HAL
open science

Signatures of Knowledge for Boolean Circuits under Standard Assumptions (Full version)

Karim Baghery, Alonso González, Zaira Pindado, Carla Ràfols

To cite this version:

Karim Baghery, Alonso González, Zaira Pindado, Carla Ràfols. Signatures of Knowledge for Boolean Circuits under Standard Assumptions (Full version). AFRICACRYPT 2020 - 12th International Conference on Cryptology in Africa, Jul 2020, Cairo / Virtual, Egypt. pp.24-44, 10.1007/978-3-030-51938-4_2 . hal-03118271

HAL Id: hal-03118271

https://inria.hal.science/hal-03118271v1

Submitted on 22 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Signatures of Knowledge for Boolean Circuits under Standard Assumptions (Full version)

Karim Baghery[1,2], Alonso González[3], Zaira Pindado[4] and Carla Ràfols[4,5]

[1] imec-COSIC, KU Leuven, Leuven, Belgium,
[2] University of Tartu, Tartu, Estonia,
karim.baghery@kuleuven.be
[3] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France, alonso.gonzalez@ens-lyon.fr
[4] Universitat Pompeu Fabra, Barcelona, Spain, zaira.pindado,carla.rafols@upf.edu
[5] Cybercat, Spain

Abstract. This paper constructs unbounded simulation sound proofs for boolean circuit satisfiability under standard assumptions with proof size $O(n+d)$ bilinear group elements, where d is the depth and n is the input size of the circuit. Our technical contribution is to add unbounded simulation soundness to a recent NIZK of González and Ràfols (ASIACRYPT'19) with very small overhead. We give two different constructions: the first one is more efficient but not tight, and the second one is tight. Our new scheme can be used to construct Signatures of Knowledge based on standard assumptions that also can be composed universally with other cryptographic protocols/primitives.
As an independent contribution we also detail a simple formula to encode Boolean circuits as Quadratic Arithmetic Programs.

Keywords: NIZK, Signatures, Bilinear Groups, CircuitSat.

Keywords: NIZK, Signatures, Bilinear Groups, CircuitSat.

1 Introduction

As one of the essential tools in modern cryptography, Non-Interactive Zero-Knowledge (NIZK) proof systems allow a party to prove that for a public statement $\vec{x}$, she knows a witness $\vec{w}$ such that $(\vec{x}, \vec{w}) \in \mathcal{R}$, for some relation $\mathcal{R}$, without leaking any information about $\vec{w}$ and without interaction with the verifier. Due to their impressive advantages and functionalities, NIZK proof systems are used ubiquitously to build larger cryptographic protocols and systems [6, 27]. Among the various constructions of NIZK arguments, there is usually a trade-off between several performance measures, in particular, between efficiency, generality and the strength of the assumptions used in the security proof.

Zero-knowledge Succinct Argument of Knowledge (zk-SNARKs) [18, 24] are among the most practically interesting NIZK proofs. They allow to generate succinct proofs for NP-complete languages (3 group elements for CircuitSat [24]) but they are constructed based on non-falsifiable assumptions (e.g. knowledge assumptions [12]). A well-known impossibility result of Gentry and Wichs [19] shows that this is unavoidable if one wants to have succinctness for general languages. Thus, non-falsifiable assumptions are an essential ingredient to have very efficient constructions, while falsifiable assumptions give stronger security guarantees and more explicit and meaningful security reductions [35].

Groth-Sahai proofs [26] also allow to prove general languages[6] under standard assumptions non-succinctly, trading security for succinctness. On the other hand, some constructions of Quasi-Adaptive NIZK (QA-NIZK) generate very efficient proofs based on falsifiable assumptions for very specific statements (e.g. membership in linear spaces). Somewhere in between, recent work by González and Ràfols [21] constructs a NIZK argument for boolean CircuitSat under falsifiable assumptions by combining techniques of QA-NIZK arguments and zk-SNARKs. The proof size of their construction is $O(n+d)$ group elements, where n is the length of the input and d is the depth of the circuit.

The primary requirements in a NIZK argument are *Completeness*, *Zero-Knowledge* (ZK), and *Soundness*. Completeness guarantees that if both parties honestly follow the protocol, the prover will convince the verifier. Zero-knowledge preserves prover's privacy and ensures that the verifier will not learn more than the truth of the statement

[6] GS proofs allow to prove satisfiability of any quadratic equation over $\mathbb{Z}_p$, where p is the order of a bilinear group. In particular, this can encode CircuitSat. The size of the resulting proof is linear in the total number of wires.

from the proof. Soundness guarantees that a dishonest prover cannot convince an honest verifier. However, in practice usually bare soundness is not sufficient and one might need stronger variations of it, known as *Knowledge Soundness*, *Simulation Soundness* or *Simulation Knowledge Soundness* (a.k.a. Simulation Extractability) [37, 22]. Knowledge soundness ensures that if an adversary manages to come up with an acceptable proof, he must *know* the witness. Simulation soundness (a.k.a. unbounded simulation soundness) ensures that an adversary cannot come up with valid proof for a false statement, even if he has seen an arbitrary number of simulated proofs. This notion basically guarantees that the proofs are sound and non-malleable. The strongest case, Simulation Extractability (SE) implies that an adversary cannot come up with a *fresh* valid proof unless he knows a witness, even if he has seen an arbitrary number of simulated proofs. In both notions knowledge soundness and simulation extractability the concept of *knowing* is formalized by showing that there exists an extraction algorithm, either non-Black-Box (nBB) or Black-Box (BB), that can extract the witness from the proof.

Zk-SNARKs (either knowledge sound ones [18, 24], or SE ones [25, 4]) are probably the best-known family of NIZK arguments. They achieve nBB extraction under non-falsifiable assumptions. While SE with nBB extraction is a stronger notion in comparison with (knowledge) soundness, it is still not sufficient for UC-security and needs to be lifted [3]. The reason is that in UC-secure NIZK arguments, to simulate the corrupted parties, the ideal-world simulator should be able to extract witnesses without getting access to the source code of environment's algorithm, which is only guaranteed by BB SE [9, 22].

SE NIZK arguments have great potential to be deployed in practice [33, 31], or construct other primitives such as Signature-of-Knowledge (SoK) [10]. In a SoK, a valid signature of a message m for some statement $\vec{x}$ and a relation $\mathcal{R}$ can only be produced if the signer knows a valid witness $\vec{w}$ such that $(\vec{x}, \vec{w}) \in \mathcal{R}$. Groth and Maller [25] constructed a SE zk-SNARK and a generic construction of a SoK from any SE NIZK argument, resulting in a SoK for CircuitSat. While their construction is for general NP relations and it is also succinct, it also relies on non-falsifiable assumptions and cannot be directly deployed in UC protocols.

This paper constructs a SE NIZK argument with BB extraction for Boolean CircuitSat which is secure under falsifiable assumptions. The proposed construction is based on the result of [21]. We show that the proposed construction adds minimal overhead to the original construction, resulting in a SE NIZK argument with BB extraction and proof size $O(n + d)$. Moreover, the proposed construction also allows one to construct a (Universally Composable) SoK of the same size.

The restriction to Boolean CircuitSat (and not arithmetic) for our SE-NIZK argument is inherited from the NIZK argument of [21] on which our argument is based. This restriction is due to the fact that we need the DLOG-based commitments to the input of the circuit to be extractable, and this is only possible (for a BB extractor) if the message space is of polynomial size. Thus, we restrict ourselves to the important special case of Boolean CircuitSat. As an independent result, in this paper we also give a simple formula to encode Boolean CircuitSat as a Quadratic Arithmetic Program [18], which we later use for our construction.

1.1 Our Contribution

Trivial Approach for Boolean CircuitSat. Let ϕ be some boolean circuit, and let a_i, b_i, c_i be the left, right and output wires of gate i. A zero-knowledge argument for Boolean CircuitSat, where the prover shows knowledge of some secret input satisfying the circuit, can be divided into three sub-arguments:

1) an argument of knowledge of some boolean input: to prove that the secret input is boolean, the prover must show that each input value satisfies some quadratic equation,
2) a set of linear constraints, which proves "correct wiring", namely that a_i, b_i are consistent with $\vec{c}$ and the specification of the circuit,
3) a set of quadratic constraints, which proves that for all i, a_i, b_i and c_i are in some quadratic relation which expresses correct evaluation of gate i.

It is straightforward to prove CircuitSat by computing perfectly binding commitments to all the wires a_i, b_i, c_i and use, for example, GS NIZK proofs for each of the three sub-arguments. However, the proof size is obviously linear in the number of wires.

New Techniques. In a recent result, González and Ràfols [21] give a proof for Boolean CircuitSat of size $O(n+d)$ group elements under falsifiable assumptions in bilinear groups. We now give an overview of their techniques, which is the main building block of our paper. The key to their result is to prove 2) and 3) succinctly for each level of the circuit. More specifically (ignoring zero-knowledge, momentarily), if L_j (resp. R_j, O_j) is a shrinking (non-hiding, deterministic) commitment to all left (resp. right, output) wires at depth j, they construct:

2') an argument that shows that the opening of L_j (resp. R_j) is in the correct linear relation (given by the wiring constraints in the circuit specification) with the input and the openings of $O_1, \ldots, O_{j-1}$,
3') an argument that shows that the opening of O_j is in the correct quadratic relation (which depends on the type of gates at level j) with the opening of L_j and R_j.

The abstraction given above of the results of [21] hides an important subtlety: "the opening of L_j" (and similarly for the other shrinking commitments O_j, R_j) is not well defined, as many openings are possible, so it is unclear what it means for these sub-arguments to be sound. However, as the authors of [21] observe, when we are using these as part of a global proof of CircuitSat, "the opening of L_j" to which we intuitively refer is well defined in terms of the openings in previous levels. In other words, in the soundness proof, 2') can be used to prove that if the reduction can extract an opening of $O_1, \ldots, O_{j-1}$ consistent with the input and the circuit, it can also extract a consistent opening of L_j (and similarly R_j). On the other hand, 3') shows that if the reduction can extract an opening of L_j and R_j consistent with the input and the circuit, it can also extract an opening of O_j. For this reason, González and Ràfols informally called 2') and 3') "arguments of knowledge transfer" (linear and quadratic, respectively): given knowledge of the input, arguments 2') and 3') can be used alternatively to transfer this knowledge to lower levels of the circuit.

Promise Problems. To formalize this intuitive notion, the authors of [21] define their sub-arguments 2') and 3') as arguments (with completeness and soundness) for certain promise problems:

2') Given the input $\vec{c}_0$ and openings $(\vec{c}_1, \ldots, \vec{c}_{j-1})$ of $O_1, \ldots, O_{j-1}$, the argument shows that L_j can be opened to some $\vec{a}_j$ with the correct linear relation to $(\vec{c}_0, \vec{c}_1, \ldots, \vec{c}_{j-1})$ (similarly for R_j).
3') Given $\vec{a}_j$ and $\vec{b}_j$, openings of L_j and R_j, the argument shows that there is an opening $\vec{c}_j$ of O_j that is in the correct quadratic relation (which depends on the type of gates at level j) with $\vec{a}_j$ and $\vec{b}_j$.

From an efficiency point of view, the interesting thing is that the arguments are of constant size. This explains the proof size $O(n + d)$: $O(n)$ is for committing to the input (with extractable commitments, which exist under falsifiable assumptions because the input is boolean), and d is the cost of doing 2') and 3') repeatedly for each level. At a conceptual level, the key issue is that the verifier never checks that the openings are correct (i.e. in 2') it never checks that $\vec{c}_i$ is a valid opening of O_i, and in 3') that $\vec{a}_j$, $\vec{b}_j$ are valid openings of L_j,R_j), which is *the promise*. Soundness is only guaranteed if the promise holds, and nothing is said when it does not hold (when the given openings are invalid). In fact, the verifier does not need these openings, they are just part of the statement to define soundness in a meaningful way, reflecting the fact that in the global argument for boolean CircuitSat, the openings at level j are uniquely determined by transferring the knowledge of the circuit to lower levels. So excluding the need to read the statement, the verifier works in constant time (it would work in linear time if it verified the statement). In particular, when using the sub-arguments in a global proof, verification of each of the sub-arguments is constant size, and the global verifier runs in time $O(n + d)$.

Security Proof. The sub-arguments 2') and 3') of [21] are not new. More specifically, for 2') the authors just use the QA-NIZK argument of linear spaces for non-witness samplable distributions of Kiltz and Wee [32], a generalization of [28, 34] and for 3') they use techniques appeared in the context of zk-SNARKs (as e.g. [18]) to write many quadratic equations as a single relation of polynomial divisibility that can be proven succinctly. The challenge they solve is to give a proof that 2') and 3') are sound for the aforementioned promise problems under falsifiable assumptions, which is not implied by the soundness of the NIZK arguments they use for 2') and 3'). More specifically, for the linear constraints the soundness of the argument of membership in a linear space does not protect from "witness switching attacks" as explained in [21]. Indeed, to prove that two shrinking commitments $\vec{c}_1$, $\vec{c}_2$ open to vectors of values with a certain linear relation, it is natural to write this as a membership proof in a linear space defined by matrices $\mathbf{M}$, $\mathbf{N}$, i.e. to prove that $\begin{pmatrix} \vec{c}_1 \\ \vec{c}_2 \end{pmatrix} \in \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix}$, which ensures that there exists some $\vec{w}$ such that $\vec{c}_1 = \mathbf{M}\vec{w}$ and $\vec{c}_2 = \mathbf{N}\vec{w}$. However, given some opening $\vec{w}$ of $\vec{c}_1$ (which in our analysis is known because of knowledge of the input and the transfer to lower levels of the circuit), the argument does not prove that $\vec{c}_2 = \mathbf{N}\vec{w}$, as it only proves that there is *some* common opening. Therefore, standard soundness does not prevent the adversary from "switching the witness": if the adversary is able to find another witness $\vec{w}'$ such that $\vec{c}_1 = \mathbf{M}\vec{w} = \mathbf{M}\vec{w}'$ it can use $\vec{w}'$ for $\vec{c}_2$, for some $\vec{w}'$ that does not satisfy the linear constraints.

This attack is easy to reduce to the binding property of commitment schemes if the reduction can extract $\vec{w}'$ from the adversary, but since the commitments are shrinking, this would require some non black-box extraction, deviating from the goal of using standard assumptions. The authors of [21] get around this by showing how to prove

soundness for the promise problems associated to linear constraints using a decisional assumption related to the matrix $\mathbf{M}$. For 3') they prove that the soundness of their argument for the promise problem is a straightforward consequence of a q-type assumption in bilinear groups.

Our Techniques: General Approach. This paper builds a SE NIZK for CircuitSat under falsifiable assumptions building on the work of [21]. There are several generic techniques to solve this problem. To the best of our knowledge, existing generic solutions are variations of the following approach, described for example in [22]: build an OR proof that given some circuit ϕ and a public input $\vec{x}_p$, either the circuit is satisfiable with public input x_p or a signature of $M = (\phi, \vec{x}_p)$ is known. The simulator uses as a trapdoor the signature secret key. We note that this approach results in a considerable (although also constant) overhead (around 20 group elements).[7] Our approach is based on the following observation: to compute "fake proofs" of satisfiability, a simulator just needs to lie either about the satisfiability of quadratic equations or linear equations, but not both. Further, it is sufficient to lie in the last gate. In particular, we choose the following strategy to simulate a proof for a circuit ϕ and a public input $\vec{x}_p$: complete the input arbitrarily, compute consistent assignments to all gates but choose the last left and right wire arbitrarily so that the last gate outputs one. Thus the simulator outputs only honest proofs except for the last linear relation, which is a simulated proof for a false statement, i.e. the simulator does not need the simulation trapdoor for sub-arguments 1) and 3') and standard soundness is sufficient. To be consistent with this strategy, our SE NIZK for boolean CircuitSat uses the construction of [21] but replaces 2'), the proof that the linear relation holds, with 2") an unbounded simulation sound proof for the same promise problem.

Recall that the argument 2') of [21] is just the QA-NIZK argument for membership in linear spaces of Kiltz and Wee for non-witness samplable distributions with a security proof is adapted for promise problems (non-trivially). We take the most efficient USS QA-NIZK argument of membership in linear spaces in the literature, also due to Kiltz and Wee [32] and we adapt the USS argument to work for bilateral linear spaces (linear spaces split among the two source groups in a bilinear group) as in [20] and for promise problems as in [21]. The overhead of the construction with respect to the original CircuitSat proof is minimal ($3|\mathbb{G}_1|$). BB extractability is achieved because of the soundness of the argument which proves that the input is boolean and the fact that ElGamal ciphertexts of 0 or 1 are BB extractable (the extraction trapdoor is the secret key).

Our approach modularly combines a USS argument of membership in linear spaces with other arguments. The USS NIZK argument of Kiltz and Wee is not tight. However, to get tight security we only need to construct a tight USS for promise problems for linear spaces (or for bilateral spaces if we want to improve efficiency). In Section 7 we give such a construction, we take the most tight QA-NIZK argument in the literature, Abe et al. [1], and we adapt the security proof to build an argument for the promise problem related to satisfiability of linear constraints. The result is a signature of knowledge for circuits with a loss of d (the circuit depth) in the reduction (inherited from [21]), but independent of the number of queries to the simulation oracle.

As Groth and Maller [25] pointed out, USS arguments for CircuitSat are very close to Signatures of Knowledge (SoK). We use the fact that our CircuitSat argument is tag-based to obtain a very simple transformation to SoK. In particular, our second construction results in a tight SoK.

Adapting the USS Argument to Promise Problems. Technically, the main challenge that we solve is to prove that the tag-based USS argument for membership in linear spaces of Kiltz and Wee [32] (in Section 6) and of Abe et al. [1] (in Section 7) is sound for the promise problem defined in [21] for linear constraints. More precisely, what we prove is that the adversary cannot create a valid proof for the statement

$$\begin{pmatrix}\vec{x}\\ \vec{y}\end{pmatrix} \in \mathbf{Im}\begin{pmatrix}\mathbf{M}\\ \mathbf{N}\end{pmatrix}$$

such that $\vec{x} = \mathbf{M}\vec{w}$ for some known $\vec{w}$ but $\vec{y} \neq \mathbf{N}\vec{w}$ even after seeing many simulated proofs. The idea is that if the linear constraints are satisfied until a certain level, they must be satisfied also at lower levels of the circuit.

In the following, we give an overview on how we adapt Kiltz and Wee USS argument for this promise problem. The tight construction based on Abe et al. in Section 7 follows the same lines. The main idea of the USS argument of Kiltz and Wee, $\Pi_{\mathsf{LIN\text{-}USS}}$ is to add a pseudorandom MAC to their QA-NIZK argument of membership in linear spaces Π_{LIN}. The soundness of the argument Π_{LIN} that proves membership in the space spanned by the columns of some matrix $\mathbf{U}$ is guaranteed by the fact that $\vec{y}^\top\mathbf{K}$ is uniformly random in the adversary's view given $\mathbf{UK}$ if $\vec{y} \notin \mathrm{Span}(\mathbf{U})$. The proof of simulation soundness of $\Pi_{\mathsf{LIN\text{-}USS}}$ shows, in the first place, that under some decisional

[7] Using OR proofs (the less efficient construction for PPE given in [36] or adding a bit as an auxiliary variable) plus the Boneh-Boyen signature for adaptive soundness.

assumption, the queries made by the adversary do not give additional information to the adversary, in particular, they do not leak additional information about the secret key other than the one in the common reference string. We can adapt this part of their argument in a straightforward way. Then their proof concludes by arguing that in the final game the common reference string information theoretically hides part of the secret key, more concretely, $\vec{y}^\top \mathbf{K}$ remains information theoretically hidden.

We need to add one extra game in the proof of $\Pi_{\mathsf{LIN\text{-}USS}}$ to account for the fact that in our case $\mathbf{U} = \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix}$ spans all of the space. In particular, on the one hand, our soundness condition is different, as explained (the adversary breaks soundness for $(\vec{x}^\top, \vec{y}^\top)$ if $\vec{x} = \mathbf{M}\vec{w}$ for some known $\vec{w}$ but $\vec{y} \neq \mathbf{N}\vec{w}$). On the other hand, the common reference string reveals all information about the secret key (since $\mathbf{U}^\top \mathbf{K}$ reveals everything about $\mathbf{K}$), so the information theoretic argument used by Kiltz and Wee to conclude the proof of $\Pi_{\mathsf{LIN\text{-}USS}}$ does not apply. We solve this in the same way as González and Ràfols [21], who show that if the Matrix Decisional DDH Assumption [16] associated to the distribution of the first block $\mathbf{M}$ holds, then we can switch to a game where $(\mathbf{0}^\top, \mathbf{N}^\top)\mathbf{K}$ is information theoretically hidden. Intuitively, this means the adversary cannot compute valid proofs such that if $\vec{x} = \mathbf{M}\vec{w}$ for some known $\vec{w}$ but $\vec{y} \neq \mathbf{N}\vec{w}$), because it does not know the projection of the secret key on the second block without involving the first block.

Generalization of Our Techniques. The observation that to add unbounded simulation soundness to NIZK arguments which prove both quadratic and linear equations it suffices to have USS in the linear part can have other applications. For example, a direct application is to give USS to the construction of Daza et al. [14], which gives a compact proof that a set of perfectly binding commitments open to 0 or 1.

A Canonical Transformation of Boolean Circuits to QAPs. To prove quadratic equations compactly, González and Ràfols adopt the idea of [18] to encode many quadratic equations as a problem of divisibility among polynomials. More in detail, in a breakthrough result building on [23], Gennaro et al. [18] introduced in 2013 two caracterizations of circuit satisfiability (Quadratic Span Programs or QSPs for boolean circuits and Quadratic Arithmetic Programs or QAPs for arithmetic circuits over $\mathbb{Z}_p$ where p should be the order of the bilinear group of the zk-SNARK, inspired by the notion of Span Programs [30]) and proposed an efficient zk-SNARK for it. The basic idea is that the correctness of all the computations of the circuit is expressed as a divisibility relation among certain polynomials which define the program. This leads to a succinct proof in the CRS model by checking the divisibility relation only in a secret point given in the CRS "in the exponent". In 2014, Danezis et al. [13] introduced Square Span Programs (SSP) for boolean circuits to simplify QSPs. The reason why special encodings for Boolean circuits exist is because these are an important special case, and they have special characteristics (a part from checking gate satisfiability, one must check that the wires are boolean). In 2016, Groth [24] introduced the most efficient zk-SNARK for QAPs, and also mentioned that QAPs can encode boolean CircuitSat but did not give an explicit transformation. González and Ràfols [21] gave an explicit encoding of Boolean CircuitSat, separating linear and quadratic constraints and dividing the encoding by layers of same depth as needed by their construction. That is, essentially they were spelling out a QAP for satisfiability of all boolean gates of the same depth.

We spell out a canonical QAP to describe boolean CircuitSat as a problem of satisfiability of polynomials. We call the transformation canonical because it is essentially the direct and simplest way to do this transformation. Although encoding Boolean CircuitSat as a QAP is not difficult and can be easily done with a computer, we give an exact formula that describes a simple QAP from the description of the gates. This is a contribution of independent interest, and when combined with Groth16's zk-SNARK it results in an argument with the polynomials that define the QAP are very simple, lagrangian polynomials or sums of them. Then, we use this transformation from boolean CircuitSat to QAP to derive a simpler transformation from Boolean CircuitSat (separated in linear and quadratic constraints for each depth) compared to González and Ràfols [21] (they needed to check a more complex quadratic equation at each depth).

Organization. In Section 2 we introduce notation and the relevant security definitions and recall the Signature of Knowledge definition and properties. In Section 3, we define a canonical QAP codification for Boolean Circuits. In Section 4 we recall the subschemes of Aggregated Proofs of Quadratic Equations and Aggregated Proofs of Linear Equations applied to our codification. In Section 5 we give our main construction, we present a framework of SE NIZK Argument for Boolean CircuitSat that uses three building blocks, a concrete instantiation of the framework in 5.1 and the UC SoK based on the instantiation in 5.2. In Section 6 we prove the USS argument of Kiltz and Wee is still secure with the promise problem. Same for Abe et al. USS argument in Section 7. Finally, in Section 8 we show how to improve the efficiency of the main construction with respect to a naive use of Groth-Sahai proofs.

Construction	Language	Signature Size	Assumption	Tightness
BFG [7]	PE	$(n_{\mathsf{PPE}} n_X, n_{\mathsf{PPE}} n_Y) + \ell_K$	Falsifiable	-
GM [25]	SAP	$(2,1) + \ell_K$	Non-falsifiable	$O(Q)$
Sec. 5.1. 6	QE	$(2n_s + 10d - 4, 6d + 4)$	Falsifiable	$O(Q)$
Sec. 5.1. 7	QE	$(2n_s + 10d + 8, 6d + 4)$	Falsifiable	$O(\log Q)$

Table 1. A comparison of our proposed SoK schemes in Sec. 5.1 with the USS argument for membership in linear spaces for in Section 6 and Section 7 respectively, with prior schemes. In the last column we show the tightness respect to the number of the queries Q for those constructions that are simulation sound. n_s denotes the secret input size in a boolean circuit, d the depth of the circuit, n_{PPE} is the number of pairing product equations (each multiplication gate in an arithmetic circuit can be encoded as a pairing product equation, in such case $n_{\mathsf{PPE}} = n$), n_X, n_Y are the number of variables in all the pairing product equations in $\mathbb{G}_1, \mathbb{G}_2$, respectively, ℓ_K is the size of the output of a hash function. PE: Pairing Equations, SAP: Square Arithmetic Equations, QE: Quadratic Equations.

Novelty. This is the full version of the work with the same title published in Africacrypt 20, [5]. The tight construction (Section 7), the details on tuning GS proofs (Section 8) and the details on the Signature of Knowledge construction (Section 5.2), and the canonical QAP for boolean circuits (Section 3) are new. Further, with respect to the Africacrypt 20 version, we have corrected minor issues about the definitions and corrected a claim about the distribution of simulated proofs. We have generalized our construction to work for all boolean circuits (and not only circuits with only NAND gates, as originally done for simplicity).

2 Preliminaries

Let PPT denote probabilistic polynomial-time, and NUPPT denote non-uniform PPT. Let $\lambda \in \mathbb{N}$ be the information-theoretic security parameter, say $\lambda = 128$. All adversaries will be stateful. For an algorithm $\mathcal{A}$, let $\mathbf{Im}(\mathcal{A})$ be the image of $\mathcal{A}$, i.e., the set of valid outputs of $\mathcal{A}$. By $y \leftarrow \mathcal{A}(x; r)$ we denote the fact that $\mathcal{A}$, given an input x and a randomizer r, outputs y. We denote by *negl* an arbitrary negligible function. For distributions A and B, $A \approx_c B$ means that they are computationally indistinguishable.

In pairing-based groups, a *bilinear group generator* $\mathsf{BGgen}(1^\lambda)$ is a PPT algorithm returns the *group key* $gk := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2)$, the description of an asymmetric bilinear group, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are additive groups of prime order p, the elements $\mathcal{P}_1, \mathcal{P}_2$ are generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear pairing, and there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Elements in $\mathbb{G}_\gamma$ are denoted implicitly as $[a]_\gamma := a\mathcal{P}_\gamma$, where $\gamma \in \{1, 2, T\}$ and $\mathcal{P}_T := e(\mathcal{P}_1, \mathcal{P}_2)$. For simplicity, we often write $[a]_{1,2}$ for the pair $[a]_1, [a]_2$. The pairing operation will be written as a product $\cdot$, that is $[a]_1 \cdot [b]_2 = [a]_1[b]_2 = e([a]_1, [b]_2) = [ab]_T$. Vectors and matrices are denoted in boldface. Given a matrix $\mathbf{T} = (t_{i,j})$, $[\mathbf{T}]_\gamma$ is the natural embedding of $\mathbf{T}$ in $\mathbb{G}_\gamma$, that is, the matrix whose (i,j)th entry is $t_{i,j}\mathcal{P}_\gamma$. We denote by $|\mathbb{G}_\gamma|$ the bit-size of the elements of $\mathbb{G}_\gamma$ and by $(\cdot, \cdot)$ the bit-size of elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ in each component.

2.1 Definitions

We recall the formal definition of QA-NIZK proofs. A QA-NIZK proof system [28] enables to prove membership in a language defined by a relation $\mathcal{R}_\rho$, which is determined by some parameter ρ sampled from a distribution $\mathcal{D}_{gk}$. While the CRS can be constructed based on ρ, the simulator of zero-knowledge is required to be a single PPT algorithm that works for the whole collection of relations $\mathcal{R}_{gk}$. For witness-relations $\mathcal{R}_{gk} = \{\mathcal{R}_\rho\}_{\rho \in \sup(\mathcal{D}_{gk})}$ with parameters sampled from a distribution $\mathcal{D}_{gk}$ over associated parameter language $\mathcal{L}_{\mathsf{par}}$, a QA-NIZK argument system Π consists of tuple of PPT algorithms $\Pi = (\mathsf{K}_0, \mathsf{K}_1, \mathsf{P}, \mathsf{V}, \mathsf{S}_0, \mathsf{S}_1, \mathcal{E})$, defined as follows,

Parameter generator, $gk \leftarrow \mathsf{K}_0(1^\lambda)$**:** K_0 is a PPT algorithm that given 1^λ generates group description gk.

CRS generator, $\mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho)$**:** K_1 is a PPT algorithm that given gk, samples string $\rho \leftarrow \mathcal{D}_{gk}$, and then uses gk, ρ and generates $(\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e)$, it also defines the tag space $\mathcal{T}$; finally output crs (that also contains parameter ρ) and stores the *simulation* trapdoor tr_s and *extraction* trapdoor tr_e as trapdoors.

Prover, $\pi \leftarrow \mathsf{P}(\mathsf{crs}, \vec{x}, \vec{w}, \tau)$**:** P is a PPT algorithm that, given $(\mathsf{crs}, \vec{x}, \vec{w}, \tau)$, where $(\vec{x}, \vec{w}) \in \mathcal{R}$ outputs an argument π with respect to a tag $\tau \in \mathcal{T}$. Otherwise, it outputs $\perp$.

Verifier, $\{0, 1\} \leftarrow \mathsf{V}(\mathsf{crs}, \vec{x}, \pi, \tau)$**:** V is a PPT algorithm that, given $(\mathsf{crs}, \vec{x}, \pi, \tau)$, returns either 0 (reject) or 1 (accept).

Prover Simulator, $\pi \leftarrow \mathsf{S}(\mathsf{crs}, \vec{x}, \mathsf{tr}_s, \tau)$**:** S is a PPT algorithm that, given $(\mathsf{crs}, \vec{x}, \mathsf{tr}_s)$, outputs a simulated argument π with respect to a tag $\tau \in \mathcal{T}$.

Extractor, $\vec{w} \leftarrow \mathcal{E}(gk, \mathsf{crs}, \vec{x}, \pi, \tau, \mathsf{tr}_e)$**:** $\mathcal{E}$ is a PPT algorithm that, given $(\mathsf{crs}, \vec{x}, \pi, \tau, \mathsf{tr}_e)$ extracts the witness $\vec{w}$; where tr_e is the extraction trapdoor.

We require an argument QA-NIZK system Π to be *quasi-adaptive complete*, *computational quasi-adaptive sound* and *computational quasi-adaptive zero-knowledge*, as defined below.

Definition 1 (Quasi-Adaptive Completeness). *A quasi-adaptive argument Π is perfectly complete for $\mathcal{R}_\rho$, if for all λ, all $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$, and all $\tau \in \mathcal{T}$,*

$$\Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda),\ \rho \leftarrow \mathcal{D}_{gk}, \\ \mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho), \pi \leftarrow \mathsf{P}(\mathsf{crs}, \vec{x}, \vec{w}, \tau) \end{bmatrix} : \mathsf{V}(\mathsf{crs}, \vec{x}, \pi, \tau) = 1 \Bigg] = 1\ .$$

Definition 2 (Computational Quasi-Adaptive Soundness). *A quasi-adaptive argument Π is computationally quasi-adaptive sound for $\mathcal{R}_\rho$, if for all λ, and for all non-uniform PPT $\mathcal{A}$,*

$$\Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda),\ \rho \leftarrow \mathcal{D}_{gk}, \\ \mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho),\ (\vec{x}, \pi, \tau) \leftarrow \mathcal{A}(gk, \mathsf{crs}) \end{bmatrix} : \begin{matrix} \mathsf{V}(\mathsf{crs}, \vec{x}, \pi, \tau) = 1 \wedge \\ (\vec{x}, \vec{w}) \notin \mathcal{R}_\rho \end{matrix} \Bigg] \approx 0$$

Definition 3 (Computational Quasi-Adaptive Zero-Knowledge). *A quasi-adaptive argument Π is computationally quasi-adaptive zero-knowledge for $\mathcal{R}_\rho$, if for all λ, all $\tau \in \mathcal{T}$, and for all non-uniform PPT $\mathcal{A}$,*

$$\Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda), \rho \leftarrow \mathcal{D}_{gk}, \\ \mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho) : \\ \mathcal{A}^{\mathcal{O}_{real}(\vec{x}, \vec{w})}(gk, \mathsf{crs}) = 1 \\ (\vec{x}, \vec{w}) \in \mathcal{R}_\rho \end{bmatrix} \approx \Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda), \rho \leftarrow \mathcal{D}_{gk}, \\ (\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathrm{K}_1(gk, \rho) : \\ \mathcal{A}^{\mathcal{O}_{sim}(\vec{x}, \vec{w})}(gk, \mathsf{crs}) = 1 \\ (\vec{x}, \vec{w}) \in \mathcal{R}_\rho \end{bmatrix}$$

where $\mathcal{O}_{real}(\vec{x}, \vec{w}, \tau)$ returns $\mathsf{P}(\mathsf{crs}, \vec{x}, \vec{w}, \tau)$ which emulates the actual prover for $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$, otherwise it outputs $\perp$; and $\mathcal{O}_{sim}(\vec{x}, \vec{w}, \tau)$ that returns $\mathsf{S}(\mathsf{crs}, \mathsf{tr}_s, \vec{x}, \tau)$ on input $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$ and $\perp$ if $(\vec{x}, \vec{w}) \notin \mathcal{R}_\rho$.

We also consider Simulation Soundness for our proofs, we take the next definition from Kiltz and Wee [32].

Definition 4 (Unbounded Simulation Adaptive Soundness). *A quasi-adaptive argument Π is unbounded simulation adaptive sound for $\mathcal{R}_\rho$, if for all λ, and for all non-uniform PPT $\mathcal{A}$,*

$$\Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda), \rho \leftarrow \mathcal{D}_{gk}, \\ (\mathsf{crs}, \mathsf{tr}) \leftarrow \mathsf{K}_1(gk, \rho); \\ (\vec{x}^*, \pi^*, \tau^*) \leftarrow \mathcal{A}^{O(\cdot)}(gk, \mathsf{crs}, \rho) \end{bmatrix} : \begin{matrix} \tau^* \notin \mathcal{Q}_{tags}\ \wedge\ (\vec{x}^*, \vec{w}^*) \notin \mathcal{R}_\rho \\ \wedge\ \mathsf{V}(\mathsf{crs}, \vec{x}^*, \pi^*, \tau^*) = 1 \end{matrix} \Bigg] \approx 0,$$

where $O(\vec{x})$ returns $\mathsf{S}(\mathsf{crs}, \mathsf{tr}, \vec{x}, \tau)$ and adds τ to the set $\mathcal{Q}_{tags}$.

Now we define a variation of definition *BB simulation extractability* for QA-NIZKs that is satisfied by our schemes.

Definition 5 (Quasi-Adaptive BB Simulation Extractability). *A non-interactive argument scheme Π is* quasi-adaptive black-box simulation-extractable *for $\mathcal{R}_\rho$, if for all λ, and for all non-uniform PPT $\mathcal{A}$, there exists a black-box extractor $\mathcal{E}$ such that,*

$$\Pr\begin{bmatrix} gk \leftarrow \mathsf{K}_0(1^\lambda),\ \rho \leftarrow \mathcal{D}_{gk}, \\ (\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathsf{K}_1(gk, \rho); \\ (\vec{x}^*, \pi^*, \tau^*) \leftarrow \mathcal{A}^{O(\cdot)}(gk, \mathsf{crs}, \rho), \\ \vec{w}^* \leftarrow \mathcal{E}(gk, \mathsf{crs}, \vec{x}^*, \pi^*, \tau^*, \mathsf{tr}_e) \end{bmatrix} : \begin{matrix} \mathsf{V}(\mathsf{crs}, \vec{x}^*, \pi^*, \tau^*) = 1 \\ \wedge\ (\vec{x}^*, \vec{w}^*) \notin \mathcal{R}_\rho \wedge\ (\vec{x}^*, \pi^*) \notin \mathcal{Q} \\ \tau^* \notin \mathcal{Q}_{tags} \end{matrix} \Bigg] \approx 0,$$

where $O(\vec{x}, \tau)$ returns $\mathsf{S}(\mathsf{crs}, \mathsf{tr}_s, \vec{x}, \tau)$ and adds $(\vec{x}, \pi)$ to the set of simulated proofs $\mathcal{Q}$ and τ to the set $\mathcal{Q}_{tags}$.

A key point about Def. 5 is that the extraction procedure is black-box and the extractor $\mathcal{E}$ works for all adversaries.

2.2 Signature of Knowledge

A Signature of Knowledge (SoK) [10] generalizes the concept of digital signature. One can sign the message just if it has a valid witness for membership in a language, in our case the NP-complete language of boolean CircuitSat. We require three properties: *Correctness* that ensures that all signers with a valid witness can always produce a signature that convinces the verifier, *Simulation-Extractability* that any adversary able to issue a new signature, even after seeing arbitrary signatures for different instances, should know a witness and *Perfect Simulatability* that ensures that the verifier learns nothing new about the witness from a signature.

We give the formal definitions of [25] in the following.

Definition 6 (Signature of Knowledge). *Let $\mathcal{R}$ be a relation generator, $\{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ a sequence of message spaces. Then, a tuple* (SSetup, SSign, SV, SS) *is a* Signature of Knowledge *scheme for $\mathcal{R}$ if it is correct, simulatable, simulation-extractable (defined in the following) and it is composed by the following algorithms:*

$\mathsf{tr}_s, \mathsf{tr}_e, \mathsf{pp} \leftarrow \mathsf{SSetup}(R)$: *the setup algorithm is a PPT algorithm that takes as input a relation $R \in \mathcal{R}_\lambda$ and returns public parameters pp, together with a simulation trapdoor tr_s and an extraction trapdoor tr_e.*

$\boldsymbol{\sigma} \leftarrow \mathsf{SSign}(\mathsf{pp}, \vec{x}, \vec{w}, \mathsf{m})$ *: the signing algorithm is a PPT algorithm that takes as input the public parameters pp, a pair $(\vec{x}, \vec{w}) \in R$ and a message $\mathsf{m} \in \mathcal{M}_\lambda$ and returns a signature $\boldsymbol{\sigma}$.*

$0/1 \leftarrow \mathsf{SV}(\mathsf{pp}, \vec{x}, \mathsf{m}, \boldsymbol{\sigma})$ *: the verification algorithm is a deterministic polynomial time algorithm that takes as input some public parameters pp, an instance $\vec{x}$, a message $\mathsf{m} \in \mathcal{M}_\lambda$, and a signature $\boldsymbol{\sigma}$ and outputs either 0 or 1 if it rejects or accepts, respectively.*

$\boldsymbol{\sigma} \leftarrow \mathsf{SS}(\mathsf{pp}, \mathsf{tr}_s, \vec{x}, \mathsf{m})$ *: the simulated signing algorithm is a PPT algorithm that takes as input some public parameters pp, a simulation trapdoor tr_s, and an instance $\vec{x}$ and returns a signature $\boldsymbol{\sigma}$.*

Definition 7. *A Signature of Knowledge is* correct *if for all $\lambda \in \mathbb{N}$, for all $R \in \mathcal{R}_\lambda$, for all $(\vec{x}, \vec{w})$ and for all $m \in \mathcal{M}_\lambda$*

$$Pr[\mathsf{pp} \leftarrow \mathsf{SSetup}(R); \boldsymbol{\sigma} \leftarrow \mathsf{SSign}(\mathsf{pp}, \vec{x}, \vec{w}, \mathsf{m}) : \mathrm{SV}(\mathsf{pp}, \vec{x}, \mathsf{m}, \boldsymbol{\sigma}) = 1] = 1,$$

Definition 8. *A Signature of Knowledge is* simulatable *if for any non-uniform PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{simul}_{SoK,\mathcal{A}}(\lambda) = 2\epsilon_{\mathcal{A}}(\lambda) - 1 \approx 0$, where*

$$\epsilon_{\mathcal{A}}(\lambda) = Pr\left[\begin{array}{c} R \leftarrow \mathcal{R}(1^\lambda), \mathsf{pp} \leftarrow \mathsf{SSetup}(R) \\ b \leftarrow \{0,1\}, b' \leftarrow \mathcal{A}^{S^b_{\mathsf{pp},\mathsf{tr}_s}(\cdot,\cdot,\cdot)}(\mathsf{pp}) \end{array} : b = b'\right],$$

where $S^b_{\mathsf{pp},\mathsf{tr}_s}(\vec{x}_i, \vec{w}_i, \mathsf{m}_i)$ checks $((\vec{x}_i, \vec{w}_i) \in R, \mathsf{m}_i \in M_\lambda)$ and returns $\boldsymbol{\sigma}_i \leftarrow \mathsf{SSign}(\mathsf{pp}, \vec{x}_i, \vec{w}_i, \mathsf{m}_i)$ if $b = 0$ and $\boldsymbol{\sigma}_i \leftarrow \mathrm{SS}(\mathsf{pp}, \mathsf{tr}_s, \vec{x}_i, \mathsf{m}_i)$ if $b = 1$.

Definition 9. *A Signature of Knowledge is* simulation-extractable *if for any non-uniform PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that*
$\mathsf{Adv}^{sim-extr}_{SoK,\mathcal{A},\mathcal{E}}(\lambda) \approx 0$, where

$$\mathsf{Adv}^{sig-extr}_{SoK,\mathcal{A},\mathcal{E}}(\lambda) = Pr\left[\begin{array}{c} R \leftarrow \mathcal{R}(1^\lambda), (\mathsf{pp}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathrm{SSetup}(R) \\ (\vec{x}, \mathsf{m}, \boldsymbol{\sigma}) \leftarrow \mathcal{A}^{\mathrm{OSign}_{\mathsf{pp},\mathsf{tr}_s}(\cdot,\cdot)}, \vec{w} \leftarrow \mathcal{E}(\mathsf{pp}, \mathsf{tr}_e, (\vec{x}, \mathsf{m}, \boldsymbol{\sigma})) \end{array} : \begin{array}{c} (\vec{x}, \vec{w}) \notin R \text{ and } (\vec{x}, m, \boldsymbol{\sigma}) \notin \mathcal{Q}, \\ 1 \leftarrow \mathrm{SV}(\mathsf{pp}, \vec{x}, \mathsf{m}, \boldsymbol{\sigma}) \end{array}\right],$$

where $\mathrm{OSign}_{\mathsf{pp},\mathsf{tr}_s}(\vec{x}_i, \mathsf{m}_i)$ returns $\vec{\sigma}_i \leftarrow \mathrm{SS}(\mathsf{pp}, \mathsf{tr}_s, \vec{x}_i, \mathsf{m}_i)$ and adds $\{(\vec{x}_i, \mathsf{m}_i, \boldsymbol{\sigma}_i)\}$ to the set $\mathcal{Q}$, which is initialized to $\emptyset$.

We emphasize that in the construction given in this paper, the extractor $\mathcal{E}$ is a PPT algorithm that only accesses $\mathcal{A}$'s output, as opposed to the work of Groth and Maller [25], where the extractor has nBB access to the adversary.

3 Canonical QAP for Boolean Circuits

Boolean circuits are acyclic directed graphs where the edges are called wires and the vertices are called gates. In this work, we consider boolean circuits $\phi : \{0,1\}^{n_{\mathsf{ipt}}} \longrightarrow \{0,1\}^{n_{\mathsf{opt}}}$, with possibly some set of public inputs n_p and some set of private inputs n_s, $n_s + n_p = n_{\mathsf{ipt}}$. Gates are arbitrary gates of fan-in two, (excluding non-interesting or trivial gate types). We denote m the total number of wires, n the number of boolean gates of the circuit, and ℓ the number of public values, where usually it will be the case that $\ell = n_p + n_{\mathsf{opt}} + 1$ and $m = n_p + n_s + n + 1$.

It is a well-known fact that, if $a, b \in \{0,1\}$, correct gate evaluation can be expressed as a quadratic equation over $\mathbb{Z}$. That, is for each gate type there exist values $\rho, \omega, \gamma, \epsilon \in \mathbb{Z}$, such that if $a, b \in \{0,1\}$, and $c = \rho ab + \omega a + \gamma b + \epsilon$, then $c \in \{0,1\}$ and c is the correct value of the gate evaluated at a, b. The constants satisfy that $\epsilon \in \{0,1\}$,

$\omega, \gamma \in \{0, \pm 1\}$, $\rho \in \{\pm 1\}$ for all gate types except for XOR and XNOR, and $\rho \in \{\pm 2\}$ for these two gates. More specifically, the important gate types are the following[8]

$\mathsf{AND}(a,b,c)$: $c = ab$.
$\mathsf{NAND}(a,b,c)$: $c = -ab + 1$
$\mathsf{OR}(a,b,c)$: $c = -ab + a + b$.
$\mathsf{NOR}(a,b,c)$: $c = ab - a - b + 1$
$\mathsf{XOR}(a,b,c)$: $c = -2ab + a + b$.
$\mathsf{XNOR}(x,y,x)$: $c = 2ab - a - b + 1$.
$\mathsf{G}_1(a,b,c) = (c = \overline{a} \wedge b) : c = -ab + b$.
$\mathsf{G}_2(a,b,c) = (c = \overline{\overline{a} \wedge b}) : c = ab - b + 1$.
$\mathsf{G}_3(a,b,c) = (c = a \wedge \overline{b}) : c = -ab + a$.
$\mathsf{G}_4(a,b,c) = (c = \overline{a \wedge \overline{b}}) : c = ab - a + 1$.

Therefore, we can express a boolean circuit of m wires and n gates as a tuple $(\mathbf{F}, \mathbf{G}, \vec{\rho}, \vec{\omega}, \vec{\gamma}, \vec{\epsilon})$, where $\mathbf{F} = (f_{ij}), \mathbf{G} = (g_{ij}) \in \{0,1\}^{m \times n}$ are the matrices which express the constraints for the left and right inputs for every gate and $\vec{\rho}, \vec{\omega}, \vec{\gamma}, \vec{\epsilon} \in \mathbb{Z}^n$ are the vectors of constants associated to every gate. That is, if a_{j_L} (resp. a_{j_R}) is the left (resp. right) wire of gate j, then $a_{j_L} = \sum_{i=1}^m f_{ij} a_i$ (resp. $a_{j_R} = \sum_{i=1}^m g_{ij} a_i$), i.e. $\vec{f_j} = (f_{1j}, \ldots, f_{mj})$ is a unit vector which selects the left wire.

We show how to encode correct boolean circuit computation to prove that some pair $(\vec{x}, \vec{y})$ satisfies that $\phi(\vec{x}) = \vec{y}$ as a simple QAP. The vector $\vec{a}$ will denote the assignment of the circuit, so $(a_1, \ldots, a_{n_{\mathsf{ipt}}}) = \vec{x}$ and $(a_{m-n_{opt}+1}, \ldots, a_m) = \vec{y}$.

Theorem 1. *Let p be some prime number, $p > 2$. Let $\phi : \{0,1\}^{n_{ipt}} \to \{0,1\}^{n_{opt}}$, be a circuit with n boolean gates, m wires, n_s secret inputs and n_p public inputs, defined by $(\mathbf{F}, \mathbf{G}, \vec{\rho}, \vec{\omega}, \vec{\gamma}, \vec{\varepsilon}) \in (\{0,1\}^{m \times n})^2 \times (\mathbb{Z}_p^n)^4$ as described above. Define the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{Z}_p^{m \times (n_s + n)}$ as*

$$\mathbf{A} = \left(\begin{array}{c|c} \mathbf{0}_{(n_p+1)\times n_s} & \vec{\gamma} \\ \mathbf{I}_{n_s} & \\ \mathbf{0}_{n \times n_s} & \mathbf{F}' \end{array}\right), \quad \mathbf{B} = \left(\begin{array}{c|c} \mathbf{0}_{(n_p+1)\times n_s} & \vec{\omega}' \\ \mathbf{I}_{n_s} & \\ \mathbf{0}_{n \times n_s} & \mathbf{G} \end{array}\right), \quad \mathbf{C} = \left(\begin{array}{c|c} \mathbf{0}_{(n_p+1)\times n_s} & \vec{\varepsilon} - \vec{\gamma} \circ \vec{\omega}' \\ \mathbf{I}_{n_s} & \mathbf{0}_{n_s \times n} \\ \mathbf{0}_{n_s \times n} & \mathbf{I}_n \end{array}\right)$$

where $\mathbf{F}' = \mathbf{F} \begin{pmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_n \end{pmatrix}$, $\vec{\omega}' = \vec{\omega} \begin{pmatrix} \rho_1^{-1} & & \\ & \ddots & \\ & & \rho_n^{-1} \end{pmatrix}$.

Then, $\vec{a} = (1, a_1, \ldots, a_m) \in \mathbb{Z}_p^{m+1}$ is a valid assignment of the circuit wires if and only if

$$(\vec{a}^\top \mathbf{A}) \circ (\vec{a}^\top \mathbf{B}) - \vec{a}^\top \mathbf{C} = \vec{0}_{n_s+n}^\top, \tag{1}$$

which is equivalent to

$$(\vec{a}'^\top \underline{\mathbf{A}} + \hat{\vec{\gamma}}) \circ (\vec{a}'^\top \underline{\mathbf{B}} + \hat{\vec{\omega}}) - \vec{a}'^\top \underline{\mathbf{C}} + \hat{\vec{\varepsilon}} - \hat{\vec{\gamma}} \circ \hat{\vec{\omega}} = \mathbf{0}_{n_s+n}^\top, \tag{2}$$

where $\vec{a}' = (a_1, \ldots, a_m)$, $\underline{\mathbf{A}}(\underline{\mathbf{B}}, \underline{\mathbf{C}}) \in \mathbb{Z}_p^m$ is the matrix $\mathbf{A}$ (resp. $\mathbf{B}$, $\mathbf{C}$) without the first row, $\hat{\vec{\gamma}} = \begin{pmatrix} \mathbf{0}_{n_s} & \vec{\gamma} \end{pmatrix}$, $\hat{\vec{\omega}} = \begin{pmatrix} \mathbf{0}_{n_s} & \vec{\omega}' \end{pmatrix}$, $\hat{\vec{\varepsilon}} = \begin{pmatrix} \mathbf{0}_{n_s} & \vec{\varepsilon} \end{pmatrix} \in \mathbb{Z}_p^{n_s+n}$.

As we will see, the first n_s equations (corresponding to the first n_s columns) prove that the secret inputs of the circuit $a_{n_p+1}, \ldots, a_{n_p+n_s}$ are boolean and the last n columns correspond with correct gate evaluation equations for the wiring corresponding with matrices $\mathbf{F}, \mathbf{G}$.

Proof. We first observe that the matrices are well defined since $\rho_j^{-1} \mod p$ is always defined because $\rho_j \neq 0$ and its absolute value is at most 2 for the type of gates considered.

We then note that when restricted to $i = n_p + 2, \ldots, n_p + n_s + 1$, $j = 1, \ldots, n_s$, all three matrices A_{ij}, B_{ij}, C_{ij} are the identity matrix $\mathbf{I}_{n_s}$. Therefore, for any assignment $\vec{a}$ the first n_s columns of equation (2) expresses the fact that the secret input is boolean. If $\vec{A}_j$, $\vec{B}_j$, $\vec{C}_j$ are the jth column of the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, for $j = 1, \ldots, n_s$ we have $(\vec{a}^\top \vec{A}_j) \circ (\vec{a}^\top \vec{B}_j) - \vec{a}^\top \vec{C}_j = a_j a_j - a_j = a_j^2 - a_j = 0$ is satisfied if and only if $a_i \in \{0,1\}$, for $i = n_p+2, \ldots, n_p+n_s+1 = n_{\mathsf{ipt}}$.

We now look at the equations determined by the last n columns of equation (2). If $\vec{F}'_j$, $\vec{G}_j$ are the jth columns of $\mathbf{F}'$, $\mathbf{G}$, then, the $(n_s + j)$th equation in expression (2) can be rewritten as:

$$(\vec{a}'^\top \vec{F}'_j + \gamma_j) \circ (\vec{a}'^\top \vec{G}_j + \omega'_j) - \vec{a}'^\top_{n|} \vec{I}_{n,j} + \varepsilon_j - \gamma_j \omega'_j = 0 \tag{3}$$

where the vector $\vec{a}'_{n|}$ contains the last n components of $\vec{a}'$, i.e. $(a'_{n_{\mathsf{ipt}}+1}, \ldots, a'_m)$.

The circuit ϕ specifies, for the jth circuit gate, a pair of indexes j_L, j_R which indicate the left and right wires. By definition of $\mathbf{F}' = (f'_{i,j}), \mathbf{G} = (g_{i,j})$, for $i = 1, \ldots, m$, $j = 1, \ldots, n$, the constants $f'_{i,j}$ and $g_{i,j}$ are 0 everywhere except for $f'_{j_L,j} = \rho_j$ and $g_{j_R,j} = 1$. Then, $\vec{a}'^\top \vec{F}'_j = \rho_j a_{j_L}$, $\vec{a}'^\top \vec{G}_j = a_{j_R}$ and $\vec{a}'^\top \vec{I}_{n_s+j} = a'_{n_{\mathsf{ipt}}+j} + \epsilon_j$.

[8] As observed in [13], the last remaining 6 gate types depend mostly on one input and are not used.

Replacing these values in equation (3), we obtain:

$$(\rho_j a_{j_L} + \gamma_j)(a_{j_R} + \omega'_j) - a_{n_{\mathsf{ipt}}+j} - \gamma_j \omega'_j + \epsilon_j = 0. \tag{4}$$

Using the fact that, by definition, $\omega'_j = (\rho_j^{-1}\omega_j) \mod p$, we can rewrite this equation as:

$$a_{n_{\mathsf{ipt}}+j} = \rho_j a_{j_L} a_{j_R} + a_{j_L}\omega_j + a_{j_R}\gamma_j + \epsilon_j, \tag{5}$$

which by definition of the constants encodes the satisfiability of gate j. □

The reason why the encoding is very simple is because the matrices $\underline{\mathbf{B}}$ and $\underline{\mathbf{C}}$ are mostly independent of the gate type, and have only $0, 1$ entries, whereas the entries of matrix $\underline{\mathbf{A}}$ are $\{0, \pm 1, \pm 2\}$. Further, matrices $\underline{\mathbf{A}}, \underline{\mathbf{B}}, \underline{\mathbf{C}}$ are as sparse as possible (with $n + n_s$ non-zero entries) and all columns have exactly one non-zero value. This is optimal, since $n + n_s$ equations are required to prove that the secret input (of size n_s) is boolean and n gates are satisfied, this is why we call it *canonical*. For completeness, in the next Theorem, we express all the quadratic equations (boolean input and correct gate evaluation) as a divisibility relation following the usual "polynomial aggregation technique" of [18].

Theorem 2. *Let $\mathcal{R} \subset \mathbb{Z}_p$ be some fixed set of cardinal n_s+n and let $\lambda_i(X)$ be the associated Lagrangian polynomials and $t(X)$ the polynomial whose roots are the elements of $\mathcal{R}$. Let $\phi : \{0,1\}^{n_{\mathsf{ipt}}} \to \{0,1\}^{n_{\mathsf{opt}}}$, be any circuit with n boolean gates, m wires, n_s secret inputs, and ℓ public values. There exist some polynomials $\{u_i(X); v_i(X); w_i(X)\}_{i=0}^m\}$such that $\vec{a} = (a_0, a_1, \ldots, a_m)$, with $a_0 = 1$, is a valid assignment to the circuit wires if and only if*

$$(\textstyle\sum_{i=0}^m a_i u_i(X)) \cdot (\sum_{i=0}^m a_i v_i(X)) - (\sum_{i=0}^m a_i w_i(X)) \equiv 0 \mod t(X). \tag{6}$$

Proof. Numerate the rows of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ from $0, \ldots, m$. For $i \in [1, m]$ set

$$u_i(X) = \sum_{j=1}^{n_s+n} A_{ij}\lambda_j(X), \quad v_i(X) = \sum_{j=1}^{n_s+n} B_{ij}\lambda_j(X), \quad w_i(X) = \sum_{j=1}^{n_s+n} C_{ij}\lambda_j(X).$$

Further, define

$$u_0(X) = \sum_{j=n_s+1}^{n_s+n} \gamma_j\lambda_j(X),\ v_0(X) = \sum_{j=n_s+1}^{n_s+n} \omega'_j\lambda_j(X),\ w_0(X) = \sum_{j=n_s+1}^{n_s+n} (\varepsilon_j - \gamma_j\omega'_j)\lambda_j(X).$$

Finally, if we let

$$u(X) = \sum_{i=1}^m a_i u_i(X) + u_0(X),\ v(X) = \sum_{i=1}^m a_i v_i(X) + v_0(X),\ w(X) = \sum_{i=1}^m a_i w_i(X) + w_0(X)$$

it holds that $\vec{a}$ satisfies equation (6) if and only if $t(X)$ divides $p(X) = u(X)v(X) - w(X)$. This is a direct consequence of the definition of the polynomials and Theorem 1. □

The simple form of matrices $\underline{\mathbf{A}}$, $\underline{\mathbf{B}}$ and $\underline{\mathbf{C}}$ translates into very simple expressions for $\{u_i(X), v_i(X), w_i(X)\}_{i=1}^m$. For instance, the $v_i(X)$'s can be computed as a sum of Lagrangian polynomials, without any exponentiation. Similarly, $u_0(X)$ has a very simple expression as $\gamma_j \in \{\pm 1\}$, $v_0(X)$ is slightly more complicated (the coefficients take values in $\{\pm 1, \pm 2^{-1} \mod p\}$) and so is $w_0(X)$.

3.1 Circuit Slicing

As we explain in Section 4 following González and Ràfols [21], the prover aggregates the proofs that all the gates are satisfied at level i (a set of quadratic equations), on the one hand, and all the linear equations that show "correct wiring", i.e. that the outputs at level at most $i-1$ are correctly transferred to inputs at level i, on the other hand.

For this, as in [21], we *slice* a boolean circuit in layers according to the depth of each gate. That is, we index the gates of ϕ by a pair (i, j), where i denotes the gate depth and j is some index in the range $1, \ldots, n_i$, where n_i is the number of gates at level i, and we write down, for each level, the set of quadratic and affine constraints that need to be satisfied. In the following, $\phi : \{0,1\}^n \to \{0,1\}$ and we call d the depth of the circuit.

We define a witness for Boolean CircuitSat as a tuple $(\vec{a}, \vec{b}, \vec{c})$ which is, respectively, a valid assignment to the left, right and output wires of ϕ when each boolean gate is written as a multiplicative constraint, as explained below. To

"slice" the circuit, each of these vectors is written as a concatenation of vectors, one for each multiplicative depth. That is, $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_d)$, $\vec{b} = (\vec{b}_1, \ldots, \vec{b}_d)$ and $\vec{c} = (\vec{c}_0, \vec{c}_1, \ldots, \vec{c}_d)$ and $\vec{y}_i = (y_{i,1}, \ldots, y_{i,n_i})$ for all $\vec{y} \in \{\vec{a}, \vec{b}, \vec{c}\}$. Gate (i, j) is described by constants $\rho_{i,j}, \omega_{i,j}, \gamma_{i,j}, \epsilon_{i,j}$, and $\vec{\rho}_i, \vec{\omega}_i, \vec{\gamma}_i, \vec{\epsilon}_i \in \mathbb{Z}^{n_i}$ are the vectors of constants associated to the n_i gates at level i.

A valid assignment should give $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ the values that prove correct gate evaluation of gate (i, j), namely, $c_{i,j} = (a_{i,j} + \gamma_{i,j})(b_{i,j} + \omega'_{i,j}) - (\gamma_{i,j}\omega'_{i,j} + \epsilon_{i,j})$ that are consistent with some boolean input $c_{0,1}, \ldots, c_{0,n}$ are some boolean values that represent a satisfying input.

We differ from [21] in that we take advantage of our work in the previous section characterizing Boolean CircuitSat as a QAP, therefore, the set of equations that need to be satisfied is simpler.

Lemma 1 breaks down CircuitSat in different items which reflect the different building blocks used by [21] and also our work. The input vector $\vec{x}$ (which corresponds to $\vec{c}_0$) is divided in two parts, the first n_p components being the public input $\vec{x}_p$ and the rest is the secret input $\vec{x}_s$ of length n_s. The main achievement of [21] is to do two aggregated proofs of all the constraints at the same depth with just two constant size proofs, one for the multiplicative and the other for the linear constraints. Therefore, items *c*) (resp. *d*)) require that for each $i = 1, \ldots, d$, a set of quadratic (resp. linear) equations holds. In the next two subsections (Section 4.1,4.2) we sketch the aggregated proofs of the sets of equations described in *c*) and *d*).

Lemma 1. *Let $\phi : \{0,1\}^n \to \{0,1\}$, be a circuit with m boolean gates. Then, for any public input $\vec{x}_p \in \{0,1\}^{n_p}$, $(\vec{a}, \vec{b}, \vec{c})$ is a valid input for satisfiability of $\phi(\vec{x}_p, \cdot)$ if and only if:*

a) $(c_{0,1}, \ldots, c_{0,n_p}) = (\vec{x}_p)$.
b) Boolean secret input: $(c_{0,n_p+1}, \ldots, c_{0,n}) = (\vec{x}_s) \in \{0,1\}^{n_s}$.
c) Correct gate evaluation at level i, for $i = 1, \ldots, d$ there exists a vector of constants $\vec{k}_i$ such that:

$$\vec{c}_i = \vec{k}_i + \vec{a}_i \circ \vec{b}_i, \qquad j = 1, \ldots, n_i,$$

d) Correct "wiring" (linear constraints) at level i: there exist some matrices $\widetilde{\mathbf{F}}_i, \widetilde{\mathbf{G}}_i$ such that $\vec{a}_i = \widetilde{\mathbf{F}}_i \vec{c}_{|i-1}$ and $\vec{b}_i = \widetilde{\mathbf{G}}_i \vec{c}_{|i-1}$, where $\vec{c}_{|i-1}^\top = (1, \vec{c}_0^\top, \ldots, \vec{c}_{i-1}^\top)$, more precisely,
e) Correct output: $c_{d,1} = 1$.

The matrices $\widetilde{\mathbf{F}}_i, \widetilde{\mathbf{G}}_i$ and the constants $k_{i,j}$ are defined naturally from the description in Theorem 1, namely:

- $\widetilde{\mathbf{F}}_i = \left(\vec{\gamma}_i \; \mathbf{F}'_i\right)$ where $\mathbf{F}'_i = \begin{pmatrix} \rho_{i,1} & & \\ & \ddots & \\ & & \rho_{i,n_i} \end{pmatrix} \mathbf{F}_i^\top$, where if $\mathbf{F}$ is the matrix given in the circuit description, $\mathbf{F}_i \in \mathbb{Z}_p^{(\sum_{j=0}^{i-1} n_j) \times n_i}$ is the matrix that describes the left wires of gates at level i.
- $\widetilde{\mathbf{G}}_i = \left(\vec{\omega}'_i \; \mathbf{G}_i^\top\right)$, where $\vec{\omega}'_i = \vec{\omega}_i \begin{pmatrix} \rho_{i1}^{-1} & & \\ & \ddots & \\ & & \rho_{in_i}^{-1} \end{pmatrix}$, and if $\mathbf{G}$ is the matrix given in the circuit description, $\mathbf{G}_i \in \mathbb{Z}_p^{(\sum_{j=0}^{i-1} n_j) \times n_i}$ is the matrix that describes the right wires of gates at level i.
- $k_{i,j} = \varepsilon_{i,j} - \gamma_{i,j}\omega_{i,j}\rho_{i,j}^{-1}$.

4 GR19 Argument for Boolean CircuitSat

In Section 3 we have described Boolean CircuitSat as a d sets of linear and quadratic constraints, where d is the depth of the circuit. In this section, we revisit the results of González and Ràfols [21] but using the simpler characterization of Boolean CircuitSat given in 3.1.Recall that GR19 shows how to give a constant size proof for each of these sets of constraints while basing security on falsifiable assumptions provided a witness of satisfiability is known for the "previous" sets of equations (ordering the sets of equations in the natural order from the input).

4.1 Aggregated Proofs of Quadratic Equations

We now describe the construction proposed in González and Ràfols [21] to prove correct gate evaluation at level i, for $i = 1, \ldots, d-1$, i.e. a proof that $c_{i,j} = k_{i,j} - a_{i,j}b_{i,j}$, for all $j = 1, \ldots, n_i$. It consists, for $k = 1, 2$, of a Groth-Sahai

NIZK Proof that some secret values $[L_{i,k}]_1, [R_{i,k}]_2, [O_{i,k}]_1, [O^*_{i,k}]_2, [H_{i,k}]_1$ satisfy the following relation[9]:

$$e([K_{i,k}]_1, [1]_2) + e([L_{i,k}]_1, [R_{i,k}]_2) - e([O_{i,k}]_1, [1]_2) = e([H_{i,k}]_1, [t_k]_2), \tag{7}$$

$$e([O_{i,k}]_1, [1]_2) = e([1]_1, [O^*_{i,k}]_2). \tag{8}$$

where if $t(X) = \prod_{r\in\mathcal{R}}(X - r)$, $t_k = t(s_k)$ and $\lambda_i(X) = \prod_{j\in\mathcal{R}\setminus\{r_i\}} \frac{(X - r_j)}{(r_i - r_j)}$ is the ith Lagrangian polynomial associated to $\mathcal{R}$, a set of $W = \max_{i=1,\ldots,d} n_i$ points used for interpolation, then

$$L_{i,k} = \sum a_j\lambda_j(s_k), \quad R_{i,k} = \sum b_j\lambda_j(s_k), \quad C_{i,k} = \sum c_j\lambda_j(s_k), \quad H_{i,k} = h_i(s_k),$$

where s_1, s_2 are random secret points specified in the CRS, $h_i(X) = (1-(\sum a_j\lambda_j(X))(\sum b_j\lambda_j(X))-\sum c_j\lambda_j(X))/t(X)$ and $[K_{i,k}]_1 = \sum k_{i,j}\lambda_j(s_k)$. Alternatively, for each n_i we define $\mathbf{\Lambda}_{n_i} = \begin{pmatrix}\lambda_1(s_1) \ldots \lambda_{n_i}(s_1)\\ \lambda_1(s_2) \ldots \lambda_{n_i}(s_2)\end{pmatrix}$,

$$[\vec{L}_i]_1 = [\mathbf{\Lambda}_{n_i}\vec{a}_i]_1, [\vec{R}_i]_2 = [\mathbf{\Lambda}_{n_i}\vec{b}_i]_2, [\vec{O}_i]_1 = [\mathbf{\Lambda}_{n_i}\vec{c}_i]_1,$$

and $\mathbf{\Lambda}$ is called Lagrangian Pedersen commitment in [21].

To the reader familiar with the literature, it is obvious that equation (7) uses SNARK techniques originally appeared in [18] (what we could call "polynomial aggregation") for proving many quadratic equations simultaneously. What is new in [21], is the security analysis, which avoids non-falsifiable assumptions.

GS proofs are necessary for zero-knowledge because $\vec{L}_i, \vec{R}_i, \vec{O}_i$ need to be deterministic for the proof to work. The authors of [21] use this proof as a building block in a larger proof, and for this we prove the following:
"if $(\vec{a}_i, \vec{b}_i)$ are valid openings of $[L_{i,k}]_1, [R_{i,k}]_2$ for $k = 1, 2$ then $\vec{k}_i + \vec{a}_i \circ \vec{b}_i$ is a valid opening of $O_{i,k}$."

Formally, we define the languages

$$\mathcal{L}^{\mathsf{quad}}_{\mathsf{YES}} = \left\{ \begin{array}{c} (\vec{a}, \vec{b}, [\vec{L}]_1, [\vec{R}]_2, [\vec{O}]_1) : \vec{k} + \vec{a} \circ \vec{b} = \vec{c}, \\ \left[\vec{L}\right]_1 = [\mathbf{\Lambda}]_1\vec{a}, \left[\vec{R}\right]_2 = [\mathbf{\Lambda}]_2\vec{b}, \left[\vec{O}\right]_1 = [\mathbf{\Lambda}]_1\vec{c} \end{array} \right\}$$

$$\mathcal{L}^{\mathsf{quad}}_{\mathsf{NO}} = \left\{ \begin{array}{c} (\vec{a}, \vec{b}, [\vec{L}]_1, [\vec{R}]_2, [\vec{O}]_1) : \vec{k} + \vec{a} \circ \vec{b} = \vec{c}, \\ \left[\vec{L}\right]_1 = [\mathbf{\Lambda}]_1\vec{a}, \left[\vec{R}\right]_2 = [\mathbf{\Lambda}]_2\vec{b}, \left[\vec{O}\right]_1 \neq [\mathbf{\Lambda}]_1\vec{c} \end{array} \right\}.$$

The argument consists of giving some values $\vec{H}, \vec{O}^*$ chosen by the prover which satisfy equations (7) for $\vec{L}, \vec{R}, \vec{O}$. *Completeness* holds for $\mathcal{L}^{\mathsf{quad}}_{\mathsf{YES}}$ and *soundness* for values $\mathcal{L}^{\mathsf{quad}}_{\mathsf{NO}}$ under the $(\mathcal{R}, m)$-Rational Strong Diffie-Hellman assumption ([21]). When(7) are proven with GS proofs, the authors argue that *zero-knowledge* also holds.

Note that the fact $[\vec{L}]_1 = [\mathbf{\Lambda}]_1\vec{a}$, or $[\vec{R}]_2 = [\mathbf{\Lambda}]_2\vec{b}$ is never checked by the verifier, this is the promise. The argument does not give any guarantee when this does not hold.

4.2 Aggregated Proofs of Linear Equations

In this section we explain the technique used in González and Ràfols [21] to prove correct "wiring" at level i, for $i = 1, \ldots, d-1$, i.e. an aggregated proof for linear constraints applied to the equations defined in 3.1. As we have seen in Lemma 1, we can express linear constraints at level i as:

$$\vec{a}_i = \widetilde{\mathbf{F}}_i\vec{c}_{|i-1}, \quad \vec{b}_i = \widetilde{\mathbf{G}}_i\vec{c}_{|i-1} \text{ for all } i = 1, \ldots, d. \tag{9}$$

Then at level i left and right constraints can be expressed, respectively as:

$$\begin{pmatrix}\vec{O}_{|i-1}\\ \vec{L}_i\end{pmatrix} = \begin{pmatrix}\mathbf{C}_i\\ \mathbf{N}^L_i\end{pmatrix}\vec{c}_{|i-1}, \quad \begin{pmatrix}\vec{O}_{|i-1}\\ \vec{R}_i\end{pmatrix} = \begin{pmatrix}\mathbf{C}_i\\ \mathbf{N}^R_i\end{pmatrix}\vec{c}_{|i-1} \tag{10}$$

[9] The second equation is added to have the element $O_{i,k}$ in both groups $\mathbb{G}_1, \mathbb{G}_2$. This will allow us to use simple QA-NIZK proofs of membership in linear spaces in $\mathbb{G}_1$ and $\mathbb{G}_2$ for the linear constraints, instead of using proofs of membership in bilateral spaces (spaces with parts in $\mathbb{G}_1$ and in $\mathbb{G}_2$.)

where $\mathbf{C}_i = \begin{pmatrix} \mathbf{I} & \vec{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{n_1} & \dots & \vec{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{\Lambda}_{n_{i-1}} \end{pmatrix}$, $\mathbf{N}_i^L = \mathbf{\Lambda}_{n_i}\widetilde{\mathbf{F}}_i$, $\mathbf{N}_i^R = \mathbf{\Lambda}_{n_i}\widetilde{\mathbf{G}}_i$ and $\mathbf{\Lambda}_{n_i}$ is the matrix of the Lagrangian Pedersen commitment key defined in the last section, and $\vec{O}_0$ is just the input of the circuit.

To make the argument zero-knowledge, the prover does never give $\vec{O}_i, \vec{L}_i$ or $\vec{R}_i$ in the clear, but rather, for $k = 1, 2$ and any $i \in [d]$, it gives GS commitments $[\vec{z}]_1$ to the input (i.e. to all components of $\vec{O}_0 = \vec{c}_0$), to the vector $\vec{O}_i$ as $[\vec{z}_{O,i}]_1$, to the vector $\vec{L}_i$ as $[\vec{z}_{L,i}]_1$ and to the vector $\vec{R}_i$ as $[\vec{z}_{R,i}]_2$ (a part from other GS commitments necessary for the quadratic proof). The matrices which define the linear relation between committed values are defined from $\mathbf{C}_i$, $\mathbf{N}_i^L = \mathbf{\Lambda}_{n_i}\widetilde{\mathbf{F}}_i$, $\mathbf{N}_i^R = \mathbf{\Lambda}_{n_i}\widetilde{\mathbf{G}}_i$ adding columns and rows to accommodate for the GS commitment keys in the relevant groups (see full details in [21]). We denote the matrix that define the left (resp. right) constraints until level $i-1$ as $\mathbf{M}_i^L$ (resp. $\mathbf{M}_i^R$), that is:

$$\mathbf{M}_i^L = \begin{pmatrix} \mathbf{C}_i \\ \mathbf{N}_1^L \\ \vdots \\ \mathbf{N}_{i-1}^L \end{pmatrix}, \qquad \mathbf{M}_i^R = \begin{pmatrix} \mathbf{C}_i \\ \mathbf{N}_1^R \\ \vdots \\ \mathbf{N}_{i-1}^R \end{pmatrix}.$$

González and Ràfols prove that the QA-NIZK argument of Kiltz and Wee [32] (with standard soundness) for membership in linear spaces for non-witness samplable distributions is an argument for the following promise problem:

$$\mathcal{L}_{\text{YES}}^{\mathsf{Lin}} = \left\{ (\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c} [\vec{x}]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{y}]_1 = [\mathbf{N}]_1\vec{w} \end{array} \right\}$$

$$\mathcal{L}_{\text{NO}}^{\mathsf{Lin}} = \left\{ (\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c} [\vec{x}]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{y}]_1 \neq [\mathbf{N}]_1\vec{w} \end{array} \right\}$$

parametrized by matrices $\mathbf{M}, \mathbf{N}$.

If we use this construction for matrices $\mathbf{M}_i^L$ and $\mathbf{N}_i^L$ (similarly for right side), this argument can be used to prove that, if we can extract $\vec{c}_{|i-1}$, then we can extract an opening $\vec{a}_i$ of $\vec{L}_i$ which is in the correct linear relation with $\vec{c}_{|i-1}$. In other words, this proves that if all the linear constraints are satisfied until level $i-1$, they must be satisfied until level i.

The authors prove completeness of the argument for statements in $\mathcal{L}_{\text{YES}}^{\mathsf{Lin}}$ and soundness for $\mathcal{L}_{\text{NO}}^{\mathsf{Lin}}$ under $\mathcal{M}_L^\top$-MDDH, $\mathcal{M}_R^\top$-MDDH and KerMDH assumption, where $\mathcal{M}_L$ (resp. $\mathcal{M}_R$) is the distribution of matrices $\mathbf{M}_i^L$ (resp. $\mathbf{M}_i^R$) described above[10].

Efficiency Improvements. We note that for simplicity, we have explained the result of [21] as proving a linear system of constraints for each level and each side (left or right), but in fact a single QA-NIZK argument for bilateral spaces for non-witness samplable distributions [20] is used in [21] to gain efficiency (the proof requires then only 2 elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ instead of $O(d)$ elements).

5 SE NIZK Argument for Boolean CircuitSat

We present our Quasi-Adaptive argument for Boolean CircuitSat for the language defined as

$$\mathcal{L}_\phi = \left\{ (\vec{x}_p) \middle| \exists \vec{x}_s \in \{0,1\}^{n_s} \text{ s.t. } \phi(\vec{x}_p, \vec{x}_s) = 1 \right\}.$$

As consequence of Lemma 1 the language $\mathcal{L}_{\phi,ck}$ can be equivalently defined as

$$\mathcal{L}_\phi = \left\{ (\vec{x}_p) \middle| \begin{array}{l} \exists \vec{x}_s \text{ s.t. } \vec{x}_s \circ (\vec{x}_s - \vec{1}) = \vec{0}; \\ \vec{c}_0 := (\vec{x}_p, \vec{x}_s); \\ \forall i \in [d], \exists \vec{a}_i, \vec{b}_i, \vec{c}_i \in \mathbb{Z}_p^{n_i} \text{ s.t. } ; \\ \vec{a}_i = \widetilde{\mathbf{F}}_i \vec{c}_{|i-1}, \vec{b}_i = \widetilde{\mathbf{G}}_i \vec{c}_{|i-1} \in \mathbb{Z}_p^{n_i}, \\ \vec{k}_i + \vec{a}_i \circ \vec{b}_i = \vec{c}_i. \end{array} \right\}.$$

[10] An important point is that these MDDH assumptions can be reduced to a decisional assumption in bilinear groups which does not depend on the circuit. In fact, $\mathbf{M}_i^L$ only depends on $n, n_1, \dots, n_s$, and the assumption can be reduced to a decisional assumption which only depends on $\mathbf{\Lambda}$ and the GS commitment key.

In the following Π_Q denotes the argument for Quadratic Equations described in Section 4.1, Π_L a tag-based USS membership argument for linear spaces that can be either the one presented in Section 6 or the one presented in Section 7 and Input an argument to prove that some BB extractable commitments to integers open to binary values.

$\underline{\mathrm{K}_0(\lambda, W, \mathcal{R})}$: On input some set $\mathcal{R} \subset \mathbb{Z}_p$ of cardinal W, choose a bilinear group gk and output (gk, W).

$\underline{\mathcal{D}_{gk,W,\mathcal{R}}}$: Pick commitment keys $(ck_1, ck_2) = ([\mathbf{\Lambda}]_1, [\mathbf{\Lambda}]_2)$ that are the Lagrangian Pedersen commitment keys associated to $\mathcal{R}$. Output $(ck_1, ck_2, \mathsf{crs}_{\mathsf{GS}})$.

$\underline{\mathrm{K}_1(gk, \phi)}$: Given $(ck_1, ck_2, \mathsf{crs}_{\mathsf{GS}}) \leftarrow \mathcal{D}_{gk,W}$ and $\phi : \{0,1\}^n \to \{0,1\}$ of maximum width W. For each $i \in [d]$ define matrices $[\mathbf{M}_i^L]_1$, $[\mathbf{M}_i^R]_2$, $[\mathbf{N}_i^L]_1$, $[\mathbf{M}_i^R]_2$ as explained in Section 4.2. Let $\mathsf{crs}_{\mathsf{Input}}$ the crs of the argument Input for a vector of size n_s is binary. Let crs_Q the crs of Π_Q for proving correct evaluation of (at most) W gates. For each $i \in [d]$, let $\mathsf{crs}_{L,i}^L$ ($\mathsf{crs}_{L,i}^R$) the crs for the USS argument of linear knowledge transfer Π_L of left (right) wires at depth i. Let $\mathsf{crs}_L = \left\{\mathsf{crs}_{L,i}^L, \mathsf{crs}_{L,i}^R\right\}_{i\in[d]}$ and $\mathsf{tr}_L = \left\{\mathsf{tr}_{L,i}^L, \mathsf{tr}_{L,i}^R\right\}_{i\in[d]}$, where $\mathsf{tr}_{L,i}^L$ ($\mathsf{tr}_{L,i}^R$) are the trapdoors of the Π_L arguments of left (right) wires at depth i, crs_L includes the tag space $\mathcal{T}$.

Output $\mathsf{crs} = (ck_1, ck_2, \mathsf{crs}_{\mathsf{GS}}, \mathsf{crs}_{\mathsf{Input}}, \mathsf{crs}_Q, \mathsf{crs}_L)$, $\mathsf{tr} = \mathsf{tr}_L$.

$\underline{\mathrm{P}\left(\mathsf{crs}, \vec{x}_p, \vec{x}_s, \vec{r}, \vec{a}, \vec{b}, \vec{c}, \tau\right)}$: Computes the commitment of the secret input $[\vec{z}]_1 = \mathsf{com}_{ck_1,ck_2}(\vec{x}_s, \vec{r})$ and constructs the proof Input for $[\vec{z}]_1$. For each $i \in [d]$ compute Lagrangian Pedersen commitments to the wires $[\vec{O}_i]_{1,2}, [\vec{L}_i]_1, [\vec{R}_i]_{1,2}$, give a GS proof $\Pi_{Q,i}$ that they satisfy the equations (7) and let $[\vec{z}_{O,i,k}]_1$, $[\vec{z}^*_{O,i,k}]_2$, $[\vec{z}_{L,i,k}]_1$, $[\vec{z}_{R,i,k}]_2, [\vec{z}^*_{R,i,k}]_1$ the correspondent GS commitments to $\vec{O}, \vec{L}, \vec{R}$, for $k = 1, 2$. Compute proofs $\Pi_{L,i}$ of correct wiring, $\Pi_{L,0}$ that the opening of $[\vec{z}]_1$ is correctly assigned to $[\vec{z}_{O,0}]_1$ and that the openings of $[\vec{z}_R]_2, [\vec{z}^*_R]_1$ and $[\vec{z}_O]_1, [\vec{z}^*_O]_2$ are equal respectively.

The proof is

$$\pi = \left([\vec{z}]_1, \mathsf{Input}, [\vec{z}_O]_1, [\vec{z}_L]_1, [\vec{z}^*_O]_2, [\vec{z}_R]_2, [\vec{z}^*_R]_1 \vec{\Pi}_L, \Pi_{L,0}, \vec{\Pi}_Q\right).$$

$\underline{\mathrm{V}(\mathsf{crs}, \vec{x}_p, \pi, \tau)}$: Verify all the proofs in π with the corresponding verification algorithms $\mathrm{V}_{\mathsf{Input}}$, V_{Π_L} (which uses τ) and check the GS proofs of equations (7).

$\underline{\mathrm{S}(\mathsf{crs}, \mathsf{tr}, \vec{x}_p, \tau)}$: Extend the input with zeros, $\vec{x} = (\vec{x}_p, 0, \ldots, 0)$ and evaluate the circuit honestly with this input to obtain the corresponding $\vec{a}_i, \vec{b}_i, \vec{c}_i$ for each $i = 1, \ldots, d$. Change the last gate values, i.e. the right and left values of the last gate at level d, $a_{d,1}$, $b_{d,1}$, and $c_{d,1}$ consequently, to have an assignment that satisfies the equation of this gate. Compute the commitment $[\vec{z}]_1 = \mathsf{com}_{ck_1,ck_2}(\vec{0}, \vec{r})$, honest proofs Input and $\Pi_{Q,i}$, and commitments $[\vec{z}_{O,i,k}]_1$, $[\vec{z}_{L,i,k}]_1$, $[\vec{z}^*_{O,i,k}]_2$, $[\vec{z}_{R,i,k}]_2$, $\left[\vec{z}^*_{R,i,k}\right]_1$ for each $i = 1, \ldots, d$. Run the simulator S_{Π_L} to obtain d simulated $\Pi^S_{L,i}, \Pi^S_{R,i}$ together with $\Pi^S_{L,0}$. Finally, $\pi^S = ([\vec{z}]_1, \mathsf{Input}, [\vec{z}_O]_1, [\vec{z}_L]_1, [\vec{z}_R]_2, [\vec{z}^*_O]_2, \Pi^S_L, \Pi^S_{L,0}, \Pi_Q)$.

Completeness is direct from the completeness of the respective subarguments.

Computational Zero-Knowledge follows from witness samplablity of the GS commitment keys and the fact that in GS proofs, commitments are dual mode commitments. This means that the common reference string can be generated in an indistinguishable way so that all commitments are perfectly hiding. In particular, in this setting, the distributions of real and simulated proofs are indistinguishable.

Unbounded Simulation Extractable Adaptive Soundness is proved in the following theorem.

Theorem 3. *If $\mathcal{A}$ is an adaptive adversary against the Unbounded Simulation BB Extractability Soundness of the Boolean CircuitSat argument described in Section 5 that makes at most Q queries to S, then there exist PPT adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ against the BB Extractable Soundness of Input, the Unbounded Simulation Soundness of Π_L argument and the soundness of Π_Q argument, respectively, such that*

$$\mathsf{Adv}_{USS}(\mathcal{A}) \leq \mathsf{Adv}_{ES\text{-}\mathsf{Input}}(\mathcal{B}_1) + d\mathsf{Adv}_{USS\text{-}\Pi_L}(\mathcal{B}_2) + 2d\mathsf{Adv}_{Sound\text{-}\Pi_Q}(\mathcal{B}_3).$$

Proof. (sketch) The simulator algorithm generates honestly the Input and Π_Q arguments and an adversary sees only simulated proofs of the linear argument Π_L. Therefore, an adversary that creates a new proof for an invalid statement breaks either the knowledge soundness of the Input, the soundness of the Π_Q arguments, or the USS of the linear arguments Π_L. □

5.1 Concrete SE QA-NIZK for Boolean CircuitSat

For the scheme described above, one can take as Input, and Π_Q the same building blocks as [21], namely the bitstring argument of Daza et al. [14] and Π_L the argument described in Section 4.1. An USS argument for promise problems either the one given in Section 6 or the one given in Section 7.

To simplify the exposition we have omitted many details that actually make the proof more efficient. In particular, instead of using two linear arguments for each depth of the circuit, we can use the linear argument for all the linear constraints of the circuit at once (as it is also done in the original work [21]). First, it is easy to see one can prove all the left (and right) constraints together, by considering a larger matrix. Second, left and right constraints can be merged in a single matrix which consists of elements in both groups, and using an argument for some promise problem in *bilateral* linear spaces. This also makes the auxiliary variable O^* (and related equations) unnecessary.

Efficiency. Then, the building blocks Input, Π_Q of our instantiation are exactly the same as in González and Ràfols [21]. The cost of committing to the input plus proving it is boolean with the argument of [14] is $(2n_s + 4, 6)$. We take the same idea for quadratic constraints proof from [21] with Zero-Knowledge applied to our equations (7), that is $(6d - 3, 2d - 1)$ for the commitments and $(4d - 4, 8d - 8)$ for the GS proofs. This is the same cost as in [21], using an approach where we add more elements in the crs, but we gain in the commitment size. This approach is explained in detail in Section 8, in our case the direct approach gives us $(12d-12, 4d-4)$ elements in the commitment, while using the approach in Section 8 we add $(4d-2, 2d-2)$ elements in the crs and the commitment size is reduced to about 25% in group $\mathbb{G}_1$. Finally, the overhead of using an USS argument for promise problems in bilateral spaces as opposed to the argument for bilateral spaces with standard soundness used in González and Ràfols [21] is only 3 elements in $\mathbb{G}_1$ in case of USS argument in 6, and 15 elements in 7.

5.2 Universally Composable Signature of Knowledge

Next, we construct a Signature of Knowledge (SoK) for boolean CircuitSat. Similarly, Groth and Maller [25](See Section 2) build a SoK using a Simulation Extractable NIZK with non-black-box extraction along with a universal one-way hash function. We use a different approach and take advantage of having a tag-based argument, and we set the tag to be the output of a hash function of the message to be signed together with the public input. The efficiency of the SoK is essentially the same as the SE-NIZK on which it relies, because we just need to add a collision resistant hash function in the public parameters and compute a hash for proving/verifying the relation.

The construction of Groth and Maller is based on knowledge assumptions and non-black box extraction, while our NIZK is based on falsifiable assumptions and the extractor is used as a black box. Following previous results that show black-box simulation extractability is sufficient to realize the idea functionality of NIZK arguments [22], our constructed SoK can also achieve UC-security [8].

UC Signature of Knowledge for circuit satisfiability under standard assumptions. We present a Signature scheme of Knowledge based in the tag-based SE-NIZK argument of Section 5 for boolean CircuitSat. To sign a message m, we use a collision resistant hash function of the message and the public statement, the result is used as the tag of the argument behind. If an adversary tries to reuse the same proof to forge a signature, it should be for a different message otherwise we have the same tag.

Given a message space $\mathcal{M}$ and a relation $R \in \mathcal{R}$, we give a signature scheme in Figure 1 that is the natural transformation of the tag-based SE-NIZK argument of Section 5 to a Signature of Knowledge for R.

$\underline{\mathrm{SSetup}(R)}$ Use $(\mathsf{crs}, \vec{\mathsf{tr}}_s, \vec{\mathsf{tr}}_e) \leftarrow \mathrm{K}_1(gk, \phi)$ where crs fixes a tag space $\mathcal{T}$. Define a collision resistant hash function H, return $\mathsf{pp} = (\mathsf{crs}, H)$.	$\underline{\mathrm{SV}(\mathsf{pp}, \vec{x}_p, \boldsymbol{\sigma}, \mathsf{m})}$ Compute $\tau = H(\vec{x}_p, \mathsf{m})$, return $\mathrm{V}(\mathsf{crs}, \vec{x}_p, \boldsymbol{\sigma}, \tau)$.
$\underline{\mathrm{SSign}(\mathsf{pp}, \vec{x}_p, \vec{w}, \mathsf{m})}$ Compute $\tau = H(\vec{x}_p, \mathsf{m})$, return $\boldsymbol{\sigma} \leftarrow \mathrm{P}(\mathsf{crs}, \vec{x}_p, \vec{w}, \tau)$.	$\underline{\mathrm{SS}(\mathsf{pp}, \vec{\mathsf{tr}}_s, \vec{x}_p)}$ Compute $\tau = H(\vec{x}_p, \mathsf{m})$, return $\boldsymbol{\sigma} \leftarrow \mathrm{S}(\mathsf{crs}, \vec{\mathsf{tr}}_s, \vec{x}_p, \tau)$.

Fig. 1. UC SoK based on the tag-based SE-NIZK of Section 5, with algorithms $(\mathrm{P}, \mathrm{V}, \mathrm{S})$ and $\mathsf{m} \in \mathcal{M}$.

6 USS QA-NIZK Arguments of Knowledge Transfer for Linear Spaces

In this section we prove that the USS argument for membership in linear spaces of Kiltz and Wee also satisfies the "knowledge transfer" property, or more technically, that it has soundness for the same promise problem described in Section 4.2. We give the argument for membership in linear spaces in one group in detail in Section 6.1 and we present the scheme for the bilateral version in Section 6.2.

6.1 USS $\mathsf{Lin}_{\mathcal{D}_k}$ argument

In this section we present $\mathsf{Lin}_{\mathcal{D}_k}$, a quasi-adaptive USS argument of membership in linear spaces in the group $\mathbb{G}_1$ for the promise problem defined by languages

$$\mathcal{L}^{\mathsf{Lin}}_{\text{YES}} = \left\{ (\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c} [\vec{x}]_1 = [\mathbf{M}]_1 \vec{w} \text{ and} \\ [\vec{y}]_1 = [\mathbf{N}]_1 \vec{w} \end{array} \right\}$$

$$\mathcal{L}^{\mathsf{Lin}}_{\text{NO}} = \left\{ (\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c} [\vec{x}]_1 = [\mathbf{M}]_1 \vec{w} \text{ and} \\ [\vec{y}]_1 \neq [\mathbf{N}]_1 \vec{w} \end{array} \right\}$$

parameterized by matrices $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$ sampled from some distributions $\mathcal{M}, \mathcal{N}$. Completeness holds for YES instances, and soundness guarantees that NO instances will not be accepted. That is, as in [21], we assume $[\vec{x}]_1 = [\mathbf{M}]_1 \vec{w}$ holds when proving soundness. In the CircuitSat context, this can be assumed because the idea is that this is proven by first proving knowledge of the input and then by "transferring" this knowledge to the lower layers via the quadratic or the linear argument we have presented. We consider the general language $\mathcal{L}$ that includes all tuples $(\vec{w}, \vec{x}, \vec{y})$ of the right dimension, some of them which are outside of $\mathcal{L}^{\mathsf{Lin}}_{\text{YES}} \cup \mathcal{L}^{\mathsf{Lin}}_{\text{NO}}$. We allow simulation queries for any tuple in $\mathcal{L}$. Note that it would be enough to allow the adversary just to ask for queries in $\mathcal{L}^{\mathsf{Lin}}_{\text{NO}}$ in some contexts, as in Section 5 for CircuitSat, but we define this more generally.

Scheme Definition. The argument is presented in Figure 2 and is just the USS QA-NIZK argument of [32] written in two blocks, which adds a pseudorandom MAC to the basic (not simulation sound, just sound) QA-NIZK argument of membership in linear spaces for general distributions also given in [32]. If in the basic arguments the proofs are of the form $[\vec{x}^\top, \vec{y}^\top]_1(\mathbf{K}_1, \mathbf{K}_2)$, in the USS variant they are given by

$$\left(\left[(\vec{x}^\top, \vec{y}^\top)(\mathbf{K}_1, \mathbf{K}_2) + \vec{r}^\top \boldsymbol{\Omega}(\boldsymbol{\Omega}_0 + \tau \boldsymbol{\Omega}_1) \right]_1, \left[\vec{r}^\top \boldsymbol{\Omega}^\top \right]_1 \right).$$

Our contribution is not in the scheme but in the security analysis. Our proof follows [21], that proved that the basic argument in [32] is complete and sound for the same promise problem under some MDDH and KerMDH assumptions related to the matrix distribution $\mathcal{M}$. Our contribution is to modify their analysis to adapt it to simulation soundness for the scheme of Figure 2.

$\underline{\text{K}\,(gk, [\mathbf{M}]_1, [\mathbf{N}]_1):}$
$\mathbf{K}_1 \leftarrow \mathbb{Z}_p^{\ell_1 \times (k+1)}, \mathbf{K}_2 \leftarrow \mathbb{Z}_p^{\ell_2 \times (k+1)},$
$\mathbf{K}^\top = \left(\mathbf{K}_1^\top, \mathbf{K}_2^\top\right)$
$\mathbf{A}, \boldsymbol{\Omega} \leftarrow \mathcal{D}_k,$
$\boldsymbol{\Omega}_0, \boldsymbol{\Omega}_1 \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$
$\mathbf{C}_1 = \mathbf{K}_1 \mathbf{A}, \mathbf{C}_2 = \mathbf{K}_2 \mathbf{A},$
$[\mathbf{B}]_1 = \left[\mathbf{M}^\top \mathbf{K}_1 + \mathbf{N}^\top \mathbf{K}_2\right]_1$
$(\mathbf{P}_0, \mathbf{P}_1) = (\boldsymbol{\Omega}^\top \boldsymbol{\Omega}_0, \boldsymbol{\Omega}^\top \boldsymbol{\Omega}_1)$
$(\mathbf{Q}_0, \mathbf{Q}_1) = (\boldsymbol{\Omega}_0 \mathbf{A}, \boldsymbol{\Omega}_1 \mathbf{A})$
Return $\mathsf{crs} = \left(gk, [\mathbf{B}]_1, [\mathbf{A}]_2, [\mathbf{P}_0]_1, [\mathbf{P}_1]_1, [\mathbf{Q}_0]_2, [\mathbf{Q}_1]_2, [\mathbf{C}_1]_2, [\mathbf{C}_2]_2, [\boldsymbol{\Omega}]_1\right)$
$\mathsf{tr} = (\mathbf{K}_1, \mathbf{K}_2)$

$\underline{\text{P}\,(\mathsf{crs}, \tau, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \boldsymbol{w}):}$
Pick $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left(\boldsymbol{w}^\top [\mathbf{B}]_1 + \vec{r}^\top [\mathbf{P}_0 + \tau \mathbf{P}_1]_1, [\vec{r}^\top \boldsymbol{\Omega}^\top]_1\right).$

$\underline{\text{V}\,(\mathsf{crs}, \tau, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \vec{\pi}):}$
Check if:
$e(\vec{\pi}_1, [\mathbf{A}]_2) - e\left(\left[\boldsymbol{x}^\top, \boldsymbol{y}^\top\right]_1, [\mathbf{C}]_2\right)$
$= e(\vec{\pi}_2, [\mathbf{Q}_0 + \tau \mathbf{Q}_1]_2)$

$\underline{\text{S}\,(\mathsf{crs}, \tau, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \mathsf{tr}):}$
Sample $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left([\vec{x}^\top, \vec{y}^\top]_1 \mathbf{K} + \vec{r}^\top [\mathbf{P}_0 + \tau \mathbf{P}_1]_1, [\vec{r}^\top \boldsymbol{\Omega}^\top]_1\right).$

Fig. 2. The $\mathsf{Lin}_{\mathcal{D}_k}$ argument for proving membership in linear spaces in blocks $[\vec{x}, \vec{y}]_1 \in \text{Im}[\mathbf{M}, \mathbf{N}]_1$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$.

Perfect Completeness, Perfect Zero-Knowledge. Our language $\mathcal{L}^{\mathsf{Lin}}_{\text{YES}}$ is the same language for membership proofs in a linear space $[\mathbf{M},\mathbf{N}]_1^\top$ used in [32]:
$\left\{(\vec{w},[\vec{x},\vec{y}]_1) : [\vec{x}^\top,\vec{y}^\top]_1^\top = [\mathbf{M},\mathbf{N}]_1^\top\,\vec{w}\right\}$, so perfect completeness and perfect zero-knowledge are immediate.

Unbounded Simulation Soundness. We use Definition 4, for any adversary $\mathcal{A}$ that sends any number Q of queries $(\vec{w}^i,[\vec{x}^i,\vec{y}^i]_1) \in \mathcal{L}$ to the query simulator oracle S, receives simulated proofs $\{\vec{\pi}^i\}_{i=1}^Q$ as described in Figure 2, the probability that the adversary $\mathcal{A}$ comes up with $(\vec{w}^*,[\vec{x}^*,\vec{y}^*]_1,\tau^*,\vec{\pi}^*)$ such that $(\vec{w}^*,[\vec{x}^*,\vec{y}^*]_1) \in \mathcal{L}^{\mathsf{Lin}}_{\text{NO}}$ different of the queried ones, different tag τ^* and $\mathrm{V}(\mathsf{crs},\tau^*,[\vec{x}^*,\vec{y}^*]_1,\vec{\pi}^*) = 1$ is negligible.

Our proof is analogous to the USS proof of [32], where the authors argue that partial information about matrix $\mathbf{K}$ is computationally hidden across all the simulated proofs. Essentially, what the authors are doing is to reduce the proof of USS to a standard soundness proof. More concretely, they switch to a game where the simulated proofs hide information theoretically the projection of $\mathbf{K}$ for vectors outside of the span of the columns of a matrix $\tilde{\mathbf{M}}$ that defines the language. Therefore, one can argue, as in the standard soundness proof, that the probability of providing a valid proof for a false statement is negligible.

Our proof combines the work of [32] to show that the queries do not provide additional information, with the work of [21] to show standard soundness to the language associated to the promise problem. Indeed, in the case we are interested in the matrix $\tilde{\mathbf{M}}$ spans the whole space so the standard soundness proof used by [32] cannot be used and we need an extra change of games to use a technique proposed by [21] that proves that the block $\mathbf{K}_{2,2}$ is hidden from the adversary. This block corresponds to the part of the statement that is not in the correct linear space. That is, for breaking soundness the adversary has to create a valid proof for $(\vec{w},[\vec{x}]_1,[\vec{y}]_2)$ such that $\vec{y} \neq \mathbf{N}\vec{w}$ and $\vec{x} = \mathbf{M}\vec{w}$), and the coordinates of this block correspond to the projection by matrix $\mathbf{N}$. Concretely, at some point in their proof, Kiltz and Wee change the key matrix uniformly sampled for another of the form $\mathbf{K}' + \vec{b}\vec{a}^\perp$, where $\mathbf{K}'$ is uniformly sampled and $\vec{a}^\perp$ is in the co-kernel of $\mathbf{A}$. We apply the same change but in blocks, $\vec{b} = (\vec{b}_1,\vec{b}_2)$, so our extra game consists in changing the projection of $\vec{b}_1$ by $\mathbf{M}^\top$ to some random vector $\vec{z}$, i.e. we change $\mathbf{M}^\top\vec{b}_1 + \mathbf{N}^\top\vec{b}_2$ to $\vec{z} + \mathbf{N}^\top\vec{b}_2$ by assuming the $\mathcal{M}^\top$-$\mathsf{MDDH}_{\mathbb{G}_1}$ assumption, where $\mathcal{M}^\top$ is the matrix that defines the distribution of $\mathbf{M}^\top$ (as in [21]). So, what the adversary can see about $\vec{b}$ is just $\mathbf{N}^\top\vec{b}_2$ but it is hidden by $\vec{z}$.

For the following theorem, we use the Computational Core Lemma of Kiltz and Wee in Section 4.1. of [32], which is independent of $\mathcal{M},\mathcal{N}$, it just assumes the $\mathcal{D}_k$-$\mathsf{MDDH}_{\mathbb{G}_1}$, so we can use it directly in our proof.

Theorem 4. *The $\mathsf{Lin}_{\mathcal{D}_k}$ scheme in Figure 2 is a Quasi-adaptive Non-Interactive Zero-Knowledge Argument with Unbounded Simulation Soundness such that for any adversary $\mathcal{A}$ that makes at most Q queries to* S *there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ against the $\mathcal{D}_k$-KerMDH, $\mathcal{M}^\top$-MDDH assumptions in $\mathbb{G}_1$ for which the advantage of $\mathcal{A}$ is bounded by*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{USS}\text{-}\mathsf{Lin}_{\mathcal{D}_k}}(\mathcal{A}) \leq & \mathsf{Adv}_{\mathcal{D}_k-\mathsf{KerMDH}_{\mathbb{G}_1}}(\mathcal{B}_1) + 2Q\mathsf{Adv}_{\mathcal{D}_k\text{-}\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_2)\\ & + \mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_3) + \frac{Q+1}{p}.\end{aligned}$$

Proof. Let $\mathcal{A}$ be an adversary that plays the game described in USS definition 4. We will proceed by changing to indistinguishable games in order to bound the advantage of $\mathcal{A}$. Let Game_0 be the real game and Adv_i the advantage of winning Game_i.

– Game_1 is the same as Game_0 except the verification algorithm V is changed to

$$\begin{array}{l}\underline{\mathrm{V}^*(\mathsf{crs},\tau,[\vec{x},\vec{y}]_1,\vec{\pi}) :}\\ \text{Check: } \vec{\pi}_1 = [\vec{x}^\top,\vec{y}^\top]_1\mathbf{K} + \vec{\pi}_2(\mathbf{\Omega}_0 + \tau\mathbf{\Omega}_1).\end{array}$$

If a tuple $([\vec{x},\vec{y}]_1,\vec{\pi})$ passes verification of V but does not pass verification of V*, it means that the value $\vec{\pi} - [\vec{x}^\top,\vec{y}^\top]_1\mathbf{K} - \vec{\pi}_2(\mathbf{\Omega}_0 + \tau\mathbf{\Omega}_1) \in \mathbb{G}_1^{k+1}$ is a non-zero vector in the cokernel of $\mathbf{A}$. Thus, there exists an adversary $\mathcal{B}_1$ against $\mathsf{KerMDH}_{\mathbb{G}_1}$ such that

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \leq \mathsf{Adv}_{\mathcal{D}_k-\mathsf{KerMDH}_{\mathbb{G}_1}}(\mathcal{B}_1).$$

– Game_2 is the same as Game_1 except the simulation algorithm S is changed to

$$\begin{array}{l}\underline{\mathrm{S}^*(\mathsf{crs},\tau,[\vec{x},\vec{y}]_1,\mathsf{tr}) :}\\ \vec{r} \leftarrow \mathbb{Z}_p^k, \mu \leftarrow \mathbb{Z}_p\\ \text{Return: } \vec{\pi} = ([(\vec{x}^\top,\vec{y}^\top)\mathbf{K} + \mu\vec{a}^\perp + \vec{r}^\top(\mathbf{P}_0 + \tau\mathbf{P}_1)]_1,[\vec{r}^\top\mathbf{\Omega}]_1),\end{array}$$

where $\vec{a}^\perp$ is an element from the Kernel of $\mathbf{A}$. Let $\mathcal{B}_2$ be an adversary against $\mathcal{D}_k$-$\mathsf{MDDH}_{\mathbb{G}_1}$. $\mathcal{B}_2$ picks $\mathbf{K}$ itself and answers queries $(\tau_i,\vec{w}_i,[\vec{x}_i,\vec{y}_i]_1)$ from $\mathcal{A}$:

- if $\tau_i \neq \tau^*$: $\mathcal{B}_2$ queries the oracle $\mathcal{O}_b$, defined in the core lemma [32], who simulates S if $b = 0$, or S* if $b = 1$.
- if $\tau_i = \tau^*$: $\mathcal{B}_2$ samples $\vec{r} \leftarrow \mathbb{Z}_p$ and computes $([(\vec{x}_i^\top, \vec{y}_i^\top)\mathbf{K} + \vec{r}^\top(\mathbf{P}_0 + \tau_i \mathbf{P}_1)]_1, [\vec{r}^\top \mathbf{\Omega}_0^\top]_1)$.

Then $\mathcal{B}_2$ queries V* to simulate verification of the final message of $\mathcal{A}$, $(\tau^*, \vec{w}^*, [\vec{x}^*, \vec{y}^*]_1)$. Now, it is easy to check if $(\vec{w}^*, [\vec{x}^*, \vec{y}^*]_1) \in \mathcal{L}_{\text{NO}}^{\mathsf{Lin}}$ by computing $[\mathbf{N}]_1 \vec{w}^*$. The difference between respective advantages is bounded using the core lemma of [32] as

$$|\mathsf{Adv}_1 - \mathsf{Adv}_2| \leq 2Q\mathsf{Adv}_{\mathcal{D}_k-\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_2) + \frac{Q}{p}.$$

– Game_3 is the same as Game_2 except the matrix $\mathbf{K} \leftarrow \mathbb{Z}_p^{(\ell_1+\ell_2)\times(k+1)}$ is changed in K to $\mathbf{K} = \mathbf{K}' + \vec{b}\vec{a}^\perp$ where $\mathbf{K}' \leftarrow \mathbb{Z}_p^{(\ell_1+\ell_2)\times(k+1)}$, $\vec{b}_1 \leftarrow \mathbb{Z}_p^{\ell_1}$, $\vec{b}_2 \leftarrow \mathbb{Z}_p^{\ell_2}$, $\vec{b}^\top = (\vec{b}_1^\top, \vec{b}_2^\top)$ and $\mathbf{B} = (\mathbf{M}^\top, \mathbf{N}^\top)\mathbf{K} + (\vec{z} + \mathbf{N}^\top \vec{b}_2)\vec{a}^\perp$, where $\vec{z} = \mathbf{M}^\top \vec{b}_1$. It is direct to see that both $\mathbf{K}$, $\mathbf{K}'$ are uniformly distributed in $\mathbb{Z}_p^{(\ell_1+\ell_2)\times(k+1)}$, so the advantages in both games are equivalent.

– Game_4 is the same as Game_3 except that now $\vec{z} \leftarrow \mathbb{Z}_p^{\ell_1}$. Let $\mathcal{B}_3$ be an adversary against $\mathcal{D}_k$-$\mathsf{MDDH}_{\mathbb{G}_1}$ that receives $([\mathbf{M}^\top]_1, [\vec{z}]_1)$ as a challenge and computes the crs as in the previous game with this $[\vec{z}]_1$ in $\mathbf{B}$ and runs $\mathcal{A}$ as in Game_3. Finally, the advantage of $\mathcal{B}_3$ to distinguish between Game_3 and Game_4 is bounded by the probability of distinguishing between a random vector from the image of the matrix $\mathbf{M}^\top$, so

$$|\mathsf{Adv}_3 - \mathsf{Adv}_4| \leq \mathsf{Adv}_{\mathcal{M}^\top-\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_3).$$

Now we bound the advantage of adversary $\mathcal{A}$ in winning Game_4. Firstly, we show what is leaked about vector $\vec{b}$ in the adversary's view:

– the matrix $\mathbf{C} = (\mathbf{K}' + \vec{b}\vec{a}^\perp)\mathbf{A}$ completely hides the vector $\vec{b}$,
– the output of S*, $(\vec{x}, \vec{y})^\top(\mathbf{K}' + \vec{b}\vec{a}^\perp) + \mu\vec{a}^\perp$ completely hides $\vec{b}$ because μ masks $(\vec{x}^\top, \vec{y}^\top)\vec{b}$,
– the matrix $\mathbf{B}$ contains information about $\vec{z} + \mathbf{N}^\top \vec{b}_2$, but $\vec{z}$ is uniformly random and independent of $\vec{b}_2$, so $\vec{z}$ masks $\vec{b}_2$.

Note that if the adversary $\mathcal{A}$ passes the verification V* with some $\vec{\pi}^*$ for a statement $(\vec{w}^*, \vec{x}^*, \vec{y}^*) \in \mathcal{L}_{\text{NO}}^{\mathsf{Lin}}$, it can compute $\vec{y} = \mathbf{N}\vec{w}^*$ and construct a valid proof $\pi = (\vec{\pi}_1^* - \vec{w}^*\mathbf{B}, \vec{\pi}_2^*)$ that the vector $(\vec{0}, \vec{y} - \vec{y}^*)$ is in the span of the columns $(\mathbf{M}^\top, \mathbf{N}^\top)$. It must hold that

$$\pi = (0, \vec{y} - \vec{y}^*)(\mathbf{K}' + \vec{b}\vec{a}^\perp) = (\vec{y} - \vec{y}^*)\mathbf{K}_2' + (\vec{y} - \vec{y}^*)\vec{b}_2\vec{a}^\perp. \qquad (*)$$

Note $\vec{y} - \vec{y}^*$ is not zero because $\vec{y} \neq \vec{y}^*$. Since $\vec{b}_2$ remains completely hidden to the adversary and $\mathbf{K}_2'$ is independent of $\vec{b}_2$, the probability than equation $(*)$ holds is less that $1/p$. □

6.2 USS $\mathsf{BLin}_{\mathcal{D}_k}$ argument

In this section we present the USS argument for membership in linear spaces in groups $\mathbb{G}_1$, $\mathbb{G}_2$, which is just an extension to bilateral spaces of the USS $\mathsf{Lin}_{\mathcal{D}_k}$ argument presented in Section 6.1 for the promise problem defined by languages

$$\mathcal{L}_{\text{YES}}^{\mathsf{Blin}} = \left\{ (\vec{w}, [\vec{x}_1]_1, [\vec{x}_2]_1, [\vec{y}]_2) : \begin{array}{c} [\vec{x}_1]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{x}_2]_1 = [\mathbf{N}]_1\vec{w}, [\vec{y}]_2 = [\mathbf{P}]_2\vec{w} \end{array} \right\}$$

$$\mathcal{L}_{\text{NO}}^{\mathsf{Blin}} = \left\{ (\vec{w}, [\vec{x}_1]_1, [\vec{x}_2]_1, [\vec{y}]_2) : \begin{array}{c} [\vec{x}_1]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{x}_2]_1 \neq [\mathbf{N}]_1\vec{w} \text{ or } [\vec{y}]_2 \neq [\mathbf{P}]_2\vec{w} \end{array} \right\}$$

parameterized by matrices $\mathbf{M} \in \mathbb{Z}_p^{\ell_1\times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2\times n}, \mathbf{P} \in \mathbb{Z}_p^{\ell_3\times n}$ sampled from some distributions $\mathcal{M}, \mathcal{N}, \mathcal{P}$. This argument is presented in Figure 3. QA-NIZK arguments of membership in linear spaces were extended to the bilateral case in [20] for both samplable and non-witness samplable distributions. In [21], the authors proved that the argument for non-witness samplable distributions of [20] is also sound and complete for this promise problem. Adding the pseudorandom MAC given in [32] we get USS. The proof is essentially the same as in 6.1, but now the linear spaces are split in two groups $\mathbb{G}_1$ and $\mathbb{G}_2$. The core lemma would be the same and the reduction of the proof of USS is bounded by $\mathsf{SKerMDH}$ and $\mathcal{D}_k$-$\mathsf{MDDH}_{\mathbb{G}_1}$ Assumptions.

$\underline{\mathrm{K}\left(gk, [\mathbf{M}]_1, [\mathbf{N}]_1, [\mathbf{P}]_2\right):}$
$\mathbf{K}_1 \leftarrow \mathbb{Z}_p^{\ell_1 \times (k+1)}, \mathbf{K}_2 \leftarrow \mathbb{Z}_p^{\ell_2 \times (k+1)},$
$\mathbf{K}_3 \leftarrow \mathbb{Z}_p^{\ell_3 \times (k+1)},$
$\mathbf{A}, \mathbf{\Omega} \leftarrow \mathcal{D}_k, \mathbf{\Gamma} \leftarrow \mathbb{Z}_p^{n \times (k+1)},$
$\mathbf{\Omega}_0, \mathbf{\Omega}_1 \leftarrow \mathbb{Z}_p^{(k+1) \times (k+1)}$
$\mathbf{C}_1 = \mathbf{K}_1\mathbf{A}, \mathbf{C}_2 = \mathbf{K}_2\mathbf{A}, \mathbf{C}_3 = \mathbf{K}_3\mathbf{A},$
$[\mathbf{B}]_1 = \left[\mathbf{M}^\top\mathbf{K}_1 + \mathbf{N}^\top\mathbf{K}_2 + \mathbf{\Gamma}\right]_1$
$[\mathbf{D}]_2 = \left[\mathbf{P}^\top\mathbf{K}_3 - \mathbf{\Gamma}\right]_2$
$(\mathbf{P}_0, \mathbf{P}_1) = (\mathbf{\Omega}^\top\mathbf{\Omega}_0, \mathbf{\Omega}^\top\mathbf{\Omega}_1)$
$(\mathbf{Q}_0, \mathbf{Q}_1) = (\mathbf{\Omega}_0\mathbf{A}, \mathbf{\Omega}_1\mathbf{A})$
Return $\mathsf{crs} = (gk, [\mathbf{B}]_1, [\mathbf{A}]_{1,2}, [\mathbf{P}_0]_2,$
$[\mathbf{P}_1]_2, [\mathbf{Q}_0]_1, [\mathbf{Q}_1]_1, [\mathbf{C}_1]_2, [\mathbf{C}_2]_2,$
$[\mathbf{C}_3]_1, [\mathbf{\Omega}]_1)$
$\boldsymbol{\theta} = [\vec{y}]_2\mathbf{K}_3^\top.$
$\mathsf{tr} = (\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3)$

$\underline{\mathrm{P}\left(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \boldsymbol{w}\right):}$
Pick $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left(\boldsymbol{w}^\top[\mathbf{B}]_1 + \vec{r}^\top[\mathbf{P}_0 + \tau\mathbf{P}_1]_1,\right.$
$\left.[\vec{r}^\top\mathbf{\Omega}^\top]_1\right),$
$\boldsymbol{\theta} = \boldsymbol{w}^\top[\mathbf{D}]_2.$
$\underline{\mathrm{V}\left(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \vec{\pi}, \boldsymbol{\theta}\right):}$
Check if: $e\left(\vec{\pi}_1, [\mathbf{A}]_2\right) - e\left([\mathbf{A}]_1, \boldsymbol{\theta}\right)$
$-e\left(\left[\boldsymbol{x}_1^\top\right]_1, [\mathbf{C}_1]_2\right) - e\left(\left[\boldsymbol{x}_2^\top\right]_1, [\mathbf{C}_2]_2\right)$
$+e\left(\left[\mathbf{C}_3\right]_1, \left[\boldsymbol{y}^\top\right]_2\right) = e\left(\vec{\pi}_2, [\mathbf{Q}_0 + \tau\mathbf{Q}_1]_2\right)$
$\underline{\mathrm{S}\left(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \mathsf{tr}\right):}$
Sample $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left([\vec{x}_1, \vec{x}_2]_1(\mathbf{K}_1^\top, \mathbf{K}_2^\top)\right.$
$\left.+\vec{r}^\top(\mathbf{P}_0 + \tau\mathbf{P}_1), [\vec{r}^\top\mathbf{\Omega}^\top]_1\right),$

Fig. 3. The $\mathsf{BLin}_{\mathcal{D}_k}$ argument for proving membership in linear spaces in blocks $([\vec{x}_1, \vec{x}_2]_1, [\vec{y}]_2) \in \mathrm{Im}\left([\mathbf{M}, \mathbf{N}]_1, [\mathbf{P}]_2,\right)$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}, \mathbf{P} \in \mathbb{Z}_p^{\ell_3 \times n}$.

7 Tight USS QA-NIZK Arguments of Knowledge Transfer for Linear Spaces

In this section we prove that the Tight USS argument of Abe et al. [1] for membership in linear spaces satisfies the knowledge transfer property explained in Section 1. The authors present a Designated Verifier (DV) QA-NIZK argument and then use a well-known conversion from DV to public verifier QA-NIZK with pairings. We follow the same approach and we further modify it to be a tag-based argument and adapt the sub-argument for disjunction spaces to the one of Couteau and Hartmann [11] for efficiency.

In Section 7.1 we prove the DV QA-NIZK of [1] is perfectly complete, perfect zero-knowledge and USS for the language associated to promise problems for linear spaces, already defined in Section 6, namely:

$$\mathcal{L}_{\mathrm{YES}}^{\mathsf{Lin}} = \left\{(\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c}[\vec{x}]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{y}]_1 = [\mathbf{N}]_1\vec{w}\end{array}\right\}$$

$$\mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}} = \left\{(\vec{w}, [\vec{x}]_1, [\vec{y}]_1) : \begin{array}{c}[\vec{x}]_1 = [\mathbf{M}]_1\vec{w} \text{ and} \\ [\vec{y}]_1 \neq [\mathbf{N}]_1\vec{w}\end{array}\right\}$$

parametrized by matrices $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$ sampled from some distributions $\mathcal{M}, \mathcal{N}$. In Section 7.2 we present its natural conversion to a publicly verifiable QA-NIZK argument. We only give the argument for membership in linear spaces in one group, the bilateral version is straightforward following the work of [20], where the authors transform QA-NIZK arguments for membership in linear spaces in one group to membership in linear spaces to both groups, namely bilateral spaces.

Security Proof: Intuition. Our construction revisits the proof of Abe et al.'s DV argument for promise problems. In this approach the secret keys are vectors $\vec{k}_0, \vec{k}_1$ and the proofs, $(\vec{x}_i^\top, \vec{y}_i^\top)(\vec{k}_0 + \tau_i\vec{k}_1)$ where τ is a different value in $\mathbb{Z}_p$ for each proof. We split the secret keys $\vec{k}_0 = (\vec{k}_{1,0}, \vec{k}_{2,0})$, $\vec{k}_1 = (\vec{k}_{1,1}, \vec{k}_{2,1})$ to indicate the components that come with $\mathbf{M}^\top$, $\vec{k}_{1,0}, \vec{k}_{1,1}$, and the others with $\mathbf{N}^\top$, $\vec{k}_{2,0}, \vec{k}_{2,1}$.

We use a similar solution as in Section 6 and argue that partial information of the secret keys necessary to produce a proof in the NO language is hidden across all the proofs. In this construction, the crs contains projections of the secret keys $\vec{k}_0, \vec{k}_1$ by matrices $\mathbf{M}^\top, \mathbf{N}^\top$. Assuming the $\mathcal{M}^\top$-$\mathsf{MDDH}_{\mathbb{G}_1}$ assumption holds, where $\mathcal{M}^\top$ is the distribution of $\mathbf{M}^\top$, as in Section 6, we change the projection by $\mathbf{M}^\top$ by a random vector $\vec{z}$, which masks completely the projection by $\mathbf{N}^\top$.

Note that in the construction of Abe et al.'s we use in this section, there are also more projections of the secret keys leaked from simulated proofs, concretely: $\vec{x}_i^\top(\vec{k}_{1,0} + \tau_i\vec{k}_{1,1}) + \vec{y}_i^\top(\vec{k}_{2,0} + \tau_i\vec{k}_{2,1})$. But we can use the same information-theoretic argument as in [1], namely, since τ_i is different each time, $\vec{k}_{1,0} + \tau_i\vec{k}_{1,1}$,$\vec{k}_{2,0} + \tau_i\vec{k}_{2,1}$ are pairwise independent, then they do not add any clue to the adversary.

7.1 Tight DV QA-NIZK Argument of Knowledge Transfer for Linear Spaces.

The DV QA-NIZK argument presented in Figure 4 is the argument for linear spaces of Abe et al. [1] written in blocks, and (trivially) modified to admit tags. Also, We use the disjunction argument of Couteau and Hartmann [11], which is 3 group elements more efficient than the one presented in [2] (used in the first construction of Abe et al. [1]), and we denote it by or.

Security. We prove it has completeness for $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{YES}}$ and USS for $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$. USS relies in the same core lemma as in Abe et al. (Lemma 3 in [1]), the security of the MAC presented in Gay et al. [17], the soundness of an argument for membership in a disjunction space of [11]. Our contribution is to combine this with the same techniques as in Section 6 to adapt the proof for promise problems.

$\underline{\mathrm{K}\,(gk, [\mathbf{M}]_1, [\mathbf{N}]_1):}$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}, H \leftarrow \mathcal{H},$
$\mathsf{crs}_{\mathsf{or}} \leftarrow \mathrm{K}(gk, \mathbf{A}_0, \mathbf{A}_1)$
$\vec{k} \leftarrow \mathbb{Z}_p^{2k}, \vec{k}_0 = (\vec{k}_{1,0}, \vec{k}_{2,0}), \vec{k}_1 = (\vec{k}_{1,1}, \vec{k}_{2,1}) \leftarrow \mathbb{Z}_p^n,$
$\vec{k}_{1,0}, \vec{k}_{1,1} \in \mathbb{Z}_p^{\ell_1}, \vec{k}_{2,0}, \vec{k}_{2,1} \in \mathbb{Z}_p^{\ell_2}$
$[\vec{p}]_1 = [\mathbf{A}_0^\top \vec{k}]_1 \in \mathbb{G}_1^k,$
$[\vec{p}_0]_1 = [\mathbf{M}^\top \vec{k}_{1,0} + \mathbf{N}^\top \vec{k}_{2,0}]_1 \in \mathbb{G}_1^n,$
$[\vec{p}_1]_1 = [\mathbf{M}^\top \vec{k}_{1,1} + \mathbf{N}^\top \vec{k}_{2,1}]_1 \in \mathbb{G}_1^n,$
$\mathsf{crs} = (\mathsf{crs}_{\mathsf{or}}, [\mathbf{A}_0]_1, [\vec{p}]_1, [\vec{p}_0]_1, [\vec{p}_1]_1, H)$
$\mathsf{tr} = (\vec{k}_0, \vec{k}_1), \mathsf{vk} = (\vec{k}, \vec{k}_0, \vec{k}_1).$

$\underline{\mathrm{S}\,(\mathsf{crs}, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \tilde{\tau}, \mathsf{tr}):}$
$\vec{s} \leftarrow \mathbb{Z}_p^k, [\vec{t}]_1 = [\mathbf{A}_0]_1 \vec{s}, [\pi_{\mathsf{or}}]_{1,2} \leftarrow \mathrm{P}_{\mathsf{or}}(\mathsf{crs}_{\mathsf{or}}, [\vec{t}]_1, \vec{s})$
$\tau = H([\vec{x}]_1, [\vec{y}]_1, [\vec{t}]_1, [\pi_{\mathsf{or}}]_{1,2}, \tilde{\tau}) \in \mathbb{Z}_p,$
$[u]_1 = [\vec{x}^\top]_1(\vec{k}_{1,0} + \tau \vec{k}_{1,1}) + [\vec{y}^\top]_1(\vec{k}_{2,0} + \tau \vec{k}_{2,1}) + \vec{s}^\top [\vec{p}]_1$
Return $[\pi]_1 = ([\vec{t}]_1, [u]_1, [\pi_{\mathsf{or}}]_{1,2})$

$\underline{\mathrm{P}\,(\mathsf{crs}, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \tilde{\tau}, \boldsymbol{w}):}$
$\vec{s} \leftarrow \mathbb{Z}_p^k, [\vec{t}]_1 = [\mathbf{A}_0]_1 \vec{s}$
$[\pi_{\mathsf{or}}]_{1,2} \leftarrow \mathrm{P}_{\mathsf{or}}(\mathsf{crs}_{\mathsf{or}}, [\vec{t}]_1, \vec{s})$
$\tau = H([\vec{x}]_1, [\vec{y}]_1, [\vec{t}]_1, [\pi_{\mathsf{or}}]_{1,2}, \tilde{\tau}) \in \mathbb{Z}_p$
$[u]_1 = [\vec{w}^\top(\vec{p}_0 + \tau \vec{p}_1) + \vec{s}^\top \vec{p}]_1$
Return $[\pi]_1 = ([\vec{t}]_1, [u]_1, [\pi_{\mathsf{or}}]_{1,2})$.

$\underline{\mathrm{V}\,(\mathsf{crs}, [\boldsymbol{x}]_1, [\boldsymbol{y}]_1, \mathsf{vk}, [\pi]_1, \tilde{\tau}):}$
Parse $[\pi] = ([\vec{t}]_1, [u]_1, [\pi_{\mathsf{or}}]_{1,2}),$
$\tau = H([\vec{x}]_1, [\vec{y}]_1, [\vec{t}]_1, [\pi_{\mathsf{or}}]_{1,2}, \tilde{\tau}) \in \mathbb{Z}_p,$
Check $[\pi_{\mathsf{or}}]_{1,2}$ and $[u]_1 = [\vec{x}^\top]_1(\vec{k}_{1,0} + \tau \vec{k}_{1,1})$
$+[\vec{y}^\top]_1(\vec{k}_{2,0} + \tau \vec{k}_{2,1}) + [\vec{t}^\top]_1 \vec{k}$
Return 0/1.

Fig. 4. Tight DV QA-NIZK Argument for membership in linear spaces of Abe et al. [1] in blocks, $[\vec{x}, \vec{y}]_1 \in \mathrm{Im}[\mathbf{M}, \mathbf{N}]_1$, where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$ and $\mathcal{H}$ a family of collision-resistant hash functions. The scheme is modified to be tag-based and is written in blocks. We use the disjunction argument or of [11] with $|\mathsf{crs}_{\mathsf{or}}| = (4n+8)|\mathbb{G}_1| + (2\ell_1+3)|\mathbb{G}_2|$, $|\pi_{\mathsf{or}}| = 8|\mathbb{G}_1| + 3|\mathbb{G}_2|$.

The scheme in Figure 4 is perfectly complete and perfect-zero knowledge for YES instances, and soundness guarantees that NO instances will not be accepted as we show in the following. As in Section 6 we consider the general language $\mathcal{L}$ that includes all tuples $(\vec{w}, \vec{x}, \vec{y})$ of the right dimension, some of them are outside of $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{YES}} \cup \mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$. We allow simulation queries for any tuple in $\mathcal{L}$.

Perfect Completeness, Perfect Zero-Knowledge. Our language $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{YES}}$ is the same language for membership proofs in a linear space $[\mathbf{M}, \mathbf{N}]_1^\top$ used in [1]: $\left\{(\vec{w}, [\vec{x}, \vec{y}]_1) : [\vec{x}, \vec{y}]_1^\top = [\mathbf{M}, \mathbf{N}]_1^\top \vec{w}\right\}$. Thus, we directly obtain perfect completeness and perfect zero-knowledge.

Unbounded Simulation Soundness. We use the definition 4 where for any adversary $\mathcal{A}$ that sends any number Q of queries $(\vec{w}_i, [\vec{x}_i, \vec{y}_i]_1, \tilde{\tau}_i)$ to the query simulator oracle S, receives simulated proofs $\{[\pi_i]_1\}_{i=1}^Q$ as described in Figure 4. The probability of the adversary $\mathcal{A}$ comes up with a proof $[\pi^*]_1$ for a statement $(\vec{w}^*, [\vec{x}^*, \vec{y}^*]_1) \in \mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$ different of the queried ones and different tag $\tilde{\tau}^*$, such that $\mathrm{V}(\mathsf{crs}, \tilde{\tau}^*, [\vec{x}^*, \vec{y}^*]_1, [\pi^*]_1) = 1$, is negligible.

Abe et al.'s construction is based in the USS Kiltz and Wee argument [32], where the security relies in three security features that we use as black-boxes: their core lemma (Lemma 3 in [1]), the security of a MAC scheme presented in Gay et al. [17], and the soundness of the or argument, all proven secure under standard assumptions.

Both [32] and [1] use a MAC scheme to add randomness to the proof. Concretely, by the Gay et al. MAC, the term $\vec{t}^\top \vec{k}$ is added to the proof, where $\vec{k}$ is uniformly random and $\vec{t} \in \mathrm{Span}(\mathbf{A}_0) \cup \mathrm{Span}(\mathbf{A}_1)$ for some fixed matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_p^{2k \times k}$ in the crs. The basic idea is the prover computes $\vec{t}$ directly in the image of $[\mathbf{A}_0]_1$, uses the argument or to prove membership of $\vec{t}$ in $\mathrm{Span}(\mathbf{A}_0) \cup \mathrm{Span}(\mathbf{A}_1)$ and uses the image space of $[\mathbf{A}_1]_1$ to add randomness in the security proof. The last is done by changing to a game where $\vec{k} \in \mathbb{Z}_p^{2k}$ is switched to $\vec{k} + \mathsf{RF}(\cdot)$, with $\mathsf{RF} : \mathbb{Z}_p \to \mathbb{Z}_p^{2k}$

a random function. Indistinguishability of both games is proven in [17], concretely, the lemma gives the following tight bound for any adversary $\mathcal{A}$ that is able to distinguish between both MAC schemes:

$$\mathsf{Adv}_{\mathsf{CL}}(\mathcal{A}) \leq (4k\lceil \log Q\rceil + 2)\mathsf{Adv}_{\mathcal{D}_{2k,k}\text{-}\mathsf{MDDH}_{\mathbb{G}_1},\mathcal{B}}(\lambda) + (2\lceil \log Q\rceil + 2)\mathsf{Adv}_{\mathsf{zk\text{-}or},\mathcal{B}'}(\lambda)$$

$$+\lceil \log Q\rceil \Delta_{\mathcal{D}_{2k,k}} + \frac{4\lceil \log Q\rceil + 2}{p-1} + \frac{\lceil \log Q\rceil Q}{p},$$

where $\Delta_{\mathcal{D}_{2k,k}}$ is statistically small term for $\mathcal{D}_{2k,k}$, $\mathcal{B}$ and $\mathcal{B}'$ are adversaries against the $\mathcal{D}_{2k,k}$-$\mathsf{MDDH}_{\mathbb{G}_1}$ assumption and zero-knowledge of argument or respectively.

Theorem 5. *The argument of Figure 4 is a Designated Verifier Quasi-Adaptive Non-Interactive Zero-Knowledge argument that guarantees USS such that for any adversary $\mathcal{A}$ that makes at most Q queries to S, there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ against collision resistance of $\mathcal{H}$, core lemma of [17] and $\mathcal{M}^\top$-$\mathsf{MDDH}_{\mathbb{G}_1}$ assumption such that*

$$\mathsf{Adv}_{USS}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{CR}}(\mathcal{B}_1) + \mathsf{Adv}_{\mathsf{CL}}(\mathcal{B}_2) + 2\mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_3) + \frac{Q}{p}.$$

Proof. We proceed via changes of games starting with Game_0 that is the real USS game of definition 4. Let Adv_i be the advantage of adversary $\mathcal{A}$ winning Game_i.

- Game_1 is the same as Game_0 except the simulator computes the element $[u]_1$ as $[\vec{x}]_1(\vec{k}_{1,0} + \tau\vec{k}_{1,1}) + [\vec{y}]_1(\vec{k}_{2,0} + \tau\vec{k}_{2,1}) + [\vec{t}^\top]_1\vec{k}$ and verification of final adversary's message $(\vec{w}^*, [\vec{x}^*]_1, [\vec{y}^*]_1, [\pi^*]_1, \tilde{\tau}^*)$ checks:
  - $(\vec{w}^*, [\vec{x}^*]_1, [\vec{y}^*]_1) \in \mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$,
  - $([\vec{x}^*]_1, [\vec{y}^*]_1) \notin \mathcal{Q}_{sim}$,
  - receives $\tilde{\tau}^*$, and checks that $\tilde{\tau}^* \notin \mathcal{Q}_{tag}$. With o.w.p, by the collision resistance of H, this implies that $\tau^* = H([\vec{x}^*]_1, [\vec{y}^*]_1, [\vec{t}^*]_1, [\pi^*_{\mathsf{or}}]_{1,2}, \tilde{\tau}^*)$ is also different from all the tags used in the simulated proofs.

  The new element $[u]_1$ just differs on the element $[\vec{t}^\top]_1\vec{k}$, which in Game_0 is $\vec{s}^\top[\vec{p}]_1$, they pass verification with same probability because they are equivalent by definition. Thus,

$$|\mathsf{Adv}_0 - \mathsf{Adv}_1| \leq \mathsf{Adv}_{\mathsf{CR}}(\mathcal{B}_1).$$

- Game_2 is the same as Game_1 except that the key $\vec{k}$ is changed to $\vec{k} + \mathsf{RF}(\cdot)$ where $\mathsf{RF} : \mathbb{Z}_p \to \mathbb{Z}_p^{2k}$ is a random function. Concretely, the element $[\vec{p}]_1 = [\mathbf{A}_0^\top\vec{k}]_1$ is switched to $[\vec{p}]_1 = [\mathbf{A}_0^\top(\vec{k} + \mathsf{RF}(0))]$ in K and the element $[u]_1$ in S is computed as $[u]_1 = [(\vec{x}_i, \vec{y}_i)(\vec{k}_0 + \tau_i\vec{k}_1) + \vec{t}_i^\top(\vec{k} + \mathsf{RF}(i))]_1$ for the i-th query. Moreover, the verifier V defines the set $\mathcal{S} = \{[(\vec{x}^*, \vec{y}^*)(\vec{k}_0 + \tau^*\vec{k}_1) + \vec{t}^{*\top}(\vec{k} + \mathsf{RF}(j^*))]_1\}_{j^*=0}^{Q}$ and checks $[u^*]_1 \in \mathcal{S}$. The indistinguishability between Game_1 and Game_2 is direct from the core lemma [1] because it is equivalent of indistinguishability between both MACs defined in the core lemma, thus

$$|\mathsf{Adv}_1 - \mathsf{Adv}_2| \leq \mathsf{Adv}_{\mathsf{CL}}(\mathcal{B}_2).$$

- Game_3 is the same as Game_2 except that the elements $[\vec{p}_0]_1 = [\mathbf{M}^\top\vec{k}_{1,0} + \mathbf{N}^\top\vec{k}_{2,0}]_1$ and $[\vec{p}_1]_1 = [\mathbf{M}^\top\vec{k}_{1,1} + \mathbf{N}^\top\vec{k}_{2,1}]_1$ are switched to $[\vec{p}_0]_1 = [\vec{z}_0 + \mathbf{N}^\top\vec{k}_{2,0}]_1$ and $[\vec{p}_1]_1 = [\vec{z}_1 + \mathbf{N}^\top\vec{k}_{2,1}]_1$ in K, where $\vec{z}_0, \vec{z}_1 \leftarrow \mathbb{Z}_p^n$. We can think in an intermediate game where we just switch $[\vec{p}_0]_1$, then for any adversary $\mathcal{B}_3$ able to distinguish between these intermediate games and Game_2 is breaking $\mathcal{M}^\top$-$\mathsf{MDDH}_{\mathbb{G}_1}$. By the same argument, $\mathcal{B}_3$ distinguishing between the intermediate game and Game_3 is breaking $\mathcal{M}^\top$-$\mathsf{MDDH}_{\mathbb{G}_1}$. Finally,

$$|\mathsf{Adv}_2 - \mathsf{Adv}_3| \leq 2\mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}_{\mathbb{G}_1}}(\mathcal{B}_3).$$

Before studying the probability of the adversary $\mathcal{A}$ wins the Game_3, note that by linearity, we observe that the proof π^* is a valid proof to prove membership in the linear space of the vector $([\vec{0}]_1, [\vec{y}^*]_1)$. For any adversary that makes a proof $[\pi^*]_1$ for $(\vec{w}^*, [\vec{x}^*]_1, [\vec{y}^*]_1) \in \mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$, the element $[\overline{u}^*]_1 = [u^*]_1 - \vec{w}^*[\vec{p}_0]_1 - \vec{w}^*[\vec{p}_1]_1\tau^*$ is a valid proof for $([\vec{0}^*]_1, [\vec{y}^* - \overline{\vec{y}}]_1)$ where $\overline{\vec{y}} = \mathbf{N}\vec{w}^*$ (with same $[\vec{t}^*]_1$ and $[\pi_{\mathsf{or}}]_{1,2}$).

Now, we use an information-theoretic argument to bound the probability of success of the adversary $\mathcal{A}$. In the first place, we study what is leaked about the secret keys. The elements $[\vec{p}_0]_1 = [\vec{z}_0 + \mathbf{N}^\top\vec{k}_{2,0}]_1$, $[\vec{p}_1]_1 = [\vec{z}_1 + \mathbf{N}^\top\vec{k}_{2,1}]_1$ in the crs do not leak information about $\mathbf{N}^\top\vec{k}_{2,0}$ and $\mathbf{N}^\top\vec{k}_{2,1}$ because the vectors $[\vec{z}_0]_1$, $[\vec{z}_1]_1$ hide completely the projections by $\mathbf{N}$. Then, the element $\vec{y}^{*\top}(\vec{k}_{2,0} + \tau^*\vec{k}_{2,1})$ in the proof, where $[\vec{y}^*]_1 \notin \mathrm{Span}[\mathbf{N}]_1$, is uniformly random in adversary's view.

The adversary $\mathcal{A}$ also learns the following projections of the secret keys from each query i: $\vec{x}_i^\top(\vec{k}_{1,0}+\tau_i\vec{k}_{1,1})+\vec{y}_i^\top(\vec{k}_{2,0}+\tau_i\vec{k}_{2,1})$, but they are pairwise independent and $\vec{y}_i \neq \vec{y}^*$ for all $i=1,\ldots,Q$. So, given $\vec{x}_i^\top(\vec{k}_{1,0}+\tau_i\vec{k}_{1,1})+\vec{y}_i^\top(\vec{k}_{2,0}+\tau_i\vec{k}_{2,1})$ from the i-th query, the term $\vec{y}^{*\top}(\vec{k}_{2,0}+\tau^*\vec{k}_{2,1})$ in the proof is distributed uniformly at random. Thus, the probability of $\mathcal{A}$ computes this term and passes verification is $1/p$. Finally, taking into account there are Q simulated proofs, we have

$$|\mathsf{Adv}_3(\mathcal{A})| = \frac{Q}{p}.$$

□

7.2 Tight USS $\mathsf{Lin}_{\mathcal{D}_k}$ QA-NIZK

The QA-NIZK argument in Figure 5 is the Tight USS QA-NIZK argument for membership in linear spaces of Abe et al. [1] written in blocks for promise problem languages $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{YES}}$ and $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$ defined in Section 4.2. It is the straightforward construction from the tight DV QA-NIZK of the previous Section 7.1 to public verifier QA-NIZK with pairings.

$\underline{\mathrm{K}\,(gk,[\mathbf{M}]_1,[\mathbf{N}]_1):}$
$\mathbf{A}_0,\mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}$,
$\mathsf{crs}_{\mathsf{or}} \leftarrow \mathrm{K}(gk,\mathbf{A}_0,\mathbf{A}_1)$
$H \leftarrow \mathcal{H}, \mathbf{A} \leftarrow \mathcal{D}_k$
$\mathbf{K} \leftarrow \mathbb{Z}_p^{2k\times k}, m=\ell_1+\ell_2,$ for $i=0,1:$
$\mathbf{K}_i = \left(\overline{\mathbf{K}_i},\underline{\mathbf{K}_i}\right)^\top \leftarrow \mathbb{Z}_p^{m\times(k+1)}$,
$\overline{\mathbf{K}_i} \in \mathbb{Z}_p^{\ell_1\times(k+1)}, \underline{\mathbf{K}_i} \in \mathbb{Z}_p^{\ell_2\times(k+1)}$.
$[\mathbf{P}]_1 = [\mathbf{A}_0^\top\mathbf{K}]_1 \in \mathbb{G}_1^{k\times(k+1)}$
$[\mathbf{P}_0]_1 = [\mathbf{M}^\top\overline{\mathbf{K}_0}+\mathbf{N}^\top\underline{\mathbf{K}_0}]_1 \in \mathbb{G}_1^{n\times(k+1)}$
$[\mathbf{P}_1]_1 = [\mathbf{M}^\top\overline{\mathbf{K}_1}+\mathbf{N}^\top\underline{\mathbf{K}_1}]_1 \in \mathbb{G}_1^{n\times(k+1)}$,
$\mathbf{C} = \mathbf{K}\mathbf{A} \in \mathbb{Z}_p^{2k\times k}$,
$\mathbf{C}_0 = \mathbf{K}_0\mathbf{A}, \mathbf{C}_1 = \mathbf{K}_1\mathbf{A} \in \mathbb{Z}_p^{m\times k}$
$\mathsf{crs} = (\mathsf{crs}_{\mathsf{or}},[\mathbf{A}_0]_1,[\mathbf{P}]_1,[\mathbf{P}_0]_1,[\mathbf{P}_1]_1,[\mathbf{A}]_2,$
$[\mathbf{C}]_2,[\mathbf{C}_0]_2,[\mathbf{C}_1]_2,H)$,
$\mathsf{tr} = (\mathbf{K}_0,\mathbf{K}_1)$.

$\underline{\mathrm{P}\,(\mathsf{crs},[\boldsymbol{x}]_1,[\boldsymbol{y}]_1,\boldsymbol{w},\tilde{\tau}):}$
$\vec{s} \leftarrow \mathbb{Z}_p^k, [\vec{t}]_1 = [\mathbf{A}_0]_1\vec{s}, [\pi_{\mathsf{or}}]_{1,2} \leftarrow \mathrm{P}_{\mathsf{or}}(\mathsf{crs}_{\mathsf{or}},[\vec{t}]_1,\vec{s})$
$\tau = H([\vec{x}]_1,[\vec{y}]_1,[\vec{t}]_1,[\pi_{\mathsf{or}}]_{1,2},\tilde{\tau}) \in \mathbb{Z}_p$
$[u]_1 = [\vec{w}^\top(\mathbf{P}_0+\tau\mathbf{P}_1)+\vec{s}^\top\mathbf{P}]_1 \in \mathbb{G}_1^{k+1}$
Return $[\pi]_1 = ([\vec{t}]_1,[u]_1,[\pi_{\mathsf{or}}]_{1,2})$.

$\underline{\mathrm{V}\,(\mathsf{crs},[\boldsymbol{x}]_1,[\boldsymbol{y}]_1,[\pi]_1,\tilde{\tau}):}$
Parse $[\pi]_1 = ([\vec{t}]_1,[u]_1,[\pi_{\mathsf{or}}]_{1,2})$,
$\tau = H([\vec{x}]_1,[\vec{y}]_1,[\vec{t}]_1,[\pi_{\mathsf{or}}]_{1,2},\tilde{\tau}) \in \mathbb{Z}_p$,
Check $[\pi_{\mathsf{or}}]_{1,2}$ and
$\left[u^\top\right]_1[\mathbf{A}]_2 = [\vec{x}^\top,\vec{y}^\top]_1[\mathbf{C}_0+\tau\mathbf{C}_1]+[\vec{t}^\top]_1\mathbf{C}$
Return 0/1.

$\underline{\mathrm{S}\,(\mathsf{crs},[\boldsymbol{x}]_1,[\boldsymbol{y}]_1,\mathsf{tr},\tilde{\tau}):}$
$\vec{s} \leftarrow \mathbb{Z}_p^k, [\vec{t}]_1 = [\mathbf{A}_0]_1\vec{s}, [\pi_{\mathsf{or}}]_{1,2} \leftarrow \mathrm{P}_{\mathsf{or}}(\mathsf{crs}_{\mathsf{or}},[\vec{t}]_1,\vec{s})$
$\tau = H([\vec{x}]_1,[\vec{y}]_1,[\vec{t}]_1,[\pi_{\mathsf{or}}]_{1,2},\tilde{\tau}) \in \mathbb{Z}_p$,
$[u]_1 = [\vec{x}^\top,\vec{y}^\top]_1(\mathbf{K}_0+\tau\mathbf{K}_1)+\vec{s}^\top[\mathbf{P}]_1$.

Fig. 5. Tight QA-NIZK Argument for membership in linear spaces of Abe et al. [1] in blocks, $[\vec{x},\vec{y}]_1 \in \mathrm{Im}[\mathbf{M},\mathbf{N}]_1$, where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1\times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2\times n}$ and $\mathcal{H}$ a family of hash functions that are collision resistant. The scheme is modified to be tag-based. We use the disjunction argument or of [11] with $|\mathsf{crs}_{\mathsf{or}}| = (4n+8)|\mathbb{G}_1|+(2\ell_1+3)|\mathbb{G}_2|$, $|\pi_{\mathsf{or}}| = 8|\mathbb{G}_1|+3|\mathbb{G}_2|$.

The security proof is analogous to the security proof of the tight QA-NIZK of Abe et al. [1]. In that construction, the authors give a tight reduction where the advantage of breaking the USS of the QA-NIZK is bounded by the advantage of breaking USS of the DV QA-NIZK and a kernel assumption. As we have seen in Section 7.1 the USS of our DV QA-NIZK is proven by a tight reduction that is linear in $\log Q$, where Q is the number of simulated queries. So, the USS of the QA-NIZK argument presented here inherits the same tightness loss linear in $\log Q$.

The bilinear QA-NIZK argument of Section 5 is a membership proof in linear spaces in two groups $\mathbb{G}_1$, $\mathbb{G}_2$, for the same languages as defined in 6.2. It is easily constructed from the bilinear version of the DV QA-NIZK argument 7.1. The reduction is analogous to the unilateral QA-NIZK reduction. We bound the advantage of breaking USS of the QA-NIZK for bilateral spaces by the advantage of breaking the USS of DV QA-NIZK for bilateral spaces and the $\mathsf{SKerMDH}$ assumption, with same tightness loss linear in $\log Q$.

Acknowledgement. The research leading to this article was partially supported by Project RTI2018-102112-B-I00 (AEI/FEDER, UE). Karim Baghery was supported by CyberSecurity Research Flanders with reference number VR20192203.

Acknowledgements. Karim Baghery was supported by CyberSecurity Research Flanders with reference number VR20192203 and Defense Advanced Research Projects Agency (DARPA) with contract number HR001120C0085.

References

1. M. Abe, C. S. Jutla, M. Ohkubo, J. Pan, A. Roy, and Y. Wang. Shorter QA-NIZK and SPS with tighter security. In S. D. Galbraith and S. Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 669–699. Springer, Heidelberg, Dec. 2019. 4, 19, 20, 21, 22
2. M. Abe, C. S. Jutla, M. Ohkubo, and A. Roy. Improved (almost) tightly-secure simulation-sound QA-NIZK with applications. In T. Peyrin and S. Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 627–656. Springer, Heidelberg, Dec. 2018. 20
3. K. Baghery. On the efficiency of privacy-preserving smart contract systems. In J. Buchmann, A. Nitaj, and T. eddine Rachidi, editors, *AFRICACRYPT 19*, volume 11627 of *LNCS*, pages 118–136. Springer, Heidelberg, July 2019. 2
4. K. Baghery. Subversion-resistant simulation (knowledge) sound NIZKs. In M. Albrecht, editor, *17th IMA International Conference on Cryptography and Coding*, volume 11929 of *LNCS*, pages 42–63. Springer, Heidelberg, Dec. 2019. 2
5. K. Baghery, A. González, Z. Pindado, and C. Ràfols. Signatures of knowledge for boolean circuits under standard assumptions. In A. Nitaj and A. M. Youssef, editors, *AFRICACRYPT 20*, volume 12174 of *LNCS*, pages 24–44. Springer, Heidelberg, July 2020. 6
6. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. 1
7. D. Bernhard, G. Fuchsbauer, and E. Ghadafi. Efficient signatures of knowledge and DAA in the standard model. In M. J. Jacobson Jr., M. E. Locasto, P. Mohassel, and R. Safavi-Naini, editors, *ACNS 13*, volume 7954 of *LNCS*, pages 518–533. Springer, Heidelberg, June 2013. 6
8. R. Canetti. Universally composable signatures, certification and authentication. Cryptology ePrint Archive, Report 2003/239, 2003. `http://eprint.iacr.org/2003/239`. 15
9. R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002. 2
10. M. Chase and A. Lysyanskaya. On signatures of knowledge. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 78–96. Springer, Heidelberg, Aug. 2006. 2, 8
11. G. Couteau and D. Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In D. Micciancio and T. Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, Aug. 2020. 19, 20, 22
12. I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, Aug. 1992. 1
13. G. Danezis, C. Fournet, J. Groth, and M. Kohlweiss. Square span programs with applications to succinct NIZK arguments. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, Dec. 2014. 5, 9
14. V. Daza, A. González, Z. Pindado, C. Ràfols, and J. Silva. Shorter quadratic QA-NIZK proofs. In D. Lin and K. Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, Apr. 2019. 5, 15
15. A. Escala and J. Groth. Fine-tuning Groth-Sahai proofs. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649. Springer, Heidelberg, Mar. 2014. 25
16. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, Aug. 2013. 5
17. R. Gay, D. Hofheinz, L. Kohl, and J. Pan. More efficient (almost) tightly secure structure-preserving signatures. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 230–258. Springer, Heidelberg, Apr. / May 2018. 20, 21
18. R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct NIZKs without PCPs. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013. 1, 2, 3, 5, 10, 12
19. C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. 1
20. A. González, A. Hevia, and C. Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629. Springer, Heidelberg, Nov. / Dec. 2015. 4, 13, 18, 19, 24
21. A. González and C. Ràfols. Shorter pairing-based arguments under standard assumptions. In S. D. Galbraith and S. Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 728–757. Springer, Heidelberg, Dec. 2019. 1, 2, 3, 4, 5, 10, 11, 12, 13, 15, 16, 17, 18
22. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, Heidelberg, Dec. 2006. 2, 4, 15
23. J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, Dec. 2010. 5
24. J. Groth. On the size of pairing-based non-interactive arguments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. 1, 2, 5

25. J. Groth and M. Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 581–612. Springer, Heidelberg, Aug. 2017. 2, 4, 6, 8, 15
26. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, Apr. 2008. 1
27. D. Hofheinz, D. Jia, and J. Pan. Identity-based encryption tightly secure under chosen-ciphertext attacks. In T. Peyrin and S. Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 190–220. Springer, Heidelberg, Dec. 2018. 1
28. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2013. 3, 6
29. C. S. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2014. 24
30. M. Karchmer and A. Wigderson. On span programs. In *Proceedings of Structures in Complexity Theory*, pages 102–111, 1993. 5
31. T. Kerber, A. Kiayias, M. Kohlweiss, and V. Zikas. Ouroboros crypsinous: Privacy-preserving proof-of-stake. In *2019 IEEE Symposium on Security and Privacy*, pages 157–174. IEEE Computer Society Press, May 2019. 2
32. E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, Apr. 2015. 3, 4, 7, 13, 16, 17, 18, 20
33. A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy*, pages 839–858. IEEE Computer Society Press, May 2016. 2
34. B. Libert, T. Peters, M. Joye, and M. Yung. Linearly homomorphic structure-preserving signatures and their applications. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 289–307. Springer, Heidelberg, Aug. 2013. 3
35. M. Naor. On cryptographic assumptions and challenges (invited talk). In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, Aug. 2003. 1
36. C. Ràfols. Stretching groth-sahai: NIZK proofs of partial satisfiability. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276. Springer, Heidelberg, Mar. 2015. 4
37. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999. 2

8 Adapting GS Proofs for Improved Efficiency

In this section we show how to add zero-knowledge to the circuit satisfiability proof. A naive use of GS proofs results in a considerable overhead.

More concretely, we need to prove many quadratic Pairing Product Equations (PPEs), i.e. equations with variables in $\mathbb{G}_1$ and $\mathbb{G}_2$. Recall that GS proofs have a commit-and-prove structure: first, given an equation, the prover commits to the witness (a solution to the equation, which is a vector $[\vec{x}]_1$ of elements in $\mathbb{G}_1$ and a vector $[\vec{y}]_2$ of elements in $\mathbb{G}_2$) and then it gives a proof that the committed values satisfy the equation. When trying to save group elements of the proof, we will save on the number of group elements necessary to commit to the witness.

We note that although there are several techniques to save on the "proof part" of GS proofs [29, 20] by aggregating proofs, they work for linear equations and not for quadratic.

In order to commit to the witness of satisfiability (a pair $[\vec{x}]_1, [\vec{y}]_2$) of an equation, individual commitments to each coordinate of these vectors are computed. We focus on the Symmetric EXternal Diffie-Hellman assumption instantiation of GS proofs for efficiency. Under this assumption, each individual commitment is either a dual-mode commitment based on DDH or an ElGamal ciphertext.

A natural idea to explore to reduce the commitment cost is to compute a single commitment to the whole vector $[\vec{x}]_1$ (and similarly for $[\vec{y}]_2$). This approach fails in general because GS proofs use some homomorphic properties of the commitments to combine them in a proof, and these are lost when using a single commitment to all of $[\vec{x}]_1$. This explains why, to the best of our knowledge, there is no technique to save on the commitment part of GS proofs which works *in general*, that is, for every set of equations of any form[11].

However, for the specific form of the equations we use in this paper, it is possible to exploit the specific form of the PPEs that we need to prove. More precisely, we can exploit that the equations, which depend on some group variables $\{L_i, R_i, O_i\}_{i=1}^d$ do not have cross terms, i.e. terms which multiply L_i with R_j, $i \neq j$.

[11] What is important in the equation form for using simultaneous commitments is the structure of the quadratic part. On the other hand, this is independent of the equation type, i.e. this remark applies to multiscalar multiplication or quadratic equations in the field as well.

More specifically, we show how to reduce the size of GS proofs for equations which can be written in this form:

$$e([k_j]_1,[1]_2)+e([x_j]_1,[y_j]_2)-e([w_j]_1,[1]_2)=e([h_j]_1,[b_j]_2), \qquad j=1,\ldots,m \tag{11}$$

for some constants $[k_j]_1,[b_j]_2$, and variables x_j,y_j,w_j,h_j (in fact in our case b_j is the same for all equations, namely $t(s)$).

GS proofs use dual mode commitments to commit to the witness, meaning that commitments are either used in perfectly hiding or perfectly binding mode. To simulate proofs, the trapdoor is the equivocation trapdoor of the commitment scheme in *both* $\mathbb{G}_1$ and $\mathbb{G}_2$. However, for this particular type of equation it is enough to use standard ElGamal encryption for $\mathbb{G}_2$ (see [15]), the reason being that the equation admits the trivial solution in $\mathbb{G}_1$. That is, it is enough for commitments in $\mathbb{G}_2$ to be computationally hiding, it is not necessary that there is a setup mode in which they are perfectly hiding. This allows us to save on the proof size ($(2,4)$ elements per equation).

The idea to save on the number of commitments is to reuse the randomness and encrypt all the variables $\vec{x}$, (resp. $\vec{y},\vec{z},\vec{w}$) with a single vector of commitments. This reduces the size of the commitments from $2m$ to $m+1$ for committing to each of the variable vectors. We define the commitment key in $\mathbb{G}_1^{m+1}$ as:

$$\mathbf{u}_1 \leftarrow \mathcal{U}_{m+1,1}, \vec{u}_2=\tau\vec{u}_1, \tau\leftarrow\mathbb{Z}_p.$$

and the commitment as:

$$\mathsf{Com}_U([\vec{x}]_1,\vec{r})=r_1[\vec{u}_1]_1+r_2[\vec{u}_2]_1+\left[\begin{pmatrix}\vec{x}\\0\end{pmatrix}\right]_1,$$

where $\vec{r}\in\mathbb{Z}_p^2$ and $\mathcal{U}_{m+1,1}$ is the uniform distribution of vectors of $\mathbb{Z}_p^{m+1}$.

On the other hand, in $\mathbb{G}_2$ the commitment key is defined as:

$$\mathbf{v}\leftarrow\mathcal{U}_{m+1,1},$$

and the commitment as

$$\mathsf{Com}_V([\vec{y}]_2,s)=s[\vec{v}]_2+\left[\begin{pmatrix}\vec{y}\\0\end{pmatrix}\right]_2.$$

The idea is that a commitment $[\vec{z}_{\vec{y}}]$ to a vector $[\vec{y}]$ can be divided into small parts $[\vec{z}_{y_i}]$, such that each part is a commitment to y_i. More precisely, components $(i,m+1)$ are a commitment to $\vec{y}_i$ with the commitment key corresponding to the components of $(i,m+1)$ of $\vec{u}_1,\vec{u}_2$ (for commitments in $\mathbb{G}_1$) and of $\vec{v}$ (for commitments in $\mathbb{G}_2$). That is, commitment keys are: $\vec{u}_1^i=\begin{pmatrix}u_{1,i}\\u_{1,m+1}\end{pmatrix}$ and $\vec{u}_2^i=\begin{pmatrix}u_{2,i}\\u_{2,m+1}\end{pmatrix}$, and $\mathsf{Com}_U([x_i]_1,\vec{r})=r_1[\vec{u}_1^i]_1+r_2[\vec{u}_2^i]_1+\left[\begin{pmatrix}x_i\\0\end{pmatrix}\right]_1$. Similarly, we can get a commitment to $[y_i]_2$ by getting the components $(i,m+1)$ of a commitment in $\mathbb{G}_2$ with respect to the key $\vec{v}^i=\begin{pmatrix}v_i\\v_{m+1}\end{pmatrix}$.

Therefore, we can now prove the equation i with different commitments keys, that is, it is as if we were using a different GS common reference string for each equation, namely, the keys $\vec{u}_1^i,\vec{u}_2^i,\vec{v}^i$.

The form of the jth verification equation is:

$$e(\left[\vec{z}^j_{\vec{x}}\right]_1,\left[\vec{z}^j_{\vec{y}}\right]_2)-e(\left[\vec{z}^j_{\vec{w}}\right]_1,[\vec{z}_1]_2)=e(\left[\vec{z}^j_{\vec{h}}\right]_1,\left[\vec{z}_{b_j}\right]_2)+\sum_{i=1}^{2}e([\vec{u}_i^j]_1,[\vec{\pi}_{i,j}]_2)+e([\vec{\theta}_j]_1,[\vec{v}^j]_2),$$

where $\vec{z}^j_{\vec{\alpha}}$ is the result of keeping the jth and the $(m+1)$th coordinate of the commitment to vector $\vec{\alpha}$ and $\vec{z}_1=\begin{pmatrix}1\\0\end{pmatrix}$, $\vec{z}_{\vec{b}_j}=\begin{pmatrix}b_j\\0\end{pmatrix}$, for $j=1,\ldots,m$.

Soundness obviously holds because the partial commitment keys define perfectly binding commitments, so the same argument as in GS proofs applies.

On the other hand, one can claim computational witness indistinguishability under the DDH Assumption in $\mathbb{G}_1$. Indeed, in the security proof of witness indistinguishability, after the setup of the common reference string, the adversary can choose two witnesses $W_0=([\vec{x}_0]_1,[\vec{y}_0]_2,[\vec{w}_0]_1,[\vec{h}_0]_1)$, and $W_1=([\vec{x}_1]_1,[\vec{y}_1]_2,[\vec{w}_1]_1,[\vec{h}_1]_1)$, and receive a proof for W_b, $b\leftarrow\{0,1\}$.

We define a sequence of games, $\{\mathsf{Game}_{i,0},\mathsf{Game}_{i,1},\mathsf{Game}_{i,2}\}_{i=1}^m$.

1. In $\mathsf{Game}_{i,0}$ the commitment key is changed to define a perfectly hiding commitment to the ith coordinate of $\mathbb{G}_1$, as $\vec{u}_2 = \tau\vec{u}_1 + \vec{e}_i$, where $\vec{e}_i$ is the ith vector in the canonical basis of $\mathbb{Z}_p^{m+1}$.
2. In $\mathsf{Game}_{i,1}$ the challenger samples a bit b but uses the witness $W^*_{i,b}$ to create the proof, where $W^*_{i,b} = ([\vec{x}_{i,b}]_1, [\vec{y}_b]_2, [\vec{w}_{i,b}]_1, [\vec{h}_{i,b}]_2)$ and $[\vec{x}_{i,b}]_1, [\vec{w}_{i,b}]_1, [\vec{h}_{i,b}]_1$ are the same as $[\vec{x}_b]_1, [\vec{w}_b]_1, [\vec{h}_b]_1$ replacing the first i coordinates with 0.
3. In $\mathsf{Game}_{i,2}$ the coordinate i is changed to define a perfectly binding commitment in $\mathbb{G}_1$, as $\vec{u}_2 = \tau\vec{u}_1$.

At the end of the sequence of Games, the part in $\mathbb{G}_1$ of the witness is changed to the all zero vector, and is independent of b.

To complete the proof, we observe that the equation is left simulatable. This means that, in particular, using the properties of GS proofs it is possible to compute a valid proof of the equation given a commitment to the part of the witness of $\mathbb{G}_2$, without knowing an opening. For this reason, in the last m games we can switch to the all-zero witness for the elements in $\mathbb{G}_2$ based on the IND-CPA security of ElGamal, namely based on the DDH Assumption in $\mathbb{G}_2$.

This argues Witness Indistinguishability, which is all we need for our Signature of Knowledge, although ZK follows immediately from the fact that the equations are trivially satisfiable.

This strategy adds to the CRS $2(m-1)$ elements in $\mathbb{G}_1$ and $m-1$ in $\mathbb{G}_2$, and, as explained, this reduces the cost of committing to the witness from $3 \cdot 2m$ elements in $\mathbb{G}_1$ and $2m$ in $\mathbb{G}_2$ to $3(m+1)$ in $\mathbb{G}_1$ and $m+1$ in $\mathbb{G}_2$.