Subversion-Resistant Simulation (Knowledge) Sound NIZKs

Karim Baghery

University of Tartu, Estonia

Abstract. In ASIACRYPT 2016, Bellare, Fuchsbauer, and Scafuro studied the security of non-interactive zero-knowledge (NIZK) arguments in the face of parameter subversion. They showed that achieving subversion soundness (soundness without trusting to the third party) and standard zero-knowledge is impossible at the same time. On the positive side, in the best case, they showed that one can achieve subversion zero-knowledge (zero-knowledge without trusting to the third party) and soundness at the same time. In this paper, we show that one can amplify their best positive result and construct NIZK arguments that can achieve subversion zero-knowledge and *simulation* (knowledge) soundness at the same time. Simulation (knowledge) soundness is a stronger notion in comparison with (knowledge) soundness, as it also guarantees non-malleability of proofs. Such a stronger security guarantee is a must in practical systems. To prove the result, we show that given a NIZK argument that achieves Sub-ZK and (knowledge) soundness, one can use an OR-based construction to define a new language and build a NIZK argument that will guarantee Sub-ZK and *simulation* (knowledge) soundness at the same time. We instantiate the construction with the state-of-the-art zk-SNARK proposed by Groth [Eurocrypt 2016] and obtain an efficient SNARK that guarantees Sub-ZK and simulation knowledge soundness.

Keywords: NIZK, subversion zero-knowledge, zk-SNARK, simulation extractability, CRS model

1 Introduction

Non-Interactive Zero-Knowledge (NIZK) proofs are one of the central design tools in cryptographically secure systems, allowing one to verify the veracity of statements without leaking extra information. Technically speaking, a NIZK allows a prover to prove that for a public statement x she knows a witness w which holds in a relation $\mathbf{R}$, $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$, without leaking any information about her witness w. In the Common Reference String (CRS) model [BFM88], a NIZK requires a setup phase which is supposed to be done by a trusted third party. Under a trusted setup phase, usually a NIZK is required to guarantee three essential properties known as *completeness*, *zero-knowledge* and *soundness*. The property *completeness* guarantees that an honest prover always convinces an honest verifier. The *soundness* ensures that a malicious prover cannot

convince the honest verifier except with negligible probability. Zero-knowledge property assures that the proof generated by prover does not leak any information about the witness w. Moreover, following some stronger requirements in practical systems, there have been various constructions for NIZKs that can achieve more stronger notions than bare soundness. The notions *knowledge soundness* and *simulation knowledge soundness* (a.k.a. simulation extractability) are two flavours of soundness that guarantee more security than what soundness achieves. Knowledge-soundness guarantees that if an adversarial prover manages to come out with an acceptable proof, there exists an efficient extractor which given some secret information can efficiently extract the witness from the proof. Zero-knowledge Succinct Non-interactive Arguments of Knowledge (zk-SNARKs) [Gro10,Lip12,PHGR13,BCTV13,Gro16,GM17,Lip19] are the most known and practically-interested NIZK arguments that guarantee knowledge soundness. By the date, the most efficient zk-SNARK is proposed by Groth [Gro16] in Eurocrypt 2016, which is constructed for Quadratic Arithmetic Programs (QAPs) and works in a bilinear group. As a stronger notion, simulation knowledge soundness guarantees that knowledge-soundness is satisfied even if the adversary already has seen an arbitrary number of simulated proofs for any statements. Roughly speaking, simulation extractability guarantees that the proofs are also non-malleable and consequently secure against man-in-the-middle attacks. In Crypto 2017, Groth and Maller [GM17] proposed the first zk-SNARK in the CRS model for Square Arithmetic Programs (SAPs) that achieves (non-black-box) simulation extractability. Recently, Atapoor and Baghery [AB19] used a folklore OR technique [BG90] with a particular instantiation from C∅C∅ framework [KZM+15] [1] and presented a variation of the state-of-the-art zk-SNARK proposed by Groth [Gro16] and showed that it can achieve (non-black-box) simulation extractability and outperforms Groth and Maller's zk-SNARK [GM17] considerably [AB19]. Concurrently, Lipmaa [Lip19] proposed several (non-black-box) simulation-extractable zk-SNARKs in the CRS model for different languages including QAPs, SAPs, Quadratic Span Programs (QSPs) and Square Span Programs (SSPs). By deploying zk-SNARKs in some bigger cryptographic systems that should guarantee universal composability (UC), some studies construct zk-SNARKs with black-box simulation extractability [KZM+15,Bag19a] which is a necessary requirement for using zk-SNARKs in the UC-secure protocols.

Importance of setup phase in the CRS model. By deploying cryptographic primitives in various applications, recently there have been various attacks or flaw reports on the setup phase of cryptographic systems that rely on public parameters supposed to be generated honestly. In some cases, attacks are caused by maliciously (or incorrectly) generated public parameters or modifying cryptographic protocol specifications to embed backdoors, with intent to violate the security of the main system [BBG+13,PLS13,Gre14,Gab19,LPT19,Hae19]. Es-

[1] A framework with practically optimized primitives which given a sound NIZK lifts it to a universally composable or more precisely a black-box simulation extractable NIZK argument [KZM+15].

pecially, after the Snowden revelations, there have been various endeavours in constructing cryptographic primitives and protocols secure against active subversion. The primitives constructed in this setting, guarantee their pre-defined security with trusted parameters, even in the case that the public parameters are subverted. Initiated by Bellare et al. [BPR14] for symmetric encryption schemes, there have been various studies about subversion resistant of various cryptographic primitives, including signature schemes [AMV15], non-interactive zero-knowledge proofs [BFS16], public-key encryption schemes [ABK18] and commitment schemes [Bag19b].

Subversion Security in NIZK arguments. In the context of NIZKs, in [BFS16], Bellare, Fuchsbauer and Scafuro tackled the discussed problem by studying how much security one can still achieve when the CRS generator cannot be trusted. They first defined three new notions called subversion witness indistinguishability (Sub-WI), subversion zero-knowledge (Sub-ZK) and subversion soundness (Sub-SND) as a variant of the standard notions witness indistinguishability (WI), zero-knowledge (ZK) and soundness (SND) in NIZK arguments. The main difference of proposed notions with the standard ones is that in the new ones the setup phase is compromised and the parameters can be generated maliciously. For instance, the notion Sub-ZK guarantees that even if an adversary generates the CRS elements, still the NIZK proof does not leak any information about the witness of the prover. Intuitively, Sub-ZK implies that the ZK is guaranteed even if an adversary generates the CRS. In the rest, Bellare et al. showed that the definitions of Sub-SND and ZK are not compatible; as the former requires that a prover should not be able to generate a fake proof even if he generates the CRS, but the latter implies that there exists a simulation algorithm that given trapdoors of CRS can generate a (fake) simulated proof indistinguishable from the real ones. This resulted in a negative result that we cannot construct a NIZK argument which will guarantee ZK and Sub-SND simultaneously.

The above negative result opened two possible directions for positive results on subversion-resistant proof systems. One direction was achieving Sub-ZK and a version of soundness (i.e. one of notions soundness, knowledge soundness or simulation knowledge soundness) and the second direction was achieving Sub-WI (the best notion weaker than ZK) and a notion of Sub-SND (one of notions subversion soundness, subversion knowledge soundness or subversion simulation knowledge soundness). Along the first direction, Bellare et al. showed that one can construct NIZK arguments that achieve Sub-ZK and SND at the same time [BFS16]. Their main idea to achieve Sub-ZK is to use a knowledge assumption in the proof of zero-knowledge to extract the trapdoors of CRS from untrusted CRS and then use them to simulate the argument. After this positive result, Abdolmaleki et al. [ABLZ17] showed that the state-of-the-art zk-SNARK [Gro16] can achieve Sub-ZK and *knowledge* soundness with minimal changes in the CRS and executing an efficient public algorithm to check the well-formedness of CRS elements. In a concurrent work, Fuchsbauer [Fuc18] showed that most of pairing-based zk-SNARKs including Groth's scheme can achieve Sub-ZK and knowledge sound-

ness simultaneously. In the same direction, Abdolmaleki et al. [ALSZ18] showed that one can achieve Sub-ZK and SND in the Quasi-Adaptive NIZK arguments which are a particular type of NIZK proof systems.

In the second direction of possible positive results, Bellare et al. [BFS16] showed that Zap schemes proposed by Groth, Ostrovsky and Sahai [GOS06] achieves Sub-WI and Sub-SND at the same time; as such proof systems do not require particular CRS (consequently they do not require a trusted setup phase) but provides weaker security guarantee than ZK. Recently, Fuchsbauer and Orru [FO18] showed that one can achieve even more in this direction, by presenting a Sub-WI and *knowledge* sound Zap scheme.

Problem statement. By considering the summarized subversion-resistant constructions, one may ask if we can construct NIZK arguments with stronger security guarantees in the face of subverted CRS. For instance, can we construct NIZK arguments that can guarantee Sub-ZK and *simulation* knowledge soundness at the same time, such that the prover will not trust a third party to achieve ZK and the verifier will obtain more security guarantee (more precisely non-malleable proofs) than knowledge soundness. In comparison with non-subversion-resistant simulation-extractable zk-SNARKs, our target constructions can eliminate the trust in CRS generators from the prover side.

Our Contribution. We answer the question discussed above positively by constructing NIZK arguments that can achieve Sub-ZK and *simulation* knowledge soundness at the same time. Such construction guarantees that the prover does not need to trust a third party to achieve ZK, on the other side, extra from knowledge soundness verifier will get sure that the proofs are non-malleable. To construct such NIZK arguments, inspired by folklore OR technique [BG90], we use a part of the C∅C∅ framework [BG90,DDO+01,KZM+15] that recently is also used by Atapoor and Baghery [AB19] to achieve simulation (knowledge) soundness in Gorth's zk-SNARK [Gro16]. We show that using such construction, given NIZK arguments that guarantee Sub-ZK and (knowledge) soundness, we can construct Sub-ZK and simulation (knowledge) sound NIZK arguments.

As an instantiation, we show that a recent variation of Groth's zk-SNARK proposed by Atapoor and Baghery [AB19] can achieve Sub-ZK with the minimal extra computational cost. The cost is that similar to NIZK arguments that achieve Sub-ZK and (knowledge) soundness [ABLZ17,Fuc18], the prover only needs to execute an efficient algorithm (CRS verification) to check the well-formedness of CRS elements before using them. If CRS verification passed, the protocol ensures that the generated proof does not leak any information about the witnesses even if CRS generators collude with the verifier. This allows a prover to achieve ZK without trusting to the CRS generators.

Tab. 1 summarizes current subversion-resistant constructions and compares with an instantiation of our result. The first row shows the negative result that achieving Sub-SND and ZK at the same time is impossible as their definitions are incompatible [BFS16]. Next rows indicate the notions achieved in various presented non-interactive proof systems [ABLZ17,Fuc18,FO18,ALSZ18].

Table 1: A comparison of our results with current subversion-resistant non-interactive proof systems and their security guarantees. WI: Witness Indistinguishable, ZK: Zero-Knowledge, SND: Soundness, KS: Knowledge Soundness, SS: Simulation Soundness, SKS: Simulation Knowledge Soundness, Sub-WI: Subversion Witness Indistinguishable, Sub-ZK: Subversion Zero-Knowledge, Sub-SND: Sub-Soundness, Sub-KS: Subversion Knowledge Soundness.

	Standard						Subversion Resistant			
Achievable? ‖ Result in	WI	ZK	SND	KS	SS	SKS	Sub-WI	Sub-ZK	Sub-SND	Sub-KS
NO ‖ [BFS16]		✓							✓	
YES ‖ [BFS16]	✓		✓				✓		✓	
YES ‖ [FO18]	✓		✓	✓			✓		✓	✓
YES ‖ [BFS16]	✓	✓	✓				✓			
YES ‖ [BFS16]	✓	✓	✓				✓	✓		
YES ‖ [ALSZ18]	✓	✓	✓				✓	✓		
YES ‖ [ABLZ17,Fuc18]	✓	✓	✓	✓			✓	✓		
YES ‖ This Work	✓	✓	✓	✓	✓	✓	✓	✓		

Our technique. In the proposed construction, we use a part of the C∅C∅ framework and show that this part can be used to construct non-interactive arguments that will satisfy Sub-ZK and (non-black-box) simulation (knowledge) soundness. We define a new language $\mathbf{L}'$ based on an OR construction (that is added to achieve non-malleability) and the original language $\mathbf{L}$ in the input non-interactive argument that guarantees Sub-ZK. Then we use the basic property of an OR construction, i.e. that OR proofs can be simulated using the trapdoors of one branch. We show that if the input NIZK argument achieves Sub-ZK, then the lifted non-interactive argument also guarantees Sub-ZK. As in the notion of Sub-ZK the prover does not trust the CRS generators and consequently, the simulator does not trust the simulation trapdoors, so in proof of Sub-ZK, different form C∅C∅ framework, we use a technique in subversion-resistant schemes and simulate the protocol. In this road, a key point is that the proofs for an OR-based language can be simulated by trapdoors of either the first or second branch. Next, as an instantiation, we use the above result and show that since the state-of-the-art zk-SNARK proposed by Groth [Gro16] achieves Sub-ZK after some verifications on CRS elements [ABLZ17,Fuc18], its recent variation proposed in [AB19] (which uses the same OR construction) can achieve Sub-ZK after some efficient verifications on CRS elements.

The rest of the paper is organized as follows; Sec. 2 introduces notations and necessary preliminaries for the paper. The proposed transformation for constructing subversion-resistant simulation (knowledge) sound NIZK arguments is described in Sec. 3. In Sec. 4, we show that recent variation of Groth's zk-SNARK [Gro16] proposed by Atapoor and Baghery [AB19] can achieve Sub-ZK and simulation knowledge soundness with the minimal extra computational cost. Finally, we conclude the paper in Sec. 5.

2 Preliminaries

Let PPT denote probabilistic polynomial-time. 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 $\mathrm{im}(\mathcal{A})$ be the image of $\mathcal{A}$, i.e. the set of valid outputs of $\mathcal{A}$, let $\mathsf{RND}(\mathcal{A})$ denote the random tape of $\mathcal{A}$, and let $r \leftarrow_r \mathsf{RND}(\mathcal{A})$ denote sampling of a randomizer r of sufficient length for $\mathcal{A}$'s needs. By $y \leftarrow \mathcal{A}(x;r)$ we denote the fact that $\mathcal{A}$, given an input x and a randomizer r, outputs y. For algorithms $\mathcal{A}$ and $\mathsf{ext}_{\mathcal{A}}$, we write $(y \,\|\, y') \leftarrow (\mathcal{A} \,\|\, \mathsf{ext}_{\mathcal{A}})(x;r)$ as a shorthand for "$y \leftarrow \mathcal{A}(x;r)$, $y' \leftarrow \mathsf{ext}_{\mathcal{A}}(x;r)$". We denote by $\mathsf{negl}(\lambda)$ an arbitrary negligible function. For distributions A and B, $A \approx_c B$ means that they are computationally indistinguishable. In pairing-based groups, we use additive notation together with the bracket notation, i.e., in group $\mathbb{G}_\mu$, $[a]_\mu = a\,[1]_\mu$, where $[1]_\mu$ is a fixed generator of $\mathbb{G}_\mu$. A *bilinear group generator* $\mathsf{BGgen}(1^\lambda)$ returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where p (a large prime) is the order of cyclic abelian groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$. Finally, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an efficient non-degenerate bilinear pairing, s.t. $\hat{e}([a]_1, [b]_2) = [ab]_T$. Denote $[a]_1 \bullet [b]_2 = \hat{e}([a]_1, [b]_2)$.

Next, we review QAPs that define NP-complete language specified by a quadratic equation over polynomials and have a reduction from the language Circuit-SAT [GGPR13,Gro16].

Quadratic Arithmetic Programs. QAP was introduced by Gennaro *et al.* [GGPR13] as a language where for an input x and witness w, $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$ can be verified by using a parallel quadratic check. Consequently, any efficient simulation-extractable zk-SNARK for QAP results in an efficient simulation-extractable zk-SNARK for Circuit-SAT.

An QAP instance $\mathcal{Q}_p$ is specified by the so defined $(\mathbb{Z}_p, m_0, \ell, \{u_j, v_j, w_j\}_{j=0}^m)$, where m_0 is the length of the statement (e.g. public inputs and outputs in an arithmetic circuit), ℓ is a target polynomial (defined based on the number of constraints, e.g. number of multiplication gates in an arithmetic circuit), and u_j, v_j, w_j are three sets of polynomials that encodes the wires in the target arithmetic circuit. More discussions about encoding an arithmetic circuit to a QAP instance can be found in [GGPR13]. A QAP instance $\mathcal{Q}_p$ defines the following relation, where we assume that $A_0 = 1$:

$$\mathbf{R} = \left\{ \begin{array}{l} (\mathsf{x}, \mathsf{w}) \colon \mathsf{x} = (A_1, \ldots, A_{m_0})^\top \wedge \mathsf{w} = (A_{m_0+1}, \ldots, A_m)^\top \wedge \\ \Big(\sum_{j=0}^m A_j u_j(X)\Big)\Big(\sum_{j=0}^m A_j v_j(X)\Big) \equiv \sum_{j=0}^m A_j w_j(X) \pmod{\ell(X)} \end{array} \right\}.$$

Alternatively, $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$ if there exists a (degree $\leq n-2$) polynomial $h(X)$, s.t.

$$\left(\sum_{j=0}^m A_j u_j(X)\right)\left(\sum_{j=0}^m A_j v_j(X)\right) - \sum_{j=0}^m A_j w_j(X) = h(X)\ell(X)\ ,$$

where $\ell(X) = \prod_{i=1}^{\mathsf{n}}(X - \omega^{i-1})$ is a polynomial related to Lagrange interpolation, and ω is an n-th primitive root of unity modulo p.

Roughly speaking, the goal of the prover of a zk-SNARK for QAP [GGPR13] is to prove that for public $(A_1, \ldots, A_{m_0})$ and $A_0 = 1$, she knows $(A_{m_0+1}, \ldots, A_m)$ and a degree $\leq n-2$ polynomial $h(X)$, such that above equation holds.

2.1 Definitions

We use the definitions of subversion secure and standard NIZK arguments from [ABLZ17,Gro16,GM17]. Let $\mathcal{R}$ be a relation generator, such that $\mathcal{R}(1^\lambda)$ returns a polynomial-time decidable binary relation $\mathbf{R} = \{(\mathsf{x}, \mathsf{w})\}$. Here, x is the statement and w is the witness. Security parameter λ can be deduced from the description of $\mathbf{R}$. The relation generator also outputs auxiliary information ξ that will be given to the honest parties and the adversary. As in [Gro16,ABLZ17], ξ is the value returned by $\mathsf{BGgen}(1^\lambda)$, so ξ is given as an input to the honest parties; if needed, one can include an additional auxiliary input to the adversary. Let $\mathbf{L_R} = \{\mathsf{x} : \exists \mathsf{w}, (\mathsf{x}, \mathsf{w}) \in \mathbf{R}\}$ be an NP-language. A (subversion-resistant) *NIZK argument system* Ψ for $\mathcal{R}$ consists a tuple of PPT algorithms $(\mathsf{K}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$, such that:

CRS generator: K is a PPT algorithm that, given $(\mathbf{R}, \xi)$ where $(\mathbf{R}, \xi) \in \mathrm{im}(\mathcal{R}(1^\lambda))$, outputs $\mathbf{crs} := (\mathbf{crs_P}, \mathbf{crs_V})$ and stores trapdoors of $\mathbf{crs}$ as $\mathbf{ts}$. We distinguish $\mathbf{crs_P}$ (needed by the prover) from $\mathbf{crs_V}$ (needed by the verifier).

CRS verifier: CV is a PPT algorithm that, given $(\mathbf{R}, \xi, \mathbf{crs})$, returns either 0 (the CRS is incorrectly formed) or 1 (the CRS is correctly formed).

Prover: P is a PPT algorithm that, given $(\mathbf{R}, \xi, \mathbf{crs_P}, \mathsf{x}, \mathsf{w})$ for $\mathsf{CV}(\mathbf{R}, \xi, \mathbf{crs}) = 1$ and $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$, outputs an argument π. Otherwise, it outputs $\perp$.

Verifier: V is a PPT algorithm that, given $(\mathbf{R}, \xi, \mathbf{crs_V}, \mathsf{x}, \pi)$, returns either 0 (reject) or 1 (accept).

Simulator: Sim is a PPT algorithm that, given $(\mathbf{R}, \xi, \mathbf{crs}, \mathsf{x}, \mathbf{ts})$, outputs a simulated argument π.

Strictly speaking, a zk-SNARK system is required to be complete, (computationally) knowledge-sound, (statistically) ZK, and succinct as in the following definitions.

Definition 1 (Perfect Completeness). *A non-interactive argument Ψ is* perfectly complete for $\mathcal{R}$, *if for all λ, all $(\mathbf{R}, \xi) \in \mathrm{im}(\mathcal{R}(1^\lambda))$, and $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$,* $\Pr\left[\mathbf{crs} \leftarrow \mathsf{K}(\mathbf{R}, \xi),\ \pi \leftarrow \mathsf{P}(\mathbf{R}, \xi, \mathbf{crs_P}, \mathsf{x}, \mathsf{w}) : \mathsf{V}(\mathbf{R}, \xi, \mathbf{crs_V}, \mathsf{x}, \pi) = 1 \right] = 1$.

Definition 2 (Computationally Knowledge-Soundness [Gro16]). *A non-interactive argument Ψ is computationally (adaptively)* knowledge-sound for $\mathcal{R}$, *if for every PPT $\mathcal{A}$, there exists a PPT extractor $\mathsf{ext}_\mathcal{A}$, s.t. for all λ,*

$$\Pr\left[\begin{array}{l}\mathbf{crs} \leftarrow \mathsf{K}(\mathbf{R}, \xi), r \leftarrow_r \mathsf{RND}(\mathcal{A}), ((\mathsf{x}, \pi) \,\|\, \mathsf{w}) \leftarrow (\mathcal{A} \,\|\, \mathsf{ext}_\mathcal{A})(\mathbf{R}, \xi, \mathbf{crs}; r) : \\ (\mathsf{x}, \mathsf{w}) \notin \mathbf{R} \wedge \mathsf{V}(\mathbf{R}, \xi, \mathbf{crs_V}, \mathsf{x}, \pi) = 1\end{array}\right] \approx_\lambda 0 \ .$$

Here, ξ can be seen as a common auxiliary input to $\mathcal{A}$ and $\mathsf{ext}_\mathcal{A}$ that is generated by using a benign [BCPR14] relation generator.

Definition 3 (Statistically Zero-Knowledge (ZK) [Gro16]). *A non-interactive argument Ψ is* statistically ZK for $\mathcal{R}$, *if for all λ, all $(\mathbf{R},\xi) \in \mathrm{im}(\mathcal{R}(1^\lambda))$, and for all PPT $\mathcal{A}$, $\varepsilon_0^{unb} \approx_\lambda \varepsilon_1^{unb}$, where*

$$\varepsilon_b = \Pr[(\mathbf{crs} \,\|\, \mathbf{ts}) \leftarrow \mathsf{K}(\mathbf{R},\xi) : \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\mathbf{R},\xi,\mathbf{crs}) = 1] \ .$$

Here, the oracle $\mathsf{O}_0(\mathsf{x},\mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x},\mathsf{w}) \notin \mathbf{R}$, and otherwise it returns $\mathsf{P}(\mathbf{R},\xi,\mathbf{crs}_\mathsf{P},\mathsf{x},\mathsf{w})$. Similarly, $\mathsf{O}_1(\mathsf{x},\mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x},\mathsf{w}) \notin \mathbf{R}$, otherwise it returns $\mathsf{Sim}(\mathbf{R},\xi,\mathbf{crs},\mathbf{ts},\mathsf{x})$. Ψ is perfect ZK *for $\mathcal{R}$ if one requires that $\varepsilon_0 = \varepsilon_1$.*

Definition 4 (Succinctness [GM17]). *A non-interactive argument Ψ is* succinct *if the proof size is polynominal in λ and the verifier's computation time is polynominal in security parameter λ and the size of instance x.*

In the rest, we recall the definition of (non-black-box) simulation extractability (or simulation knowledge soundness) and Sub-ZK [ABLZ17] that we aim to achieve in new constructions.

Definition 5 ((Non-Black-Box) Simulation Extractability [GM17]). *A non-interactive argument Ψ is* (non-black-box) simulation-extractable *for $\mathcal{R}$, if for any PPT $\mathcal{A}$, there exists a PPT extractor $\mathsf{ext}_\mathcal{A}$ s.t. for all λ,*

$$\Pr\left[\begin{matrix}\mathbf{crs} \leftarrow \mathsf{K}(\mathbf{R},\xi), r \leftarrow_r \mathsf{RND}(\mathcal{A}), ((\mathsf{x},\pi) \,\|\, \mathsf{w}) \leftarrow (\mathcal{A}^{\mathsf{O}(.)} \,\|\, \mathsf{ext}_\mathcal{A})(\mathbf{R},\xi,\mathbf{crs};r) : \\ (\mathsf{x},\pi) \notin Q \wedge (\mathsf{x},\mathsf{w}) \notin \mathbf{R} \wedge \mathsf{V}(\mathbf{R},\xi,\mathbf{crs}_\mathsf{V},\mathsf{x},\pi) = 1\end{matrix}\right] \approx_\lambda 0 \ .$$

Here, Q is the set of (x,π)-pairs generated by the adversary's queries to $\mathsf{O}(.)$. Note that *(non-black-box) simulation extractability* implies *knowledge-soundness* (given in Def. 2), as the former additionally allows the adversary to send query to the proof simulation oracle.

Definition 6 (Statistically Subversion Zero-Knowledge [ABLZ17]). *A non-interactive argument Ψ is* statistically Sub-ZK *for $\mathcal{R}$, if for any PPT subverter X there exists a PPT extractor ext_X, such that for all λ, all $(\mathbf{R},\xi) \in \mathrm{im}(\mathcal{R}(1^\lambda))$, and for all PPT $\mathcal{A}$, $\varepsilon_0 \approx_\lambda \varepsilon_1$, where*

$$\Pr\left[\begin{matrix}r \leftarrow_r \mathsf{RND}(\mathsf{X}), (\mathbf{crs},\xi_\mathsf{X} \,\|\, \mathbf{ts}) \leftarrow (\mathsf{X} \,\|\, \mathsf{ext}_\mathsf{X})(\mathbf{R},\xi;r) : \\ \mathsf{CV}(\mathbf{R},\xi,\mathbf{crs}) = 1 \wedge \mathcal{A}^{\mathsf{O}_b(\cdot,\cdot)}(\mathbf{R},\xi,\mathbf{crs},\mathbf{ts},\xi_\mathsf{X}) = 1\end{matrix}\right] \ .$$

Here, ξ_X is auxiliary information generated by subverter X, and the oracle $\mathsf{O}_0(\mathsf{x},\mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x},\mathsf{w}) \notin \mathbf{R}$, and otherwise it returns $\mathsf{P}(\mathbf{R},\xi,\mathbf{crs}_\mathsf{P},\mathsf{x},\mathsf{w})$. Similarly, $\mathsf{O}_1(\mathsf{x},\mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x},\mathsf{w}) \notin \mathbf{R}$, and otherwise it returns $\mathsf{Sim}(\mathbf{R},\xi,\mathbf{crs},\mathbf{ts},\mathsf{x})$. Ψ is perfectly Sub-ZK *for $\mathcal{R}$ if one requires that $\varepsilon_0 = \varepsilon_1$.*

3 Subversion-Resistant Simulation-Extractable NIZKs

As we discussed in the introduction, currently we have NIZK constructions that can achieve Sub-ZK (defined in Def. 6) and knowledge soundness (defined in

Def. 2) at the same time [ABLZ17,Fuc18], which means prover achieves ZK *without trusting* to a third party and verifier achieves knowledge soundness but *with trusting* to the CRS generator. On the other hand, currently, there are *simulation* knowledge sound (defined in Def. 5) NIZK arguments [GM17,Lip19,AB19] but none of them are known to achieve Sub-ZK, which means both prover and verifier needs to trust the CRS generators.

In this section, we aim to construct NIZK arguments that similar to NIZKs studied in [ABLZ17,Fuc18], the prover does not need to trust CRS generators to achieve ZK, but the protocol will guarantee *simulation* knowledge soundness, as in simulation-extractable zk-SNARKs [GM17,Lip19]. Recently, the scheme proposed in [Lip19] also was updated to achieve Sub-ZK. However, in the rest, we will observe that our proposed construction can be instantiated with any of current subversion-resistant NIZKs which basically allows us to use it as a framework to achieve simulation (knowledge) soundness in all NIZKs that guarantee Sub-ZK and (knowledge) soundness. Subversion ZK in new constructions guarantees that even an adversary generates the CRS, still it cannot break the ZK of the protocol. To mitigate the level of trust or to improve security in the setup phase even more, particularly for verifier, one can use Multi-Party Computation (MPC) protocols for CRS generation [BCG$^+$15,ABL$^+$19].

3.1 Construction

In this section, we presented the proposed construction which can act as a compiler that transforms Sub-ZK and (knowledge) sound NIZKs into ones that satisfy Sub-ZK and simulation (knowledge) soundness. We use folklore OR technique with a particular instantiation proposed in [KZM$^+$15,AB19]. Indeed, the proposed OR compiler can be viewed as using the Bellare-Goldwasser paradigm [BG90], which is proposed to construct signatures from NIZK arguments, in a non-black-box way.

Consider a subversion-resistant NIZK argument Ψ for $\mathcal{R}_{\mathbf{L}}$ which consists of PPT algorithms $(\mathsf{K}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ and guarantees Sub-ZK and (knowledge) soundness. Let $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{SigVerify})$ be a one-time secure signature scheme and $(\mathsf{Com}, \mathsf{ComVerify})$ be a perfectly binding commitment scheme. Using a variation of a construction proposed by Bellare-Goldwasser [BG90] (used in [KZM$^+$15,AB19], given the language $\mathbf{L}$ with the corresponding NP relation $\mathbf{R}_{\mathbf{L}}$, we define a new language $\mathbf{L}'$ such that $((\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (\mathsf{w}, s, r)) \in \mathbf{R}_{\mathbf{L}'}$ iff:

$$((\mathsf{x}, \mathsf{w}) \in \mathbf{R}_{\mathbf{L}} \vee (\mu = f_s(pk_{\mathsf{Sign}}) \wedge \rho = \mathsf{Com}(s, r))) \,,$$

where $\{f_s : \{0,1\}^* \to \{0,1\}^\lambda\}_{s \in \{0,1\}^\lambda}$ is a pseudo-random function family. Due to OR-based construction of new language $\mathbf{L}'$, a user to be able to generate an acceptable proof will require either the witness w or the trapdoors of CRS, and since it is assumed that the CRS trapdoors are kept secret, so soundness is guaranteed as long as CRS trapdoors are secret. We note that due to using the pseudo-random function f_s with a random secret trapdoor s, the output of f_s is indistinguishable from the outputs of a truly random function. By

CRS Generator $\mathsf{K}'(\mathbf{R_L}, \xi)$**:** Call CRS generator of the subversion-resistant NIZK Ψ and sample $(\mathbf{crs} \,\|\, \mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $\mathbf{crs}' := (\mathbf{crs}, \rho)$.

CRS Verifier $\mathsf{CV}'(\mathbf{R_L}, \xi, \mathbf{crs}')$**:** Parse $\mathbf{crs}' := (\mathbf{crs}, \rho)$; abort if $\rho = 0$; call CV algorithm of the subversion-resistant NIZK Ψ and return $b \leftarrow \mathsf{CV}(\mathbf{R_L}, \xi, \mathbf{crs})$.

Prover $\mathsf{P}'(\mathbf{R_L}, \xi, \mathbf{crs}', \mathsf{x}, \mathsf{w})$**:** Parse $\mathbf{crs}' := (\mathbf{crs}, \rho)$; abort if $\mathsf{CV}'(\mathbf{R_L}, \xi, \mathbf{crs}') \neq 1$ and $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R_L}$; generate $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; sample $z_0, z_1, z_2 \leftarrow_r \{0,1\}^\lambda$; generate $\pi \leftarrow \mathsf{P}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, z_0, pk_{\mathsf{Sign}}, \rho), (\mathsf{w}, z_1, z_2))$ using the prover of subversion-resistant NIZK Ψ; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, z_0, \pi))$; and return $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$.

Verifier $\mathsf{V}'(\mathbf{R_L}, \xi, \mathbf{crs}', \mathsf{x}, \pi')$**:** Parse $\mathbf{crs}' := (\mathbf{crs}, \rho)$ and $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$; abort if $\mathsf{SigVerify}(pk_{\mathsf{Sign}}, (\mathsf{x}, z_0, \pi), \sigma) = 0$; call the verifier of input subversion-resistant NIZK argument $\mathsf{V}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, z_0, pk_{\mathsf{Sign}}, \rho), \pi)$ and abort if it outputs 0.

Simulator $\mathsf{Sim}'(\mathbf{R_L}, \xi, \mathbf{crs}', \mathbf{ts}, \mathsf{x})$**:** Parse $\mathbf{crs}' := (\mathbf{crs}, \rho)$; call the extraction algorithm ext_X constructed in simulation of subversion-resistant NIZK Ψ and extract simulation trapdoors $\mathbf{ts}$ of Ψ from the CRS generator X; generate $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; sample $z_0 \leftarrow_r \{0,1\}^\lambda$; generate $\pi \leftarrow \mathsf{Sim}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, z_0, pk_{\mathsf{Sign}}, \rho), \mathbf{ts})$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, z_0, \pi))$; and output $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$.

Fig. 1: An extension of the proposed construction in [AB19] that outputs a Sub-ZK and simulation (knowledge) sound NIZK argument Ψ'. Note that in our construction, we assumed that the input NIZK Ψ guarantees Sub-ZK and (knowledge) soundness. Due to this fact, we have a new algorithm CV' to verify the well-formedness of CRS elements, and a new simulation procedure by Sim' to simulate the proofs without trusting to the CRS generators.

considering new language, the subversion-resistant NIZK argument Ψ for the relation $\mathbf{R_L}$ with PPT algorithms $(\mathsf{K}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ that achieves Sub-ZK and (knowledge) soundness, can be lifted to a subversion-resistant NIZK Ψ' with PPT algorithms $(\mathsf{K}', \mathsf{CV}', \mathsf{P}', \mathsf{V}', \mathsf{Sim}')$ that guarantees Sub-ZK and *simulation* (knowledge) soundness. Based on the language $\mathbf{L}'$, the construction of NIZK Ψ' and the corresponding algorithms are described in Fig. 1.

Recently, Atapoor and Baghery [AB19] used the same construction to achieve *simulation knowledge soundness* in Groth's zk-SNARK [Gro16] which led to the most efficient zk-SNARK which also guarantees non-malleability of proofs. Here, we show that such OR-based language can be extended and used to build subversion-resistant NIZK arguments which will also guarantee simulation (knowledge) soundness.

Recall that one of two main differences between a Sub-ZK NIZK argument and a common NIZK argument is the existence of a public CRS verification algorithm CV in the former ones. Basically, given a CRS $\mathbf{crs}$ the algorithm CV verifies the well-formedness of its elements. Additionally, in the simulation of a Sub-ZK NIZK argument, there exists a (non-black-box) extractor ext_X that can

extract the simulation trapdoors from a (possibly malicious) CRS generator X. More detailed discussions can be found in [BFS16,ABLZ17,Fuc18,ALSZ18].

So similar to other subversion-resistant NIZK arguments, as we aim to achieve Sub-ZK (not standard ZK) and simulation (knowledge) soundness in our constructions, so there are two key differences between new proposed constructions and the one presented in [AB19] (that are shown in highlighted form in Fig. 1). The first key difference is that we have an extra algorithm CV' which will be used by prover to check the well-formedness of CRS elements before using them. This is actually the cost that prover needs to pay insted of trusting to the CRS generators. The second key difference is that in new constructions, the simulator Sim' does not get simulation trapdoors directly, as the prover does not trust to the CRS generators in this setting. Instead, the simulator Sim' calls the extraction algorithm $\mathsf{ext}_{\mathsf{X}}$ constructed for the input NIZK argument Ψ and extracts simulation trapdoors $\mathbf{ts}$ of it, and then uses them for the rest of simulation.

3.2 Efficiency

In the rest, by considering Fig. 1, we consider efficiency of new constructions for different important metrics in (subversion-resistant) NIZK arguments.

Setup phase. The setup needs to be done for a new language $\mathbf{L}'$. Roughly speaking, it means the setup phase of the original NIZK Ψ needs to be executed for a new arithmetic circuit that has a slightly larger number of gates. Again with a particular instantiation [KMS+16,AB19], new changes will add around 52.000 multiplication gates to the QAP-based arithmetic circuits that encode $\mathbf{L}$. The number of gates comes from the case that both commitment scheme and pseudo-random function used in construction are instantiated with a SHA512 hash function [KMS+16,AB19]. Implementations show that this will add a constant amount (e.g. 10 megabytes) to the size of original CRS, that for arithmetic circuits with a large number of gates (that usually is the case in practical application) the overhead is negligible[AB19].

CRS Verification. In new construction, in order to verify the well-formedness of CRS elements one needs to execute CV' algorithm which almost has the same efficiency as CV algorithm in original NIZK argument Ψ. In practice, it is shown that the running time of CV can be less than running time of P [ABLZ17].

Prover. In new schemes, prover needs to give a proof for the new language $\mathbf{L}'$ and sign the proof with a one-time secure signature scheme. Empirical performances presented in [AB19] show that the overhead for a QAP-based zk-SNARK is very small in practical cases. For instance, in the fixed setting, for an arithmetic circuit that prover already needed 83 seconds to generate a proof, in new construction the proof generation took 90.1 seconds.

Proof size. In new constructions the size of new proof $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$ will be equal to the size of original proof π plus the size of three elements (z_0, $pk_{\mathsf{Sign}}, \sigma$). Similarly, with a particular instantiation for 128-bit security (e.g. the one in [AB19]), these three elements totally can add less than 128 bytes to the size of original proof π.

Verifier. Extra from the verification of NIZK argument Ψ, the verifier of argument Ψ' will verify the validity of a one-time secure signature. Similarly, for a particular instantiation [AB19], verification of the signature scheme adds 1 equation to the verification of the original scheme that needs two pairings and one exponentiation. They show that a proper instantiation can even give a verification faster than the verification of current simulation knowledge sound NIZKs arguments in the CRS model [GM17,Lip19].

3.3 Security Proofs

In this section, we show that the given a subversion-resistant NIZK argument that guarantees completeness, Sub-ZK, and (knowledge) soundness, the described construction in Sec. 3.1 results in a NIZK argument that achieves completeness, Sub-ZK and *simulation* (knowledge) soundness.

Theorem 1 (Completeness). *If the NIZK argument Ψ ensures completeness, Sub-ZK, and (knowledge) soundness, and the one-time signature scheme* $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{SigVerify})$ *is secure, then the proposed construction in Fig. 1 guarantees completeness.*

Proof. By considering the completeness of NIZK argument Ψ and the fact that $\mathsf{SigVerify}(pk_{\mathsf{Sign}}, m, \mathsf{Sign}(m, sk_{\mathsf{Sign}})) = 1$, we conclude that the output NIZK argument Ψ' guarantees *completeness.* □

Theorem 2 (Subversion Zero-Knowledge). *If the NIZK argument Ψ guarantees completeness, Sub-ZK, and (knowledge) soundness, the pseudo-random function family is secure, and the one-time signature scheme is secure, then the proposed construction in Fig. 1 achieves Sub-ZK.*

Before going through the proof, it is worth to mention that proving Sub-ZK of a subversion-resistant NIZK argument is a bit tricky than proving the standard notion of ZK. The reason is that in the proof of standard ZK, CRS generator is trusted and the CRS trapdoors (simulation trapdoors) are honestly provided to the simulator Sim. But in proving Sub-ZK, since the prover does not trust to the CRS generator anymore, consequently the simulator Sim cannot trust to the provided trapdoors. To address this issue, the proposed solution [BFS16] is that the prover checks the well-formedness of CRS elements before using them and in simulating proofs, the simulator uses a non-black-box extraction procedure to extract the simulation trapdoors directly from the (possibly malicious) CRS generator and then uses them for the simulation [BFS16,ABLZ17,Fuc18,ALSZ18,Bag19b]. The non-black-box extraction

usually is done using various knowledge assumptions [Dam92,BFS16]. For instance [ABLZ17] used Bilinear Diffie-Hellman Knowledge of Exponents (BDH-KE) assumption [2] to prove Sub-ZK of Groth's zk-SNARK [Gro16], while Fuchsbauer [Fuc18] did the same but with the Square Knowledge of Exponent (SKE) assumption[3] which led to prove Sub-ZK of Groth's zk-SNARK [Gro16] without modifying its CRS. But, intuitively in all cases, one relies on the fact that if a (malicious) CRS generator manages to come out with some *well-formed* CRS elements, there exists a non-black-box extractor that given access to the source code and random coins of the (malicious) CRS generator, it can extract the simulation trapdoors. Once the simulation trapdoors are extracted, the simulator Sim can simulate the proofs. It is worth to mention that the well-formedness of CRS elements are checked by a public efficient CRS verification algorithm known as CV. Using different knowledge-assumptions in proving Sub-ZK of particular NIZK arguments might lead to different CV algorithms, as Abdolmaleki et al. [ABLZ17] and Fuchsbauer [Fuc18] proposed two different CV algorithms for Groth's zk-SNARK [Gro16].

Proof. Sub-ZK in the input NIZK implies that there exists a simulation algorithm Sim which first uses the extraction algorithm $\mathsf{ext}_{\mathsf{X}}$ and extracts the simulation trapdoors from (malicious) CRS generator X and then uses the extracted trapdoors to simulate the argument. We note that due to OR-based construction of new language $\mathbf{L}'$, the proofs for new language can be simulated using either simulation trapdoors of NIZK argument Ψ (first branch) or simulation trapdoors (s, r) of that are hidden in the committed value $\rho := \mathsf{Com}(s, r)$ (second branch). Here for simulation, we use simulation trapdoors of NIZK argument Ψ which can be extracted by $\mathsf{ext}_{\mathsf{X}}$. Now, consider the following experiences,

$\underline{\mathsf{EXP}_1^{zk}}$(simulator):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathbf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $\mathbf{crs}' := (\mathbf{crs}, \rho)$; where $\mathbf{ts}$ contains trapdoors of CRS $\mathbf{crs}$ and (s, r) are hidden trapdoors of the committed value ρ.
- *Define function* $\mathsf{O}(\mathsf{x}, \mathsf{w})$: Abort if $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R}_{\mathbf{L}}$; call the extraction algorithm $\mathsf{ext}_{\mathsf{X}}$ constructed in simulation of subversion-resistant NIZK Ψ and extract simulation trapdoors $\mathbf{ts}$ of Ψ from CRS generator X; generate $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; sample $z_0 \leftarrow_r \{0,1\}^\lambda$; generate $\pi \leftarrow \mathsf{Sim}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, z_0, pk_{\mathsf{Sign}}, \rho), \mathbf{ts})$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, z_0, \pi))$; return $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $b \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x},\mathsf{w})}(\mathbf{crs}')$

[2] It states that in an asymmetric bilinear group, given $[1]_1$ and $[1]_1$, if an adversary manages to come out with $[a]_1$ and $[a]_2$, he must know a. Knowing a is formalized by showing that there exists an efficient non-black-box extractor that given access to source code and random coins of the adversary, it can extract a [Dam92].

[3] It states that in asymmetric bilinear group, given $[1]_1$, if an adversary manages to come out with $[a]_1$ and $\left[a^2\right]_1$, he must know a.

$\underline{\mathsf{EXP}_2^{zk}}$(prover):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathbf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $\mathbf{crs}' := (\mathbf{crs}, \rho)$; where $\mathbf{ts}$ contains trapdoors of CRS $\mathbf{crs}$ and (s, r) are hidden trapdoors of the committed value ρ.
- *Define function* $\mathsf{O}(\mathsf{x}, \mathsf{w})$: Abort if $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R}_{\mathbf{L}}$; generate $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; sample $z_0, z_1, z_2 \leftarrow_r \{0,1\}^\lambda$; generate $\pi \leftarrow \mathsf{P}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, z_0, pk_{\mathsf{Sign}}, \rho), (\mathsf{w}, z_1, z_2))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, z_0, \pi))$; return $\pi' := (z_0, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $b \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x},\mathsf{w})}(\mathbf{crs}')$

Lemma 1. *If the NIZK argument Ψ guarantees Sub-ZK, and the one-time signature scheme is secure, for two above experiments we have* $\Pr[\mathsf{EXP}_1^{zk}] \approx \Pr[\mathsf{EXP}_2^{zk}]$.

Proof. Two experiments EXP_2^{zk} and EXP_1^{zk} model the real prover and simulator of new construction described in Fig. 1. As the NIZK argument Ψ guarantees Sub-ZK, consequently it guarantees ZK, so one can conclude that the proof generated by prover in experiment EXP_2^{zk} is indistinguishable from the proof generated by the simulator in EXP_1^{zk}. Intuitively, this is because of OR-based construction of new language $\mathbf{L}'$, and consequently the fact that all new elements added to the new construction are chosen randomly and independently. □

This results that the constructed NIZK arguments in Sec. 3.1 ensures Sub-ZK. □

Theorem 3 ((Non-Black-Box) Simulation Knowledge Soundness). *If the NIZK argument Ψ is complete, Sub-ZK, and knowledge sound, then the proposed construction in Fig. 1 guarantees (non-black-box) simulation knowledge soundness.*

Before going through the proof of the theorem, recall that simulation knowledge soundness states that given an honestly generated CRS $\mathbf{crs}$, an adversary cannot come out with valid fresh proof, even if he access to an oracle which returns simulated proofs for an arbitrary statement, unless he knows the witness. The existing of an oracle which returns simulated proofs shows that the protocol is simulatable, so the proofs should be zero-knowledge. In the last theorem, we observed that the constructed NIZK argument in Sec. 3.1 guarantees Sub-ZK and consequently ZK. In proving this theorem, as verifier trusts to the CRS generator and as Sub-ZK implies ZK, so we use the simulator of standard ZK to prove this theorem.

Proof. The proof is simplified and generalized version of the proof of simulation knowledge soundness presented in [KZM+15] and in [AB19], respectively, but for all (Sub-)ZK and (knowledge) sound NIZK arguments. For the sake of completeness, we provide the proof in the rest. The proof is for the case that the input NIZK argument guarantees knowledge soundness. Similarly, we write consecutive hybrid experiments and at the end show that success probability of

an adversary to break the simulation knowledge soundness of new constructions are negligible. Consider the following experiments,

$\underline{\mathsf{EXP}_1^{SimExt}}$(main experiment):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $(\mathbf{crs}' \,\|\, \mathsf{ts}') := ((\mathbf{crs}, \rho) \,\|\, (\mathsf{ts}, (s, r)))$; where ts' is new CRS trapdoor.
- *Define function* $\mathsf{O}(\mathsf{x})$: $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; set $\mu = f_s(pk_{\mathsf{Sign}})$; generate $\pi \leftarrow \mathsf{P}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (s, r))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, \mu, \pi))$; return $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $(\mathsf{x}, \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x})}(\mathbf{crs}')$.
- Parse $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$; $\mathsf{w} \leftarrow \mathsf{ext}_{\mathcal{A}}(\mathbf{crs}', \mathsf{x}, \pi, \xi)$.
- Return 1 iff $((\mathsf{x}, \pi') \notin Q) \wedge \ (\mathsf{V}'(\mathbf{R}_{\mathbf{L}}, \xi, \mathbf{crs}', \mathsf{x}, \pi') = 1) \wedge ((\mathsf{x}, \mathsf{w}) \notin \mathbf{R}_{\mathbf{L}})$; where Q shows the set of statment-proof pairs generated by $\mathsf{O}(\mathsf{x})$.

$\underline{\mathsf{EXP}_2^{SimExt}}$(relaxing the return checking):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $(\mathbf{crs}' \,\|\, \mathsf{ts}') := ((\mathbf{crs}, \rho) \,\|\, (\mathsf{ts}, (s, r)))$; where ts' is new CRS trapdoor.
- *Define function* $\mathsf{O}(\mathsf{x})$: $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; set $\mu = f_s(pk_{\mathsf{Sign}})$; generate $\pi \leftarrow \mathsf{P}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (s, r))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, \mu, \pi))$; return $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $(\mathsf{x}, \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x})}(\mathbf{crs}')$.
- Parse $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$; $\mathsf{w} \leftarrow \mathsf{ext}_{\mathcal{A}}(\mathbf{crs}', \mathsf{x}, \pi, \xi)$.
- Return 1 iff $((\mathsf{x}, \pi') \notin Q) \wedge \ (\mathsf{V}'(\mathbf{R}_{\mathbf{L}}, \xi, \mathbf{crs}', \mathsf{x}, \pi') = 1) \wedge (pk_{\mathsf{Sign}} \notin \mathcal{PK}) \wedge (\mu = f_s(pk_{\mathsf{Sign}}))$; where Q is the set of statment-proof pairs and $\mathcal{PK}$ is the set of signature verification keys, both generated by $\mathsf{O}(\mathsf{x})$.

Lemma 2. *If the underlying one-time signature scheme is strongly unforgeable, and the NIZK argument guarantees knowledge-soundness, then* $\Pr[\mathsf{EXP}_2^{SimExt}] \leq \Pr[\mathsf{EXP}_1^{SimExt}] + \mathsf{negl}(\lambda)$.

Proof. We note that if $(\mathsf{x}, \pi') \notin Q$ and "pk_{Sign} from (x, π'), has been generated by $\mathsf{O}(\cdot)$", then the (x, μ, π) is a valid message/signature pair. Therefore by security of the signature scheme, we know that $(\mathsf{x}, \pi) \notin Q$ and "pk_{Sign} has been generated by $\mathsf{O}(\cdot)$" happens with negligible probability, which allows us to focus on $pk_{\mathsf{Sign}} \notin \mathcal{PK}$.

Now, due to knowledge soundness of the original scheme (there is an extractor $\mathsf{ext}_{\mathcal{A}}$ where can extract witness from $\mathcal{A}$), if some witness is valid for $\mathbf{L}'$ and $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R}_{\mathbf{L}}$, so we conclude it must be the case that there exists some s', such that ρ is valid commitment of s' and $\mu = f_{s'}(pk_{\mathsf{Sign}})$, which by perfectly binding property of the commitment scheme, it implies $\mu = f_s(pk_{\mathsf{Sign}})$. □

$\underline{\mathsf{EXP}_3^{SimExt}}$(simulator):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s, r)$; and output $(\mathbf{crs}' \,\|\, \mathsf{ts}') := ((\mathbf{crs}, \rho) \,\|\, (\mathsf{ts}, (s, r)))$; where ts' is new CRS trapdoor.

- *Define function* $\mathsf{O}(\mathsf{x})$: $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; set $\mu = f_s(pk_{\mathsf{Sign}})$; generate $\pi \leftarrow \mathsf{Sim}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (\mathsf{ts} \,\|\, (s, r)))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, \mu, \pi))$; return $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $(\mathsf{x}, \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x})}(\mathbf{crs}')$.
- Parse $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$; $\mathsf{w} \leftarrow \mathsf{ext}_{\mathcal{A}}(\mathbf{crs}', \mathsf{x}, \pi, \xi)$.
- Return 1 iff $((\mathsf{x}, \pi') \notin Q) \wedge \ (\mathsf{V}'(\mathbf{R}_{\mathbf{L}}, \xi, \mathbf{crs}', \mathsf{x}, \pi') = 1) \wedge (pk_{\mathsf{Sign}} \notin \mathcal{PK}) \wedge (\mu = f_s(pk_{\mathsf{Sign}}))$; where Q is the set of statment-proof pairs and $\mathcal{PK}$ is the set of signature verification keys, both generated by $\mathsf{O}(\mathsf{x})$.

Lemma 3. *If the NIZK argument Ψ guarantees zero-knowledge, then for two experiments EXP_3^{SimExt} and EXP_2^{SimExt}, we have* $\Pr[\mathsf{EXP}_3^{SimExt}] \leq \Pr[\mathsf{EXP}_2^{SimExt}] + \mathsf{negl}(\lambda)$.

Proof. As the NIZK argument Ψ ensures Sub-ZK and consequently ZK, so it implies no polynomial time adversary can distinguish a proof generated by the simulator from a proof that is generated by the prover. So, as we are running in polynomial time, thus two experiments are indistinguishable. □

EXP_4^{SimExt}(separating secret key of pseudo random function):

- *Setup:* Sample $(\mathbf{crs} \,\|\, \mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; $s', s, r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s', r)$; and output $(\mathbf{crs}' \,\|\, \mathsf{ts}') := ((\mathbf{crs}, \rho) \,\|\, (\mathsf{ts}, (s, s', r)))$; where ts' is new trapdoor.
- *Define function* $\mathsf{O}(\mathsf{x})$: $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; set $\mu = f_s(pk_{\mathsf{Sign}})$; generate $\pi \leftarrow \mathsf{Sim}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (\mathsf{ts} \,\|\, (s, r)))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, \mu, \pi))$; return $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $(\mathsf{x}, \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x})}(\mathbf{crs}')$.
- Parse $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$; $\mathsf{w} \leftarrow \mathsf{ext}_{\mathcal{A}}(\mathbf{crs}', \mathsf{x}, \pi, \xi)$.
- Return 1 iff $((\mathsf{x}, \pi') \notin Q) \wedge \ (\mathsf{V}'(\mathbf{R}_{\mathbf{L}}, \xi, \mathbf{crs}', \mathsf{x}, \pi') = 1) \wedge (pk_{\mathsf{Sign}} \notin \mathcal{PK}) \wedge (\mu = f_s(pk_{\mathsf{Sign}}))$; where Q is the set of statment-proof pairs and $\mathcal{PK}$ is the set of signature verification keys, both generated by $\mathsf{O}(\mathsf{x})$.

Lemma 4. *If the commitment scheme used in the CRS generation is computationally hiding, then* $\Pr[\mathsf{EXP}_4^{SimExt}] \leq \Pr[\mathsf{EXP}_3^{SimExt}] + \mathsf{negl}(\lambda)$.

Proof. Computationally hiding of a commitment scheme implies that $\mathsf{Com}(m_1, r)$ and $\mathsf{Com}(m_2, r)$ are computationally indistinguishable, as in this lemma. □

EXP_5^{SimExt}(replace pseudo random function $f_s(\cdot)$ with true random function $F(\cdot)$):

- *Setup:* Sample $(\mathbf{crs}||\mathsf{ts}) \leftarrow \mathsf{K}(\mathbf{R}_{\mathbf{L}'}, \xi)$; s', $\not{s}$, $r \leftarrow_r \{0,1\}^\lambda$; $\rho := \mathsf{Com}(s', r)$; and output $(\mathbf{crs}'||\mathsf{ts}') := ((\mathbf{crs}, \rho)||(\mathsf{ts}, (\not{s}, s', r)))$; where ts' is new CRS trapdoor.
- *Define function* $\mathsf{O}(\mathsf{x})$: $(pk_{\mathsf{Sign}}, sk_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$; set $\mu = F(pk_{\mathsf{Sign}})$; generate $\pi \leftarrow \mathsf{Sim}(\mathbf{R}_{\mathbf{L}'}, \xi, \mathbf{crs}, (\mathsf{x}, \mu, pk_{\mathsf{Sign}}, \rho), (\mathsf{ts}||(s, r)))$; sign $\sigma \leftarrow \mathsf{Sign}(sk_{\mathsf{Sign}}, (\mathsf{x}, \mu, \pi))$; return $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$.
- $(\mathsf{x}, \pi') \leftarrow \mathcal{A}^{\mathsf{O}(\mathsf{x})}(\mathbf{crs}')$.
- Parse $\pi' := (\mu, \pi, pk_{\mathsf{Sign}}, \sigma)$; $\mathsf{w} \leftarrow \mathsf{ext}_{\mathcal{A}}(\mathbf{crs}', \mathsf{x}, \pi, \xi)$.

– Return 1 iff $((\mathsf{x}, \pi') \notin Q) \wedge (\mathsf{V}'(\mathbf{R_L}, \xi, \mathbf{crs}', \mathsf{x}, \pi') = 1) \wedge (pk_{\mathsf{Sign}} \notin \mathcal{PK}) \wedge (\mu = F(pk_{\mathsf{Sign}}))$; where Q is the set of statment-proof pairs and $\mathcal{PK}$ is the set of signature verification keys, both generated by $\mathsf{O}(\mathsf{x})$.

Lemma 5. *If the underlying truly random function $F(\cdot)$ is secure, then* $\Pr[\mathsf{EXP}_4^{SimExt}] \leq \Pr[\mathsf{EXP}_5^{SimExt}]$.

Proof. By assuming function $F(\cdot)$ is secure, we can conclude no polynomial time adversary can distinguish an output of the true random function $F(\cdot)$ from an output of the pseudo random function $f_s(\cdot)$. Indeed, experiment EXP_5^{SimExt} can be converted to an adversary for the game of a *true random function.* □

Claim. For experiment EXP_5^{SimExt}, we have $\Pr[\mathsf{EXP}_5^{SimExt}] \leq 2^{-\lambda}$.

Proof. From verification we know $pk_{\mathsf{Sign}} \notin \mathcal{PK}$, therefore $F(pk_{\mathsf{Sign}})$ has not been queried already. Thus, we will see $F(pk_{\mathsf{Sign}})$ as a newly generated random string independent from μ, which implies adversary only can guess. □

This completes proof of the main theorem. □

4 A Sub-ZK Simulation-Extractable SNARK

In this section, we aim to instantiate the construction proposed in Sec. 3.1, and achieve a NIZK argument that can guarantee Sub-ZK and simulation knowledge soundness. In such a scheme, the prover will get sure that the proof is ZK without trusting the CRS generators but to achieve simulation knowledge soundness they will trust the CRS generators. Based on discussions in Sec. 3.1 and Fig. 1, we need to instantiate some primitives including the pseudo-random function, the commitment scheme, and the one-time secure signature scheme and also use a subversion-resistant NIZK argument as a subroutine.

A Subversion-Resistant NIZK. Currently, Groth's zk-SNARK is the most efficient subversion-resistant NIZK argument that is proven to achieve Sub-ZK [ABLZ17,Fuc18] and knowledge soundness [Gro16] in the generic group model. While proving Sub-ZK, Abdolmaleki et al. [ABLZ17] and Fuchsbauer [Fuc18] have proposed two different CRS verification CV algorithms for Groth's zk-SNARK, which the later works for original version but the first one requires some changes in the CRS of Groth's zk-SNARK.

Instantiation of Primitives. As mentioned before, recently Atapoor and Baghery [AB19] used a similar construction to achieve simulation knowledge soundness in Groth's zk-SNARK [Gro16]. Their main goal was to construct an efficient version of Groth's zk-SNARK that can also guarantee the non-malleability of proofs and outperforms Groth and Maller's zk-SNARK [GM17]. They instantiate pseudo-random function and commitment scheme with SHA512 hash function which requires an arithmetic circuit with $\approx$ 26.000 multiplication gates [KMS+16]. They used Boneh and Boyen's signature scheme [BB08] to

CRS of Groth's zk-SNARK [Gro16]:

$$(\mathbf{crs}_{\mathsf{P}}, \mathbf{crs}_{\mathsf{V}}) := \mathbf{crs} \leftarrow \begin{pmatrix} [\alpha, \beta, \delta, (\chi^i)_{i=0}^{n-1}, (\frac{u_j(\chi)\beta + v_j(\chi)\alpha + w_j(\chi)}{\gamma})_{j=0}^{m_0}, \\ (\frac{u_j(\chi)\beta + v_j(\chi)\alpha + w_j(\chi)}{\delta})_{j=m_0+1}^{m}, (\chi^i \ell(\chi)/\delta)_{i=0}^{n-2}]_1, \\ [\beta, \gamma, \delta, (\chi^i)_{i=0}^{n-1}]_2, \ [\alpha\beta]_T \end{pmatrix}.$$

CRS of the variation of Groth's zk-SNARK in [AB19]: $\mathbf{crs}' := (\mathbf{crs}, \rho)$.

$\mathsf{CV}'(\mathbf{R}_{\mathbf{L}}, \xi, \mathbf{crs}' := (\mathbf{crs}, \rho))$:

1. For $\iota \in \{1, \alpha, \beta, \delta, \ell(\chi)/\delta\}$ and $\iota' \in \{1, \gamma\}$: check that $[\iota]_1 \neq [0]_1$ and $[\iota']_2 \neq [0]_1$
2. For $i = 1$ to $n-1$: check that $\left[\chi^i\right]_1 \bullet [1]_2 = [1]_1 \bullet \left[\chi^i\right]_2$, and $\left[\chi^i\right]_1 \bullet [1]_2 = \left[\chi^{i-1}\right]_1 \bullet [\chi]_2$,
3. For $\iota \in \{\alpha, \beta, \delta\}$: check that $[\iota]_1 \bullet [1]_2 = [1]_1 \bullet [\iota]_2$,
4. For $j = m_0 + 1$ to m: check that $[(u_j(\chi)\beta + v_j(\chi)\alpha + w_j(\chi))/\delta]_1 \bullet [\delta]_2 = [u_j(\chi)]_1 \bullet [\beta]_2 + [\alpha]_1 \bullet [v_j(\chi)]_2 + [w_j(\chi)]_1 \bullet [1]_2$,
5. For $i = 0$ to $n-2$: check that $\left[\chi^i \ell(\chi)/\delta\right]_1 \bullet [\delta]_2 = \sum_{j=0}^{n-1} L_j \left[\chi^j\right]_1 \bullet \left[\chi^i\right]_2$,
6. For $j = 0$ to m_0: check that $[(u_j(\chi)\beta + v_j(\chi)\alpha + w_j(\chi))/\gamma]_1 \bullet [\gamma]_2 = [u_j(\chi)]_1 \bullet [\beta]_2 + [\alpha]_1 \bullet [v_j(\chi)]_2 + [w_j(\chi)]_1 \bullet [1]_2$,
7. Check that $[\alpha]_1 \bullet [\beta]_2 = [\alpha\beta]_T$, and $[\delta]_2$ in $\mathbf{crs}_{\mathsf{P}}$ and $\mathbf{crs}_{\mathsf{V}}$ are the same,
8. Check that $\rho \neq 0$.

If all above checks succeeded then return 1 (the CRS is correctly formed) and otherwise 0 (the CRS is incorrectly formed).

Fig. 2: A CRS verification algorithm for the variation of Groth's zk-SNARK [Gro16] proposed by Atapoor and Baghery [AB19]. Note that $\mathbf{crs}' := (\mathbf{crs}, \rho)$, where $\rho := \mathsf{Com}(s, r)$ and $\mathbf{crs}$ is the CRS of Groth's zk-SNARK that is shown above the figure.

instantiate the one-time secure signature scheme which is a pairing-based signature scheme and works as follows:

Key Generation, $(\mathsf{pk}_{\mathsf{Sign}}, \mathsf{sk}_{\mathsf{Sign}}) \leftarrow \mathsf{KGen}(1^\lambda)$: Given a bilinear group description $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, selects $\mathsf{sk} \leftarrow_r \mathbb{Z}_p^*$, and returns $(\mathsf{pk}_{\mathsf{Sign}}, \mathsf{sk}_{\mathsf{Sign}}) := ([\mathsf{sk}]_1, \mathsf{sk})$.

Signing, $[\sigma]_2 \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{Sign}}, m)$: Given the group description, $\mathsf{sk}_{\mathsf{Sign}}$, and a message m, returns $[\sigma]_2 = [1/(m + \mathsf{sk})]_2$.

Verification, $\{1, 0\} \leftarrow \mathsf{SigVerify}(\mathsf{pk}_{\mathsf{Sign}}, [\sigma]_2)$: Given $\mathsf{pk}_{\mathsf{Sign}}$, m, and $[\sigma]_2$, checks whether $[m + \mathsf{sk}]_1 \bullet [1/(m + \mathsf{sk})]_2 = [1]_T$ and returns either 1 or 0.

Subversion-Resistant Simulation-Extractable SNARK. In the rest, we use the instantiations used in [AB19] and the CV algorithm proposed by Fuchsbauer [Fuc18] to construct a variation of Groth's zk-SNARK that will guarantee Sub-ZK and simulation knowledge soundness. In Fig. 2 we presented a CRS verification algorithm CV' which basically is a minimally changed version of the CV

algorithm proposed by Fuchsbauer [Fuc18]. In CV' we also check whether $\rho \neq 0$, and basically this is the only difference between CV' and CV algorithms.

Finally, as we used the same instantiations used in [AB19], so the other three algorithms $(\mathsf{K}', \mathsf{P}', \mathsf{V}')$ will be the same as in the variation of Groth's zk-SNARK proposed in [AB19]. Basically, we propose to add the new algorithm CV' to their variation and with minimal computational cost, achieve Sub-ZK as well. To this end, the prover first needs to check the well-formedness of CRS elements with executing the algorithm CV' (shown in Fig. 2) and if it returned 1 (the CRS is well-formed) it continues and generates the proof as described in Fig. 1. Abodlmaleki et al. [ABLZ17] showed that using batching techniques a similar CV' algorithm can be executed very efficiently; especially faster than running time of prover.

5 Conclusion

Recently, Atapoor and Baghery [AB19] defined an OR-based language [BG90] with a particular instantiation [KZM+15,AB19] to construct a variation of Groth's zk-SNARK [Gro16] that achieves simulation knowledge soundness; consequently it guarantees non-malleability of proofs.

In this paper, we showed that the same technique can be used to amplify the best result in constructing subversion-resistant NIZK arguments [BFS16,ABLZ17,Fuc18,FO18,ALSZ18]. Technically speaking, we proved that if the input NIZK argument already achieves Sub-ZK (ZK without trusting to a third party) and (knowledge) soundness, by applying the mentioned technique, the lifted NIZK argument will *also* guarantee Sub-ZK and *simulation* (knowledge) soundness. Simulation knowledge soundness (a.k.a. simulation-extractability) ensures non-malleability of proofs which is a necessary requirement in practical applications.

We emphasize that, the used compiler can be applied on any subversion-resistant NIZK argument, e.g. the ones studied in [Fuc18], but we focused on the state-of-the-art zk-SNARK which is proposed by Groth in [Gro16]. From a different perspective, basically we showed that the recent construction proposed by Atapoor and Baghery [AB19] can also achieve Sub-ZK with minimal efficiency loss. The cost is that prover will check the well-formedness of CRS elements by an efficient CRS verification algorithm before using them.

To sum up, we note that as currently Atapoor and Baghery's variation [AB19] of Groth's zk-SNARK is the most efficient simulation-extractable zk-SNARK in the CRS model, so adding Sub-ZK to their scheme will result the most efficient SNARK that can guarantee Sub-ZK and simulation-extractability.

Acknowledgment. This work was supported in part by the Estonian Research Council grant PRG49.

References

AB19. Shahla Atapoor and Karim Baghery. Simulation extractability in Groth's zk-SNARK. In Cristina Perez-Sola, Guillermo Navarro-Arribas, Alex Biryukov, and Joaquin Garcia-Alfaro, editors, *Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2019 International Workshops, DPM 2019 and CBT 2019, Luxembourg, September 26-27, 2019, Proceedings*, volume 11737 of *Lecture Notes in Computer Science*, pages 336–354. Springer, 2019.

ABK18. Benedikt Auerbach, Mihir Bellare, and Eike Kiltz. Public-key encryption resistant to parameter subversion and its realization from efficiently-embeddable groups. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 348–377. Springer, Heidelberg, March 2018.

ABL+19. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, Janno Siim, and Michal Zajac. UC-secure CRS generation for SNARKs. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje-eddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2019 - 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9-11, 2019, Proceedings*, volume 11627 of *Lecture Notes in Computer Science*, pages 99–117. Springer, 2019.

ABLZ17. Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, volume 10626 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2017.

ALSZ18. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. *IACR Cryptology ePrint Archive*, 2018:877, 2018. `http://eprint.iacr.org/2018/877`.

AMV15. Giuseppe Ateniese, Bernardo Magri, and Daniele Venturi. Subversion-resilient signatures: Definitions, constructions and applications. Cryptology ePrint Archive, Report 2015/517, 2015. `http://eprint.iacr.org/2015/517`.

Bag19a. Karim Baghery. On the efficiency of privacy-preserving smart contract systems. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje-eddine Rachidi, editors, *Progress in Cryptology - AFRICACRYPT 2019 - 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9-11, 2019, Proceedings*, volume 11627 of *Lecture Notes in Computer Science*, pages 118–136. Springer, 2019.

Bag19b. Karim Baghery. Subversion-resistant commitment schemes: Definitions and constructions. Cryptology ePrint Archive, Report 2019/1065, 2019. `http://eprint.iacr.org/2019/1065`.

BB08. Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, April 2008.

BBG+13. James Ball, Julian Borger, Glenn Greenwald, et al. Revealed: how us and uk spy agencies defeat internet privacy and security. *The Guardian*, 6:2–8, 2013.

BCG+15. Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure sampling of public parameters for succinct zero knowledge proofs. In *2015 IEEE Symposium on Security and Privacy*, pages 287–304. IEEE Computer Society Press, May 2015.

BCPR14. Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In David B. Shmoys, editor, *46th ACM STOC*, pages 505–514. ACM Press, May / June 2014.

BCTV13. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct non-interactive arguments for a von neumann architecture. Cryptology ePrint Archive, Report 2013/879, 2013. `http://eprint.iacr.org/2013/879`.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, Heidelberg, December 2016.

BG90. Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 194–211. Springer, Heidelberg, August 1990.

BPR14. Mihir Bellare, Kenneth G. Paterson, and Phillip Rogaway. Security of symmetric encryption against mass surveillance. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 1–19. Springer, Heidelberg, August 2014.

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

DDO+01. Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 566–598. Springer, Heidelberg, August 2001.

FO18. Georg Fuchsbauer and Michele Orrù. Non-interactive zaps of knowledge. In *ACNS 18*, LNCS, pages 44–62. Springer, Heidelberg, 2018.

Fuc18. Georg Fuchsbauer. Subversion-zero-knowledge SNARKs. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part I*, volume 10769 of *LNCS*, pages 315–347. Springer, Heidelberg, March 2018.

Gab19. Ariel Gabizon. On the security of the BCTV pinocchio zk-snark variant. *IACR Cryptology ePrint Archive*, 2019:119, 2019.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013.

GM17. Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 581–612. Springer, Heidelberg, August 2017.

GOS06. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006.

Gre14. Glenn Greenwald. *No place to hide: Edward Snowden, the NSA, and the US surveillance state*. Macmillan, 2014.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

Hae19. Rolf Haenni. Swiss post public intrusion test: Undetectable attack againstvote integrity and secrecy `https://e-voting.bfh.ch/app/download/7833162361/PIT2.pdf?t=1552395691`. 2019.

KMS+16. Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos 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.

KZM+15. Ahmed E. Kosba, Zhichao Zhao, Andrew Miller, Yi Qian, T.-H. Hubert Chan, Charalampos Papamanthou, Rafael Pass, Abhi Shelat, and Elaine Shi. CØCØ: A Framework for Building Composable Zero-Knowledge Proofs. Technical Report 2015/1093, November 10, 2015. `http://eprint.iacr.org/2015/1093`, last accessed version from 9 Apr 2017.

Lip12. Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012.

Lip19. Helger Lipmaa. Simulation-extractable SNARKs revisited. Cryptology ePrint Archive, Report 2019/612, 2019. `http://eprint.iacr.org/2019/612`.

LPT19. Sarah Jamie Lewis, Olivier Pereira, and Vanessa Teague. Trapdoor commitments in the swisspost e-voting shuffle proof, `https://people.eng.unimelb.edu.au/vjteague/SwissVote`. 2019.

PHGR13. Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013.

PLS13. Nicole Perlroth, Jeff Larson, and Scott Shane. Nsa able to foil basic safeguards of privacy on web. *The New York Times*, 5, 2013.