Signatures of Knowledge for Boolean Circuits Under Standard Assumptions

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

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

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. Our new scheme can be used to construct the most efficient Signature-of-Knowledge based on standard assumptions that also can be composed universally with other cryptographic protocols/primitives.

Keywords: NIZK · Signatures · Bilinear groups

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 [2,16]. 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) [8,13] are among the most practically interesting NIZK proofs. They allow to generate succinct proofs for NP-complete languages (3 group elements for CircuitSat [13]) but they are constructed based on non-falsifiable assumptions (e.g. knowledge assumptions [6]). A well-known impossibility result of Gentry and Wichs [9]

A. Nitaj and A. Youssef (Eds.): AFRICACRYPT 2020, LNCS 12174, pp. 24–44, 2020.
https://doi.org/10.1007/978-3-030-51938-4_2

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 [22].

Groth-Sahai proofs [15] also allow to prove general languages[1] under standard assumptions non-succinctly, trading security for succinctness. On the other extreme, 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 [11] constructs a NIZK argument for boolean CircuitSat under falsifiable assumptions by combining techniques of QA-NIZK arguments and zk-SNARKs of size $O(n + d)$ 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 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) [12,24]. 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 [8,13], or SE ones [1,14]) are the best-known family of NIZK arguments that achieve nBB extraction which is achieved 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. 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 is BB SE [4,12].

[1] 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.

SE NIZK arguments have great potential to be deployed in practice [18,20], or construct other primitives such as Signature-of-Knowledge (SoK) [5]. In an 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 [14] constructed a SE zk-SNARK and a generic construction of an SoK from any SE NIZK argument, resulting in an 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 used in the UC framework.

Briefly speaking, 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 [11]. 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 $O(n + d)$. That the proposed construction also allows one to construct a (universally composable) SoK of roughly the same size. A comparison of our SoK with prior schemes can be found in Table 1.

Table 1. A comparison of our proposed SoK with prior schemes, where n_s 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.

Construction	Language	Signature Size	Assumption
BFG [3]	PE	$(n_{\mathsf{PPE}} n_X, n_{\mathsf{PPE}} n_Y) + \ell_K$	Falsifiable
GM [14]	SAP	3	Non-falsifiable
Sect. 3.2	QE	$(2n_s + 6d + 13, 2d + 11) + \ell_K$	Falsifiable

1.1 Our Contribution

Trivial Approach for Boolean CircuitSat. Let ϕ some boolean circuit, and let a_i, b_i, c_i be the left, right and output wires of gate i. An 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 [11] give a proof for Boolean CircuitSat of size $O(n + d)$ group elements. 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 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 [11] 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 [11] 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 [11] 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 $(\vec{c}_1, \ldots, \vec{c}_{j-1})$ (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 circuit sat, 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 [11] 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 [19], a generalization of [17,21] and for 3') they use techniques appeared in the context of zk-SNARKs (as e.g. [8]) 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. For 2'), they prove that soundness holds under some decisional assumption related to the matrix which defines the linear relations and for 3') they prove this 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 [11]. 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 [12]: 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).[2] 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

[2] Using OR proofs (the less efficient construction for PPE given in [23] or adding a bit as an auxiliary variable) plus the Boneh-Boyen signature for adaptive soundness.

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 [11] 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 [11] 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 [19] 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 [10] and for promise problems as in [11]. 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). Using the generic transformation of Groth and Maller [14], the result gives directly an SoK for boolean CircuitSat.

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. [7], which gives a compact proof that a set of perfectly binding commitments open to 0 or 1. Second, we observe that the advantage of our approach is that to get tight security we only need to construct a tight USS for promise problems in bilateral linear spaces, which we leave for future work. The result would be a signature of knowledge for circuits with a loss of d (the circuit depth) in the reduction (inherited from [11]).

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, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ 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 $\mathbb{G}_\gamma$.

2.1 Definitions

We recall the formal definition of QA-NIZK proofs. A QA-NIZK proof system [17] 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 $\mathcal{R}_{gk} = \{\mathcal{R}_\rho\}_{\rho \in \mathrm{sup}(\mathcal{D}_{gk})}$ 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, sample string $\rho \leftarrow \mathcal{D}_{gk}$, and then uses gk, ρ and generate $(\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e)$; finally output crs (that also contains parameter ρ) and store *simulation* trapdoor tr_s and *extraction* trapdoor tr_e as the trapdoors of CRS.

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

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

CRS Simulator, $(\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathsf{S}_1(gk, \rho)$**:** S_1 is a PPT algorithm that, given (gk, ρ), output $(\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e)$, where tr_s is *simulation* trapdoor and tr_e is the *extraction* trapdoor.

Prover Simulator, $\pi \leftarrow \mathsf{S}_2(\mathsf{crs}, \vec{x}, \mathsf{tr}_s)$**:** S_2 is a PPT algorithm that for valid statements, given $(\mathsf{crs}, \vec{x}, \mathsf{tr}_s)$, output a simulated argument π.

Extractor, $\vec{w} \leftarrow \mathcal{E}(gk, \mathsf{crs}, \vec{x}, \pi, \mathsf{tr}_e)$**:** $\mathcal{E}$ is a PPT algorithm that, given $(\mathsf{crs}, \vec{x}, \pi, \mathsf{tr}_e)$ 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*, *computationally quasi-adaptive sound* and *perfectly 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, and all $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$,*

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

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 $\mathcal{A}$,*

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{K}_0(1^\lambda),\ \rho \leftarrow \mathcal{D}_{gk}, \\ \mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho),\ (\vec{x}, \pi) \leftarrow \mathcal{A}(gk, \mathsf{crs}) \end{array} : \begin{array}{c} \mathsf{V}(\mathsf{crs}, \vec{x}, \pi) = 1 \wedge \\ (\vec{x}, \vec{w}) \notin \mathcal{R}_\rho \end{array}\right] \approx 0$$

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

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{K}_0(1^\lambda), \rho \leftarrow \mathcal{D}_{gk}, \\ \mathsf{crs} \leftarrow \mathsf{K}_1(gk, \rho) : \\ \mathcal{A}^{\mathsf{P}(\mathsf{crs}, \cdot, \cdot)}(gk, \mathsf{crs}) = 1 \end{array}\right] = \Pr\left[\begin{array}{l} gk \leftarrow \mathsf{K}_0(1^\lambda), \rho \leftarrow \mathcal{D}_{gk}, \\ (\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathsf{S}_1(gk, \rho) : \\ \mathcal{A}^{\mathsf{S}_2(\mathsf{crs}, \mathsf{tr}_s, \cdot, \cdot)}(gk, \mathsf{crs}) = 1 \end{array}\right]$$

where $\mathsf{P}(\mathsf{crs}, \cdot, \cdot)$ emulates the actual prover, and given $(\mathsf{crs}, \vec{x}, \vec{w})$ outputs a proof π if $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$, otherwise it outputs $\perp$; and $\mathsf{S}_2(\mathsf{crs}, \mathsf{tr}_s, \cdot, \cdot)$ is an oracle that given $(\mathsf{crs}, \mathsf{tr}_s, \vec{x}, \vec{w})$, it outputs a simulated proof $\mathsf{S}_2(\mathsf{crs}, \mathsf{tr}_s, \vec{x})$ if $(\vec{x}, \vec{w}) \in \mathcal{R}_\rho$ 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 [19].

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 $\mathcal{A}$,*

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

where $O(\vec{x})$ returns $(\vec{x}, \pi) \leftarrow \mathsf{S}_2(\mathsf{crs}, \mathsf{tr}, \tau, \vec{x})$ and adds τ to the set $\mathcal{Q}_{tags}$.

Now we define a variation of definition *BB simulation extractability* for QA-NIZKs that is used in the construction of new schemes. To the best of our knowledge, this is the first time that this definition is defined for QA-NIZKs.

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,*

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathsf{K}_0(1^\lambda),\ \rho \leftarrow \mathcal{D}_{gk}, \\ (\mathsf{crs}, \mathsf{tr}_s, \mathsf{tr}_e) \leftarrow \mathsf{S}_1(gk, \rho); \\ (\vec{x}^*, \pi^*) \leftarrow \mathcal{A}^{O(\cdot,\cdot)}(gk, \mathsf{crs}, \rho), \\ \vec{w} \leftarrow \mathcal{E}(gk, \mathsf{crs}, \vec{x}^*, \pi^*, \mathsf{tr}_e) \end{array} : \begin{array}{c} \mathsf{V}(\mathsf{crs}, \vec{x}^*, \pi^*) = 1 \\ \wedge\ (\vec{x}, \vec{w}) \notin \mathcal{R}_\rho \wedge\ (\vec{x}^*, \pi^*) \notin \mathcal{Q} \end{array}\right] \approx 0,$$

where $O(\vec{x})$ *returns* $(\vec{x}, \pi) \leftarrow \mathsf{S}_2(\mathsf{crs}, \vec{x}, \mathsf{tr}_s)$ *and adds* $(\vec{x}, \pi)$ *to the set of simulated proofs* $\mathcal{Q}$.

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

2.2 Boolean Circuits

As in González and Ràfols [11], we *slice* a boolean circuit in layers according to the level of each gate. Throughout the paper, $\phi : \{0,1\}^n \rightarrow \{0,1\}$ is a boolean circuit with m gates of fan-in two and d is the depth. To simplify the exposition of our result in limited space, we consider only NAND gates, but it is immediate to extend our result to include gates of ϕ of any type as was done in fact in [11].

The gates of ϕ are indexed 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.

In Lemma 1 we now express in equations what it means for a tuple $(\vec{a}, \vec{b}, \vec{c})$ to be a valid assignment to the left, right and output wires of ϕ respectively, where $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_d)$, $\vec{b} = (\vec{b}_1, \ldots, \vec{b}_d)$ 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 $\vec{y} \in \{\vec{a}, \vec{b}, \vec{c}\}$. A valid assignment should give $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$ the values of the left, right and output wires of the gate indexed by (i, j) and $c_{0,1}, \ldots, c_{0,n}$ some boolean values which represent a satisfying input.

Lemma 1 breaks down CircuitSat in different items which reflect the different building blocks used by [11] 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 [11] is to do two aggregated proof 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 (Sect. 2.3, 2.4) we sketch the aggregated proofs of the sets of equations described in *c*) and *d*).

Lemma 1. *Let* $\phi : \{0,1\}^n \rightarrow \{0,1\}$*, be a circuit with* m *NAND 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* $i = 1, \ldots, d$*:*

$$c_{i,j} = 1 - a_{i,j} b_{i,j}, \qquad j = 1, \ldots, n_i,$$

d) Correct "wiring" (linear constraints) at level i*:*

$$a_{i,j} = c_{k_L,\ell_L}, \qquad b_{i,j} = c_{k_R,\ell_R},$$

for some indexes $0 \leq k_L, k_R < i$*,* $\ell_L \in \{1, \ldots, n_{k_L}\}$ *and* $\ell_R \in \{1, \ldots, n_{k_R}\}$*. In other words, for all* i*, there exist some matrices* $\mathbf{F}_i, \mathbf{G}_i$ *such that* $\vec{a}_i = \mathbf{F}_i \vec{c}_{|i-1}$ *and* $\vec{b}_i = \mathbf{G}_i \vec{c}_{|i-1}$*, where* $\vec{c}_{|i-1}^{\top} = (\vec{c}_0^{\top}, \ldots, \vec{c}_{i-1}^{\top})$.
e) Correct output: $c_{d,1} = 1$.

2.3 Aggregated Proofs of Quadratic Equations

We now describe the construction proposed in González and Ràfols [11] to prove correct gate evaluation at level i, for $i = 1, \ldots, d-1$, i.e. a proof that $c_{i,j} = 1 - a_{i,j}b_{i,j}$, 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[3]:

$$[1]_T - e([L_{i,k}]_1, [R_{i,k}]_2) - e([O_{i,k}]_1, [1]_2) = e([H_{i,k}]_1, [t_k]_2), \quad (1)$$

$$e([O_{i,j}]_1, [1]_2) = e([1]_1, [O^*_{i,j}]_2). \quad (2)$$

where if $t(X) = \prod_{r\in\mathcal{R}}(X - r)$, $t_k = t(s_k)$ and $\lambda_i(X) = \prod_{j\in\mathcal{R}\setminus\{r_i\}} \dfrac{(X - r_j)}{(r_i - r_j)}$ 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, and $h_i(X) = (1 - (\sum a_j\lambda_j(X))(\sum b_j\lambda_j(X)) - \sum c_j\lambda_j(s_k))/t(X)$. 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 [11].

To the reader familiar with the literature, it is obvious that Eq. (1) uses SNARK techniques originally appeared in [8] (what we could call "polynomial aggregation") for proving many quadratic equations simultaneously. What is new in [11], 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 [11] use this proof as a building block in a larger proof, and for this they 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 $1 - \vec{a}_i \circ \vec{b}_i$ is a valid opening of $O_{i,k}$."

Formally, the authors define the languages

$$\mathcal{L}^{\mathsf{quad}}_{\mathsf{YES}} = \left\{ \begin{array}{c} (\vec{a}, \vec{b}, [\vec{L}]_1, [\vec{R}]_2, [\vec{O}]_1) : \vec{1} - \vec{a} \circ \vec{b} = \vec{c}, \\ \left[\vec{L}\right]_1 = [\mathbf{\Lambda}]_1\vec{a}, \left[\vec{R}\right]_2 = [\mathbf{\Lambda}]_2\vec{b}, \left[\vec{O}\right]_1 = [\mathbf{\Lambda}]_1\vec{c} \end{array} \right\}$$

[3] The second equation is added to have the element $O_{i,j}$ 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 $\mathbb{G}_2$.).

$$\mathcal{L}_{\mathsf{NO}}^{\mathsf{quad}} = \left\{ \begin{array}{c} (\vec{a}, \vec{b}, [\vec{L}]_1, [\vec{R}]_2, [\vec{O}]_1) : \vec{1} - \vec{a} \circ \vec{b} = \vec{c}, \\ \left[\vec{L}\right]_1 = [\mathbf{\Lambda}]_1 \vec{a}, \left[\vec{R}\right]_2 = [\mathbf{\Lambda}]_2 \vec{b}, \left[\vec{O}\right]_1 \neq [\mathbf{\Lambda}]_1 \vec{c} \end{array} \right\}.$$

The argument consists of giving some values $\vec{H}, \vec{O}^*$ chosen by the prover which satisfy Eq. (1) for $\vec{L}, \vec{R}, \vec{O}$. *Completeness* holds for $\mathcal{L}_{\mathsf{YES}}^{\mathsf{quad}}$ and *soundness* for values $\mathcal{L}_{\mathsf{NO}}^{\mathsf{quad}}$ under the $(\mathcal{R}, m)$-Rational Strong Diffie-Hellman assumption [11]. When (1) are proven with GS proofs, they 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.

2.4 Aggregated Proofs of Linear Equations

In this section we explain the technique used in González and Ràfols [11] to prove correct "wiring" at level i, for $i = 1, \ldots, d-1$, i.e. an aggregated proof for linear constraints. As we have seen in Lemma 1, we can express linear constraints at level i as:

$$\vec{a}_i = \mathbf{F}_i \vec{c}_{|i-1}, \quad \vec{b}_i = \mathbf{G}_i \vec{c}_{|i-1} \text{ for all } i = 1, \ldots, d. \tag{3}$$

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

$$\begin{pmatrix} \vec{O}_{|i-1} \\ \vec{L}_i \end{pmatrix} = \begin{pmatrix} \mathbf{C}_i \\ \tilde{\mathbf{F}}_i \end{pmatrix} (\vec{c}_{|i-1}), \quad \begin{pmatrix} \vec{O}_{|i-1} \\ \vec{R}_i \end{pmatrix} = \begin{pmatrix} \mathbf{C}_i \\ \tilde{\mathbf{G}}_i \end{pmatrix} (\vec{c}_{|i-1}) \tag{4}$$

where $\mathbf{C}_i = \begin{pmatrix} \mathbf{I} & \vec{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{n_1} & \ldots & \vec{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{\Lambda}_{n_{i-1}} \end{pmatrix}$, $\tilde{\mathbf{F}}_i = \mathbf{\Lambda}_{n_i} \mathbf{F}_i$, $\tilde{\mathbf{G}}_i = \mathbf{\Lambda}_{n_i} \mathbf{G}_i$ and $\mathbf{\Lambda}_{n_i}$ is the matrix of the Lagrangian Pedersen commitment key defined in 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 $\mathbf{C}_i$, $\tilde{\mathbf{F}}_i = \mathbf{\Lambda}_{n_i} \mathbf{F}_i$, $\tilde{\mathbf{G}}_i = \mathbf{\Lambda}_{n_i} \mathbf{G}_i$ adding columns and rows to accommodate for the GS commitment keys in the relevant groups (see full details in [11]). We denote these matrices $\mathbf{M}_i^L, \mathbf{N}_i^L$ for the left constraints and $\mathbf{M}_i^R, \mathbf{N}_i^R$ for the right constraints.

González and Ràfols prove that the QA-NIZK argument of Kiltz and Wee [19] (with standard soundness) for membership in linear spaces for non-witness samplable distributions is an argument for the following promise problem:

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

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

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 $\vec{c}_{|i-1}$.

The authors prove completeness of the argument for statements in $\mathcal{L}_{\mathrm{YES}}^{\mathsf{Lin}}$ and soundness for $\mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}}$ 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[4].

We note that for simplicity, we have explained the result of [11] 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 [10] is used in [11] to gain efficiency (the proof requires then only 2 elements in $\mathbb{G}_1$ and $\mathbb{G}_2$ instead of $O(d)$ elements).

3 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

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

In the following Π_Q denotes the argument for Quadratic Equations described in Sect. 2.3, Π_L the USS membership argument for linear spaces presented in Sect. 4 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 $(ck_1, ck_2, \mathsf{crs}_{\mathsf{GS}})$.

[4] 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 $n, n_1, \ldots, n_s$, and the assumption can be reduced to a decisional assumption which only depends on $\mathbf{\Lambda}$ and the GS commitment key.

$\underline{\mathrm{K}_1\,(gk,\phi)}$: Given $(ck_1, ck_2, \mathsf{crs}_{\mathsf{GS}}) \leftarrow \mathcal{D}_{gk,W}$ and $\phi : \{0,1\}^n \rightarrow \{0,1\}$ 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 Sect. 2.4. 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.

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

$\underline{\mathrm{P}\left(\mathsf{crs}, \vec{x}_s, \vec{r}, \vec{a}, \vec{b}, \vec{c}, \vec{x}_p\right)}$: 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 $[\vec{O}_i]_{1,2}, [\vec{L}_i]_1, [\vec{R}_i]_2$, give a GS proof $\Pi_{Q,i}$ that they satisfy Eq. (1) and let $[\vec{z}_{O,i,k}]_1$, $[\vec{z}^*_{O,i,k}]_2$, $[\vec{z}_{L,i,k}]_1$, $[\vec{z}_{R,i,k}]_2$ the correspondent GS commitments to $\vec{O}, \vec{L}, \vec{R}$, for $k = 1, 2$. Compute proofs $\Pi_{L,i}$ of correct wiring and $\Pi_{L,0}$ that the opening of $[\vec{z}]_1$ is correctly assigned to $[\vec{z}_{O,0}]_1$. Outputs $\pi = \left([\vec{z}]_1, \mathsf{Input}, [\vec{z}_O]_1, [\vec{z}_L]_1, [\vec{z}^*_O]_2, [\vec{z}_R]_2, \vec{\Pi}_L, \Pi_{L,0}, \vec{\Pi}_Q\right)$.

$\underline{\mathrm{V}\,(\mathsf{crs}, \vec{x}_p, \pi)}$: Verify all the proofs in π with the corresponding verification algorithms $\mathrm{V}_{\mathsf{Input}}$, V_{Π_L} and check Eq. (1).

$\underline{\mathrm{S}\,(\mathsf{crs}, \vec{x}_p, \mathsf{tr})}$: 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 to $\hat{\vec{a}}_d = 0$, $\hat{\vec{b}}_d = 1$, $\hat{\vec{c}}_d = 1$. 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 $[\vec{z}_{O,i,k}]_1, [\vec{z}_{L,i,k}]_1, [\vec{z}^*_{O,i,k}]_2, [\vec{z}_{R,i,k}]_2$ 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 and Zero-Knowledge are directly from the completeness and zero-knowledge of the respective subarguments.

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

Theorem 1. *If $\mathcal{A}$ is an adaptive adversary against the Unbounded Simulation BB Extractability Soundness of the Boolean CircuitSat argument described in Sect. 3 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. □

3.1 Concrete USES QA-NIZK for Boolean CircuitSat

For the scheme described above, one can take as Input, and Π_Q the same building blocks as [11], namely the bitstring argument of Daza et al. [7] and the argument described in Sect. 2.3. The USS for promise problems given in Sect. 4.

To simplify the exposition we have ommitted 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). 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 (1), (2) of our instantiation are exactly the same as in González and Ràfols [11]. The cost of committing to the input plus proving it is boolean with the argument of [7] is $(2n_s + 4)|\mathbb{G}_1| + 6|\mathbb{G}_2|$. We take the same quadratic constraints proof from [11] with Zero-Knowledge that is $12d|\mathbb{G}_1| + 4d|\mathbb{G}_2|$ for the commitments and $8d|\mathbb{G}_1| + 4d|\mathbb{G}_2|$ for the GS proofs. This is the same cost as in [11], but in the full version we will give different tradeoffs to reduce the proof size at the cost of increasing the common reference string. In any case, 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 [11] is only $3|\mathbb{G}_1|$.

3.2 Universally Composable Signature of Knowledge

Following the same approach as Groth and Maller [14], the SE NIZK argument with BB extractability together with a universal one-way hash function allows to construct a UC secure SoK for boolean CircuitSat based on falsifiable assumptions in bilinear groups in a straightforward way. The full details of this construction will appear in the full version of the paper.

4 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 Sect. 2.4.

We give the argument for membership in linear spaces in one group in detail in Sect. 4.1 and we present the scheme for the bilateral version in Sect. 4.2.

4.1 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

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

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

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 [11], 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 $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{YES}} \cup \mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$. 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 $\mathcal{L}^{\mathsf{Lin}}_{\mathrm{NO}}$ in some contexts, as in Sect. 3 for CircuitSat, but we define for general statements.

Scheme Definition. The argument is presented in Fig. 1 and is just the USS QA-NIZK argument of [19] written in two blocks, which adds a pseudorandom MAC to the basic (not simulation sound, just sound) QA-NIZK argument of membership linear spaces for non-witness samplable distributions also given in [19]. If in the basic arguments 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

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

Our contribution is not in the scheme but in the security analysis. Our proof follows [11] that proved that the basic argument in [19] is complete and sound for the same promise problem under some MDDH and KerMDH assumptions related to the matrix $\mathcal{M}$. Our contribution is to modify their analysis to adapt it to simulation soundness for the scheme of Fig. 1.

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 [19]: $\left\{ (\vec{w}, [\vec{x}, \vec{y}]_1) : [\vec{x}^\top, \vec{y}^\top]_1^\top = [\mathbf{M}, \mathbf{N}]_1^\top \vec{w} \right\}$, so perfect completeness and perfect zero-knowledge are immediate.

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

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

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

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

Fig. 1. The $\mathsf{Lin}_{\mathcal{D}_k}$ argument for proving membership in linear spaces in blocks $[\vec{x}, \vec{y}]_1 \in \mathrm{Im}[\mathbf{M}, \mathbf{N}]_1$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$.

Unbounded Simulation Soundness. 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 Fig. 1, the probability that the adversary $\mathcal{A}$ comes up with a proof $\vec{\pi}^*$ for a statement $(\vec{w}^*, [\vec{x}^*, \vec{y}^*]_1) \in \mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}}$ different of the queried ones and different tag τ^*, such that $\mathrm{V}(\mathsf{crs}, \tau^*, [\vec{x}^*, \vec{y}^*]_1, \vec{\pi}^*) = 1$ is negligible.

We use Definition 4 and our proof is analogous to USS proof of [19], where the authors argue that partial information about matrix $\mathbf{K}$ is hidden across all the simulated proofs which fits perfectly with the soundness argumentation in [11], where the authors prove the block $\mathbf{K}_{2,2}$ is hidden from the adversary. We need an extra change of games because our matrices admit more rows than columns and we have to assure the projection of our matrices does not reveal information of $\mathbf{K}_2$.

For the following theorem, we use the Computational Core Lemma of Kiltz and Wee in Sect. 4.1. of [19], 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 2. *The $\mathsf{Lin}_{\mathcal{D}_k}$ scheme in Fig. 1 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*

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

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

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

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 $\vec{\pi} - [\vec{x}^\top, \vec{y}^\top]_1 \mathbf{K} - \vec{\pi}_2(\mathbf{\Lambda}_0 + \tau \mathbf{\Lambda}_1) \in \mathbb{G}_1^{k+1}$ 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

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

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 [19], 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 $([(\vec{x}_i^\top, \vec{y}_i^\top)\mathbf{K} + \vec{r}^\top(\mathbf{P}_0 + \tau_i \mathbf{P}_1)]_1, [\vec{r}^\top \mathbf{\Lambda}_0^\top]_1)$.

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 $(\vec{w}^*, [\vec{x}^*, \vec{y}^*]_1) \in \mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}}$ by computing $[\mathbf{M}]_1 \vec{w}^*$ and $[\mathbf{N}]_1 \vec{w}^*$. The difference between respective advantages is bounded using the core lemma of [19] 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 of 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 previous game with this $[\vec{z}]_1$ in $\mathbf{B}$ and runs $\mathcal{A}$ as in Game_3. Finally, when 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 $\vec{b}$ for 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 an statement $(\vec{w}^*, \vec{x}^*, \vec{y}^*) \in \mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}}$, it can also construct a valid proof $\pi = (\vec{\pi}_1^* - \vec{w}^*\mathbf{B}, \vec{\pi}_2^*)$ for the statement $(\vec{w}^*, \vec{0}, \vec{y} - \vec{y}^*) \in \mathcal{L}_{\mathrm{NO}}^{\mathsf{Lin}}$ where $\vec{y} = \mathbf{N}\vec{w}^*$. 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 that equation $(*)$ holds is less that $1/p$. □

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

$\underline{\mathrm{P}\,(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \boldsymbol{w})}$:
Pick $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left(\boldsymbol{w}^{\top}[\mathbf{B}]_1 + \vec{r}^{\top}[\mathbf{P}_0 + \tau\mathbf{P}_1]_1, [\vec{r}^{\top}\mathbf{\Lambda}^{\top}]_1\right)$,
$\boldsymbol{\theta} = \boldsymbol{w}^{\top}[\mathbf{D}]_2$.

$\underline{\mathrm{V}\,(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \vec{\pi}, \boldsymbol{\theta})}$:
Check if: $e\,(\vec{\pi}_1, [\mathbf{A}]_2) - e\,([\mathbf{A}]_1, \boldsymbol{\theta})$
$-e\left(\left[\boldsymbol{x}_1^{\top}\right]_1, [\mathbf{C}_1]_2\right) - e\left(\left[\boldsymbol{x}_2^{\top}\right]_1, [\mathbf{C}_2]_2\right)$
$+e\left([\mathbf{C}_3]_1, \left[\boldsymbol{y}^{\top}\right]_2\right) = e\,(\vec{\pi}_2, [\mathbf{Q}_0 + \tau\mathbf{Q}_1]_2)$

$\underline{\mathrm{S}\,(\mathsf{crs}, \tau, [\boldsymbol{x}_1]_1, [\boldsymbol{x}_2]_1, [\boldsymbol{y}]_2, \mathsf{tr})}$:
Sample $\vec{r} \leftarrow \mathbb{Z}_p^k$ and return
$\vec{\pi} = \left([\vec{x}_1, \vec{x}_2]_1(\mathbf{K}_1^{\top}, \mathbf{K}_2^{\top}) + \vec{r}^{\top}(\mathbf{P}_0 + \tau\mathbf{P}_1), [\vec{r}^{\top}\mathbf{\Lambda}^{\top}]_1\right)$,

Fig. 2. The $\mathsf{BLin}_{\mathcal{D}_k}$ argument for proving membership in linear spaces in blocks $([\vec{x}_1, \vec{x}_2]_1, [\vec{y}]_2) \in \mathrm{Im}\,([\mathbf{M}, \mathbf{N}]_1, [\mathbf{P}]_2,)$ where $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}, \mathbf{P} \in \mathbb{Z}_p^{\ell_3 \times n}$.

4.2 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 Sect. 4.1 for the promise problem defined by languages

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

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

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 Fig. 2. QA-NIZK arguments of membership in linear spaces were extended to the bilateral case in [10] for both samplable and non-witness samplable distributions. In [11], the authors proved that the argument for non-witness samplable distributions of [10] is also sound and complete for this promise problem. Adding the pseudorandom MAC given in [19] we get USS. The proof is essentially the same as in Sect. 4.1, but now the linear spaces are split in two groups $\mathbb{G}_1$ and $\mathbb{G}_2$. The core lemma would be the analogous one and the reduction of the proof of USS is bounded by $\mathsf{SKerMDH}$ and $\mathcal{D}_k$-$\mathsf{MDDH}_{\mathbb{G}_1}$ Assumptions.

Acknowledgement. Karim Baghery was supported by CyberSecurity Research Flanders with reference number VR20192203.

References

1. Baghery, K.: Subversion-resistant simulation (knowledge) sound NIZKs. In: Albrecht, M. (ed.) IMACC 2019. LNCS, vol. 11929, pp. 42–63. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35199-1_3
2. Ben-Sasson, E., et al.: Zerocash: decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy, pp. 459–474. IEEE Computer Society Press, May 2014
3. Bernhard, D., Fuchsbauer, G., Ghadafi, E.: Efficient signatures of knowledge and DAA in the standard model. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 518–533. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38980-1_33
4. Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. In: 34th ACM STOC, pp. 494–503. ACM Press, May 2002
5. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 78–96. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_5
6. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_36
7. Daza, V., González, A., Pindado, Z., Ràfols, C., Silva, J.: Shorter quadratic QA-NIZK proofs. In: Lin, D., Sako, K. (eds.) PKC 2019, Part I. LNCS, vol. 11442, pp. 314–343. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17253-4_11
8. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_37

9. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC, pp. 99–108. ACM Press, June 2011
10. González, A., Hevia, A., Ràfols, C.: QA-NIZK arguments in asymmetric groups: new tools and new constructions. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 605–629. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_25
11. González, A., Ràfols, C.: Shorter pairing-based arguments under standard assumptions. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11923, pp. 728–757. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_25
12. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006). https://doi.org/10.1007/11935230_29
13. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_11
14. Groth, J., Maller, M.: Snarky signatures: minimal signatures of knowledge from simulation-extractable SNARKs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 581–612. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_20
15. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_24
16. Hofheinz, D., Jia, D., Pan, J.: Identity-based encryption tightly secure under chosen-ciphertext attacks. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 190–220. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_7
17. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_1
18. Kerber, T., Kiayias, A., Kohlweiss, M., Zikas, V.: Ouroboros crypsinous: privacy-preserving proof-of-stake. In: 2019 IEEE Symposium on Security and Privacy, pp. 157–174. IEEE Computer Society Press, May 2019
19. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_4
20. Kosba, A.E., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: 2016 IEEE Symposium on Security and Privacy, pp. 839–858. IEEE Computer Society Press, May 2016
21. Libert, B., Peters, T., Joye, M., Yung, M.: Linearly homomorphic structure-preserving signatures and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 289–307. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_17
22. Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_6

23. Ràfols, C.: Stretching groth-sahai: NIZK proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 247–276. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_10
24. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS, pp. 543–553. IEEE Computer Society Press, October 1999