}_{k,i}) = 1_{\mathbb{G}_T}$ holds and 0 otherwise.

Combine$(PK, \mathbf{VK}, M, \{(i,\sigma_i)\}_{i\in S})$: Given a $(t+1)$-set with valid shares $\{(i,\sigma_i)\}_{i\in S}$, parse the signature share σ_i as $\big(z_i, r_i\big) \in \mathbb{G}^2$ for each $i \in S$. Then, compute

$$(z, r) = \big(\prod_{i\in S} z_i^{\Delta_{i,S}(0)},\ \prod_{i\in S} r_i^{\Delta_{i,S}(0)}\big)$$

by Lagrange interpolation in the exponent. Return the pair $(z, r) \in \mathbb{G}^2$.

Verify$\big(PK, M, \sigma\big)$: Given a purported signature $\sigma = (z, r) \in \mathbb{G}^2$, compute $(H_1, H_2) = H(M) \in \mathbb{G}\times\mathbb{G}$ and return 1 if and only if the following equality holds:

$$e(z, \hat{g_z})\cdot e(r, \hat{g_r})\cdot e(H_1, \hat{g_1})\cdot e(H_2, \hat{g_2}) = 1_{\mathbb{G}_T}\ .$$

If the scheme is instantiated using Barreto-Naehrig curves [5] at the 128-bit security level, each signature consists of 512 bits. For the same security level, RSA-based threshold signatures like [68, 4] require 3076 bits. The scheme is also very efficient from a computational standpoint. Each server only has to compute two multi-exponentiations with two base elements and two "hash-on-curve" operations. The verifier has to compute a product of four pairings.

At the end of the key generation phase, each player only needs to store a private key share $SK_i = \{(A_k(i), B_k(i))\}_{k=1}^2$ of constant-size —whereas solutions like [4] incur the storage of $O(n)$ elements at each player— and can erase all intermediate values, including the polynomials $A_{ik}[X]$ and $B_{ik}[X]$. However, we insist that the security analysis does not require reliable erasures. When a player is corrupted, we assume that the adversary learns the entire history of this player.

3.2 Security

Although the public key is not guaranteed to be uniform due to the use of Pedersen's DKG protocol, the key homomorphic property allows the reduction to turn the adversary's forgery into a valid signature with respect to some uniformly random public key obtained by multiplying honest users' contributions to the public key. This is sufficient for solving a given Double Pairing instance.

THEOREM 3.1. *The scheme provides adaptive security under the SXDH assumption in the random oracle model. For any PPT adversary $\mathcal{A}$, there exist DDH distinguishers $\mathcal{B}_1$ and $\mathcal{B}_2$ with comparable running time in the groups $\mathbb{G}$ and $\hat{\mathbb{G}}$, respectively.*

A detailed proof of Theorem 3.1 is available in the full version of the paper [53]. It proceeds with a sequence of games which can be outlined as follows.

The first game is the real game where the challenger assumes the role of all honest players in the distributed key generation phase. Since it controls a majority of players, the challenger knows $\{A_{jk}[X], B_{jk}[X]\}_{j\in\mathcal{Q},k\in\{1,2\}}$ and the private key shares $\{SK_j\}_{j\in\mathcal{Q}}$ of all non-disqualified players —either because it obtained at least $t+1$ polynomial shares $\{(A_{jk}(i), B_{jk}(i))\}_{i\in\mathcal{G},k\in\{1,2\}}$ for each $j \in \mathcal{Q}$ or because it chose the polynomials itself— at the end of Dist-Keygen.

In subsequent games, the challenger applies Coron's proof technique for Full Domain Hash signatures [20]. At each random oracle query $H(M)$, it flips a coin $\vartheta_M \in \{0,1\}$ that takes the value 0 with probability $1/q_s$ and the value 1 with probability $1/(q_s+1)$, where q_s is the number of signing queries. If $\vartheta_M = 1$, the challenger defines $H(M)$ to be a random vector of $\mathbb{G}^2$. If $\vartheta_M = 0$, the message M is hashed to a subspace of dimension 1. We prove that, although H does no longer behave as an actual random oracle, this change should not affect the adversary's view if the DDH assumption holds in $\mathbb{G}$. Coron's analysis [20] shows that, with probability $\Omega(1/q_s)$, the following conditions are fulfilled: (i) The adversary only obtains partial signatures on messages $M_1,\ldots,M_{q_s}$ that are hashed in a one-dimensional subspace; (ii) The adversary's forgery involves a message $M^\star$ such that $(H_1^\star, H_2^\star) = H(M^\star)$ is linearly independent of the vectors $\{(H_{1,i}, H_{2,i}) = H(M_i)\}_{i=1}^{q_s}$. Condition (i) ensures that the adversary obtains little information about the private key shares $\{SK_i\}_{i\in\mathcal{G}}$ and the additive shares $\{a_{ik0}, b_{ik0}\}_{i\in\mathcal{G},k\in\{1,2\}}$ of honest players. Hence, if the challenger computes the additive contribution of honest play-

ers to a signature on the vector $(H_1^\star, H_2^\star) = H(M^\star)$, this contribution is completely unpredictable by the adversary due to condition (ii). With overwhelming probability, this contribution does *not* coincide with the one that can be extracted (using the additive shares $\{a_{jk0}, b_{jk0}\}_{j\in\mathcal{Q}\setminus\mathcal{G},\ k\in\{1,2\}}$ that the reduction knows from the key generation phase) from the adversary's forgery $(z^\star, r^\star)$ using the key homomorphic property of the scheme. The challenger thus obtains two distinct linearly homomorphic signatures on the vector $(H_1^\star, H_2^\star)$, which allows solving an instance of the Double Pairing problem.

The proof of Theorem 3.1 goes through if, during the DKG phase, each player P_i additionally publicizes the pair $(Z_{i0}, R_{i0}) = \big(g^{-a_{i10}}h^{-a_{i20}}, g^{-b_{i10}}h^{-b_{i20}}\big)$, for public generators $g, h \in \mathbb{G}$, which satisfies

$$e(Z_{i0}, \hat{g}_z) \cdot e(R_{i0}, \hat{g}_r) \cdot e(h, \hat{W}_{i10}) \cdot e(g, \hat{W}_{i20}) = 1_{\mathbb{G}_T}$$

and forms a LHSPS on (g, h) for the public key $\{\hat{W}_{ik0}\}_{k=1}^2$. Based on this observation, we describe a simple modification of the scheme that supports signature aggregation in the full version of the paper [53].

3.3 Adding Proactive Security

The scheme readily extends to provide proactive security [61, 47, 37] against mobile adversaries that can potentially corrupt all the players at some point as long as it never controls more than t players at any time. By having the players refreshing all shares (without changing the secret) at discrete time intervals, the scheme remains secure against an adversary corrupting up to t players during the same period. This is achieved by having all players run a new instance of Pedersen's DKG protocol where the shared secret is $\{(0,0)\}_{k=1}^2$ and locally add the resulting shares to their local shares before updating $\{VK_i\}_{i=1}^n$ accordingly.

The techniques of [46, Section 4] can also be applied to detect parties holding a corrupted share (due to a crash during an update phase or an adversarial behavior) and restore the correct share if necessary.

4. A CONSTRUCTION IN THE STANDARD MODEL

This section gives a round-optimal construction in the standard model. We remark that, under the Decision Linear assumption [11], any one-time LHSPS in symmetric bilinear groups can be turned into a full-fledged digital signature, as shown in the full version of the paper. In the threshold setting, we need to rely on specific properties of the underlying LHSPS in order to achieve adaptive security without relying on a trusted dealer.

The scheme relies on the Groth-Sahai non-interactive witness indistinguishable (NIWI) proof systems [45], which are recalled in Appendix A. In its centralized version, a signature consists of a NIWI proof of knowledge —somewhat in the spirit of Okamoto's signature scheme [60]— of a one-time linearly homomorphic signature on a fixed vector $g \in \mathbb{G}$ of dimension $n = 1$. To generate this proof, the signer forms a Groth-Sahai [45] common reference string (CRS) $(\vec{f}, \vec{f}_M)$ using the bits of the message M, according to a technique suggested by Malkin *et al.* [58]. Due to the witness indistinguishability property of Groth-Sahai proofs, no information leaks about the private key of the underlying one-time homomorphic signature. For this reason, when the adversary creates a fake signature, the reduction is able to extract a different homomorphic signature than the one it can compute. Hence, it obtains two distinct signatures on the same vector, which allows solving an instance of the DP problem.

The scheme can also be seen as a threshold version of (a variant of) the signature presented in [58]. In order to distribute the signing process, we take advantage of the homomorphic properties of Groth-Sahai proofs. More precisely, we use the fact that linear pairing product equations and their proofs can be linearly combined in order to obtain a valid proof for the desired statement when performing a Lagrange interpolation in the exponent. In order to avoid interaction during the signing process, we leverage the property that the centralized signature scheme is key homomorphic.

However, we have to prove that the scheme remains adaptively secure when the DKG phase uses Pedersen's protocol [62]. To this end, we take further advantage of the key homomorphic property. In the security proof, we show that, if the adversary can forge a signature for a non-uniform public key PK, we can turn this forgery into one for another public key PK', which is uniformly distributed.

In the following notations, for each $h \in \mathbb{G}$ and any vector $\vec{g} = (g_1, g_2) \in \mathbb{G}^2$, we denote by $E(\vec{g}, \hat{h})$ the vector $(e(g_1, \hat{h}), e(g_2, \hat{g})) \in \mathbb{G}_T^2$.

Here, we assume public parameters params made of asymmetric bilinear groups $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with generators $g \in_R \mathbb{G}$, $\hat{g}_z, \hat{g}_r \in_R \hat{\mathbb{G}}$ and vectors $\vec{f} = (f, h) \in \mathbb{G}^2$ and $\vec{f_i} = (f_i, h_i) \xleftarrow{R} \mathbb{G}^2$ for $i = 0$ to L, where $L \in \mathsf{poly}(\lambda)$.

$\mathsf{Dist\text{-}Keygen}(\mathsf{params}, \lambda, t, n)$: This protocol proceeds as in the scheme of Section 3. Namely, given common parameters $\mathsf{params} = \{(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T), g, \hat{g}_z, \hat{g}_r, \vec{f}, \{\vec{f_i}\}_{i=0}^L\}$, a security parameter λ and integers $t, n \in \mathbb{N}$ such that $n \geq 2t+1$, each player P_i conducts the following steps.

1. Each player P_i shares a random pair (a_{i0}, b_{i0}) according to the following step:
   (a) Pick random polynomials $A_i[X] = a_{i0} + a_{i1}X + \cdots + a_{it}X^t$, $B_i[X] = b_{i0} + b_{i1}X + \cdots + b_{it}X^t$ of degree t and broadcast $\hat{W}_{i\ell} = \hat{g_z}^{a_{i\ell}}\hat{g_r}^{b_{i\ell}}$ for all $\ell \in \{0, \ldots, t\}$.
   (b) For $j = 1$ to n, send $(A_i(j), B_i(j))$ to P_j.
2. For each received shares $(A_j(i), B_j(i))$, player P_i verifies that
$$\hat{g_z}^{A_j(i)}\hat{g_r}^{B_j(i)} = \prod_{\ell=0}^{t} \hat{W}_{j\ell}^{\ i^\ell} \quad . \tag{2}$$
   If the latter equality does not hold, P_i broadcasts a complaint against P_j.
3. Any player receiving more than t complaints is disqualified. Each player P_i who received a complaint from another player P_j responds by returning the correct shares $(A_i(j), B_i(j))$. If any of these new shares fails to satisfy (2), the faulty P_i is expelled. Let $\mathcal{Q} \subset \{1, \ldots, n\}$ be the set of non-disqualified players at the end of step 3.
4. The public key PK is obtained as $PK = \hat{g}_1$, where $\hat{g}_1 = \prod_{i\in\mathcal{Q}} \hat{W}_{i0} = \hat{g}_z^{\sum_{i\in\mathcal{Q}} a_{i0}}\hat{g}_r^{\sum_{i\in\mathcal{Q}} b_{i0}}$. Each player P_i locally defines his private key share as $SK_i = (A(i), B(i)) = \big(\sum_{j\in\mathcal{Q}} A_j(i), \sum_{j\in\mathcal{Q}} B_j(i)\big)$ and anyone can publicly compute his verification key as

$VK_i = \hat{V}_i = \hat{g}_z{}^{A(i)}\hat{g}_r{}^{B(i)} = \prod_{j\in\mathcal{Q}}\prod_{\ell=0}^{t}\hat{W}_{j\ell}{}^{i^\ell}$. Any disqualified player $i \in \{1,\ldots,n\}\setminus\mathcal{Q}$ is implicitly assigned the share $SK_i = (0,0)$ and the matching verification key $VK_i = 1_{\hat{\mathbb{G}}}$.

At the end of the protocol, the vector of private key shares is $\mathbf{SK} = (SK_1,\ldots,SK_n)$ and the corresponding vector of verification keys $\mathbf{VK} = (VK_1,\ldots,VK_n)$. The public key consists of $PK = \big(\mathsf{params},\ \hat{g_1}\big)$.

Share-Sign(SK_i, M): To generate a partial signature on a L-bit message $M = M[1]\ldots M[L] \in \{0,1\}^L$ using $SK_i = (A(i), B(i))$, define $(z_i, r_i) = (g^{-A(i)}, g^{-B(i)})$.

1. Using the bits $M[1]\ldots M[L]$ of $M \in \{0,1\}^L$, define the vector $\vec{f}_M = \vec{f}_0 \cdot \prod_{i=1}^{L}\vec{f}_i^{M[i]}$ so as to assemble a Groth-Sahai CRS $\mathbf{f}_M = (\vec{f}, \vec{f}_M)$.
2. Using the CRS $\mathbf{f}_M = (\vec{f}, \vec{f}_M)$, compute Groth-Sahai commitments $\vec{C}_{z,i} = (1_{\mathbb{G}}, z_i)\cdot\vec{f}_1^{\nu_{z,1}}\cdot\vec{f}_M^{\nu_{z,2}}$ and $\vec{C}_{r,i} = (1_{\mathbb{G}}, r_i)\cdot\vec{f}^{\nu_{r,1,i}}\cdot\vec{f}_M^{\nu_{r,2,i}}$ to the group elements z_i and r_i, respectively. Then, generate a NIWI proof $\vec{\hat{\pi}}_i = (\hat{\pi}_{1,i}, \hat{\pi}_{2,i}) \in \hat{\mathbb{G}}^2$ that committed elements $(z_i, r_i) \in \mathbb{G}^2$ satisfy the verification equation $1_{\mathbb{G}_T} = e(z_i, \hat{g_z})\cdot e(r_i, \hat{g_r})\cdot e(g, \hat{V}_i)$. This proof is obtained as

$$\begin{aligned}\vec{\hat{\pi}}_i &= (\hat{\pi}_{1,i}, \hat{\pi}_{2,i})\\ &= \big(\hat{g_z}^{-\nu_{z,1,i}}\cdot\hat{g_r}^{-\nu_{r,1,i}},\ \hat{g_z}^{-\nu_{z,2,i}}\cdot\hat{g_r}^{-\nu_{r,2,i}}\big)\end{aligned}$$

Return $\sigma_i = (\vec{C}_{z,i}, \vec{C}_{r,i}, \vec{\hat{\pi}}_i) \in \mathbb{G}^4\times\hat{\mathbb{G}}^2$.

Share-Verify$(PK, \mathbf{VK}, M, (i,\sigma_i))$: Given $M \in \{0,1\}^L$ and a candidate σ_i, parse σ_i as $\sigma_i = (\vec{C}_{z,i}, \vec{C}_{r,i}, \vec{\hat{\pi}}_i)$. Define $\vec{f}_M = \vec{f}_0\cdot\prod_{i=1}^{L}\vec{f}_i^{M[i]}$ and return 1 if $\vec{\hat{\pi}}_i = (\hat{\pi}_{1,i}, \hat{\pi}_{2,i})$ satisfies

$$\begin{aligned}&E\big((1_{\mathbb{G}}, g), \hat{V}_i\big)^{-1}\\ &= E\big(\vec{C}_{z,i}, \hat{g_z}\big)\cdot E\big(\vec{C}_{r,i}, \hat{g_r}\big)\cdot E(\vec{f}, \hat{\pi}_{1,i})\cdot E(\vec{f}_M, \hat{\pi}_{2,i})\end{aligned}$$

and 0 otherwise.

Combine$(PK, \mathbf{VK}, M, \{(i,\sigma_i)\}_{i\in S})$: Given a $(t+1)$-set with valid shares $\{(i,\sigma_i)\}_{i\in S}$, parse each signature share σ_i as $(\vec{C}_{z,i}, \vec{C}_{r,i}, \vec{\hat{\pi}}_i) \in \mathbb{G}^4\times\hat{\mathbb{G}}^2$, where $\vec{\hat{\pi}}_i = (\hat{\pi}_{1,i}, \hat{\pi}_{2,i})$, for all $i \in S$. Then, compute $(\vec{C}'_z, \vec{C}'_r, \hat{\pi}'_1, \hat{\pi}'_2)$ as

$$\Big(\prod_{i\in S}\vec{C}_{z,i}^{\Delta_{i,S}(0)},\ \prod_{i\in S}\vec{C}_{r,i}^{\Delta_{i,S}(0)},\ \prod_{i\in S}\hat{\pi}_{1,i}^{\Delta_{i,S}(0)},\ \prod_{i\in S}\hat{\pi}_{2,i}^{\Delta_{i,S}(0)}\Big)$$

by Lagrange interpolation in the exponent. Finally, re-randomize $(\vec{C}'_z, \vec{C}'_r, \hat{\pi}'_1, \hat{\pi}'_2)$ and output the resulting re-randomized full signature $\sigma = (\vec{C}_z, \vec{C}_r, \hat{\pi}_1, \hat{\pi}_2)$.

Verify(PK, M, σ): Given a message $M \in \{0,1\}^L$ and a purported signature σ, parse σ as $(\vec{C}_z, \vec{C}_r, \vec{\hat{\pi}}) \in \mathbb{G}^4\times\hat{\mathbb{G}}^2$. Define $\vec{f}_M = \vec{f}_0\cdot\prod_{i=1}^{L}\vec{f}_i^{M[i]}$ and return 1 if and only if $\vec{\hat{\pi}} = (\hat{\pi}_1, \hat{\pi}_2)$ satisfies

$$\begin{aligned}&E\big((1_{\mathbb{G}}, g), \hat{g}_1\big)^{-1}\\ &= E\big(\vec{C}_z, \hat{g_z}\big)\cdot E\big(\vec{C}_r, \hat{g_r}\big)\cdot E(\vec{f}, \hat{\pi}_1)\cdot E(\vec{f}_M, \hat{\pi}_2)\ .\end{aligned}$$

The scheme can be simplified by having each player set his private key share as $SK_i = (g^{-A(i)}, g^{-B(i)})$ so as to spare two exponentiations in the signing phase. In the description, we defined SK_i as $(A(i), B(i))$ to insist that no reliable erasures are needed. At each corruption query, the adversary obtains $(A(i), B(i))$ and, not only $(g^{-A(i)}, g^{-B(i)})$. In any case, each player only needs to store two elements of $\mathbb{Z}_p$.

At the 128-bit security level, if each element of $\mathbb{G}$ (resp. $\hat{\mathbb{G}}$) has a 256-bit (resp. 512 bit) representation on Barreto-Naehrig curves [5], we only need 2048 bits per signature.

THEOREM 4.1. *The scheme provides adaptive security under the SXDH assumption in the standard model. For any PPT adversary $\mathcal{A}$, there exist DDH distinguishers $\mathcal{B}_1$ and $\mathcal{B}_2$ with comparable running time in $\mathbb{G}$ and $\hat{\mathbb{G}}$, respectively.* (The proof is available in the full version of the paper).

Acknowledgements

The first author's work has been supported in part by the ERC Starting Grant ERC-2013-StG-335086-LATTAC.

5. REFERENCES

[1] M. Abe, S. Fehr. Adaptively secure Feldman VSS and applications to universally-composable threshold cryptography. In *Crypto '04*, *LNCS* 3152, pp. 317–334, 2004.

[2] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Crypto '10*, *LNCS* 6223, pp. 209–236, 2010.

[3] M.Abe, K. Haralambiev, M. Ohkubo. Signing on elements in bilinear groups for modular protocol design. In Cryptology ePrint Archive: Report 2010/133, 2010.

[4] J. Almansa, I. Damgård, J.-B. Nielsen. Simplified threshold RSA with adaptive and proactive security. In *Eurocrypt '06*, *LNCS* 4004, pp. 593–611, 2006.

[5] P. Barreto, M. Naehrig. Pairing-friendly elliptic curves of prime order. In *SAC '05*, *LNCS* 3897, pp. 319–331, 2005.

[6] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *Crypto '09*, *LNCS* 5677, pp. 108–125, 2009.

[7] M. Bellare, C. Namprempre, G. Neven. Unrestricted aggregate signatures. In *ICALP '07*, *LNCS* 4596, pp. 411–422, 2007.

[8] M. Bellare, T. Ristenpart. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In *Eurocrypt '09*, *LNCS* 5479, pp. 407–424, 2009.

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

[10] A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In *Public-Key Cryptography 2003 (PKC '03)*, *LNCS* 2567, pp. 31–46, 2003.

[11] D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *Crypto '04*, *LNCS* 3152, pp. 41–55, 2004.

[12] D. Boneh, M. Franklin. Efficient Generation of Shared RSA Keys. In *Crypto '97*, *LNCS* 1924, pp. 425–439, 1997.

[13] D. Boneh, C. Gentry, B. Lynn, H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Eurocrypt '03*, *LNCS* 2656, pp. 416–432, 2003.

[14] D. Boneh, B. Lynn, H. Shacham. Short signatures from the Weil pairing. *J. Cryptology* 17(4), pp. 297–319, 2004. Earlier version in *Asiacrypt '01*, *LNCS* 2248, pp. 514–532, 2001.

[15] C. Boyd. Digital Multisignatures. In *Cryptography and Coding*, Oxford University Press, pp. 241–246, 1989.

[16] R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Adaptive security for threshold cryptosystems. In *Crypto '99*, *LNCS* 1666, pp. 98–115, 1999.

[17] R. Canetti, S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *Eurocrypt '99*, *LNCS* 1592, pp. 90–106, 1999.

[18] Distributed CA for Visa-MC SET Infrastructure announcement, 1997 see: `http://www.geocities.ws/rayvaneng/w0597_09.htm`

[19] J. Chen, H.-W. Lim, S. Ling, H. Wang, H. Wee. Shorter IBE and signatures via asymmetric pairings. In *Pairing '12*, *LNCS* 7708, pp. 122–140, 2012.

[20] J.-S. Coron. On the exact security of full domain hash. In *Crypto '00*, *LNCS* 1880, pp. 229–235, 2000.

[21] V. Cortier, D. Galindo, S. Glondu, M. Izabachène. Distributed ElGamal à la Pedersen: Application to Helios. In *WPES '13*, pp. 131–142, 2013.

[22] R. Cramer, M. Franklin, B. Schoenmakers, M. Yung. Multi-autority secret-ballot elections with linear work. In *Eurocrypt '96*, *LNCS* 1070, pp. 72–83, 1996.

[23] R. Cramer, R. Gennaro, B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Eurocrypt '97*, *LNCS* 1233, pp. 103–118, 1997.

[24] R. Cramer, I. Damgård, J.-B. Nielsen. Multiparty computation from threshold homomorphic encryption. In *Eurocrypt '01*, *LNCS* 2045, pp. 280–299, 2001.

[25] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, T. Rabin. Efficient multi-party computations secure against an adaptive adversary. In *Eurocrypt '99*, *LNCS* 1592, pp. 311–326, 1999.

[26] I. Damgård, G. Mikkelsen. Efficient, robust and constant-round distributed RSA key generation. In *TCC '10*, *LNCS* 5978, pp. 183–200, 2010.

[27] A. Dent. A Note On Game-Hopping Proofs. Cryptology ePrint Archive: Report 2006/260.

[28] A. De Santis, Y. Desmedt, Y. Frankel, M. Yung. How to share a function securely. In *STOC '94*, pp. 522-533, 1994.

[29] Y. Desmedt. Society and group oriented cryptography: A new concept. In *Crypto '87*, *LNCS* 293, pp. 120–127, 1987.

[30] Y. Desmedt, Y. Frankel. Threshold cryptosystems. In *Crypto '89*, *LNCS* 435, pp. 307–315, Springer, 1990.

[31] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Crypto '84*, *LNCS* 196, pp. 10–18, 1984.

[32] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *FOCS '87*, pp. 427–437, 1987.

[33] P.-A. Fouque, J. Stern. Fully Distributed Threshold RSA under standard assumptions. In *Asiacrypt '01*, *LNCS* 2248, pp. 310–330, Springer, 2001.

[34] Y. Frankel, P. MacKenzie, M. Yung. Robust efficient distributed RSA-key generation. In *STOC '98*, pp. 663–672, 1998.

[35] Y. Frankel, P. MacKenzie, M. Yung. Adaptively-secure distributed public-key systems. In *ESA '99*, *LNCS* 1643, pp. 4–27, 1999.

[36] Y. Frankel, P. MacKenzie, M. Yung. Adaptively-secure optimal-resilience proactive RSA. In *Asiacrypt '99*, *LNCS* 1716, pp. 180–194, 1999.

[37] Y. Frankel, P. Gemmell, P. MacKenzie, M. Yung. Optimal resilience proactive public-key cryptosystems. In *FOCS '97*, pp. 384–393, 1997.

[38] D. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *Eurocrypt '10*, *LNCS* 6110, pp. 44-61, 2010.

[39] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Robust threshold DSS signatures. In *Eurocrypt '96*, *LNCS* 1070, pp. 354–371, 1996.

[40] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Robust and efficient sharing of RSA functions. In *Crypto '96*, *LNCS* 1109, pp. 157–172, 1996.

[41] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *Eurocrypt '99*, *LNCS* 1592, pp. 295–310, 1999.

[42] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Secure applications of Pedersen's distributed key generation protocol. In *CT-RSA '03*, *LNCS* 2612, pp. 373–390, 2003.

[43] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *J. Cryptology* 20(1), pp. 51–83, 2007.

[44] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin. Threshold RSA for dynamic and ad-hoc groups. In *Eurocrypt '08*, *LNCS* 4965, pp. 88–107, 2008.

[45] J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt '08*, *LNCS* 4965, pp. 415–432, 2008.

[46] A. Herzberg, S. Jarecki, H. Krawczyk, M. Yung. Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In *Crypto '95*, *LNCS* 963, pp. 339-352, 1995.

[47] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, M. Yung. Proactive public key and signature systems. In *ACM-CCS '97*, pp. 100–110, 1997.

[48] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao. OceanStore: An architecture for global-scale persistent storage. In *ASPLOS '00*, pp. 190-201, 2000.

[49] S. Jarecki, A. Lysyanskaya. Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In *Eurocrypt '00*, *LNCS* 1807, pp. 221–242, 2000.

[50] J. Katz, M. Yung. Threshold cryptosystems based on factoring. In *Asiacrypt '02*, *LNCS* 2501, pp. 199-205, 2002.

[51] A. Lewko, B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC '10*, *LNCS* 5978, pp. 455–479, 2010.

[52] A. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *Eurocrypt '12*, *LNCS* 5978, pp. 318–33, 2012.

[53] B. Libert, M. Joye, M. Yung. Born and raised distributively: fully distributed non-interactive adaptively-secure threshold signatures with short shares. Full version. Available from `http://hal.inria.fr/hal-00983149`.

[54] B. Libert, T. Peters, M. Joye, M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In *Crypto '13*, *LNCS* 8043, pp. 289–307, 2013.

[55] B. Libert, M. Yung. Adaptively secure non-interactive threshold cryptosystems. *Theoretical Computer Science*, vol. 478, pp. 76–100, March 2013. Extended abstract in *ICALP 2011*, *LNCS* 6756, pp. 588–600, 2011.

[56] B. Libert, M. Yung. Non-interactive CCA2-secure threshold cryptosystems with adaptive security: New framework and constructions. In *TCC '12*, *LNCS* 7194, pp. 75–93, Springer, 2012.

[57] A. Lysyanskaya, C. Peikert. Adaptive security in the threshold setting: From cryptosystems to signature schemes. In *Asiacrypt '01*, *LNCS* 2248, pp. 331–350, 2001.

[58] T. Malkin, I. Teranishi, Y. Vahlis, M. Yung. Signatures resilient to continual leakage on memory and computation. In *TCC '11*, *LNCS* 6597, pp. 89–106, 2011.

[59] M. Naor, O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS '97*, pp. 458–467, 1997.

[60] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Crypto '92*, *LNCS* 740, pp. 31–53, 2011.

[61] R. Ostrovksy, M. Yung. How to withstand mobile virus attacks. In *PODC '91*, pp. 51–59, 1991.

[62] T. Pedersen. A threshold cryptosystem without a trusted party. *Eurocrypt '91*, *LNCS* 547, pp. 522–526, 1991.

[63] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. *Crypto '91*, *LNCS* 576, pp. 129–140, 1991.

[64] T. Rabin. A simplified approach to threshold and proactive RSA. In *Crypto '98*, *LNCS* 1462, pp. 89–104, 1998.

[65] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, J. Kubiatowicz. Pond: The OceanStore prototype. In *2003 USENIX Workshop on File and Storage Technologies*, 2003.

[66] M. Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number. Cryptology ePrint Archive: Report 2002/164.

[67] A. Shamir. How to share a secret. In *Commun. ACM* 22(11), pp. 612–613, 1979.

[68] V. Shoup. Practical threshold signatures. In *Eurocrypt 2000*, *LNCS* 1807, pp. 207–220, 2000.

[69] H. Wee. Threshold and revocation cryptosystems via extractable hash proofs. In *Eurocrypt '11*, *LNCS* 6632, pp. 589–609, 2011.

[70] Z. Wang, H. Qian, Z. Li. Adaptively secure threshold signature scheme in the standard model. In *Informatica* 20(4), pp. 591-612, 2009.

[71] B. Waters. Efficient identity-based encryption without random oracles. In *Eurocrypt '05*, *LNCS* 3494, 2005.

[72] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Crypto '09*, *LNCS* 5677, pp. 619–636, 2009.

APPENDIX

A. GROTH-SAHAI NON-INTERACTIVE PROOF SYSTEMS

In [45], Groth and Sahai described efficient non-interactive witness indistinguishable (NIWI) proof systems of which one instantiation relies on the SXDH assumption. This instantiation uses prime order groups and a common reference string containing three vectors $\vec{f}_1, \vec{f}_2 \in \mathbb{G}^2$, where $\vec{f}_1 = (g, f_1)$, $\vec{f}_2 = (h, f_2)$, for some $g, h, f_1, f_2 \in \mathbb{G}$. To commit to a group element $X \in \mathbb{G}$, the prover chooses $r, s \xleftarrow{R} \mathbb{Z}_p$ and computes $\vec{C} = (1, X) \cdot \vec{f}_1^{\,r} \cdot \vec{f}_2^{\,s}$. On a perfectly sound common reference string, we have $\vec{f}_2 = \vec{f}_1^{\,\xi}$, for some $\xi \in \mathbb{Z}_p$. Commitments $\vec{C} = (g^{r+\xi s}, f_1^{r+\xi s} \cdot X)$ are extractable as their distribution coincides with that of an Elgamal ciphertexts [31] and the committed X can be extracted using $\beta = \log_g(f_1)$. In the witness indistinguishability (WI) setting, the vector $\vec{f}_2$ is chosen so that $(\vec{f}_1, \vec{f}_2)$ are linearly independent vectors and $\vec{C}$ is a perfectly hiding commitment. Under the DDH assumption in $\mathbb{G}$, the two kinds of CRS can be exchanged for one another without the adversary noticing.

To convince the verifier that committed variables satisfy a set of relations, the prover computes one commitment per variable and one proof element per equation. Such NIWI proofs can be efficiently generated for linear pairing-product equations, which are relations of the type

$$\prod_{i=1}^{n} e(\mathcal{X}_i, \mathcal{A}_i) = t_T \,, \tag{3}$$

for variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \in \mathbb{G}$ and constants $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \hat{\mathbb{G}}$, $t_T \in \mathbb{G}_T$, $a_{ij} \in \mathbb{Z}_p$, for $i, j \in \{1, \ldots, n\}$.

Under the SXDH assumption, proving an an equation like (3) requires two elements of $\hat{\mathbb{G}}$.

In [6], Belenkiy *et al.* showed that Groth-Sahai proofs are perfectly randomizable. Given commitments $\{\vec{C}_{\mathcal{X}_i}\}_{i=1}^n$ and a NIWI proof $\vec{\pi}_{\text{PPE}}$ that committed $\{\mathcal{X}\}_{i=1}^n$ satisfy (3), anyone can publicly compute re-randomized commitments $\{\vec{C}_{\mathcal{X}'_i}\}_{i=1}^n$ and a re-randomized proof $\vec{\pi}'_{\text{PPE}}$ of the same statement. Moreover, $\{\vec{C}_{\mathcal{X}'_i}\}_{i=1}^n$ and $\vec{\pi}'_{\text{PPE}}$ are distributed as freshly generated commitments and proof.