Published in P. Q. Nguyen and E. Oswald, Eds, *Advances in Cryptology – EUROCRYPT 2014*, vol. 8441 of *Lecture Notes in Computer Science*, pp. 514-532, Springer, 2014.

Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures

Benoît Libert[1], Thomas Peters[2] ⋆, Marc Joye[1], and Moti Yung[3]

[1] Technicolor
[2] Université catholique de Louvain, Crypto Group
[3] Google Inc. and Columbia University

Abstract. Verifiability is central to building protocols and systems with integrity. Initially, efficient methods employed the Fiat-Shamir heuristics. Since 2008, the Groth-Sahai techniques have been the most efficient in constructing non-interactive witness indistinguishable and zero-knowledge proofs for algebraic relations in the standard model. For the important task of proving membership in linear subspaces, Jutla and Roy (Asiacrypt 2013) gave significantly more efficient proofs in the quasi-adaptive setting (QA-NIZK). For membership of the row space of a $t \times n$ matrix, their QA-NIZK proofs save $\Omega(t)$ group elements compared to Groth-Sahai. Here, we give QA-NIZK proofs made of a *constant* number group elements – regardless of the number of equations or the number of variables – and additionally prove them *unbounded* simulation-sound. Unlike previous unbounded simulation-sound Groth-Sahai-based proofs, our construction does not involve quadratic pairing product equations and does not rely on a chosen-ciphertext-secure encryption scheme. Instead, we build on structure-preserving signatures with homomorphic properties. We apply our methods to design new and improved CCA2-secure encryption schemes. In particular, we build the first efficient threshold CCA-secure keyed-homomorphic encryption scheme (*i.e.*, where homomorphic operations can only be carried out using a dedicated evaluation key) with publicly verifiable ciphertexts.

1 Introduction

Non-interactive zero-knowledge proofs [6] play a fundamental role in the design of numerous cryptographic protocols. Unfortunately, until breakthrough results in the last decade [19–21], it was not known how to construct them efficiently without appealing to the random oracle methodology [5]. Groth and Sahai [21] described very efficient non-interactive witness indistinguishable (NIWI) and zero-knowledge (NIZK) proof systems for algebraic relations in groups equipped with a bilinear map. For these specific languages, the methodology of [21] does

⋆ This author was supported by the CAMUS Walloon Region Project.

not require any proof of circuit satisfiability but rather leverages the properties of homomorphic commitments in bilinear groups. As a result, the length of each proof only depends on the number of equations and the number of variables.

While dramatically more efficient than general NIZK proofs, the GS techniques remain significantly more expensive than non-interactive proofs obtained from the Fiat-Shamir heuristic [18] in the random oracle model [5]: for example, proving that t variables satisfy a system of n linear equations demands $\Theta(t+n)$ group elements where Σ-protocols allow for $O(t)$-size proofs. In addition, GS proofs are known to be malleable which, although useful in certain applications [3, 11], is undesirable when NIZK proofs serve as building blocks for non-malleable protocols. To construct chosen-ciphertext-secure encryption schemes [36], for example, the Naor-Yung/Sahai [32, 37] paradigm requires NIZK proofs satisfying a property called *simulation-soundness* [37]: informally, this property captures the inability of the adversary to prove false statements, even after having observed simulated proofs for possibly false statements of its choice.

Groth-Sahai proofs can be made simulation-sound using ideas suggested in [20, 9, 22]. However, even when starting from a linear equation, these techniques involve proofs for quadratic equations, which results in longer proofs. One-time simulation-soundness (*i.e.*, where the adversary only sees one simulated proof) is more economical to achieve as shown in [25, 27]. Jutla and Roy suggested a more efficient way to achieve a form of one-time simulation-soundness [23].

Quasi-Adaptive NIZK Proofs. For languages consisting of linear subspaces of a vector space, Jutla and Roy [24] recently showed how to significantly improve upon the efficiency of the GS paradigm in the *quasi-adaptive* setting. In quasi-adaptive NIZK proofs (QA-NIZK) for a class of languages $\{\mathcal{L}_\rho\}$ parametrized by ρ, the common reference string (CRS) is allowed to depend on the particular language $\mathcal{L}_\rho$ of which membership must be proved. At the same time, a single simulator should be effective for the whole class of languages $\{\mathcal{L}_\rho\}$. As pointed out in [24], QA-NIZK proofs are sufficient for many applications of Groth-Sahai proofs. In this setting, Jutla and Roy [24] gave very efficient QA-NIZK proofs of membership in linear subspaces. If $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ is a matrix or rank $t < n$, in order to prove membership of the language $\mathcal{L} = \{\boldsymbol{v} \in \mathbb{G}^n \mid \exists \boldsymbol{x} \in \mathbb{Z}_p^t \text{ s.t. } \boldsymbol{v} = g^{\boldsymbol{x}\cdot\mathbf{A}}\}$, the Jutla-Roy proofs only take $O(n-t)$ group elements – instead of $O(n+t)$ in [21] – at the expense of settling for computational soundness. While highly efficient in the case $t \approx n$, these proofs remain of linear size in n and may result in long proofs when $t \ll n$, as is the case in, *e.g.*, certain applications of the Naor-Yung paradigm [9]. In the general case, we are still lacking a method for building proofs of size $O(t)$ – at least without relying on non-falsifiable assumptions [31] – which contrasts with the situation in the random oracle model.

The problem is even harder if we aim for simulation-soundness. While the Jutla-Roy solutions [24] nicely interact with their one-time simulation-sound proofs [23], they do not seem to readily extend into unbounded simulation-sound (USS) proofs (where the adversary can see an arbitrary number of simulated proofs before outputting a proof of its own) while retaining the same efficiency. For this reason, although they can be applied in specific cases like [9], we can-

not always use them in a modular way to build IND-CCA2-secure encryption schemes when security definitions involve many challenge ciphertexts.

Our Contributions. Recently [28], it was observed that structure-preserving signatures (SPS) [2, 1] with homomorphic properties have unexpected applications in the design of non-malleable structure-preserving commitments. Here, we greatly extend their range of applications and demonstrate that they can surprisingly be used (albeit non-generically) in the design of strongly non-malleable primitives like simulation-sound proofs and CCA2-secure cryptosystems.

Concretely, we describe unbounded simulation-sound QA-NIZK proofs of *constant-size* for linear subspaces. The length of a proof does not depend on the number of equations or the number of variables, but only on the underlying assumption. Like those of [24], our proofs are computationally sound under standard assumptions. Somewhat surprisingly, they are even asymptotically shorter than random-oracle-based proofs derived from Σ-protocols.

Moreover, our construction provides *unbounded* simulation-soundness. Under the Decision Linear assumption [7], we obtain QA-NIZK arguments made of 15 group elements and a one-time signature with its verification key. As it turns out, it is also the first unbounded simulation-sound proof system that does not involve quadratic pairing product equations or a CCA2-secure encryption scheme. Efficiency comparisons show that we only need 20 group elements per proof where the best USS extension [9] of Groth-Sahai costs $6t + 2n + 52$ group elements. Under the k-linear assumption, the proof length becomes $O(k^2)$ and thus avoids any dependency on the subspace dimension. Our system builds on the linearly homomorphic structure-preserving signatures of Libert *et al.* [28], which can sign vectors of group elements without knowing their discrete logarithms.

For applications, like CCA2 security [32, 37], where only one-time simulation-soundness is needed, we further optimize our proof system and obtain a relatively simulation-sound QA-NIZK proof system, as defined in [23], with constant-size proofs. Under the DLIN assumption (resp. the k-linear assumption), we achieve relative simulation-soundness with only 4 (resp. $k + 2$) group elements!

As a first application of our USS proofs, we build a chosen-ciphertext-secure keyed-homomorphic system with threshold decryption. Keyed-homomorphic encryption is a primitive, suggested by Emura *et al.* [16], where homomorphic ciphertext manipulations are only possible to a party holding a devoted evaluation key SK_h which, by itself, does not enable decryption. The scheme should provide IND-CCA2 security when the evaluation key is unavailable to the adversary and remain IND-CCA1 secure when SK_h is exposed. Other approaches to reconcile homomorphism and non-malleability were taken in [33–35, 8, 11] but they inevitably satisfy weaker security notions than adaptive chosen-ciphertext security [36]. The results of [16] showed that CCA2-security does not rule out homomorphicity when the capability to compute over encrypted data is restricted.

Emura *et al.* [16] gave chosen-ciphertext-secure keyed-homomorphic schemes based on hash proof systems [13]. However, these do not readily enable threshold decryption – as would be desirable in voting protocols – since valid ciphertexts are not publicly recognizable, which makes it harder to prove CCA security in

the threshold setting. Also, these solutions are not known to satisfy the strongest security definition of [16]. The reason is that this definition seemingly requires a form of unbounded simulation-soundness. Our QA-NIZK proofs fulfill this requirement and provide an efficient CCA2-secure threshold keyed-homomorphic system where ciphertexts are 65% shorter than in instantiations of the same high-level idea using previous simulation-sound proofs. The resulting system can be used in scalable elections [26] where the vote aggregation is organized within a hierarchy and can only be performed by tallying authorities in a certain order.

Using our relatively simulation-sound QA-NIZK proofs, we build new adaptively secure non-interactive threshold cryptosystems [14, 15] with CCA2 security and improved efficiency. The constructions of [27] were improved by Escala *et al.* [17]. So far, the most efficient solution is obtained from the Jutla-Roy results [23, 24] via relatively sound proofs [23]. Using our relatively sound QA-NIZK proofs, we shorten ciphertexts by $\Theta(k)$ elements under the k-linear assumption.

Our Techniques. In our unbounded simulation-sound proofs, each QA-NIZK proof can be seen as a Groth-Sahai NIWI proof of knowledge of a one-time linearly homomorphic signature on the vector that allegedly belongs to the linear subspace. Here, the NIWI proof is generated for a Groth-Sahai CRS that depends on the verification key of a one-time signature (following an idea of Malkin *et al.* [30] which extends Waters' techniques [39]), the private key of which is used to sign the entire proof so as to prevent re-randomizations. The reason why it provides unbounded simulation-soundness is that, with non-negligible probability, the CRS is perfectly hiding on all simulated proofs and extractable in the adversarially-generated fake proof. Hence, if the adversary manages to prove membership of a vector outside the linear subspace, the reduction is able to extract a homomorphic signature that it would not have been able to compute itself, thereby breaking the DLIN assumption. At a high level, the system can be seen as a two-tier proof system made of a non-malleable proof of knowledge of a malleable proof of membership.

In our optimized relatively-sound proofs, we adapt ideas of Jutla and Roy [23] and combine the one-time linearly homomorphic signature of [28] with a smooth-projective hash function [13].

Our threshold keyed-homomorphic scheme combines a hash proof system and a publicly verifiable USS proof that the ciphertext is well-formed. The keyed-homomorphic property is achieved by using the simulation trapdoor of the proof system as an evaluation key SK_h, allowing the evaluator to generate proofs without the witnesses. As implicitly done in [16] using hash proof systems, the simulation trapdoor is used in the scheme and not only in the security proof.

2 Background and Definitions

2.1 Quasi-Adaptive NIZK Proofs

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS is allowed to depend on the specific language for which proofs have to be generated.

The CRS is divided into a fixed part Γ, produced by an algorithm K_0, and a language-dependent part ψ. However, there should be a single simulator for the entire class of languages.

Let λ be a security parameter. For public parameters Γ produced by K_0, let $\mathcal{D}_\Gamma$ be a probability distribution over a set of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ with an associated language $\mathcal{L}_\rho = \{x \mid \exists w : R_\rho(x, w) = 1\}$.

We consider proof systems where the prover and the verifier both take a label lbl as additional input. For example, this label can be the message-carrying part of an Elgamal-like encryption. Formally, a tuple of algorithms $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a QA-NIZK proof system for $\mathcal{R}$ if there exists a PPT simulator $(\mathsf{S}_1, \mathsf{S}_2)$ such that, for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, we have the following properties:

Quasi-Adaptive Completeness:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, w, \mathsf{lbl}) \leftarrow \mathcal{A}_1(\Gamma, \psi, \rho);\\ \pi \leftarrow \mathsf{P}(\psi, x, w, \mathsf{lbl}) : \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \text{ if } R_\rho(x, w) = 1] = 1.$$

Quasi-Adaptive Soundness:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_2(\Gamma, \psi, \rho) :\\ \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1 \ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)] \in \mathsf{negl}(\lambda).$$

Quasi-Adaptive Zero-Knowledge:

$$\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{P}(\psi,\cdot,\cdot)}(\Gamma, \psi, \rho) = 1] \approx\\ \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho)\ :\ \mathcal{A}_3^{\mathsf{S}(\psi,\tau_{sim},\cdot,\cdot,\cdot)}(\Gamma, \psi, \rho) = 1],$$

where

- $\mathsf{P}(\psi, ., ., .)$ emulates the actual prover. It takes as input (x, w) and lbl and outputs a proof π if $(x, w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $\mathsf{S}(\psi, \tau_{sim}, ., ., .)$ is an oracle that takes as input (x, w) and lbl. It outputs a simulated proof $\mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ if $(x, w) \in R_\rho$ and $\perp$ if $(x, w) \notin R_\rho$.

We assume that the CRS ψ contains an encoding of ρ, which is thus available to V. The definition of Quasi-Adaptive Zero-Knowledge requires a single simulator for the entire family of relations $\mathcal{R}$.

2.2 Simulation-Soundness and Relative Soundness

It is often useful to have a property called *simulation-soundness*, which requires that the adversary be unable to prove false statements even after having seen simulated proofs for possibly false statements.

Unbounded Simulation-Soundness: For any PPT adversary $\mathcal{A}_4$,

$$\Pr[\ \Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau_{sim}) \leftarrow \mathsf{S}_1(\Gamma, \rho);\\ (x, \pi, \mathsf{lbl}) \leftarrow \mathcal{A}_4^{\mathsf{S}_2(\psi,\tau_{sim},\cdot,\cdot)}(\Gamma, \psi, \rho)\ :\ \mathsf{V}(\psi, x, \pi, \mathsf{lbl}) = 1\\ \wedge\ \neg(\exists w : R_\rho(x, w) = 1)\ \wedge\ (x, \pi, \mathsf{lbl}) \notin Q] \in \mathsf{negl}(\lambda),$$

where the adversary is allowed unbounded access to an oracle $\mathsf{S}_2(\psi, \tau, ., .)$ that takes as input statement-label pairs (x, lbl) (where x may be outside $\mathcal{L}_\rho$) and outputs simulated proofs $\pi \leftarrow \mathsf{S}_2(\psi, \tau_{sim}, x, \mathsf{lbl})$ before updating the set $Q = Q \cup \{(x, \pi, \mathsf{lbl})\}$, which is initially empty.

In the weaker notion of one-time simulation-soundness, only one query to the S_2 oracle is allowed.

In some applications, one may settle for a weaker notion, called *relative soundness* by Jutla and Roy [23], which allows for more efficient proofs, especially in the single-theorem case. Informally, relatively sound proof systems involve both a public verifier *and* a private verification algorithm, which has access to a trapdoor. For hard languages, the two verifiers should almost always agree on any adversarially-created proof. Moreover, the private verifier should not accept a non-trivial proof for a false statement, even if the adversary has already seen proofs for false statements.

A labeled single-theorem relatively sound QA-NIZK proof system is comprised of a quasi-adaptive labeled proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ along with an efficient private verifier W and an efficient simulator $(\mathsf{S}_1, \mathsf{S}_2)$. Moreover, the following properties should hold for any PPT adversaries $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$.

Quasi Adaptive Relative Single-Theorem Zero-Knowledge:

$$\begin{aligned}
&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ \psi \leftarrow \mathsf{K}_1(\Gamma, \rho);\ (x, w, \mathsf{lbl}, s) \leftarrow \\
&\qquad \mathcal{A}_1^{\mathsf{V}(\psi,.,.)}(\Gamma, \psi, \rho); \pi \leftarrow \mathsf{P}(\psi, \rho, x, w, \mathsf{lbl}) : \mathcal{A}_2^{\mathsf{V}(\psi,.,.)}(\pi, s) = 1] \\
&\approx \Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau) \leftarrow \mathsf{S}_1(\Gamma, \rho);\ (x, w, \mathsf{lbl}, s) \leftarrow \\
&\qquad \mathcal{A}_1^{\mathsf{W}(\psi,\tau,.,.)}(\Gamma, \psi, \rho); \pi \leftarrow \mathsf{S}_2(\psi, \rho, \tau, x, \mathsf{lbl}) : \mathcal{A}_2^{\mathsf{W}(\psi,\tau,.,.)}(\pi, s) = 1],
\end{aligned}$$

Here, $\mathcal{A}_1$ is restricted to choosing (x, w) such that $R_\rho(x, w) = 1$.

Quasi Adaptive Relative Single-Theorem Simulation-Soundness:

$$\begin{aligned}
&\Pr[\Gamma \leftarrow \mathsf{K}_0(\lambda);\ \rho \leftarrow D_\Gamma;\ (\psi, \tau) \leftarrow \mathsf{S}_1(\Gamma, \rho);\ (x, \mathsf{lbl}, s) \leftarrow \mathcal{A}_3^{\mathsf{W}(\psi,\tau,.,.)}(\Gamma, \psi, \rho); \\
&\pi \leftarrow \mathsf{S}_2(\psi, \rho, \tau, x, \mathsf{lbl}) : (x', \mathsf{lbl}', \pi') \leftarrow \mathcal{A}_4^{\mathsf{W}(\psi,\tau,.,.)}(s, \pi) : (x, \pi, \mathsf{lbl}) \neq (x', \pi', \mathsf{lbl}') \\
&\qquad \wedge \quad \nexists w' \text{ s.t. } R_\rho(x', w') = 1 \ \wedge\ \mathsf{W}(\psi, \tau, x', \mathsf{lbl}', \pi') = 1] \in \mathsf{negl}(\lambda)
\end{aligned}$$

2.3 Definitions for Threshold Keyed-Homomorphic Encryption

A (t, N)-threshold keyed-homomorphic encryption scheme consists of the following algorithms.

Keygen(λ, t, N)**:** inputs a security parameter λ and integers $t, N \in \mathsf{poly}(\lambda)$ (with $1 \leq t \leq N$), where N is the number of decryption servers and $t \leq N$ is the decryption threshold. It outputs a public key PK, a homomorphic evaluation key SK_h, a vector of private key shares $\mathbf{SK}_d = (SK_{d,1}, \ldots, SK_{d,N})$ and a vector of verification keys $\mathbf{VK} = (VK_1, \ldots, VK_N)$. For each i, the decryption server i is given the share $(i, SK_{d,i})$. The verification key VK_i will be used to check the validity of decryption shares generated using $SK_{d,i}$.

Encrypt(PK, M)**:** takes a input a public key PK and a plaintext M. It outputs a ciphertext C.

Ciphertext-Verify(PK, C)**:** takes as input a public key PK and a ciphertext C. It outputs 1 if C is deemed valid w.r.t. PK and 0 otherwise.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on input of a public key PK, a ciphertext C and a private-key share $(i, SK_{d,i})$, this (possibly randomized) algorithm outputs a special symbol $(i, \perp)$ if **Ciphertext-Verify**$(PK, C) = 0$. Otherwise, it outputs a decryption share $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify(PK, VK_i, C, μ_i)**:** takes in PK, the verification key VK_i, a ciphertext C and a purported decryption share $\mu_i = (i, \hat{\mu}_i)$. It outputs either 1 or 0. In the former case, μ_i is said to be a *valid* decryption share. We adopt the convention that $(i, \perp)$ is an invalid decryption share.

Combine$(PK, \mathbf{VK}, C, \{\mu_i\}_{i\in S})$**:** takes as input $(PK, \mathbf{VK}, C)$ and a t-subset $S \subset \{1, \ldots, N\}$ with decryption shares $\{\mu_i\}_{i\in S}$. It outputs either a plaintext M or $\perp$ if $\{\mu_i\}_{i\in S}$ contains invalid shares.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** takes as input PK, the evaluation key SK_h and ciphertexts $C^{(1)}$, $C^{(2)}$. If **Ciphertext-Verify**$(PK, C^{(j)}) = 0$ for some $j \in \{1, 2\}$, the algorithm returns $\perp$. Otherwise, it conducts a binary homomorphic operation over $C^{(1)}$ and $C^{(2)}$ and outputs a ciphertext C.

The above syntax assumes a trusted dealer. It generalizes that of ordinary threshold cryptosystems. By setting $SK_h = \varepsilon$ and discarding the evaluation algorithm, we obtain a classical threshold system.

Definition 1. *A threshold keyed-homomorphic public-key cryptosystem is secure against chosen-ciphertext attacks (or KH-CCA secure) if no PPT adversary has noticeable advantage in this game:*

1. *The challenger runs* **Keygen**(λ) *to obtain a public key PK, vectors $\mathbf{SK}_d$ and $\mathbf{VK}$ and a homomorphic evaluation key SK_h. It gives PK and $\mathbf{VK}$ to the adversary $\mathcal{A}$ and keeps $(SK_h, \mathbf{SK}_d)$ to itself. In addition, the challenge initializes a set $\mathcal{D} \leftarrow \emptyset$, which is initially empty.*

2 *On multiple occasions, $\mathcal{A}$ adaptively invokes the following oracles:*
   - *Corruption query: at any time, $\mathcal{A}$ may decide to corrupt a decryption server. To this end, it specifies an index $i \in \{1, \ldots, N\}$ and obtains the private key share $SK_{d,i}$.*
   - *Evaluation query: $\mathcal{A}$ can invoke the evaluation oracle $\mathsf{Eval}(SK_h, .)$ on a pair $(C^{(1)}, C^{(2)})$ of ciphertexts of its choice. If there exists $j \in \{1, 2\}$ such that* **Ciphertext-Verify**$(PK, C^{(j)}) = 0$*, return $\perp$. Otherwise, the oracle $\mathsf{Eval}(SK_h, .)$ computes $C \leftarrow$* **Eval**$(SK_h, C^{(1)}, C^{(2)})$ *and returns C. In addition, if $C^{(1)} \in \mathcal{D}$ or $C^{(2)} \in \mathcal{D}$, it sets $\mathcal{D} \leftarrow \mathcal{D} \cup \{C\}$.*
   - *Reveal query: at any time, $\mathcal{A}$ may also decide to corrupt the evaluator by invoking the RevHK oracle on a unique occasion. The oracle responds by returning SK_h.*
   - *Partial decryption query: $\mathcal{A}$ can also choose arbitrary ciphertexts C and indexes $i \in \{1, \ldots, n\}$. If* **Ciphertext-Verify**$(PK, C) = 0$ *or if $C \in \mathcal{D}$, the oracle returns $\perp$. Otherwise, the oracle returns the decryption share $\mu_i \leftarrow$* **Share-Decrypt**$(PK, i, SK_{d,i}, C)$.

3. *The adversary $\mathcal{A}$ chooses two equal-length messages M_0, M_1 and obtains $C^\star = \mathbf{Encrypt}(PK, M_\beta)$ for some random bit $\beta \xleftarrow{R} \{0,1\}$. In addition, the challenger sets $\mathcal{D} \leftarrow \mathcal{D} \cup \{C^\star\}$.*
4. *$\mathcal{A}$ makes further queries as in step 2 with some restrictions. Namely, $\mathcal{A}$ cannot corrupt more than $t-1$ servers throughout the entire game. Moreover, if $\mathcal{A}$ chooses to obtain SK_h (via the* RevHK *oracle) at some point, no more post-challenge decryption query is allowed beyond that point.*
5. *$\mathcal{A}$ outputs a bit β' and is deemed successful if $\beta' = \beta$. As usual, $\mathcal{A}$'s advantage is measured as the distance $\mathbf{Adv}(\mathcal{A}) = |\Pr[\beta' = \beta] - \frac{1}{2}|$.*

Note that, even if $\mathcal{A}$ chooses to obtain SK_h immediately after having seen the public key PK, it still has access to the decryption oracle *before* the challenge phase. In other words, the scheme should remain IND-CCA1 if $\mathcal{A}$ is given PK and SK_h at the outset of the game. After the challenge phase, decryption queries are allowed until the moment when the adversary obtains SK_h.

In [16], Emura *et al.* suggested a weaker definition where the adversary is not allowed to query the evaluation oracle on derivatives of the challenge ciphertext. As a consequence, the set $\mathcal{D}$ is always the singleton $\{C^\star\}$ after step 3. In this paper, we will stick to the stronger definition.

2.4 Hardness Assumptions

We will use symmetric bilinear maps $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ over groups of prime order p, but extensions to the asymmetric setting $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ are possible.

Definition 2 ([7]). *The* **Decision Linear Problem** *(DLIN) in $\mathbb{G}$, is to distinguish the distributions $(g^a, g^b, g^{ac}, g^{bd}, g^{c+d})$ and $(g^a, g^b, g^{ac}, g^{bd}, g^z)$, where $a, b, c, d \xleftarrow{R} \mathbb{Z}_p$, $z \xleftarrow{R} \mathbb{Z}_p$.*

We sometimes use the Simultaneous Double Pairing (SDP) assumption, which is weaker than DLIN. As noted in [10], any algorithm solving SDP immediately yields a DLIN distinguisher.

Definition 3. *The* **Simultaneous Double Pairing problem** *(SDP) in $(\mathbb{G}, \mathbb{G}_T)$ is, given group elements $(g_z, g_r, h_z, h_u) \in \mathbb{G}^4$, to find a non-trivial triple $(z, r, u) \in \mathbb{G}^3 \backslash \{(1_\mathbb{G}, 1_\mathbb{G}, 1_\mathbb{G})\}$ such that $e(g_z, z) \cdot e(g_r, r) = 1_{\mathbb{G}_T}$ and $e(h_z, z) \cdot e(h_u, u) = 1_{\mathbb{G}_T}$.*

2.5 Linearly Homomorphic Structure-Preserving Signatures

Linearly homomorphic SPS schemes are homomorphic signatures where messages and signatures live in the domain group $\mathbb{G}$ (see [28] for syntactic definitions) of a bilinear map. Libert *et al.* [28] described the following one-time construction and proved its security under the SDP assumption. By "one-time", we mean that only one linear subspace can be signed using a given key pair.

Keygen(λ, n): given a security parameter λ and the dimension $n \in \mathbb{N}$ of vectors to be signed, choose bilinear group $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$. Choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$. Then, for $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z{}^{\chi_i} g_r{}^{\gamma_i}$ and $h_i = h_z{}^{\chi_i} h_u{}^{\delta_i}$. The private key is $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ while the public key is $\mathsf{pk} = \big(g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^n\big)$.

Sign($\mathsf{sk}, (M_1, \ldots, M_n)$): to sign $(M_1, \ldots, M_n) \in \mathbb{G}^n$ using $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$, return $(z, r, u) = \big(\prod_{i=1}^n M_i^{-\chi_i}, \prod_{i=1}^n M_i^{-\gamma_i}, \prod_{i=1}^n M_i^{-\delta_i}\big) \in \mathbb{G}^3$.

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

Verify($\mathsf{pk}, \sigma, (M_1, \ldots, M_n)$): given a signature $\sigma = (z, r, u) \in \mathbb{G}^3$ 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, u) satisfy the equalities $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i, M_i)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i, M_i)$.

One particularity of this scheme is that, even if the private key is available, it is difficult to find two distinct signatures on the same vector if the SDP assumption holds: by dividing out the two signatures, one obtains the solution of an SDP instance (g_z, g_r, h_z, h_u) contained in the public key.

Two constructions of full-fledged (as opposed to one-time) linearly homomorphic SPS were given in [28]. One of these will serve as a basis for our proof system. In these constructions, all algorithms additionally input a tag which identifies the dataset that vectors belongs to. Importantly, only vectors associated with the same tag can be homomorphically combined.

3 Unbounded Simulation-Sound Quasi-Adaptive NIZK Arguments

In the following, vectors are always considered as row vectors unless stated otherwise. If $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ is a matrix, we denote by $g^\mathbf{A} \in \mathbb{G}^{t \times n}$ the matrix obtained by exponentiating g using the entries of $\mathbf{A}$.

We consider public parameters $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$ consisting of bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ with a generator $g \in \mathbb{G}$. Like [24], we will consider languages $\mathcal{L}_\rho = \{g^{\boldsymbol{x} \cdot \mathbf{A}} \in \mathbb{G}^n \mid \boldsymbol{x} \in \mathbb{Z}_p^t\}$ that are parametrized by $\rho = g^\mathbf{A} \in \mathbb{G}^{t \times n}$, where $\mathbf{A} \in \mathbb{Z}_q^{t \times n}$ is a $t \times n$ matrix of rank $t < n$.

As in [24], we assume that the distribution $\mathcal{D}_\Gamma$ is efficiently samplable: there exists a PPT algorithm which outputs a pair $(\rho, \mathbf{A})$ describing a relation R_ρ and its associated language $\mathcal{L}_\rho$ according to $\mathcal{D}_\Gamma$. One example of such a distribution is obtained by picking a uniform matrix $\mathbf{A} \xleftarrow{R} \mathbb{Z}_p^{t \times n}$ – which has full rank with overwhelming probability – and setting $\rho = g^\mathbf{A}$.

Our construction builds on the homomorphic signature recalled in Section 2.5. Specifically, the language-dependent CRS ψ contains one-time linearly homomorphic signatures on the rows of the matrix $\rho \in \mathbb{G}^{t \times n}$. For each vector $\boldsymbol{v} \in \mathcal{L}_\rho$, the prover can use the witness $\boldsymbol{x} \in \mathbb{Z}_p^t$ to derive and prove knowledge of a one-time homomorphic signature (z, r, u) on $\boldsymbol{v}$. This signature (z, r, u) is

already a QA-NIZK proof of membership but it is not simulation-sound. To acquire this property, we follow [30] and generate a NIWI proof of knowledge of (z, r, u) for a Groth-Sahai CRS that depends on the verification key of an ordinary one-time signature. The latter's private key is used to sign the NIWI proof so as to prevent unwanted proof manipulations. Using the private key of the homomorphic one-time signature as a trapdoor, the simulator is also able to create proofs for vectors $\boldsymbol{v} \notin \mathcal{L}_\rho$. Due to the use of perfectly NIWI proofs, these fake proofs do not leak any more information about the simulation key than the CRS does. At the same time, the CRS can be prepared in such a way that, with non-negligible probability, it becomes perfectly binding on an adversarially-generated proof, which allows extracting a non-trivial signature on a vector $\boldsymbol{v} \notin \mathcal{L}_\rho$.

Like [24], our quasi-adaptive NIZK proof system $(\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V})$ is a split CRS construction in that K_1 can be divided into two algorithms $(\mathsf{K}_{10}, \mathsf{K}_{11})$. The first one K_{10} outputs some state information s and a first CRS $\mathbf{CRS}_2$ which is only used by the verifier and does not depend on the language $\mathcal{L}_\rho$. The second part K_{11} of K_1 inputs the state information s and the output of Γ of K_0 and outputs $\mathbf{CRS}_1$ which is only used by the prover. The construction goes as follows.

$\mathsf{K}_0(\lambda)$**:** choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$

The dimensions (t, n) of the matrix $\mathbf{A} \in \mathbb{Z}_p^{t \times n}$ can be either fixed or part of the language, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as a matrix $\rho = \big(G_{i,j}\big)_{1 \leq i \leq t,\ 1 \leq j \leq n} \in \mathbb{G}^{t \times n}$.

1. Generate a key pair $(\mathsf{pk}_{ots}, \mathsf{sk}_{ots})$ for the homomorphic signature of Section 2.5 in order to sign vectors of $\mathbb{G}^n$ and incorporate a set of Groth-Sahai common reference strings in the public key pk_{ots}. In details:
   a. Choose $g_z, g_r, h_z, h_u \xleftarrow{R} \mathbb{G}$. For $i = 1$ to n, pick $\chi_i, \gamma_i, \delta_i \xleftarrow{R} \mathbb{Z}_p$ and compute $g_i = g_z^{\chi_i} g_r^{\gamma_i}$ and $h_i = h_z^{\chi_i} h_u^{\delta_i}$.
   b. Generate $L + 1$ Groth-Sahai CRSes, for some $L \in \mathsf{poly}(\lambda)$. To this end, choose $f_1, f_2 \xleftarrow{R} \mathbb{G}$ and define vectors $\boldsymbol{f}_1 = (f_1, 1, g) \in \mathbb{G}^3$, $\boldsymbol{f}_2 = (1, f_2, g) \in \mathbb{G}^3$. Then, pick $\boldsymbol{f}_{3,i} \xleftarrow{R} \mathbb{G}^3$ for $i = 0$ to L.

   Let $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^n$ be the private key and the public key is

$$\mathsf{pk}_{ots} = \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^n,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big) \Big).$$

2. Use sk_{ots} to generate one-time homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^t$ on the rows $\boldsymbol{\rho}_i = (G_{i1}, \ldots, G_{in}) \in \mathbb{G}^n$ of ρ. These are obtained as $(z_i, r_i, u_i) = \big(\prod_{j=1}^n G_{i,j}^{-\chi_j}, \prod_{j=1}^n G_{i,j}^{-\gamma_j}, \prod_{j=1}^n G_{i,j}^{-\delta_j}\big)$ for all $i \in \{1, \ldots, t\}$.
3. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings.
4. The CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$ consists of two parts which are defined as

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{ots},\ \{(z_i, r_i, u_i)\}_{i=1}^t,\ \Sigma\Big), \qquad \mathbf{CRS}_2 = \Big(\mathsf{pk}_{ots},\ \Sigma\Big),$$

while the simulation trapdoor τ_{sim} is $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{n}$.

$\mathsf{P}(\Gamma, \psi, \boldsymbol{v}, x, \mathsf{lbl})$**:** given $\boldsymbol{v} \in \mathbb{G}^n$ and a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x}\cdot\mathbf{A}}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$.

1. Using $\{(z_j, r_j, u_j)\}_{j=1}^{t}$, derive a one-time linearly homomorphic signature (z, r, u) on $\boldsymbol{v}$. Namely, set $z = \prod_{i=1}^{t} z_i^{x_i}$, $r = \prod_{i=1}^{t} r_i^{x_i}$ and $u = \prod_{i=1}^{t} u_i^{x_i}$.
2. Using $\mathsf{SVK} \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^{L} \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u$ to the components of (z, r, u) along with NIWI proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ that (z, r, u) is a valid homomorphic signature for $\boldsymbol{v}$. Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting commitments and proofs.
3. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}))$ and output

$$\pi = (\mathsf{SVK}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \tag{1}$$

$\mathsf{V}(\Gamma, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** parse π as per (1). Return 1 if (i) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ forms a valid NIWI proof for the Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$; (ii) $\mathcal{V}(\mathsf{SVK}, (\boldsymbol{v}, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \mathsf{lbl}), \sigma) = 1$. If either condition fails to hold, return 0.

In order to simulate a proof for a given vector $\boldsymbol{v} \in \mathbb{G}^n$, the simulator uses $\tau_{sim} = \mathsf{sk}_{ots}$ to generate a fresh one-time homomorphic signature on $\boldsymbol{v} \in \mathbb{G}^n$ and proceeds as in steps 2-3 of algorithm P.

The proof π only consists of 15 group elements and a one-time pair (SVK, σ). Remarkably, its length does not depend on the number of equations n or the number of variables t. In comparison, Groth-Sahai proofs already require $3t+2n$ group elements in their basic form and become even more expensive when it comes to achieve unbounded simulation-soundness. The Jutla-Roy techniques [24] reduce the proof length to $2(n-t)$ elements – which only competes with our proofs when $t \approx n$ – but it is unclear how to extend them to get unbounded simulation-soundness without affecting their efficiency. Our CRS consists of $O(t+n+L)$ group elements against $O(t(n-t))$ in [24].

Interestingly, the above scheme even outperforms Fiat-Shamir-like proofs derived from Σ-protocols which would give $O(t)$-size proofs here. The construction readily extends to rely on the k-linear assumption for $k > 2$. In this case, the proof comprises $(k+1)(2k+1)$ elements and its size thus only depends on k.

The verification algorithm only involves *linear* pairing product equations whereas all known unbounded simulation-sound extensions of GS proofs require either quadratic equations or a linearization step involving extra variables.

We finally remark that, if we give up the simulation-soundness property, the proof length drops to $k+1$ group elements under the k-linear assumption.

Theorem 1. *The scheme is an unbounded simulation-sound QA-NIZK proof system if the DLIN assumption holds in $\mathbb{G}$ and Σ is strongly unforgeable.* (The proof is given in the full version of the paper [29]).

We note that the above construction is not tightly secure as the gap between the simulation-soundness adversary's advantage and the probability to break

the DLIN assumption depends on the number of simulated proofs obtained by the adversary. For applications like tightly secure public-key encryption [22], it would be interesting to modify the proof system to obtain tight security.

4 Single-Theorem Relatively Sound Quasi-Adaptive NIZK Arguments

In applications where single-theorem relatively sound NIZK proofs suffice, we can further improve the efficiency. Under the k-linear assumption, the proof length reduces from $O(k^2)$ elements to $O(k)$ elements. Under the DLIN assumption, each proof fits within 4 elements and only costs $2n+6$ pairings to verify. In comparison, the verifier needs $2(n-t)(t+2)$ pairing evaluations in [24].

As in [23], we achieve relative soundness using smooth projective hash functions [13]. To this end, we encode the matrix $\rho \in \mathbb{G}^{t\times n}$ as a $2t \times (2n+1)$ matrix.

$\mathsf{K}_0(\lambda)$**:** choose groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$ with $g \xleftarrow{R} \mathbb{G}$. Then, output $\Gamma = (\mathbb{G}, \mathbb{G}_T, g)$.

Again, the dimensions of $\mathbf{A} \in \mathbb{Z}_p^{t\times n}$ can be either fixed or part of $\mathcal{L}_\rho$, so that t, n can be given as input to the CRS generation algorithm K_1.

$\mathsf{K}_1(\Gamma, \rho)$**:** parse Γ as $(\mathbb{G}, \mathbb{G}_T, g)$ and ρ as $\rho = \big(G_{ij}\big)_{1\leq i\leq t,\ 1\leq j\leq n} \in \mathbb{G}^{t\times n}$.

1. Choose vectors $\boldsymbol{d} = (d_1, \ldots, d_n) \xleftarrow{R} \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \xleftarrow{R} \mathbb{Z}_p^n$. Define $\boldsymbol{W} = (W_1, \ldots, W_t) = g^{\mathbf{A}\cdot\boldsymbol{d}^\top} \in \mathbb{G}^t$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_t) = g^{\mathbf{A}\cdot\boldsymbol{e}^\top} \in \mathbb{G}^t$, which will be used to define a projective hash function.
2. Generate a key pair for the one-time homomorphic signature of Section 2.5 in order to sign vectors in $\mathbb{G}^{2n+1}$. Let $\mathsf{sk}_{ots} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^{2n+1}$ be the private key and let $\mathsf{pk}_{ots} = \big((\mathbb{G}, \mathbb{G}_T), g_z, g_r, h_z, h_u, \{(g_i, h_i)\}_{i=1}^{2n+1}\big)$ be the corresponding public key.
3. Use sk_{ots} to generate one-time homomorphic signatures $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$ on the independent vectors below, which are obtained from the rows of the matrix $\rho = \big(G_{i,j}\big)_{1\leq i\leq t,\ 1\leq j\leq n}$.

$$\begin{aligned} \boldsymbol{H}_{2i-1} &= (G_{i,1}, \ldots, G_{i,n}, Y_i,\ 1\ ,\ \ \ldots\ \ ,1\ ) \in \mathbb{G}^{2n+1} \qquad i \in \{1, \ldots, t\} \\ \boldsymbol{H}_{2i} &= (1\ ,\ \ \ldots\ \ ,1\ , W_i, G_{i,1}, \ldots, G_{i,n}) \in \mathbb{G}^{2n+1} \end{aligned}$$

4. Choose a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_p$.
5. The CRS $\psi = (\mathbf{CRS}_1, \mathbf{CRS}_2)$ consists of a first part $\mathbf{CRS}_1$ that is only used by the prover and a second part $\mathbf{CRS}_2$ which is only used by the verifier. These are defined as $\mathbf{CRS}_2 = \big(\mathsf{pk}_{ots}, \boldsymbol{W}, \boldsymbol{Y}, H\big)$ and

$$\mathbf{CRS}_1 = \Big(\rho,\ \mathsf{pk}_{ots}, \boldsymbol{W}, \boldsymbol{Y}, \{(z_i, r_i, u_i)\}_{i=1}^{2t},\ H\Big).$$

The simulation trapdoor τ_{sim} is sk_{ots} and the private verification trapdoor is $\tau_v = \{\boldsymbol{d}, \boldsymbol{e}\}$.

$\mathsf{P}(\Gamma, \psi, \boldsymbol{v}, x, \mathsf{lbl})$**:** given $\boldsymbol{v} \in \mathbb{G}^n$, a witness $\boldsymbol{x} = (x_1, \ldots, x_t) \in \mathbb{Z}_p^t$ such that $\boldsymbol{v} = g^{\boldsymbol{x} \cdot \mathbf{A}}$ and a label lbl, compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$. Then, using $\{(z_i, r_i, u_i)\}_{i=1}^{2t}$, derive a one-time homomorphic signature (z, r, u) on the vector $\tilde{\boldsymbol{v}} = \big(v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha\big) \in \mathbb{G}^{2n+1}$, where $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$. Namely, output $\pi = (z, r, u, \pi_0) \in \mathbb{G}^4$, where $z = \prod_{i=1}^t (z_{2i-1} \cdot z_{2i}^\alpha)^{x_i}$, $r = \prod_{i=1}^t (r_{2i-1} \cdot r_{2i}^\alpha)^{x_i}$, $u = \prod_{i=1}^t (u_{2i-1} \cdot u_{2i}^\alpha)^{x_i}$ and $\pi_0 = \prod_{i=1}^t (W_i^\alpha Y_i)^{x_i}$.

$\mathsf{V}(\Gamma, \psi, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** parse $\boldsymbol{v}$ as $(v_1, \ldots, v_n) \in \mathbb{G}^n$ and π as $(z, r, u, \pi_0) \in \mathbb{G}^4$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl})$ and return 1 if and only if (z, r, u) is a valid signature on $\tilde{\boldsymbol{v}} = (v_1, \ldots, v_n, \pi_0, v_1^\alpha, \ldots, v_n^\alpha) \in \mathbb{G}^{2n+1}$. Namely, it should satisfy the equalities $1_{\mathbb{G}_T} = e(g_z, z) \cdot e(g_r, r) \cdot \prod_{i=1}^n e(g_i \cdot g_{i+n+1}^\alpha, v_i) \cdot e(g_{n+1}, \pi_0)$ and $1_{\mathbb{G}_T} = e(h_z, z) \cdot e(h_u, u) \cdot \prod_{i=1}^n e(h_i \cdot h_{i+n+1}^\alpha, v_i) \cdot e(h_{n+1}, \pi_0)$.

$\mathsf{W}(\Gamma, \psi, \tau_v, \boldsymbol{v}, \pi, \mathsf{lbl})$**:** given $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{G}^n$, parse π as $(z, r, u, \pi_0) \in \mathbb{G}^4$ and τ_v as $\{\boldsymbol{d}, \boldsymbol{e}\}$, with $\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_p^n$ and $\boldsymbol{e} = (e_1, \ldots, e_n) \in \mathbb{Z}_p^n$. Compute $\alpha = H(\rho, \boldsymbol{v}, \mathsf{lbl}) \in \mathbb{Z}_p$ and return 0 if the public verification test V fails. Otherwise, return 1 if $\pi_0 = \prod_{j=1}^n v_j^{e_j + \alpha d_j}$ and 0 otherwise.

We note that, while the proving algorithm is deterministic, each statement has many valid proofs. However, finding two valid proofs for the same statement is computationally hard, as will be shown in the proof of Theorem 2.

The scheme readily extends to rest on the k-linear assumption with $k > 2$. In this case, the proof requires $k + 2$ group elements – whereas combining the techniques of [23, 24] demands $k(n + 1 - t)$ elements per proof – and a CRS of size $O(k(n + t))$. We prove the following result in the full version of the paper.

Theorem 2. *The above proof system is a relatively sound QA-NIZK proof system if the SDP assumption holds in* $(\mathbb{G}, \mathbb{G}_T)$ *and if* H *is a collision-resistant hash function.*

As an application, we describe a new adaptively secure CCA2-secure non-interactive threshold cryptosystem based on the DLIN assumption in the full version of the paper. Under the k-linear assumption, the scheme provides ciphertexts that are $\Theta(k)$ group elements shorter than in previous such constructions. Under the DLIN assumption, ciphertexts consist of 8 elements of $\mathbb{G}$, which spares one group element w.r.t. the best previous variants [23, 24] of Cramer-Shoup with publicly verifiable ciphertexts.

5 An Efficient Threshold Keyed-Homomorphic KH-CCA-Secure Encryption Scheme from the DLIN Assumption

The use of linearly homomorphic signatures as publicly verifiable proofs of ciphertext validity in the Cramer-Shoup paradigm [12, 13] was suggested in [28]. However, the latter work only discusses non-adaptive (*i.e.*, CCA1) attacks. In the CCA2 case, a natural idea is to proceed as in our unbounded simulation-sound proof system and use the verification key of a on-time signature as the

tag of a homomorphic signature: since cross-tag homomorphic operations are disallowed, the one-time signature will prevent illegal ciphertext manipulations after the challenge phase. To obtain the desired keyed-homomorphic property, we use the simulation trapdoor of a simulation-sound proof system as the homomorphic evaluation key. This approach was already used by Emura *et al.* [16] in the context of designated verifier proofs. Here, publicly verifiable proofs are obtained from a homomorphic signature scheme of which the private key serves as an evaluation key: anyone equipped with this key can multiply two ciphertexts (or, more precisely, their built-in homomorphic components), generate a new tag and sign the resulting ciphertext using the private key of the homomorphic signature. Moreover, we can leverage the fact that the latter private key is always available to the reduction in the security proof of the homomorphic signature [28]. In the game of Definition 1, the simulator can thus hand over the evaluation key SK_h to the adversary upon request.

Emura *et al.* [16] gave constructions of KH-CCA secure schemes based on hash proof systems [13]. However, these constructions are only known to provide a relaxed flavor of KH-CCA security where evaluation queries should not involve derivatives of the challenge ciphertext. The reason is that 2-universal hash proof systems [13] only provide a form of one-time simulation soundness whereas the model of Definition 1 seemingly requires unbounded simulation-soundness. Indeed, when the evaluation oracle is queried on input of a derivative of the challenge ciphertext in the security proof, the homomorphic operation may result in a ciphertext containing a vector outside the language $\mathcal{L}_\rho$. Since the oracle has to simulate a proof for this vector, each homomorphic evaluation can carry a proof for a potentially false statement. In some sense, each output of the evaluation oracle can be seen as yet another challenge ciphertext. In this setting, our efficient unbounded simulation-sound QA-NIZK proof system comes in handy.

It remains to make sure that CCA1 security is always preserved, should the adversary obtain the evaluation key SK_h at the outset of the game. To this end, we include a second derived one-time homomorphic signature (Z, R, U) in the ciphertext without including its private key in SK_h.

Keygen(λ, t, N)**:** Choose bilinear groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$.

1. Pick $f, g, h \xleftarrow{R} \mathbb{G}$, $x_0, x_1, x_2 \xleftarrow{R} \mathbb{Z}_p$ and set $X_1 = f^{x_1} g^{x_0}$, $X_2 = h^{x_2} g^{x_0}$. Then, define $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$.
2. Pick random polynomials $P_1[Z], P_2[Z], P[Z] \in \mathbb{Z}_p[Z]$ of degree $t-1$ such that $P_1(0) = x_1$, $P_2(0) = x_2$ and $P(0) = x_0$. For each $i \in \{1, \ldots, N\}$, compute $VK_i = (Y_{i,1}, Y_{i,2})$ where $Y_{i,1} = f^{P_1(i)} g^{P(i)}$ and $Y_{i,2} = h^{P_2(i)} g^{P(i)}$.
3. Choose $f_{r,1}, f_{r,2} \xleftarrow{R} \mathbb{G}$ and define $\boldsymbol{f}_{r,1} = (f_{r,1}, 1, g)$, $\boldsymbol{f}_{r,2} = (1, f_{r,2}, g)$ and $\boldsymbol{f}_{r,3} = \boldsymbol{f}_{r,1}^{\phi_1} \cdot \boldsymbol{f}_{r,2}^{\phi_2} \cdot (1, 1, g)^{-1}$, where $\phi_1, \phi_2 \xleftarrow{R} \mathbb{Z}_p$. These vectors will be used as a Groth-Sahai CRS for the generation of NIZK proofs showing the validity of decryption shares.
4. Choose a strongly unforgeable one-time signature $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys consisting of L-bit strings, for some $L \in \mathsf{poly}(\lambda)$.

5. Generate a key pair for the one-time homomorphic signature of Section 2.5 with $n = 3$. Let $\mathsf{pk}_{ot} = \big(G_z, G_r, H_z, H_u, \{(G_i, H_i)\}_{i=1}^3\big)$ be the public key and let $\mathsf{sk}_{ot} = \{(\varphi_i, \vartheta_i, \varpi_i)\}_{i=1}^3$ be the corresponding private key.
6. Generate one-time homomorphic signatures $\{(Z_j, R_j, U_j)\}_{j=1,2}$ on the vectors $\boldsymbol{f} = (f, 1, g)$ and $\boldsymbol{h} = (1, h, g)$ and erase sk_{ot}.
7. Generate another linearly homomorphic key pair $(\mathsf{pk}, \mathsf{sk})$ with $n = 3$. The public key pk is augmented so as to contain a set of Groth-Sahai CRSes as in step 1 of the proof system in Section 3. Let $\mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$ be the private key for which the corresponding public key is
$$\mathsf{pk} = \Big(g_z,\ g_r,\ h_z,\ h_u,\ \{(g_i, h_i)\}_{i=1}^3,\ \mathbf{f} = \big(\boldsymbol{f}_1, \boldsymbol{f}_2, \{\boldsymbol{f}_{3,i}\}_{i=0}^L\big)\ \Big).$$
8. Use sk to generate one-time homomorphic signatures $\{(z_j, r_j, u_j)\}_{j=1,2}$ on the independent vectors $\boldsymbol{f} = (f, 1, g) \in \mathbb{G}^3$ and $\boldsymbol{h} = (1, h, g) \in \mathbb{G}^3$.
9. The public key is defined to be
$$PK = \Big(g,\ \boldsymbol{f},\ \boldsymbol{h},\ \boldsymbol{f}_{r,1},\ \boldsymbol{f}_{r,2},\ \boldsymbol{f}_{r,3},\ X_1,\ X_2,\ \mathsf{pk}_{ot},\ \mathsf{pk},\ \{(Z_j, R_j, U_j)\}_{j=1}^2,\ \{(z_j, r_j, u_j)\}_{j=1}^2 \Big).$$
The evaluation key is $SK_h = \mathsf{sk} = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$ while the i-th decryption key share is defined to be $SK_{d,i} = (P_1(i), P_2(i), P(i))$. The vector of verification keys is defined as $\mathbf{VK} = (VK_1, \ldots, VK_N)$, where $VK_i = (Y_{i,1}, Y_{i,2})$ for $i = 1$ to N.

Encrypt(M, PK)**:** to encrypt $M \in \mathbb{G}$, generate a one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$.

1. Set $(C_0, C_1, C_2, C_3) = (M \cdot X_1^{\theta_1} \cdot X_2^{\theta_2}, f^{\theta_1}, h^{\theta_2}, g^{\theta_1+\theta_2})$, with $\theta_1, \theta_2 \xleftarrow{R} \mathbb{Z}_p$.
2. Compute a first homomorphic signature (Z, R, U) on $(C_1, C_2, C_3) \in \mathbb{G}^3$. Namely, compute $Z = Z_1^{\theta_1} \cdot Z_2^{\theta_2}$, $R = R_1^{\theta_1} \cdot R_2^{\theta_2}$ and $U = U_1^{\theta_1} \cdot U_2^{\theta_2}$.
3. Using $\{(z_j, r_j, u_j)\}_{j=1,2}$, derive another homomorphic signature (z, r, u) on (C_1, C_2, C_3). Namely, compute $(z, r, u) = (z_1^{\theta_1} \cdot z_2^{\theta_2}, r_1^{\theta_1} \cdot r_2^{\theta_2}, u_1^{\theta_1} \cdot u_2^{\theta_2})$.
4. Using $\mathsf{SVK} \in \{0,1\}^L$, define the vector $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$ and assemble a Groth-Sahai CRS $\mathbf{f}_{\mathsf{SVK}} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$. Using $\mathbf{f}_{\mathsf{SVK}}$, generate commitments $\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u$ to the components of $(z, r, u) \in \mathbb{G}^3$ along with proofs $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ as in step 2 of the proving algorithm of Section 3. Let $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ be the resulting NIWI proof.
5. Generate $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$ and output
$$C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma) \qquad (2)$$

Ciphertext-Verify$\big(PK, C\big)$**:** parse C as in (2). Return 1 if and only if: (i) $\mathcal{V}(\mathsf{SVK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2), \sigma) = 1$; (ii) (Z, R, U) is a valid homomorphic signature on (C_1, C_2, C_3); (iii) $(\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \in \mathbb{G}^{15}$ is a valid proof w.r.t. the CRS $(\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_{\mathsf{SVK}})$ that committed (z, r, u) form a valid homomorphic signature for the vector $(C_1, C_2, C_3) \in \mathbb{G}^3$. Here, we define $\boldsymbol{f}_{\mathsf{SVK}} = \boldsymbol{f}_{3,0} \cdot \prod_{i=1}^L \boldsymbol{f}_{3,i}^{\mathsf{SVK}[i]}$.

Share-Decrypt$(PK, i, SK_{d,i}, C)$**:** on inputs $SK_{d,i} = (P_1(i), P_2(i), P(i)) \in \mathbb{Z}_p^3$ and C, return $(i, \perp)$ if **Ciphertext-Verify**$\big(PK, C\big) = 0$. Otherwise, compute $\hat{\mu}_i = \big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ which consists of a partial decryption $\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}$ as well as commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ to exponents $P_1(i), P_2(i), P(i) \in \mathbb{Z}_p$ and a proof π_{ν_i} that these satisfy the equations

$$\nu_i = C_1^{P_1(i)} \cdot C_2^{P_2(i)} \cdot C_3^{P(i)}, \qquad Y_{i,1} = f^{P_1(i)} g^{P(i)}, \qquad Y_{i,2} = h^{P_2(i)} g^{P(i)}.$$

The commitments $\boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P$ and the proof π_{ν_i} are generated using the CRS $(\boldsymbol{f}_{r,1}, \boldsymbol{f}_{r,2}, \boldsymbol{f}_{r,3})$. Then, return $\mu_i = (i, \hat{\mu}_i)$.

Share-Verify$\big(PK, VK_i, C, (i, \hat{\mu_i})\big)$**:** parse C as in (2) and VK_i as $(Y_{i,1}, Y_{i,2})$. If $\hat{\mu_i} = \perp$ or $\hat{\mu_i}$ cannot be parsed as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$, return 0. Otherwise, return 1 if and only if π_{μ_i} is valid.

Combine$(PK, \mathbf{VK}, C, \{(i, \hat{\mu}_i)\}_{i \in S})$**:** for each index $i \in S$, parse the share $\hat{\mu_i}$ as $\big(\nu_i, \boldsymbol{C}_{P_1}, \boldsymbol{C}_{P_2}, \boldsymbol{C}_P, \pi_{\mu_i}\big)$ and return $\perp$ if **Share-Verify**$\big(PK, C, (i, \hat{\mu_i})\big) = 0$. Otherwise, compute $\nu = \prod_{i \in S} \nu_i^{\Delta_{i,S}(0)} = C_1^{x_1} \cdot C_2^{x_2} \cdot C_3^{x_0} = X_1^{\theta_1} \cdot X_2^{\theta_2}$ and output $M = C_0/\nu$.

Eval$(PK, SK_h, C^{(1)}, C^{(2)})$**:** parse SK_h as $\{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$. For each $j \in \{1, 2\}$, parse $C^{(j)}$ as

$$(\mathsf{SVK}^{(j)}, C_0^{(j)}, C_1^{(j)}, C_2^{(j)}, C_3^{(j)}, Z^{(j)}, R^{(j)}, U^{(j)}, \boldsymbol{C}_z^{(j)}, \boldsymbol{C}_r^{(j)}, \boldsymbol{C}_u^{(j)}, \boldsymbol{\pi}_1^{(j)}, \boldsymbol{\pi}_2^{(j)}, \sigma^{(j)})$$

and return $\perp$ if either $C^{(1)}$ or $C^{(2)}$ is invalid. Otherwise,

1. Compute $C_0 = \prod_{j=1}^2 C_0^{(j)}$, $C_1 = \prod_{j=1}^2 C_1^{(j)}$, $C_2 = \prod_{j=1}^2 C_2^{(j)}$ and $C_3 = \prod_{j=1}^2 C_3^{(j)}$ as well as $Z = \prod_{j=1}^2 Z^{(j)}$, $R = \prod_{j=1}^2 R^{(j)}$ and $U = \prod_{j=1}^2 U^{(j)}$.
2. Generate a new one-time signature key pair $(\mathsf{SVK}, \mathsf{SSK}) \leftarrow \mathcal{G}(\lambda)$. Using $SK_h = \{(\chi_i, \gamma_i, \delta_i)\}_{i=1}^3$, generate proof elements $\boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ on the vector (C_1, C_2, C_3) using the simulator of the proof system in Section 3 with the one-time verification key SVK.
3. Return $C = (\mathsf{SVK}, C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \sigma)$ where $\sigma = \mathcal{S}(\mathsf{SSK}, (C_0, C_1, C_2, C_3, Z, R, U, \boldsymbol{C}_z, \boldsymbol{C}_r, \boldsymbol{C}_u, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2))$.

In the full version of the paper [29], we prove the KH-CCA security of the scheme assuming that Σ is a strongly unforgeable one-time signature and that the DLIN assumption holds in $\mathbb{G}$.

In some applications, it may be desirable to add an extra randomization step to the evaluation algorithm in order to make sure that derived ciphertexts will be indistinguishable from freshly generated encryption (similarly to [34]). It is straightforward to modify the scheme to obtain this property.

If the scheme is instantiated using Groth's one-time signature [20], the ciphertext consists of 25 elements of $\mathbb{G}$ and two elements of $\mathbb{Z}_p$. It is interesting to compare the above system with an instantiation of the same design principle using the best known Groth-Sahai-based unbounded simulation-sound proof [9][Appendix A.2], which requires 65 group elements in this specific case. With this proof system, we end up with 77 group elements per ciphertexts under the

DLIN assumption (assuming that an element of $\mathbb{Z}_p$ has the same length as the representation of a group element). The above realization thus saves 50 group elements and compresses ciphertexts to 35% of their original length.

References

1. 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.
2. M.Abe, K. Haralambiev, M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. In Cryptology ePrint Archive: Report 2010/133, 2010.
3. 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.
4. 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.
5. M. Bellare, P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pp. 62–73, 1993.
6. M. Blum, P. Feldman, S. Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC'88*, pp. 103-112, 1988.
7. D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *Crypto'04*, LNCS 3152, pp. 41–55, 2004.
8. D. Boneh, G. Segev, B. Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS 2012*, pp. 350–366, 2012.
9. J. Camenisch, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt'09*, LNCS 5479, pp. 351–368, 2009.
10. J. Cathalo, B. Libert, M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt'09*, *LNCS* 5912, pp. 179–196, 2009.
11. M. Chase, M. Kohlweiss, A. Lysyanskaya, S. Meiklejohn. Malleable Proof Systems and Applications. In *Eurocrypt'12*, *LNCS* 7237, pp. 281–300, 2012.
12. R. Cramer, V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto'98*, LNCS 1462, pp. 13–25, 1998.
13. R. Cramer, V. Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt'02*, *LNCS* 2332, pp. 45–64, 2002.
14. Y. Desmedt. Society and Group Oriented Cryptography: A New Concept. In *Crypto'87*, *LNCS* 293, pp. 120–127, 1987.
15. Y. Desmedt, Y. Frankel. Threshold Cryptosystems. In *Crypto'89*, *LNCS* 435, pp. 307–315, 1989.
16. K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, S. Yamada. Chosen Ciphertext Secure Keyed-Homomorphic Public-Key Encryption. In *PKC'13*, *LNCS* 7778, pp. 32–50, 2013.
17. A. Escala, G. Herold, E. Kiltz, C. Ràfols, J. Villar. An Algebraic Framework for Diffie-Hellman Assumptions. In *Crypto'13*, *LNCS* 8043, pp. 129–147, 2013.
18. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto'86*, LNCS 263, pages 186–194, 1986.

19. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Eurocrypt'06*, *LNCS* 4004, pp. 339–358, 2006.
20. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *Asiacrypt'06*, *LNCS* 4284, pp. 444–459, 2006.
21. J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt'08*, LNCS 4965, pp. 415–432, 2008.
22. D. Hofheinz, T. Jager. Tightly Secure Signatures and Public-Key Encryption. In *Crypto'12*, *LNCS* 7417, pp. 590–607, 2012.
23. C. Jutla, A. Roy. Relatively-Sound NIZKs and Password-Based Key-Exchange. In *PKC'12*, *LNCS* 7293, pp. 485–503, 2012.
24. C. Jutla, A. Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *Asiacrypt'13*, *LNCS* 8269, pp. 1–20, 2013.
25. J. Katz, V. Vaikuntanathan. Round-Optimal Password-Based Authenticated Key Exchange. In *TCC'11*, LNCS 6597, pp. 293–310, 2011.
26. A. Kiayias, M. Yung. Tree-Homomorphic Encryption and Scalable Hierarchical Secret-Ballot Elections. In *FC 2010*, LNCS 6052, pp. 257–271, 2011.
27. B. Libert, M. Yung. Non-Interactive CCA2-Secure Threshold Cryptosystems with Adaptive Security: New Framework and Constructions. In *TCC 2012*, LNCS 7194, pp. 75–93, 2012.
28. B. Libert, T. Peters, M. Joye, M. Yung. Linearly Homomorphic Structure-Preserving Signatures and their Applications. In *Crypto 2013*, *LNCS* 8043, pp. 289–307, 2013.
29. B. Libert, T. Peters, M. Joye, M. Yung. Non-Malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures. In Cryptology ePrint Archive: Report 2013/691.
30. 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.
31. M. Naor. On cryptographic assumptions and challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109, 2003.
32. M. Naor, M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC'90*, pp. 427–437, 1990.
33. M. Prabhakaran, M. Rosulek. Rerandomizable RCCA Encryption. *Crypto 2007*, *LNCS* 4622, pp. 517–534, 2007.
34. M. Prabhakaran, M. Rosulek. Homomorphic Encryption with CCA Security. *ICALP 2008*, *LNCS* 5126, pp. 667–678, 2008.
35. M. Prabhakaran, M. Rosulek. Towards Robust Computation on Encrypted Data. *Asiacrypt 2008*, *LNCS* 5350, pp. 216–233, 2008.
36. C. Rackoff, D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto'91*, *LNCS* 576, pp. 433–444, 1991.
37. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *FOCS'99*, pp. 543–553, 1999.
38. V. Shoup, R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *J. of Cryptology*, 15(2), pp. 75–96, 2002. Earlier version in *Eurocrypt'98*, *LNCS* 1403, pp. 1–16, 1998.
39. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, 2005.