Designated Verifier/Prover and Preprocessing NIZKs from Diffie-Hellman Assumptions

Shuichi Katsumata[1,3], Ryo Nishimaki[2]
, Shota Yamada[1], and Takashi Yamakawa[2]

[1] AIST, Tokyo, Japan
yamada-shota@aist.go.jp
[2] NTT Secure Platform Laboratories, Tokyo, Japan
{ryo.nishimaki.zk,takashi.yamakawa.ga}@hco.ntt.co.jp
[3] The University of Tokyo, Tokyo, Japan
shuichi_katsumata@it.k.u-tokyo.ac.jp

Abstract. In a non-interactive zero-knowledge (NIZK) proof, a prover can non-interactively convince a verifier of a statement without revealing any additional information. Thus far, numerous constructions of NIZKs have been provided in the common reference string (CRS) model (CRS-NIZK) from various assumptions, however, it still remains a long standing open problem to construct them from tools such as pairing-free groups or lattices. Recently, Kim and Wu (CRYPTO'18) made great progress regarding this problem and constructed the first lattice-based NIZK in a relaxed model called NIZKs in the preprocessing model (PP-NIZKs). In this model, there is a trusted statement-independent preprocessing phase where secret information are generated for the prover and verifier. Depending on whether those secret information can be made public, PP-NIZK captures CRS-NIZK, designated-verifier NIZK (DV-NIZK), and designated-prover NIZK (DP-NIZK) as special cases. It was left as an open problem by Kim and Wu whether we can construct such NIZKs from weak paring-free group assumptions such as DDH. As a further matter, all constructions of NIZKs from Diffie-Hellman (DH) type assumptions (regardless of whether it is over a paring-free or paring group) require the proof size to have a multiplicative-overhead $|C| \cdot \mathsf{poly}(\kappa)$, where $|C|$ is the size of the circuit that computes the **NP** relation.
In this work, we make progress of constructing (DV, DP, PP)-NIZKs with varying flavors from DH-type assumptions. Our results are summarized as follows:

- DV-NIZKs for **NP** from the CDH assumption over pairing-free groups. This is the first construction of such NIZKs on pairing-free groups and resolves the open problem posed by Kim and Wu (CRYPTO'18).
- DP-NIZKs for **NP** with short proof size from a DH-type assumption over pairing groups. Here, the proof size has an additive-overhead $|C|+\mathsf{poly}(\kappa)$ rather then an multiplicative-overhead $|C|\cdot\mathsf{poly}(\kappa)$. This is the first construction of such NIZKs (including CRS-NIZKs) that does not rely on the LWE assumption, fully-homomorphic encryption, indistinguishability obfuscation, or non-falsifiable assumptions.
- PP-NIZK for **NP** with short proof size from the DDH assumption over pairing-free groups. This is the first PP-NIZK that achieves a

short proof size from a weak and static DH-type assumption such as DDH. Similarly to the above DP-NIZK, the proof size is $|C| + \mathsf{poly}(\kappa)$. This too serves as a solution to the open problem posed by Kim and Wu (CRYPTO'18).

Along the way, we construct two new homomorphic authentication (HomAuth) schemes which may be of independent interest.

1 Introduction

1.1 Background

Zero-knowledge (ZK) proof system [57] is an interactive protocol where a prover convinces the validity of a statement to a verifier without providing any additional knowledge. A non-interactive zero-knowledge (NIZK) proof (or argument[4]) [13] is a ZK proof (or argument) where a prover can generate a proof to the validity of a statement *without* interacting with a verifier. Due to the absence of interaction, NIZKs have found tremendous number of applications in cryptography including (but not limited to) chosen-ciphertext secure public key encryption [76,45,84], group/ring signatures [37,9,81], anonymous credentials [35,40], and multi-party computations (MPC) [55]. Furthermore, aside from its practical interests, due to its theoretically appealing nature, studying the types of assumptions which imply NIZKs has also been an active research area for NIZKs [47,63,12,82]. Below, we briefly review the current state of affairs concerning NIZKs.

NIZKs in the CRS model. It is well known that NIZKs for non-trivial languages do not exist in the plain model where there is no trusted setup [56]. Therefore NIZKs for all of **NP** are constructed either in the common reference string (CRS) model [47] or the random oracle model [48,79]. In the former type of NIZK, the prover and the verifier have access to a CRS generated by a trusted third party (hereafter referred to as CRS-NIZK). Thus far, known constructions of CRS-NIZK for **NP** are based on (doubly-enhanced) trapdoor permutation [47,10,53], pairing [63,64], or indistinguishability obfuscation [86,11,12]. Constructing CRS-NIZKs based on other assumptions such as pairing-free groups and lattices remains to be a long standing open problem.

NIZKs in the designated verifier/prover model. As an alternative line of research, NIZKs in a relaxed model have been considered: *designated verifier* NIZKs (DV-NIZKs) and *designated prover* NIZKs (DP-NIZKs). Both notions of NIZKs retain most of the useful security properties of NIZKs with some relaxation. In DV-NIZKs, anybody can generate a proof, but the proof can only be verified by a designated party in possession of a *verification key*. On the other hand, in DP-NIZKs only a designated party in possession of a *proving key* can generate a proof, but the proof can be verified by anybody. Although the two types of NIZKs are relaxation of CRS-NIZKs, they showed to be no easier to construct. There have

[4] NIZK arguments are a relaxed notion of NIZK proofs where soundness only holds against computationally bounded adversaries. Throughout the introduction, we simply refer to them as NIZKs.

been a long line of work concerning DV-NIZKs [77,42,91,33,74,32,34], however, many of these schemes do not satisfy soundness against multiple theorems, which in brief means that soundness does not hold against a cheating prover given unbounded access to a verification oracle (See Sec. 1.4 for more details). Moreover, DV-NIZKs satisfying soundness against multiple theorems [32,34] are built on tools which are already known to imply CRS-NIZKs. It was not until recently that Kim and Wu [71] in a breakthrough result showed how to construct DP-NIZKs supporting **NP** languages from lattices; this is the first NIZKs for all of **NP** in any model that is based on lattice assumptions. They showed a generic construction of DP-NIZKs from homomorphic signatures (HomSig) and instantiated it with the lattice-based HomSig of [60]. However, despite these recent developments, basing the construction of DV-NIZKs or DP-NIZKs for all of **NP** on pairing-free groups still remains unsolved, and Kim and Wu [71] have stated it as an open problem to construct such NIZKs from the decisional Diffie-Hellman (DDH) assumption.

First Contribution. One of our main contributions is solving this open problem and constructing the first DV-NIZKs from the computational Diffie-Hellman (CDH) assumption over *paring-free groups*. As our scheme is DV-NIZKs and not DP-NIZKs, our techniques depart from [71] and follows more closely to the classical techniques of [47]. More details will be provided in Sec. 1.2.

NIZKs with short proof size. An equally important topic for NIZKs is constructing NIZKs with short proof size. Our construction above solves the open problem of constructing DV or DP-NIZKs from paring-free groups, however, the size of proof is rather large. Namely, it is of size $\mathsf{poly}(\kappa, |C|)$, where κ is the security parameter and $|C|$ is the size of circuit computing the **NP** relation $\mathcal{R}$. In particular, the proof size incurs at least a *multiplicative*-overhead of $O(|C|\kappa)$. As far as we know, the only (CRS, DV, DP)-NIZKs for **NP** in the standard model with a short proof size, i.e., a proof with *additive*-overhead $O(|C|) + \mathsf{poly}(\kappa)$ rather than $O(|C|) \cdot \mathsf{poly}(\kappa)$, either requires a knowledge assumption [62], (fully-)homomorphic encryption (FHE) [52], indistinguishability obfuscation (iO) [86], or HomSig with additional compactness properties [71].[5] Notably, we do not know how to construct (CRS, DV, DP)-NIZKs with short proof size from standard assumptions from paring-free groups. In fact, this is the case even if we were to consider paring groups [25,63,1] as none of the aforementioned heavy machineries are implied from such groups. In other words, it is not known whether DH-type assumptions can be used to construct DV or DP-NIZKs with short proof size.

Second Contribution. Our second contribution is constructing a DP-NIZK *with short proof size* from a DH-type assumption over paring groups by proposing a compact HomSig scheme from a new non-static DH-type assumption (proven to hold in the generic group model) and following the general conversion from HomSig to DP-NIZK by Kim and Wu [71]. More details will be provided in Sec. 1.2.

[5] In fact, as we show in Table 1, all of these approaches lead to a much more succinct proof size of $|w| + \mathsf{poly}(\kappa)$, where w is the witness.

Our second scheme achieves the first DP-NIZK with short proof size from any DH-type assumptions, however, one caveat is that the assumption is *non-static* and rather strong, and furthermore requires *paring groups*. Therefore, desirably we would like to construct any type of NIZKs with short proof size from weaker and *static* assumptions such as the DDH assumption while only requiring *paring-free groups*. To this end, we consider a further relaxation of NIZKs in the *preprocessing model* (hereafter referred to as PP-NIZK). In this model, there is a trusted preprocessing setup that generates a verification *and* proving key, where only those with the proving (resp. verification) key can generate (resp. verify) proofs. Analogously to the history of DV and DP-NIZKs, even with this added relaxation, PP-NIZKs turned out to be a rather difficult primitive to construct. There have been several works concerning PP-NIZKs [43,70,73,41,39,66], however, all of them were only *bounded-theorem* in the sense that either the soundness or zero-knowledge property hold in a bounded manner. The problem of constructing *unbounded-theorem* PP-NIZKs, which meets the standard criteria of NIZK, was only recently resolved in the aforementioned paper [71], where Kim and Wu showed a generic construction of PP-NIZKs using homomorphic MACs (HomMAC). In particular, depending on whether the signature can be verified publicly (HomSig) or not (HomMAC), their generic construction leads to a DP-NIZK or a PP-NIZK, respectively. In fact, it was observed in [71] that using the compact HomMAC proposed by Catalano and Fiore [27] based on the non-static ℓ-computational DH inversion (ℓ-CDHI) assumption [14,21], we can construct PP-NIZKs from a non-static DH-type assumption over paring-free groups. However, they left it as an open problem to construct HomMAC that suffices for PP-NIZKs (with short proof size) from a weaker static assumption such as DDH.

Final Contribution. Our final contribution is constructing a PP-NIZK *with short proof size* from the DDH assumption over *paring-free groups*. We first construct a non-compact HomMAC from the DDH assumption and exploit extra structures in our HomMAC to achieve short proof size when converting it into a PP-NIZK. More details will be provided in Sec. 1.2.

Motivation for studying different types of NIZKs. Although (DV, DP, PP)-NIZKs may be more restricted compared to CRS-NIZKs, they can be useful nonetheless. For example, applications of CRS-NIZKs including group signatures [37,9], anonymous credentials [35,40], electronic cash [36], anonymous authentication [89] may lead to a designated verifier or prover variant by using DV or DP-NIZKs. In some natural scenarios where we do not require public verifiability or require everybody to be able to construct proofs, these alternatives may suffice. Furthermore, as stated in [71], PP-NIZKs can be used instead of CRS-NIZKs to boost semi-honest security to malicious security [55]. Finally, we believe studying different types of NIZKs and understanding which assumptions imply them will provide us with new insights on realizing the long standing open problem of constructing CRS-NIZKs from paring-free groups or lattices.

1.2 Our Results in Detail

As briefly mentioned above, we give new constructions of DV-NIZK, DP-NIZK, and PP-NIZK with different flavors from DH-type assumptions. Our first and third schemes are instantiated on a pairing-free group, and the second scheme requires a pairing group.

1. We construct DV-NIZKs for **NP** from the CDH assumption over pairing-free groups that resists the verifier rejection attack. This is the first construction of such (DV, DP)-NIZK on pairing-free groups and resolves the open problem posed by Kim and Wu [71].
2. We construct DP-NIZKs for **NP** with short proof size from a newly defined non-static (n, m)-computational DH exponent and ratio (CDHER) assumption (proven in the generic group model) over pairing groups. This is the first NIZK in the standard model to achieve a short proof size without assuming the LWE assumption, fully-homomorphic encryption, indistinguishability obfuscation, or non-falsifiable assumptions. The proof size has an additive-overhead $|C| + \mathsf{poly}(\kappa)$ rather then a multiplicative-overhead $|C| \cdot \mathsf{poly}(\kappa)$ where $|C|$ is the size of the circuit that computes the **NP** relation (See Table 1). Moreover, if we make a slight relaxation in the assumption that the **NP** relation is expressed by a "leveled circuit" [20], then the proof size can be made as short as $|w| + |C|/\log\kappa + \mathsf{poly}(\kappa)$ where $|w|$ denotes the witness size. This is the first NIZK (including PP-NIZKs) that achieves *sublinear* proof size in $|C|$. We note that by applying the same technique to the ℓ-CDHI-based construction of PP-NIZK stated in Kim and Wu [71], we can make their proof size sublinear as well, as long as the **NP** relation can be expressed by a leveled circuit.
3. We construct PP-NIZKs for **NP** with short proof size from the DDH assumption over pairing-free groups that are multi-theorem. This is the first PP-NIZK that achieves a short proof size from a weak and static DH-type assumption such as DDH. (In fact, this construction also serves as a solution to the open problem posed by Kim and Wu [71].) Similarly to the above DP-NIZK, the proof size is $|C| + \mathsf{poly}(\kappa)$. Moreover, going through the same technique with additional observations, in case the **NP** relation can be expressed by a leveled circuit, we are able to make the proof size sublinear $|w| + |C|/\log\kappa + \mathsf{poly}(\kappa)$.

Perhaps of an independent interest, along the way to achieve our second result, we propose an HomSig scheme that simultaneously achieves compactness, context-hiding, and online-offline efficiency under the (n, m)-CDHER assumption. This is the first construction of such HomSig schemes on pairing groups.

The comparison table among existing and our NIZK is given in Table 1. We note that we omit schemes that do not support all of **NP**, do not resist the verifier rejection attack, or do not achieve unbounded-theorem soundness or zero-knowledge from the table.

Table 1. Comparison of NIZKs for **NP**.

Reference	Soundness	ZK	Proof size	Model	Assumption
FLS [47]	stat.	comp.	$\mathsf{poly}(\kappa, \lvert C\rvert)$	CRS	trapdoor permutation‡
Groth [62]	stat.	comp.	$\lvert C\rvert \cdot k_{\mathsf{tpm}} \cdot \mathsf{polylog}(\kappa) + \mathsf{poly}(\kappa)$	CRS	trapdoor permutation‡
Groth [62]	stat.	comp.	$\lvert C\rvert \cdot \mathsf{polylog}(\kappa) + \mathsf{poly}(\kappa)$	CRS	Naccache-Stern PKE
GOS [63]	perf.	comp.	$O(\lvert C\rvert\kappa)$	CRS	DLIN/SD
GOS [63]	comp.	perf.	$O(\lvert C\rvert\kappa)$	CRS	DLIN/SD
CHK,DN,Abu [25,46,1]	stat.	comp.	$\mathsf{poly}(\kappa, \lvert C\rvert)$	CRS	CDH
Groth [62]	comp.	perf.	$O(\kappa)$	CRS	q-PKE and q-CPDH
GGIPSS [52]	stat.	comp.	$\lvert w\rvert + \mathsf{poly}(\kappa)$	CRS	FHE and CRS-NIZK
SW [86]	comp.	perf.	$O(\kappa)$	CRS	iO+OWF
KW [71]	stat.*	comp.	$\lvert w\rvert + \mathsf{poly}(\kappa, d)$	DP	LWE
CF+KW [27]+[71]	comp.	comp.	$\lvert C\rvert + \mathsf{poly}(\kappa)$	PP	ℓ-CDHI (pairing-free)
Sec. 3	stat.	comp.	$\mathsf{poly}(\kappa, \lvert C\rvert)$	DV	CDH (pairing-free)
Sec. 4	comp.	comp.	$\lvert C\rvert + \mathsf{poly}(\kappa)$	DP	(n, m)-CDHER
Sec. 4†	comp.	comp.	$\lvert w\rvert + \lvert C\rvert / \log(\kappa) + \mathsf{poly}(\kappa)$	DP	(n, m)-CDHER
Sec. 5	stat.	comp.	$\lvert C\rvert + \mathsf{poly}(\kappa)$	PP	DDH (pairing-free)
Sec. 5†	stat.	comp.	$\lvert w\rvert + \lvert C\rvert / \log(\kappa) + \mathsf{poly}(\kappa)$	PP	DDH (pairing-free)

In column "Soundness" (resp. "ZK"), perf., stat., and comp. means perfect, statistical, and computational soundness (resp. zero-knowledge), respectively. In column "Proof size", κ is the security parameter, $|w|$ is the witness-size, and $|C|$ and d are the size and depth of circuit computing the **NP** relation. In column "Assumption", DLIN stands for the decisional linear assumption, SD stands for the subgroup decision assumption, q-PKE stands for the q-power knowledge of exponent assumption, and q-CPDH stands fo the q-computational power Diffie-Hellman assumption.

* Though their primary construction only has computational soundness, they sketched a variant that achieves statistical soundness in the latest version [72, Remark 4.10]

† Applicable only when C is a leveled circuit.

‡ If the domain of the permutation is not $\{0, 1\}^n$, we further assume they are doubly-enhanced [53].

1.3 Technical Overview

We rely on mainly two approaches to achieve our results. The first approach is an extension of the construction of CRS-NIZKs from trapdoor permutations by Feige, Lapidot, and Shamir [47] (we call it the FLS construction) to the DV setting. The second approach is constructing (DP, PP)-NIZKs using the the Kim-Wu conversion [71] from homomorphic authenticators (HomAuth), where HomAuth are shorthand for HomSig and HomMAC. Specifically, we provide new instantiations of context-hiding HomAuth schemes. Our first result is obtained by the first approach, and the second and third results are obtained by the second approach. In the following, we explain these approaches.

Part 1: DV-NIZK from CDH via FLS paradigm. Our DV-NIZK is based on the Feige-Lapidot-Shamir (FLS) paradigm [47], which enables to construct CRS-NIZKs based on trapdoor permutations (TDP). However, we can not directly use the FLS paradigm since we currently do not know how to achieve TDPs from the CDH assumption. In this study, we present a variant of the FLS construction in the DV setting that can be instantiated by the CDH assumption over paring-free groups.

Our starting point is the CRS-NIZK based on the CDH assumption *over pairing groups* [25,46,1]. The idea is to use a function f_ι defined as follows instead of a TDP for the FLS construction: $f_\iota(X, Z) := X$ if (g, X, Y, Z) is a DH tuple and otherwise $\perp$, where $\iota := (g, Y = g^\tau)$. Though f_ι is not a TDP, it is a trapdoor function (TDF) with a structure that is sufficient for implementing the FLS construction. Below, we take a closer look at the construction.

NIZK in the Hidden Bits Model. Before explaining the construction, we recall the notion of NIZK proof systems in the hidden bits model (hereafter referred to as HBM-NIZK) [47]. In [47], HBM-NIZKs is used as a building block for the final CRS-NIZK. In HBM-NIZK, a prover is provided with a randomly generated string $\rho \xleftarrow{\$} \{0,1\}^\ell$ (referred to as a *hidden random string*) independently from the statement x and witness w for the **NP** language $\mathcal{L}$. Then it generates a proof π_{hbm} along with an index set I indicating the positions in the hidden random string. A verifier given a sub-string $\rho_{|I}$ of the hidden random string ρ on positions corresponding to the index set I along with the statement x and a proof π_{hbm}, either accepts or rejects. Soundness requires that no adversary can generate a valid proof π_{hbm} with an index set I if $x \notin \mathcal{L}$, and the zero-knowledge property requires that a proof provides no additional knowledge to the verifier beyond that $x \in \mathcal{L}$ *if all bits of ρ on positions corresponding to $[\ell] \setminus I$ are hidden* to the verifier. Feige et al. proved that HBM-NIZKs for all of **NP** exist unconditionally.

CRS-NIZK from CDH with pairings We now describe the CRS-NIZK based on the CDH assumption over pairing groups [25,46,1]. We give a direct (high-level) description without using the abstraction by TDFs for clarity.

$\mathsf{Setup}(1^\kappa)$: Output a CRS crs consisting of a group description $(\mathbb{G}, p, g)$ and random group elements $(X_1, ..., X_\ell) \xleftarrow{\$} \mathbb{G}^\ell$ where ℓ is the length of the hidden random string of the underlying HBM-NIZK.

$\mathsf{Prove}(\mathsf{crs}, x, w)$: The prover samples $\tau \xleftarrow{\$} \mathbb{Z}_p$, computes $Z_i := X_i^\tau$ and lets ρ_i be the hardcore bit of Z_i for all $i \in [\ell]$. Then it uses $\rho := \rho_1 \| \cdots \| \rho_\ell$ as a hidden random string to generate a proof π_{hbm} along with an index set $I \subset [\ell]$ by the proving algorithm of the underlying HBM-NIZK on (x, w). It outputs a proof $\pi = (\pi_{\mathsf{hbm}},\ I,\ \{Z_i\}_{i \in I}, Y := g^\tau)$.

$\mathsf{Verify}(\mathsf{crs}, x, \pi)$ Given a statement x and a proof $\pi = (\pi_{\mathsf{hbm}}, I, \{Z_i\}_{i \in I}, Y := g^\tau)$, the verification algorithm verifies (g, X_i, Y, Z_i) is a DH-tuple for all $i \in I$ by using pairing, and rejects if it is not the case. Then it computes the hardcore bit ρ_i of Z_i for all $i \in I$, and verifies π_{hbm} by the verification algorithm of the underlying HBM-NIZK.

Roughly speaking, soundness and zero-knowledge follow from those of the underlying HBM-NIZK since a hidden random string ρ is somehow "committed" in $(X_1, ..., X_\ell)$ once τ is fixed, and only the sub-string of them corresponding to I is revealed to the verifier.[6] Clearly, the above construction relies on pairing to

[6] Though a cheating prover can arbitrarily choose $\tau \in \mathbb{Z}_p$, we can negligibly bound its success probability by the union bound if the success probability of a cheating prover of the underlying HBM-NIZK is bounded by $p^{-1} \cdot \mathsf{negl}(\kappa)$.

check if (g, X_i, Y, Z_i) is a DH-tuple during verification. We note that this check is essential since without it, a cheating prover can arbitrarily choose Z_i for $i \in I$ to control $\rho_{|I}$ to any value, in which case soundness of HBM-NIZK ensures nothing.

Getting rid of pairing. Now, we explain how to get rid of the use of pairing from the above construction in the DV setting. Our main idea is to use the twin-DH technique [26]. Intuitively, the twin-DH technique enables a designated entity to verify whether a tuple $(g, X, Y, Z) \in \mathbb{G}^4$ is a DH-tuple without knowing the discrete logarithm of X or Y and without using pairings, where (g, X) is public, and (Y, Z) may be chosen arbitrarily. More precisely, suppose that an extra element $\widehat{X} := g^{\beta}/X^{\alpha}$ is published in addition to (g, X) where $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$. Then for $Y = g^{\tau}$, we may consider $(Z = X^{\tau}, \widehat{Z} = \widehat{X}^{\tau})$ to be a "proof" that (g, X, Y, Z) is a DH-tuple. Namely, a designated verifier who holds α and β can verify the validity of the "proof" by checking if $Z^{\alpha}\widehat{Z} = Y^{\beta}$ holds. The main implication of the twin-DH technique is that the above verification is essentially equivalent to checking if $Z = X^{\tau}$ and $\widehat{Z} = \widehat{X}^{\tau}$ hold conditioned on the fact that $(Y, Z, \widehat{Z})$ is chosen by a "prover" who does not know $(\alpha . \beta)$.

With this technique in hand, we describe how to modify the above construction to achieve DV-NIZK without pairing: We add extra elements $\widehat{X}_i := g^{\beta_i}/X^{\alpha_i}$ where $\alpha_i, \beta_i \xleftarrow{\$} \mathbb{Z}_p$ for $i \in [\ell]$ in the CRS, give $\{\alpha_i, \beta_i\}_{i \in [\ell]}$ as the verification key to the designated verifier, and add extra elements $\widehat{Z}_i := \widehat{X}_i^{\tau}$ for $i \in I$ in the proof. Then the verifier can verify that (g, X_i, Y, Z_i) is a DH-tuple by checking if $Z^{\alpha_i}\widehat{Z} = Y^{\beta_i}$ holds *without using pairing*. This enables us to achieve DV-NIZK without pairing.

On adaptive zero-knowledge. Though our main idea is as presented above, the above described construction only achieves *non-adaptive zero-knowledge* which requires an adversary to choose the statement x independently of the CRS. To achieve adaptive zero-knowledge, we need to add some extra structures using the technique of non-committing encryption [24,46]. See Sec. 3 for technical details. We note that the original FLS NIZK proof system is also adaptive zero-knowledge, but it uses specific properties of the underlying HBM-NIZK. Though a similar analysis may also yield alternative construction of DV-NIZKs with adaptive zero-knowledge from the CDH assumption without pairing, we choose the above approach where we do not assume any structure on the underlying HBM-NIZK for a conceptually simpler and modular construction.

Part 2: PP-NIZK via context-hiding HomAuth. Kim and Wu [71] showed a conversion from any context-hiding HomAuth scheme to PP-NIZKs. In particular, they noted that context-hiding HomAuth scheme for $\mathbf{NC}^1$ suffices to instantiate their conversion. In this part, we propose new constructions of context-hiding HomAuth schemes for $\mathbf{NC}^1$, and plug them into their conversion. First, we recall the definition of HomAuth. Roughly speaking, a HomAuth scheme is a digital signature or MAC scheme with a homomorphic property. Namely, given a vector of signatures $\boldsymbol{\sigma}$ for a vector of messages $\mathbf{x}$, anyone can publicly evaluate the signature on a circuit C to generate an evaluated signature σ for a message $C(\mathbf{x})$. We say that a HomAuth scheme is a HomSig scheme if verification can be done publicly, and is a HomMAC scheme otherwise. As a security requirement of HomAuth scheme, we require that an adversary given $\mathbf{x}$ cannot generate a pair of an evaluated signature σ^* and a circuit C^* such that σ^* is a valid signature for a message $z \neq C^*(\mathbf{x})$ even if the adversary is given access to a verification oracle. In addition, we say that a HomAuth scheme is context-hiding if σ for a message z generated by evaluating a circuit C on a vector of signatures $\boldsymbol{\sigma}$ for $\mathbf{x}$ does not reveal information of $\mathbf{x}$ beyond that $C(\mathbf{x}) = z$.

In this paper, we propose two new constructions of HomAuth schemes for $\mathbf{NC}^1$. The first one is a HomSig scheme based on a new assumption that we call (n, m)- CDHER assumption on a pairing-group. A nice feature of this HomSig scheme is that the size of an evaluated signature is compact (i.e., does not depend on the message vector length or the circuit to evaluate), and has online-offline efficiency. The second one is a HomMAC scheme based on the DDH assumption on a pairing-free group. The function class the second scheme supports is arithmetic circuits over $\mathbb{Z}_p$ of polynomial degree, which is larger than $\mathbf{NC}^1$, and we take advantage of this extra freedom to improve the proof size. We explain these constructions below.

HomSig from CDHER. Here, we informally explain how an attribute-based encryption (ABE) scheme with some special properties can be converted into a HomSig scheme. Our HomSig scheme from the CDHER assumption can be seen as an instantiation of this conversion.

To explain the idea, we first recall the notion of (key-policy) ABE. In an ABE scheme, one can encrypt a message M with respect to some string $\mathbf{x} \in \{0,1\}^\ell$ using some public parameter pp. Furthermore, a secret key is associated with some policy $C : \{0,1\}^\ell \to \{0,1\}$ and the decryption is possible if and only if $C(\mathbf{x}) = 1$. As for security, we require the selective *one-way* security. In a selective one-way security game, an adversary has to declare its target $\mathbf{x}^\star$ at the beginning of the game before seeing the public parameter pp. An adversary can further query secret keys for C such that $C(\mathbf{x}^\star) = 0$ unbounded polynomially many times throughout the game, and we require that an adversary given an encryption of a *random* message $\mathsf{M}^\star$ under the string $\mathbf{x}^\star$ cannot recover $\mathsf{M}^\star$.

We first observe that the security proofs for most selectively secure schemes such as those proposed in [87,61,92,59,18] can be abstracted in the following manner:[7] At the beginning of the game, the reduction algorithm is given a problem instance Ψ of some hard problem (e.g., the bilinear Diffie-Hellman problem). Then, it first runs the adversary to obtain the target $\mathbf{x}^\star$. Given Ψ and $\mathbf{x}^\star$, the reduction algorithm generates pp along with some simulation trapdoor $\mathsf{td}_{\mathbf{x}^\star}$. The reduction algorithm can perfectly simulate the game using $\mathsf{td}_{\mathbf{x}^\star}$. Namely, given $\mathsf{td}_{\mathbf{x}^\star}$, it can generate correctly distributed secret key sk_C for any C such that $C(\mathbf{x}^\star) = 0$. Furthermore, given $\mathsf{td}_{\mathbf{x}^\star}$, it can embed the problem instance Ψ into

[7] Actually, these previous works prove the standard indistinguishability security notion rather than one-wayness. However, one-wayness is sufficient for our application.

the challenge ciphertext so that it can extract the answer of the hard problem whenever the adversary succeeds in extracting $\mathsf{M}^{\star}$.

Our basic idea for constructing HomSig is to use the above reduction algorithm in the real world. To sign on a message $\mathbf{x}$, we generate $\mathsf{td}_{\mathbf{x}}$ and set $\boldsymbol{\sigma} := \mathsf{td}_{\mathbf{x}}$. To evaluate the signature $\boldsymbol{\sigma}$ on a circuit C such that $C(\mathbf{x}) = 0$, we run the reduction algorithm of the ABE scheme on input $\mathsf{td}_{\mathbf{x}}$ to generate sk_C and set $\sigma := \mathsf{sk}_C$. Here, evaluation of signatures can be done publicly since $\mathsf{td}_{\mathbf{x}}$ is the only secret state required to run the reduction algorithm. A subtle problem with this approach is that we cannot evaluate the signature on a circuit C such that $C(\mathbf{x}) = 1$ since the reduction algorithm does not work for such C. This problem can be easily fixed by defining the scheme so that when evaluating a signature on such C, we generate $\mathsf{sk}_{\neg C}$ instead of sk_C, where $\neg C$ is a circuit that is obtained by flipping the output bit of C by applying the NOT gate. Now, for the signature $\sigma = \mathsf{sk}_C$ to be publicly verifiable, we require it to be possible to efficiently check whether σ is a correctly generated secret key of the ABE given (C, σ). However, this is not such a strong restriction since it is satisfied by many selectively secure ABE schemes such as the ones listed above.

We recall that given $\mathsf{td}_{\mathbf{x}}$, the reduction algorithm can perfectly simulate the selective security game for ABE where $\mathbf{x}$ is the target chosen by the adversary. This in particular implies that sk_C simulated by $\mathsf{td}_{\mathbf{x}}$ follows the same distribution as sk_C generated in the real system which does not use information of $\mathbf{x}$. Then, the context-hiding property of the scheme follows from this fact. Namely, the distribution of $\sigma = \mathsf{sk}_C$ only depends on C and pp, not on $\mathbf{x}$. In other words, σ does not leak any information of $\mathbf{x}$, which meets the requirements of the context-hiding security. Furthermore, the unforgeability of the scheme follows from the one-wayness of the ABE: If the adversary can forge a signature $\sigma = \mathsf{sk}_{C^{\star}}$ for $C^{\star}$ such that $C^{\star}(x) = 1$, then $\mathsf{sk}_{C^{\star}}$ can be used to decrypt the challenge ciphertext, which contradicts the security of the ABE. We note that the circuit class of the allowed homomorphic evaluation for the resulting HomSig scheme is roughly the same as the circuit class supported by the original ABE scheme.

In order to obtain the aforementioned HomSig scheme for $\mathbf{NC}^1$ with compact signatures, we need a key-policy ABE scheme with constant-size secret keys. Unfortunately, the only construction of ABE scheme [7] which meets the efficiency (i.e., compactness) property we require does not conform to our template that uses the simulation trapdoor $\mathsf{td}_{\mathbf{x}}$. Therefore, we construct a new ABE scheme with the required property which conforms to our template based on the CDHER assumption. The structure of our ABE scheme is inspired by the ciphertext-policy ABE scheme with constant-size ciphertexts (*not* secret keys) due to Agrawal and Chase [3]. To turn their scheme into an ABE scheme with constant-size secret keys, at a high level, we swap the ciphertexts and secret keys of their construction. Since the security of the resulting scheme is not guaranteed by that of the original one, we directly prove its security by adding considerable modification to the previous proof techniques [83,4].

HomMAC from DDH. Here, we explain the construction of HomMAC under the DDH assumption. Our idea is to add the context-hiding property to the non-context-hiding HomMAC proposed by Catalano and Fiore [27] by using functional encryption for inner products (IPFE). First, we recall their non-context-hiding HomMAC, which supports all arithmetic circuits of polynomially bounded degree.[8] The signing/verification key of their construction are $\mathbf{r} \in \mathbb{Z}_p^\ell$ and $s \in \mathbb{Z}_p^*$ where ℓ is the arity of arithmetic circuits it supports, and the evaluation key is a prime p. A signature $\boldsymbol{\sigma} \in \mathbb{Z}_p^\ell$ for a message $\mathbf{x} \in \mathbb{Z}_p^\ell$ is set to be $\boldsymbol{\sigma} := (\mathbf{r} - \mathbf{x})s^{-1} \mod p$.[9] Given an arithmetic circuit f of degree D, a message $\mathbf{x}$, and a signature $\boldsymbol{\sigma}$, the evaluation algorithm computes the coefficients $(c_1, ..., c_D) \in \mathbb{Z}_p^D$ that satisfy $f(\mathbf{r}) = f(\mathbf{x}) + \sum_{j=1}^{D} c_j s^j$, and sets $\sigma := (c_1, ..., c_D)$ as an evaluated signature. We remark that this can be done by using $\mathbf{x}$, $\boldsymbol{\sigma}$, and p without knowing $(r_1, ..., r_n)$ or s since the signatures satisfy $s\boldsymbol{\sigma} + \mathbf{x} = \mathbf{r} \mod p$. To verify the evaluated signature, the verifier simply checks if the above equation holds by using $\mathbf{r}$ and s included in the verification key. Though the construction is very simple, the scheme satisfies unforgeability even against unbounded-time adversaries. Unfortunately, this construction cannot yet be used for the purpose of PP-NIZKs, since in general it is not context-hiding.

Here, we observe that in the above construction, what a verifier has to know for the verification is only $\sum_{j=1}^{D} c_j s^j$, and not the entire $(c_1, ..., c_D)$. Moreover, $\sum_{j=1}^{D} c_j s^j$ does not convey any information on $\mathbf{x}$ beyond $f(\mathbf{x})$ because the term is determined solely by $\mathbf{r}$ and $f(\mathbf{x})$. Therefore if there exists a way to only transfer $\sum_{j=1}^{D} c_j s^j$ to the verifier, then context-hiding is guaranteed. We remark that a trivial idea of publishing s does not work because it completely breaks the unforgeability. In particular, we want to find a way to let a verifier only know $\sum_{j=1}^{D} c_j s^j$ without providing s to the evaluator. To solve this problem we rely on IPFE. In an IPFE scheme, both a ciphertext and a secret key are associated with a vector. If we decrypt a ciphertext of a vector $\mathbf{x}$ by a secret key associated with $\mathbf{y}$, then the decryption result is $\langle \mathbf{x}, \mathbf{y} \rangle$, which is an inner product of $\mathbf{x}$ and $\mathbf{y}$. We convert the above non-context-hiding HomMAC to a context-hiding one by using IPFE as follows: In the setup, we additionally generate a public parameter pp and a master secret key msk of IPFE. Then a verifier is provided with a secret key $\mathsf{sk}_{(s,...,s^D)}$ for a vector $(s, ..., s^D)$, and an evaluator is provided with pp. The evaluator sets the evaluated signature to be an encryption ct of $(c_1, ..., c_D)$ instead of $(c_1, , , ., c_D)$ itself. Now, a verifier only learns $\sum_{j=1}^{D} c_j s^j$ due to the security of IPFE, and thus context-hiding is achieved.

Given the above overview, it may seem that any IPFE scheme suffices for the construction. Moreover, since only one secret key is needed in the construction, it seems that one-key IPFE suffices. Since there are constructions of one-key secure FE even for all circuits based on any PKE scheme [85,58], one may think that we can implement the above construction based on any PKE scheme. However, this is in fact not the case because these FE schemes are malleable. Namely, the standard

[8] Though the original construction by Catalano and Fiore [27] is based on PRF, we present an information theoretically secure variant of it in a simplified setting where the arity of an arithmetic circuit is bounded.

[9] Though the scheme is not publicly verifiable, we call $\boldsymbol{\sigma}$ a "signature" for compatibility to HomSig.

security notion of FE does not prevent a malicious encryptor from generating an invalid ciphertext. Put differently, the decryption result may be controlled. In the context of the above construction, the fact that an evaluator generates a ciphertext ct by the secret key $\mathsf{sk}_{(s,\ldots,s^D)}$ that is decrypted to T does not necessarily mean that it knows $(c_1,\ldots.,c_D)$ such that $\sum_{j=1}^{D} c_j s^j = T$. Therefore, although the construction seems to work, we cannot prove unforgeability of the above scheme. To solve this problem, we introduce a notion which we call *extractability* for IPFE. Extractability requires that for any (possibly malformed) ciphertext ct that is decrypted to T with a secret key sk associated with a vector $\mathbf{y}$, we can extract $\mathbf{x}$ such that $\langle \mathbf{x}, \mathbf{y} \rangle = T$ from ct. It is clear that the above problem is resolved if we have an extractable IPFE.

Here, we observe that the IPFE scheme based on the DDH assumption proposed by Agrawal, Libert, and Stehlé [5] satisfies extractability. A subtle problem of their construction is that a decryptor must compute a discrete logarithm for computing a decryption result, and thus the size of the decryption result must be limited to being relatively small. Fortunately, this does not matter in our application since the verification is done by simply checking if a decryption result of IPFE satisfies a certain linear equation which can be performed on the exponent. Concretely, we only need a variant of IPFE that enables a decryptor to learn inner-product on the exponent. Putting all the ideas together, we obtain a context-hiding HomMAC for arithmetic circuits of polynomial degree (which includes $\mathbf{NC}^1$) based on the DDH assumption, which further combined with [71] leads to PP-NIZK proofs based on the DDH assumption. Moreover, we can make the proof size of the PP-NIZK short by incorporating the idea by Katsumata [69]. Namely, the proof size of the resulting PP-NIZK is $|C| + \mathsf{poly}(\kappa)$ where $|C|$ is the size of a circuit that computes a relation to prove. See the full version for details.

PP-NIZK with sublinear proof size. Direct adaptations of the Kim-Wu conversion to compact context-hiding HomAuth for $\mathbf{NC}^1$ yield PP-NIZK with proof sizes $|C| + \mathsf{poly}(\kappa)$. Here, we explain that this can be further reduced to *sublinear size* $|w| + |C|/\log\kappa + \mathsf{poly}(\kappa)$ by making a slight relaxation that a circuit C computing the $\mathbf{NP}$ relation is expressed as a *leveled circuit* [20]; a circuit whose gates are partitioned into $D + 1$ levels and all incoming wires to a gate of level $i + 1$ come from gates of level i for each $i \in [D]$. To explain this, we first briefly review the Kim-Wu conversion. In their construction, a prover is provided with a secret key K of a symmetric key encryption (SKE) scheme as its proving key, and to prove that (x, w) satisfies $C(x, w) = 1$ for a circuit C, it encrypts w by using K to generate a ciphertext ct, and generates an evaluated signature σ on message "1" under the function $f_{\mathsf{ct},x}$ defined by $f_{\mathsf{ct},x}(K') := C(x, \mathsf{Dec}(K', \mathsf{ct}))$ where Dec is the decryption algorithm of the SKE scheme. A proof consists of ct and σ. A verifier simply verifies that the evaluated signature σ is a valid signature on message "1" under the function $f_{\mathsf{ct},x}$. To implement this construction based on HomAuth for $\mathbf{NC}^1$, we have to express a circuit that computes the $\mathbf{NP}$ relation in $\mathbf{NC}^1$. This is in general possible by "expanding" the witness to values corresponding to all wires of $C(x, \cdot)$. However, since the size of the expanded witness is as large as the circuit size $|C|$, the proof size of the resulting PP-NIZK is linear in $|C|$. Now, we observe that we actually need not expand the witness to all wires, and we can choose a portion of them based on a similar idea used in [20]. Namely, for a leveled circuit C of depth D, we divide $[D]$ into $\log \kappa$ intervals of length $D/\log \kappa$, and choose "special levels" i in each interval so that the number of gates of level i is the smallest among those in the interval. Then we set an expanded witness to be the original witness appended by values corresponding to all wires *of special levels* of $C(x, \cdot)$. We observe that the consistency of the expanded witness generated in this way still can be verified in $\mathbf{NC}^1$ since successive special levels are at most $2 \log \kappa$ apart from each other. Moreover, the size of the expanded witness is at most $|w| + |C|/\log \kappa$ since the number of gates of special levels is at most $|C|/\log \kappa$ by the choice of special levels. Thus, by applying the Kim-Wu conversion with the above expanded witness, we obtain PP-NIZK with proof size $|w| + |C|/\log \kappa + \mathsf{poly}(\kappa)$.

1.4 Other Related Works

Concurrent Works. There are two concurrent and independent works [38,80] that contain similar results to our first result, namely, multi-theorem DV-NIZK from CDH assumption in pairing-free groups. We summarize differences of these results below.

- Couteau and Hofheinz [38] additionally give a construction of (CRS,DV)-NIZK assuming the LWE assumption and a (CRS,DV)-non-interactive witness indistinguishable proof system for bounded distance decoding.
- Quach, Rothblum, and Wichs [80] additionally consider a stronger variant of DV-NIZK called malicious DV-NIZK, and construct it based on a stronger assumption called the one-more CDH assumption in pairing-free groups.
- Constructions of (DP,PP)-NIZKs with compact proofs are unique to this paper.

CRS-NIZK from Lattices. Very recently, Peikert and Shiehian [78] constructed the first CRS-NIZKs for **NP** under standard lattice assumptions following the line of researches [68,23,65,22] to instantiate the Fiat-Shamir transform [48] in the standard model.

More discussions on existing (DV, DP, PP)-NIZK. Unlike CRS-NIZKs where proving statements and verifying proofs can be done publicly, in (DV, DP, PP)-NIZKs since we have the notion of secret states, it is not uncommon to have a bound on the number of statements (i.e., theorems) one can prove without compromising soundness or zero-knowledge. In DV-NIZKs, a common issue have been the bound on the number of time the prover can query the verification oracle. Namely, a prover can break the soundness of a DV-NIZK if the verifier uses the same verification key to verify multiple statements. Due to this fact, such DV-NIZKs that require a bound on the number of time a prover can query the verification oracle are called *bounded-theorem*. If the verifier can keep using the same key for multiple statements, then it is called *multi-theorem*. Almost all previous DV-NIZKs for all of **NP** [77,42,91,33,74] suffered from this issue of being bounded-theorem. There are more recent works that avoid the above issue based on a certain type of additively homomorphic encryption [32] or a primitive called oblivious linear-function evaluation [34]. However, instantiating either of these primitives require an assumption that is already known to imply a CRS-NIZK. DP and PP-NIZKs share similar problems, where in this case, zero-knowledge does not hold if the prover uses the same proving key multiple statements. Other than the recent schemes by Kim and Wu [71] and Boyle et al [19], all previous DP or PP-NIZKs [43,70,73,41,39,66] are known to be bounded-theorem. Though it is known that we can convert any bounded theorem NIZK to unbounded theorem NIZK in the CRS setting [47], the conversion heavily relies on the fact that proofs can be generated publicly, and does not seem to work in the PP model. We refer to [71] for more discussions.

Homomorphic authenticators. The notion of homomorphic authenticators (MACs or signatures) originates to Desmedt [44] and was first formalized by Johnson et al. [67]. In the beginning, HomAuth was considered extensively in the context of network coding where the homomorphism were focused on linear functions, yielding a long line of interesting works such as [2,15,50,16,8,17,29,49,31,28]. HomAuth for linear functions has also been considered for proofs of retrievability for outsourced storage [6,88]. Boneh and Freeman [16] were the first to consider homomorphism beyond linear functions, showing the first scheme for polynomial function based on lattices. Since then numerous improvements on HomAuth have been made [29,51,60,27]. Gorbunov et al. [60] constructed a HomSig that supports arbitrary circuits with bounded-depth from lattices and Catalano et al. [27] constructed a HomMAC that supports arbitrary arithmetic circuits with bounded-degree from PRFs or DH-type assumptions.

Recently, Tsabary [90] showed a generic conversion of an attribute-based signature (ABS) to HomSig. Using their construction, we may obtain a HomSig with compact signatures starting from an ABS with short signatures. However, the two ABS schemes with short signatures are not a complete fit for the conversion: The scheme by Attrapadung et al. [7] is only selectively-secure and the above conversion is not applicable. The scheme by [75] is constructed on composite-order groups, which is not desirable from the view points of security and efficiency.

Finally, we also mention that our idea of viewing some types of ABE as HomSig seems to be applicable for other ABE schemes such as [61]. This leads to a context-hiding HomSig scheme from the CDH assumption and thus DP-NIZK from the same assumption via the transformation due to Kim and Wu [71]. In addition, we observe that if we start from the ABE for circuits from lattices due to Boneh et al. [18], we recover the existing HomSig scheme by Gorbunov, Vaikuntanathan, and Wichs [60]. While this is not a new result, the observation provides new insights into the connection between them.

2 Preliminaries

We omit basic notations and knowldege on cryptography due to limited space.

2.1 Preprocessing NIZKs

Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a polynomial time recognizable binary relation. For $(x,w) \in \mathcal{R}$, we call x as the statement and w as the witness. Let $\mathcal{L}$ be the corresponding **NP** language $\mathcal{L} = \{x \mid \exists w \text{ s.t. } (x,w) \in \mathcal{R}\}$. We also write $\mathcal{R}(x,w) \in \{0,1\}$ as the output of the polynomial time decision algorithm $\mathcal{R}$ on input (x,w), where 0 is for reject and 1 is for accept. Below, we define (adaptive multi-theorem) preprocessing NIZKs for **NP** languages. Some discussions on our presentation of NIZKs are provided below.

Definition 2.1 (NIZK Proofs). *A non-interactive zero-knowledge (NIZK) proof in the preprocessing model Π_{PPNIZK} for the relation $\mathcal{R}$ is defined by the following three polynomial time algorithms:*

$\mathsf{Setup}(1^\kappa) \to (\mathsf{crs}, k_{\mathsf{P}}, k_{\mathsf{V}})$*: The setup algorithm takes as input the security parameter 1^κ and outputs a common reference string* crs*, a proving key k_{P}, and a verification key k_{V}. This algorithm is executed as the "preprocessing" step.*

$\mathsf{Prove}(\mathsf{crs}, k_{\mathsf{P}}, x, w) \to \pi$*: The prover's algorithm takes as input a common reference string* crs*, a proving key k_{P}, a statement x, and a witness w and outputs a proof π.*

$\mathsf{Verify}(\mathsf{crs}, k_{\mathsf{V}}, x, \pi) \to \top$ *or* $\bot$*: The verifier's algorithm takes as input a common reference string, a verification key k_{V}, a statement x, and a proof π and outputs $\top$ to indicate acceptance of the proof and $\bot$ otherwise.*

Moreover, an (adaptive multi-theorem) NIZK proof in the preprocessing model Π_{PPNIZK} is required to satisfy the following properties, where the probabilities are taken over the random choice of the algorithms:

Completeness. *For all pairs $(x,w) \in \mathcal{R}$, if we run $(\mathsf{crs}, k_{\mathsf{P}}, k_{\mathsf{V}}) \leftarrow \mathsf{Setup}(1^\kappa)$, then we have*

$$\Pr[\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, k_{\mathsf{P}}, x, w) : \mathsf{Verify}(\mathsf{crs}, k_{\mathsf{V}}, x, \pi) = \top] = 1.$$

Soundness. *For all (possibly inefficient) adversaries $\mathcal{A}$, if we run $(\mathsf{crs}, k_{\mathsf{P}}, k_{\mathsf{V}}) \leftarrow \mathsf{Setup}(1^\kappa)$, then we have*

$$\Pr[(x,\pi) \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs},k_{\mathsf{V}},\cdot,\cdot)}(1^\kappa, \mathsf{crs}, k_{\mathsf{P}}) : x \notin \mathcal{L} \land \mathsf{Verify}(\mathsf{crs}, k_{\mathsf{V}}, x, \pi) = \top] = \mathsf{negl}(\kappa).$$

Here, in case soundness only holds for computationally bounded adversaries $\mathcal{A}$, we say it is a NIZK argument.

(Non-Programmable CRS) Zero-Knowledge. *For all PPT adversaries $\mathcal{A}$, there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that if we run $(\mathsf{crs}, k_{\mathsf{P}}, k_{\mathsf{V}}) \leftarrow \mathsf{Setup}(1^\kappa)$ and $\tau_{\mathsf{V}} \leftarrow \mathcal{S}_1(1^\kappa, \mathsf{crs}, k_{\mathsf{V}})$, then we have*

$$\left| \Pr[\mathcal{A}^{\mathcal{O}_0(\mathsf{crs},k_{\mathsf{P}},\cdot,\cdot)}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\mathsf{crs},k_{\mathsf{V}},\tau_{\mathsf{V}},\cdot,\cdot)}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}}) = 1] \right| = \mathsf{negl}(\kappa),$$

where $\mathcal{O}_0(\mathsf{crs}, k_\mathsf{P}, x, w)$ *outputs* $\mathsf{Prove}(\mathsf{crs}, k_\mathsf{P}, x, w)$ *if* $(x, w) \in \mathcal{R}$ *and* $\bot$ *otherwise, and* $\mathcal{O}_1(\mathsf{crs}, k_\mathsf{V}, \tau_\mathsf{V}, x, w)$ *outputs* $\mathcal{S}_2(\mathsf{crs}, k_\mathsf{V}, \tau_\mathsf{V}, x)$ *if* $(x, w) \in \mathcal{R}$ *and* $\bot$ *otherwise.*

Remark 2.1 (Programmable Zero-Knowledge). As also discussed in [72], we can define a slightly weaker variant of zero-knowledge where the simulator is provided the freedom of programming the common reference string crs and verification key k_V.

(Programmable CRS) Zero-Knowledge For all PPT adversaries $\mathcal{A}$, there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that if we run $(\mathsf{crs}, k_\mathsf{P}, k_\mathsf{V}) \leftarrow \mathsf{Setup}(1^\kappa)$ and $(\overline{\mathsf{crs}}, \bar{k}_\mathsf{V}, \bar{\tau}_\mathsf{V}) \leftarrow \mathcal{S}_1(1^\kappa)$, then we have

$$\left|\Pr[\mathcal{A}^{\mathcal{O}_0(\mathsf{crs}, k_\mathsf{P}, \cdot, \cdot)}(1^\kappa, \mathsf{crs}, k_\mathsf{V}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\bar{\mathsf{crs}}, \bar{k}_\mathsf{V}, \bar{\tau}_\mathsf{V}, \cdot, \cdot)}(1^\kappa, \overline{\mathsf{crs}}, \bar{k}_\mathsf{V}) = 1]\right| = \mathsf{negl}(\kappa),$$

where $\mathcal{O}_0(\mathsf{crs}, k_\mathsf{P}, x, w)$ outputs $\mathsf{Prove}(\mathsf{crs}, k_\mathsf{P}, x, w)$ if $(x, w) \in \mathcal{R}$ and $\bot$ otherwise, and $\mathcal{O}_1(\overline{\mathsf{crs}}, \bar{k}_\mathsf{V}, \bar{\tau}_\mathsf{V}, x, w)$ outputs $\mathcal{S}_2(\overline{\mathsf{crs}}, \bar{k}_\mathsf{V}, \bar{\tau}_\mathsf{V}, x)$ if $(x, w) \in \mathcal{R}$ and $\bot$ otherwise.

This definition captures the zero-knowledge property used in standard NIZKs in the common reference string (CRS) model. In the CRS model, the Setup algorithm outputs a CRS σ used by both the prover and verifier, and the zero-knowledge simulator is allowed to program the CRS σ. Specifically, the proving key and verification key are both set as the CRS σ.

Remark 2.2 (Different types of NIZKs). The definition is general enough to capture many of the existing types of NIZKs. In case $k_\mathsf{P} = k_\mathsf{V} = \bot$, the above definition captures the standard NIZKs in the common reference string (CRS) model, which we refer to as CRS-NIZKs hereafter. Specifically anybody can construct a proof using the public CRS and those proofs are publicly verifiable [47]. On the other hand, in case $k_\mathsf{P} = \bot$ but k_V is required to be kept secret, the above definition captures *designated verifier* NIZKs (DV-NIZKs) [77,42]. Moreover, in case $k_\mathsf{V} = \bot$ but k_P is required to be kept secret, the above definition captures designated prover NIZKs (DP-NIZKs) [71]. Finally, in case both k_P and k_V must be kept secret, it is simply called preprocessing NIZKs (PP-NIZKs) [39].

Remark 2.3 (Bounded and Multi-Theorem NIZK). Unlike CRS-NIZKs where there are nothing to be kept secret, (DV, DP, PP)-NIZKs take more subtle care to construct. Specifically, the latter types of NIZKs may possibly leak secret information when constructing a proof (DP-NIZKs) or verifying a proof (DV-NIZKs). We say the scheme is *bounded-theorem* if the number of statements supported by the scheme to guarantee soundness or zero-knowledge is bounded before setup. Otherwise, we say the scheme is *multi-theorem.* All the NIZKs we construct in this paper are multi-theorem. Finally, we call the scheme *single-theorem* if it only supports one statement.

Remark 2.4 (Adaptive and Non-Adaptive NIZK). One often considers weaker security called *non-adaptive* soundness and zero-knowledge. In non-adaptive soundness, an adversary has to declare the statement x on which he forges a proof before seeing a common reference string. In non-adaptive zero-knowledge, an adversary has to declare a pair of a statement x and its witness w to query the proving oracle before seeing a common reference string. All the NIZKs we construct in this paper satisfy adaptive soundness and zero-knowledge.

NIZKs for Bounded Languages. Throughout this paper, we mainly consider the weaker variant of PP-NIZKs which we call PP-NIZKs *for bounded languages* as was done by Kim and Wu [71]. PP-NIZKs for bounded languages enable one to generate a proof for $(x, w) \in \mathcal{R} \cap (\{0,1\}^{n(\kappa)} \times \{0,1\}^{m(\kappa)})$ for a priori bounded polynomials $n(\cdot)$ and $m(\cdot)$. For clarity, we say PP-NIZKs *for unbounded languages* to express PP-NIZKs that do not have the above limitation. As discussed in the full version, we can generically convert any PP-NIZKs for bounded languages to PP-NIZKs for unbounded languages at the cost of making the proof size larger. However, we note that since the conversion makes the proof size larger, the distinction between PP-NIZKs for bounded and unbounded languages are meaningful if we start to consider proof sizes.

3 DV-NIZK from CDH via FLS Transform

In this section, we construct a DV-NIZK from the CDH assumption over pairing-free groups based on the FLS construction [47] for CRS-NIZKs from TDPs. More formally, we prove the following theorem.

Theorem 3.1. *If the CDH assumption holds on a pairing-free group, then there exists an (adaptive multi-theorem) DV-NIZK proof system for all* ***NP*** *languages.*

The theorem is proven in the following steps:

1. We first construct a variant of DV-NIZK proof system (which we call the *base proof system*) with a special syntax satisfying a relaxed notion of soundness and adaptive *single-theorem* zero-knowledge. We construct it from a NIZK proof system in the hidden-bits model based on the CDH assumption over pairing-free groups. This is done by applying the FLS construction [47] along with the twin-DH technique. A relaxed notion of adaptive zero-knowledge is achieved by using a technique often used in non-committing encryption.
2. We then construct an adaptive *designated-verifier non-interactive witness indistinguishable* (DV-NIWI) proof for all **NP** languages by running many copies of the base proof system in parallel.
3. Finally, we transform our adaptive DV-NIWI proofs into adaptive multi-theorem DV-NIZK proofs by using pseudorandom generators via the transformation of Feige, Lapidot, and Shamir [47] (i.e., the technique of FLS is applicable to the DV-NIZK setting).

3.1 Preliminaries

We introduce the Goldreich-Levin hardcore function $\mathsf{GL}(a; r)$. This is defined by $\mathsf{GL}(a; r) := \langle a, r \rangle := \bigoplus_{j=1}^{u} (a_j \cdot r_j)$ where $a, r \in \{0,1\}^u$ and σ_j denotes the j-th bit of a string σ. In fact, we use groups in our construction and the input to GL is an element in $\mathbb{G}$. Thus, we interpret a group element $g^{r_i} \in \mathbb{G}$ as a u-bit-string.

Theorem 3.2 (Goldreich-Levin Theorem (adapted) [54]). *Assuming that the CDH assumption holds, it holds that*

$$\left|\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{GL\text{-}cdh}}(\kappa, 0) = 1] - \Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{GL\text{-}cdh}}(\kappa, 1) = 1]\right| \leq \mathsf{negl}(\kappa),$$

where the experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{GL\text{-}cdh}}(\kappa, \mathsf{coin})$ *is defined as follows.*

```
$\underline{\mathsf{Expt}_{\mathcal{A}}^{\mathsf{GL\text{-}cdh}}(\kappa, \mathsf{coin})}$
Samples $(\mathbb{G}, p, g) \xleftarrow{\$} \mathsf{GGen}(1^\kappa)$, $R \xleftarrow{\$} \{0,1\}^u$, and $x, y \xleftarrow{\$} \mathbb{Z}_p$.
If $\mathsf{coin} = 1$, then $\rho \xleftarrow{\$} \{0,1\}$, else if $\mathsf{coin} = 0$, then $\rho := \mathsf{GL}(g^{xy}; R)$.
Output $\mathsf{coin}' \leftarrow \mathcal{A}(1^\kappa, \mathbb{G}, p, g, g^x, g^y, R, \rho)$
```

Next, we introduce a theorem called twin-DH trapdoor test which enables one to check if a tuple (g, X, Y, Z) is a DH-tuple without knowing the discrete logarithm of X or Y by using a special trapdoor.

Theorem 3.3 (Twin-DH Trapdoor Test [26]). *For any* $(\mathbb{G}, p, g) \leftarrow \mathsf{GGen}(\kappa)$ *and function F, it holds that*

$$\Pr\left[(Z^\alpha \widehat{Z} \overset{?}{=} Y^\beta) \neq ((Z \overset{?}{=} Y^x) \wedge (\widehat{Z} \overset{?}{=} Y^{\hat{x}})) \;\middle|\; \begin{array}{c} X \xleftarrow{\$} \mathbb{G}, \\ \alpha, \beta \xleftarrow{\$} \mathbb{Z}_p, \;\; \widehat{X} := g^\beta / X^\alpha, \\ (Y, Z, \widehat{Z}) \leftarrow F((\mathbb{G}, p, g), X, \widehat{X}) \end{array} \right] \leq 1/p,$$

where $X = g^x$ and $\widehat{X} = g^{\hat{x}}$.

We introduce the notion of witness indistinguishability.

Definition 3.1 (Adaptive WI (in the DV model)). *We say that a proof system Π satisfies adaptive witness indistinguishability if for all PPT adversaries $\mathcal{A}$ that makes arbitrary number of queries (resp. at most 1 query), if we run* $(\mathsf{crs}, k_{\mathsf{V}}) \leftarrow \mathsf{Setup}(1^\kappa)$*, then we have*

$$\left|\Pr[\mathcal{A}^{\mathcal{O}_0(\mathsf{crs},\cdot,\cdot,\cdot)}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\mathsf{crs},\cdot,\cdot,\cdot)}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}}) = 1]\right| = \mathsf{negl}(\kappa),$$

where $\mathcal{O}_b(\mathsf{crs}, x, w_0, w_1)$ *outputs* $\mathsf{Prove}(\mathsf{crs}, x, w_b)$ *if* $(x, w_0) \in \mathcal{R} \wedge (x, w_1) \in \mathcal{R}$ *and* $\perp$ *otherwise.*

Definition 3.2 (Adaptive NIWI). *We say that a proof system Π is adaptive designated-verifier non-interactive witness indistinguishable proof system if Π satisfies completeness, soundness in Definition 2.1 (in the designated-verifier model), and adaptive witness indistinguishability in Definition 3.1.*

We then formally define a NIZK proof in the hidden-bits model, which will be used as a building block in our construction.

Definition 3.3. *A NIZK proof in the hidden-bits model (HBM) for $\mathcal{L}$ is defined by the following two polynomial time algorithms:*

$\mathsf{Prove}(1^\kappa, x, w, \rho) \rightarrow (\pi, I)$: *The prover's algorithm takes as input the security parameter 1^κ, a statement x, a witness w, and a hidden random string $\rho \in \{0,1\}^{\ell_{\mathsf{hrs}}(\kappa)}$, and outputs a proof π and a set of indices $I \subseteq [\ell_{\mathsf{hrs}}(\kappa)]$ where $\ell_{\mathsf{hrs}}(\cdot)$ is a polynomial of κ.*

$\mathsf{Verify}(1^\kappa, x, \pi, I, \rho_{|I}) \rightarrow \top$ *or* $\bot$: *The verifier's algorithm takes as input the security parameter, a statement x, a proof π, an index set I, a substring $\rho_{|I} := \{\rho_i\}_{i \in I}$, where ρ_i is the i-th bit of ρ, and outputs $\top$ to indicate acceptance of the proof and $\bot$ otherwise.*

Completeness. *For all $x \in \mathcal{L}$ and w such that $(x, w) \in \mathcal{R}$, we have*

$$\Pr[\rho \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{hrs}}(\kappa)}, (\pi, I) \leftarrow \mathsf{Prove}(1^\kappa, x, w, \rho) : \mathsf{Verify}(1^\kappa, x, \pi, I, \rho_{|I}) = \top] = 1.$$

Soundness. *For all (possibly inefficient) adversaries $\mathcal{A}$, we have*

$$\epsilon_{\mathsf{HBM}} := \Pr[\rho \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{hrs}}(\kappa)}, (x, \pi, I) \leftarrow \mathcal{A}(1^\kappa, \rho) : x \notin \mathcal{L} \land \mathsf{Verify}(1^\kappa, x, \pi, I, \rho_{|I}) = \top] = \mathsf{negl}(\kappa).$$

We call ϵ_{HBM} soundness error.

Zero-Knowledge. *There exists a PPT simulator $\mathcal{S}$ such that for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we have*

$$\Big|\Pr[(x, w) \leftarrow \mathcal{A}_1(1^\kappa), \rho \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{hrs}}(\kappa)}, (\pi, I) \leftarrow \mathsf{Prove}(1^\kappa, x, w, \rho) : \mathcal{A}_2(x, \pi, I, \rho_{|I}) = 1]$$
$$- \Pr[(x, w) \leftarrow \mathcal{A}_1(1^\kappa), (\pi, I, \rho_{|I}) \leftarrow \mathcal{S}(1^\kappa, x) : \mathcal{A}_2(x, \pi, I, \rho_{|I}) = 1]\Big| = \mathsf{negl}(\kappa).$$

Theorem 3.4 (NIZK for all NP languages in the HBM [47]). *Unconditionally, there exists NIZK proof systems for all **NP** languages in the HBM with soundness error $\epsilon_{\mathsf{HBM}} \leq 2^{-cn\kappa}$ where $c > 1$ is a constant, n is polynomially related to the size of the circuit computing the **NP** language, κ is the security parameter, and $\ell_{\mathsf{hrs}} = \mathsf{poly}(\kappa, n)$.*

3.2 Constructing DV-NIWI

The goal of this subsection is proving the following theorem.

Theorem 3.5. *Assume that the CDH assumption over paring-free group holds, then there exists an adaptive DV-NIWI for all **NP** languages.*

Here, we sketch our high-level construction. First, we present our so-called base proof system bP, and then convert it into an adaptive DV-NIWI proof system. Here, the base proof system bP is *not* a standard DV-NIWI proof system since it has a slightly different syntax. Namely, the proving and verification algorithms of the base proof system take an auxiliary string s as input in addition to (crs, x, w) and $(\mathsf{crs}, k_{\mathsf{V}}, x, \pi)$, respectively. We show that the base proof system satisfies two properties called *relaxed* soundness, which means that an adversary cannot forge a proof *if s is fixed*, and *relaxed* zero-knowledge, which means that a proof can be simulated without a witness *if s is randomly chosen.* Observe that if we were to convert the prover to sample s on its own and include it in the proof, then the syntax fits that of DV-NIWI. However, such a simple conversion of our base proof system bP into a DV-NIWI will not work as the acquired DV-NIWI will not have soundness. Namely, the relaxed soundness of bP does not prevent a cheating prover from forging a proof if he is allowed to choose s himself. To resolve this problem, we use a similar idea used by Dwork and Naor [46]. Our construction of an adaptive DV-NIWI proof system consists of running many copies of the base proof system using a single common auxiliary input s for all copies. Then, when the number of copies is sufficiently large, soundness of the scheme can be proven from the union bound on all possible s. Moreover, since the relaxed zero-knowledge implies witness indistinguishability, and witness indistinguishability is preserved under parallel repetitions, we can prove the witness indistinguishability of our DV-NIWI.

Base proof system. First, we introduce the syntax and security properties of the base proof system bP. Note that bP is merely an intermediate system introduced for a modular exposition and not a standard NIZK proof system.

Definition 3.4 (Syntax of base proof system). *A base proof system* bP *consists of the following three polynomial time algorithms.*

$\mathsf{bP.Setup}(1^\kappa) \to (\mathsf{crs}, k_\mathsf{V})$*: The setup algorithm takes as input the security parameter 1^κ and outputs a common reference string* crs*, and a verification key k_V.*

$\mathsf{bP.Prove}(\mathsf{crs}, x, w, s) \to \pi$*: The prover's algorithm takes as input a common reference string* crs*, a statement x, a witness w, and a fixed string $s \in \{0,1\}^{\ell_\mathsf{hrs}(\kappa)}$, and outputs a proof π.*

$\mathsf{bP.Verify}(\mathsf{crs}, k_\mathsf{V}, x, \pi, s) \to \top$ *or* $\bot$*: The verifier's algorithm takes as input a common reference string* crs*, a verification key k_V, a statement x, a proof π, and a fixed string $s \in \{0,1\}^{\ell_\mathsf{hrs}(\kappa)}$, and outputs $\top$ to indicate acceptance of the proof and $\bot$ otherwise.*

Definition 3.5 (Security of base proof system). *A base proof system is required to satisfy the following three properties.*

Correctness: *For all pairs $(x, w) \in \mathcal{R}$ and $s \in \{0,1\}^{\ell_\mathsf{hrs}(\kappa)}$, if we run* $(\mathsf{crs}, k_\mathsf{V}) \xleftarrow{\$} \mathsf{bP.Setup}(1^\kappa)$*, then we have*

$$\Pr[\pi \xleftarrow{\$} \mathsf{bP.Prove}(\mathsf{crs}, x, w, s) : \mathsf{bP.Verify}(\mathsf{crs}, k_\mathsf{V}, x, \pi, s) = \top] = 1$$

Relaxed ϵ-soundness: *For any fixed $s \in \{0,1\}^{\ell_\mathsf{hrs}}$, it holds that all (possibly inefficient) adversaries $\mathcal{A}$,*

$$\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{r\text{-}snd}}(1^\kappa, s) = \top] < \epsilon,$$

where ϵ is the soundness error of bP*, and the experiment* $\mathsf{Expt}^{\mathsf{r\text{-}snd}}_{\mathcal{A}}(1^\kappa)$ *is defined as follows.*

```
$\underline{\mathsf{Expt}^{\mathsf{r\text{-}snd}}_{\mathcal{A}}(1^\kappa, s)}$
$(\mathsf{crs}, k_{\mathsf{V}}) \leftarrow \mathsf{bP.Setup}(1^\kappa)$,
$(x^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{bP.Verify}(\mathsf{crs}, k_{\mathsf{V}}, \cdot, \cdot, s)}(1^\kappa, \mathsf{crs}, s)$,
If $x^* \notin \mathcal{L} \wedge \mathsf{bP.Verify}(\mathsf{crs}, k_{\mathsf{V}}, x^*, \pi^*, s) = \top$, then outputs $1$,
Otherwise, outputs $0$.
```

This is basically the same as the standard soundness except that $\mathcal{A}$ must use a fixed s.

Relaxed zero-knowledge: *There exists a PPT simulation algorithm* $\mathsf{bP}.\mathcal{S} = (\mathsf{bP}.\mathcal{S}_1, \mathsf{bP}.\mathcal{S}_2)$ *that satisfies the following. For all (stateful) PPT adversaries $\mathcal{A}$, we have*

$$\left| \Pr[\mathsf{Expt}^{\mathsf{r\text{-}real}}_{\mathcal{A}}(1^\kappa) = 1] - \Pr[\mathsf{Expt}^{\mathsf{r\text{-}sim}}_{\mathcal{A},\mathcal{S}}(1^\kappa) = 1] \right| = \mathsf{negl}(\kappa),$$

where experiments $\mathsf{Expt}^{\mathsf{r\text{-}real}}_{\mathcal{A}}$ *and* $\mathsf{Expt}^{\mathsf{r\text{-}sim}}_{\mathcal{A},\mathcal{S}}$ *are defined as follows.*

```
$\underline{\mathsf{Expt}^{\mathsf{r\text{-}real}}_{\mathcal{A}}}$
$(\mathsf{crs}, k_{\mathsf{V}}) \leftarrow \mathsf{bP.Setup}(1^\kappa)$,
$(x, w) \leftarrow \mathcal{A}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}})$,
$s \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{hrs}}}$,
If $(x, w) \in \mathcal{R}$, $\pi \leftarrow \mathsf{bP.Prove}(\mathsf{crs}, x, w, s)$,
otherwise $\pi := \bot$,
$b' \leftarrow \mathcal{A}(\pi, s)$
outputs $b'$
```

```
$\underline{\mathsf{Expt}^{\mathsf{r\text{-}sim}}_{\mathcal{A},\mathcal{S}}}$
$(\mathsf{crs}, k_{\mathsf{V}}, \tau_{\mathsf{V}}) \leftarrow \mathsf{bP}.\mathcal{S}_1(1^\kappa)$,
$(x, w) \leftarrow \mathcal{A}(1^\kappa, \mathsf{crs}, k_{\mathsf{V}})$,

If $(x, w) \in \mathcal{R}$, $(\pi, s) \leftarrow \mathsf{bP}.\mathcal{S}_2(\mathsf{crs}, k_{\mathsf{V}}, \tau_{\mathsf{V}}, x)$,
otherwise $\pi := \bot$,
$b' \leftarrow \mathcal{A}(\pi, s)$
outputs $b'$
```

We present a base proof system $\mathsf{bP} := (\mathsf{bP.Setup}, \mathsf{bP.Prove}, \mathsf{bP.Verify})$ based on a NIZK proof system in the HBM $(\mathsf{HBM.Prove}, \mathsf{HBM.Verify})$ (with hidden-random-string-length $\ell_{\mathsf{hrs}}(\kappa)$) and the CDH assumption. Note that we use the $\mathsf{GGen}(1^\kappa)$ algorithm to generate $(\mathbb{G}, p, g)$ where $2^{2\kappa} \leq p$ throughout Sec. 3. Hereafter, we simply write ℓ_{hrs} instead of $\ell_{\mathsf{hrs}}(\kappa)$ for ease of notation.

$\mathsf{bP.Setup}(1^\kappa)$**:** This algorithm generates the following parameters.

1. Samples $(\mathbb{G}, p, g) \xleftarrow{\$} \mathsf{GGen}(1^\kappa)$.
2. Samples $(\alpha_{i,b}, \beta_{i,b}) \xleftarrow{\$} \mathbb{Z}_p^2$ for all $i \in [\ell_{\mathsf{hrs}}]$ and $b \in \{0,1\}$ and a common reference string $\overline{\mathsf{crs}} := \{X_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}} \xleftarrow{\$} \mathbb{G}^{2\ell_{\mathsf{hrs}}}$ uniformly at random.
3. Sets $\widehat{\mathsf{crs}} := \{\widehat{X}_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}} := \{X_{i,b}^{-\alpha_{i,b}} \cdot g^{\beta_{i,b}}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$.
4. Samples $R_i \xleftarrow{\$} \{0,1\}^u$ for all $i \in [\ell_{\mathsf{hrs}}]$ and sets $\overline{R} := \{R_i\}_{i \in [\ell_{\mathsf{hrs}}]}$.
5. Outputs a common reference string $\mathsf{crs} := (\mathbb{G}, p, g) \| \overline{\mathsf{crs}} \| \widehat{\mathsf{crs}} \| \overline{R}$ and a verification key $k_{\mathsf{V}} := \{(\alpha_{i,b}, \beta_{i,b})\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$.

We can interpret crs as $(\{X_{i,b}, \widehat{X}_{i,b}, R_i\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}) \in \mathbb{G}^{4\ell_{\mathsf{hrs}}} \times \{0,1\}^{\ell_{\mathsf{hrs}} u}$, where u is the length of the binary representation of a group element.

$\mathsf{bP.Prove}(\mathsf{crs}, x, w, s)$**:** This algorithm does the following.

1. Parses $\mathsf{crs} = (\mathbb{G}, p, g) \| \overline{\mathsf{crs}} \| \widehat{\mathsf{crs}} \| \overline{R}$ where $\overline{\mathsf{crs}} = \{X_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$, $\widehat{\mathsf{crs}} = \{\widehat{X}_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$, $\overline{R} = \{R_i\}_{i \in [\ell_{\mathsf{hrs}}]}$, and $s \in \{0,1\}^{\ell_{\mathsf{hrs}}}$.

2. Samples $\tau \xleftarrow{\$} \mathbb{Z}_p$.
3. Sets $Z_i := (X_{i,s_i})^\tau$ and $\widehat{Z}_i := (\widehat{X}_{i,s_i})^\tau$ and $\rho_i = \mathsf{GL}(Z_i; R_i)$ for $i \in [\ell_{\mathsf{hrs}}]$.
4. Generates $(\pi_{\mathsf{hbm}}, I) \leftarrow \mathsf{HBM.Prove}(1^\kappa, x, w, \rho)$ where $\rho := \rho_1 \| \cdots \| \rho_{\ell_{\mathsf{hrs}}}$.
5. Outputs a proof $\pi := (\pi_{\mathsf{hbm}}, I, \{(Z_i, \widehat{Z}_i)\}_{i \in I}, g^\tau)$.

$\mathsf{bP.Verify}(\mathsf{crs}, k_{\mathsf{V}}, x, \pi, s)$**:** This algorithm parses $\pi = (\pi_{\mathsf{hbm}}, I, \{(Z_i, \widehat{Z}_{,i})\}_{i \in I}, T)$, $k_{\mathsf{V}} := \{(\alpha_{i,b}, \beta_{i,b})\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$, $\mathsf{crs} = (\mathbb{G}, p, g) \| \overline{\mathsf{crs}} \| \widehat{\mathsf{crs}} \| \overline{R}$ where $\overline{\mathsf{crs}} = \{X_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$, $\widehat{\mathsf{crs}} = \{\widehat{X}_{i,b}\}_{i \in [\ell_{\mathsf{hrs}}], b \in \{0,1\}}$, $\overline{R} = \{R_i\}_{i \in [\ell_{\mathsf{hrs}}]}$, and $s \in \{0,1\}^{\ell_{\mathsf{hrs}}}$. This algorithm does the following.

- For all $i \in I$,
  1. Verifies that $\mathsf{Test}_{\mathsf{TDH}}((\alpha_{i,s_i}, \beta_{i,s_i}), X_{i,s_i}, \widehat{X}_{i,s_i}, T, Z_i, \widehat{Z}_i) = \top$, where $\mathsf{Test}_{\mathsf{TDH}}$ is defined in Figure 1. If any one of the equations does not hold, then outputs $\bot$.
  2. Computes $\rho_i = \mathsf{GL}(Z_i; R_i)$.
- If the proof passes all the tests above, then this algorithm outputs $\mathsf{HBM.Verify}(1^\kappa, x, \pi_{\mathsf{hbm}}, I, \rho_{|I})$.

The trapdoor test $\mathsf{Test}_{\mathsf{TDH}}((\alpha, \beta), X, \widehat{X}, Y, Z, \widehat{Z})$

1. Verifies that $Z^\alpha \cdot \widehat{Z} = Y^\beta$. If it holds, then outputs $\top$, else $\bot$.

Fig. 1. The algorithm $\mathsf{Test}_{\mathsf{TDH}}((\alpha, \beta), X, \widehat{X}, Y, Z, \widehat{Z})$ verifies that $Z = Y^x$ and $\widehat{Z} = Y^{\widehat{x}}$, that is (g, Y, X, Z) and $(g, Y, \widehat{X}, \widehat{Z})$ where $X = g^x$ and $\widehat{X} = g^{\widehat{x}}$ are DDH-tuples without $(x, \widehat{x})$.

Unlike the idea outlined in the introduction, the CRS in bP consists of a doubled-line of random elements $(X_{i,0}, \widehat{X}_{i,0})$ and $(X_{i,1}, \widehat{X}_{i,1})$ for each $i \in [\ell_{\mathsf{hrs}}]$. These doubled-line of random elements are crucial for achieving *adaptive* zero-knowledge. If we only had a singled-line of random elements as the CRS in the introduction, then we would have the following issue: The only way for the ZK-simulator of bP $\mathcal{S}_{\mathsf{bP}}$ to use the ZK-simulator $\mathcal{S}_{\mathsf{hbm}}$ of the NIZK in the HBM, is to feed $\mathcal{S}_{\mathsf{hbm}}$ the statement x output by the adversary. Now, for the simulated proof π, index set I, and hidden bits $\rho_{|I}$ output by $\mathcal{S}_{\mathsf{hbm}}$ to be useful, we must have $\rho_i = \mathsf{GL}(X_i^\tau; R_i)$ for all $i \in I$ where τ is some element simulated by $\mathcal{S}_{\mathsf{bP}}$. However, due to soundness, if the CRS was only a single-line of random elements $(X_i, \widehat{X}_i)$, then there exists no τ with overwhelming probability such that the above condition holds. Therefore, $\mathcal{S}_{\mathsf{bP}}$ must choose τ and program the singled-line of random elements $(X_i, \widehat{X}_i)$ in the CRS conditioned on $\rho_i = \mathsf{GL}(X_i^\tau; R_i)$ for all $i \in I$ in order to appropriately use $\mathcal{S}_{\mathsf{hbm}}$. However, since ρ_i is only output as the result of feeding $\mathcal{S}_{\mathsf{hbm}}$ with the statement x, $\mathcal{S}_{\mathsf{bP}}$ can only set the CRS *after* it is given the statement x from the adversary. To overcome this problem, we use the technique of non-committing encryption. Namely, we let CRS be a doubled-line of random elements $(X_{i,0}, \widehat{X}_{i,0})$ and $(X_{i,1}, \widehat{X}_{i,1})$. In the real-scheme the fixed string $s \in \{0,1\}^{\ell_{\mathsf{hrs}}}$ dictates which ℓ_{hrs}-random elements $(X_{i,s_i}, \widehat{X}_{i,s_i})_{i \in [\ell_{\mathsf{hrs}}]}$ a prover must use. Then during the adaptive ZK proof, $\mathcal{S}_{\mathsf{bP}}$ will prepare the CRS so that $\{\mathsf{GL}(X^{\tau}_{i,0}; R_i), \mathsf{GL}(X^{\tau}_{i,1}; R_i)\} = \{0,1\}$ *without* seeing the statement x. Then after the adversary outputs the statement x, it runs $\mathcal{S}_{\mathsf{hbm}}$, and samples a string s so that $\rho_i = \mathsf{GL}(X^{\tau}_{i,s_i}; R_i)$ for all $i \in I$.

Security of bP. The following lemmas address the correctness and security of our base proof system. Due to limited space the proof will appear in the full version.

Lemma 3.1 (Correctness). *Our base proof system* bP *satisfies the correctness in Definition 3.5.*

Lemma 3.2 (Relaxed Soundness). *If* HBM *is sound, then* bP *satisfies the relaxed $(p \cdot \epsilon_{\mathsf{HBM}} + (q_v + 1)/p)$-soundness defined in Definition 3.5.*

Lemma 3.3 (Relaxed ZK). *If the CDH assumption over pairing-free group holds, then* bP *satisfies the relaxed ZK defined in Definition 3.5.*

Construction of DV-NIWI. Here, we present our adaptive DV-NIWI proof system $\Pi := (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ based on the base proof system $\mathsf{bP} := (\mathsf{bP.Setup}, \mathsf{bP.Prove}, \mathsf{bP.Verify})$ that has relaxed ϵ-soundness for some $\epsilon < 1$. We note that we proved that the base proof system satisfies relaxed $(p \cdot \epsilon_{\mathsf{HBM}} + (q_v + 1)/p)$-soundness in Lemma 3.2, and we can make $(p \cdot \epsilon_{\mathsf{HBM}} + (q_v + 1)/p) < 1$ by choosing a parameter for HBM so that $p \cdot \epsilon_{\mathsf{HBM}}$ is negligible. (This is possible by Theorem 3.4). We set an integer ℓ' so that we have $2^{\ell_{\mathsf{hrs}}} \cdot \epsilon^{\ell'} \le 2^{-\kappa}$. Then Π is described as follows.

$\mathsf{Setup}(1^{\kappa})$**:** This algorithm samples $(\mathsf{crs}_j, k_{\mathsf{V}}^{(j)}) \leftarrow \mathsf{bP.Setup}(1^{\kappa})$ for $j \in [\ell']$. It sets $\mathsf{crs} := \mathsf{crs}_1 \| \cdots \| \mathsf{crs}_{\ell'}$ and $k_{\mathsf{V}} := k_{\mathsf{V}}^{(1)} \| \cdots \| k_{\mathsf{V}}^{(\ell')}$, and outputs $(\mathsf{crs}, k_{\mathsf{V}})$.

$\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$**:** This algorithm does the following:

1. chooses $s \xleftarrow{\$} \{0,1\}^{\ell_{\mathsf{hrs}}}$,
2. generates $\pi_j \leftarrow \mathsf{bP.Prove}(\mathsf{crs}_j, x, w, s)$ for all $j \in [\ell']$,
3. outputs a proof $\pi := (\pi_1, \ldots, \pi_{\ell'}, s)$.

$\mathsf{Verify}(\mathsf{crs}, k_{\mathsf{V}}, x, \pi) \to \top$ **or** $\bot$**:** This algorithm parses $\pi = (\pi_1, \ldots, \pi_{\ell'}, s)$. For all $j \in [\ell']$, it verifies that $\top = \mathsf{bP.Verify}(\mathsf{crs}_j, k_{\mathsf{V}}^{(j)}, x, \pi_j, s)$. If the proof passes all the tests, then this algorithm outputs $\top$, otherwise $\bot$.

Our adaptive DV-NIWI proof system Π is complete, sound, and adaptively witness-indistinguishable. The proofs can be found in the full version.

3.3 Transformation from DV-NIWI into Multi-Theorem DV-NIZK

To complete the proof of Theorem 3.1, it remains to show the following theorem.

Theorem 3.6. *If there exists an adaptive DV-NIWI proof systems for all **NP** languages and pseudorandom generators, then there exists an adaptive multi-theorem DV-NIZK proof system for all **NP** languages.*

We omit the proof since the transformation is essentially the same as that of Feige et al. [47] (from NIWI to multi-theorem NIZK), with the exception that we consider the *designated-verifier* setting.

4 Constructing HomSig from ABE-Simulation Paradigm

We construct a context-hiding HomSig for $\mathbf{NC}^1$ from a new non-static (q-type) assumption on pairing groups that we call the CDHER assumption. Specifically, we first construct a new ABE scheme from the same assumption and then apply the (semi-generic) conversion sketched in Sec. 1.2. We directly give a construction of HomSig instead of constructing it via the new ABE. Using the transformation by Kim and Wu [71], we obtain a DP-NIZK from the same assumption.

Theorem 4.1. *If the CDHER assumption holds on a pairing group, then there exists DP-NIZK for all* $\boldsymbol{NP}$ *languages with proof size* $|C| + \mathsf{poly}(\kappa)$*, where* $|C|$ *denotes the size of the circuit that computes the relation being proved.*

As far as we know, this is the first DP-NIZK scheme with short proofs without assuming the LWE assumption, fully-homomorphic encryption, indistinguishability obfuscation, or non-falsifiable assumptions. Furthermore, if the proven $\mathbf{NP}$ relation can be expressed as a leveled circuit, we can reduce the proof size to $|w| + |C|/\log \kappa + \mathsf{poly}(\kappa)$, where $|w|$ is the length of the witness of the proven relation and a leveled circuit refers to a circuit whose gates can be divided into layers and only gates from the consecutive layers are connected by wires. See the full version for the details.

Besides being a building-block for PP-NIZKs, our HomSig scheme alone may be of an independent interest. In the full version, we extend the scheme to the multi-data setting and demonstrate that it achieves online-offline efficiency. This greatly improves the HomSig scheme with the same properties from the multi-linear map [30] in terms of efficiency and security.

5 HomMAC from Inner Product Functional Encryption

In this section, we give a construction of HomMAC based on a variant of functional encryption for inner-products (IPFE) which we call a *functional encryption for inner-product on exponent* (expIPFE). Namely, we show that an expIPFE scheme that satisfies a property called extractability suffices for constructing statistically unforgeable and computationally context-hiding HomMAC. We also show that the IPFE scheme by Agrawal et al. [5] can be seen as an instantiation of an extractable expIPFE scheme under the DDH assumption. As a result, we obtain a statistically unforgeable and computationally context-hiding HomMAC based on the DDH assumption, which yields statistically sound and computationally (non-programmable CRS) zero-knowledge PP-NIZK based on the DDH assumption (over paring-free groups). Since our HomMAC is not compact, a simple adaptation of their transformation yields PP-NIZK with proof size $O(|C|\kappa) + \mathsf{poly}(\kappa)$. However, by taking advantage of the fact our scheme can deal with arithmetic circuits over $\mathbb{Z}_p$ of polynomial degree, which is larger than $\mathbf{NC}^1$, and incorporating the technique by Katsumata [69], we can reduce the proof size to $|C| + \mathsf{poly}(\kappa)$. See the full version for details. Then we obtain the following theorem.

Theorem 5.1. *If the DDH assumption holds on a pairing free group, then there exists PP-NIZK for all* **NP** *languages with proof size* $|C| + \mathsf{poly}(\kappa)$*, where* $|C|$ *denotes the size of circuit that computes the relation being proved.*

Similarly to the case in Sec. 4, if the proven **NP** relation can be expressed as a leveled circuit, we can further reduce the proof size to $|w| + |C|/\log\kappa + \mathsf{poly}(\kappa)$. See the full version for the details.

Acknowledgement

We would like to thank Geoffroy Couteau for helpful comments on related works and anonymous reviewers of Eurocrypt 2019 for their valuable comments. The first author was partially supported by JST CREST Grant Number JPMJCR1302 and JSPS KAKENHI Grant Number 17J05603. The third author was supported by JST CREST Grant No. JPMJCR1688 and JSPS KAKENHI Grant Number 16K16068.

References

1. H. Abusalah. Generic instantiations of the hidden bits model for non-interactive zero-knowledge proofs for NP, 2013. Master's thesis, RWTH-Aachen University.
2. S. Agrawal and D. Boneh. Homomorphic MACs: MAC-based integrity for network coding. In M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, editors, *ACNS 09*, volume 5536 of *LNCS*, pages 292–305. Springer, Heidelberg, June 2009.
3. S. Agrawal and M. Chase. A study of pair encodings: Predicate encryption in prime order groups. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 259–288. Springer, Heidelberg, Jan. 2016.
4. S. Agrawal and M. Chase. Simplifying design and analysis of complex predicate encryption schemes. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Heidelberg, Apr. / May 2017.
5. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 333–362. Springer, Heidelberg, Aug. 2016.
6. G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. Song. Provable data possession at untrusted stores. In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *ACM CCS 2007*, pages 598–609. ACM Press, Oct. 2007.
7. N. Attrapadung, G. Hanaoka, and S. Yamada. Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 575–601. Springer, Heidelberg, Nov. / Dec. 2015.
8. N. Attrapadung and B. Libert. Homomorphic network coding signatures in the standard model. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 17–34. Springer, Heidelberg, Mar. 2011.

9. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, Heidelberg, May 2003.
10. M. Bellare and M. Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *Journal of Cryptology*, 9(3):149–166, June 1996.
11. N. Bitansky and O. Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, Heidelberg, Mar. 2015.
12. N. Bitansky, O. Paneth, and D. Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In E. Kushilevitz and T. Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 474–502. Springer, Heidelberg, Jan. 2016.
13. M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.
14. D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, Heidelberg, May 2004.
15. D. Boneh, D. Freeman, J. Katz, and B. Waters. Signing a linear subspace: Signature schemes for network coding. In S. Jarecki and G. Tsudik, editors, *PKC 2009*, volume 5443 of *LNCS*, pages 68–87. Springer, Heidelberg, Mar. 2009.
16. D. Boneh and D. M. Freeman. Homomorphic signatures for polynomial functions. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 149–168. Springer, Heidelberg, May 2011.
17. D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 1–16. Springer, Heidelberg, Mar. 2011.
18. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.
19. E. Boyle, G. Couteau, N. Gilboa, and Y. Ishai. Compressing vector OLE. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, Oct. 2018.
20. E. Boyle, N. Gilboa, and Y. Ishai. Breaking the circuit size barrier for secure computation under DDH. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539. Springer, Heidelberg, Aug. 2016.
21. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer, Heidelberg, May 2005.
22. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs. Fiat-Shamir: from practice to theory. 2019.
23. R. Canetti, Y. Chen, L. Reyzin, and R. D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 91–122. Springer, Heidelberg, Apr. / May 2018.

24. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.
25. R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, July 2007.
26. D. Cash, E. Kiltz, and V. Shoup. The twin Diffie-Hellman problem and applications. *Journal of Cryptology*, 22(4):470–504, Oct. 2009.
27. D. Catalano and D. Fiore. Practical homomorphic message authenticators for arithmetic circuits. *Journal of Cryptology*, 31(1):23–59, Jan. 2018.
28. D. Catalano, D. Fiore, and L. Nizzardo. Programmable hash functions go private: Constructions and applications to (homomorphic) signatures with shorter public keys. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 254–274. Springer, Heidelberg, Aug. 2015.
29. D. Catalano, D. Fiore, and B. Warinschi. Efficient network coding signatures in the standard model. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 680–696. Springer, Heidelberg, May 2012.
30. D. Catalano, D. Fiore, and B. Warinschi. Homomorphic signatures with efficient verification for polynomial functions. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 371–389. Springer, Heidelberg, Aug. 2014.
31. D. Catalano, A. Marcedone, and O. Puglisi. Authenticating computation on groups: New homomorphic primitives and applications. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 193–212. Springer, Heidelberg, Dec. 2014.
32. P. Chaidos and G. Couteau. Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 193–221. Springer, Heidelberg, Apr. / May 2018.
33. P. Chaidos and J. Groth. Making sigma-protocols non-interactive without random oracles. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 650–670. Springer, Heidelberg, Mar. / Apr. 2015.
34. M. Chase, Y. Dodis, Y. Ishai, D. Kraschewski, T. Liu, R. Ostrovsky, and V. Vaikuntanathan. Reusable non-interactive secure computation. *IACR Cryptology ePrint Archive*, 2018:940, 2018.
35. D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.
36. D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 319–327. Springer, Heidelberg, Aug. 1990.
37. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer, Heidelberg, Apr. 1991.
38. G. Couteau and D. Hofheinz. Designated-verifier pseudorandom generators, and their applications. In *Eurocrypt*, pages ???–??? Springer, 2019.
39. R. Cramer and I. Damgård. Secret-key zero-knowlegde and non-interactive verifiable exponentiation. In M. Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 223–237. Springer, Heidelberg, Feb. 2004.
40. I. Damgård. On the randomness of legendre and jacobi sequences. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 163–172. Springer, Heidelberg, Aug. 1990.

41. I. Damgård. Non-interactive circuit based proofs and non-interactive perfect zero-knowledge with proprocessing. In R. A. Rueppel, editor, *EUROCRYPT'92*, volume 658 of *LNCS*, pages 341–355. Springer, Heidelberg, May 1993.
42. I. Damgård, N. Fazio, and A. Nicolosi. Non-interactive zero-knowledge from homomorphic encryption. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 41–59. Springer, Heidelberg, Mar. 2006.
43. A. De Santis, S. Micali, and G. Persiano. Non-interactive zero-knowledge with preprocessing. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 269–282. Springer, Heidelberg, Aug. 1990.
44. Y. Desmedt. Computer security by redefining what a computer is. In *NSPW*, pages 160–166. ACM, 1993.
45. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
46. C. Dwork and M. Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.
47. U. Feige, D. Lapidot, and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.
48. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, Aug. 1987.
49. D. M. Freeman. Improved security for linearly homomorphic signatures: A generic framework. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 697–714. Springer, Heidelberg, May 2012.
50. R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin. Secure network coding over the integers. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 142–160. Springer, Heidelberg, May 2010.
51. R. Gennaro and D. Wichs. Fully homomorphic message authenticators. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 301–320. Springer, Heidelberg, Dec. 2013.
52. C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, Oct. 2015.
53. O. Goldreich. Foundations of cryptography: Volume 2, basic applications. 2004.
54. O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.
55. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In A. Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.
56. O. Goldreich and Y. Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, Dec. 1994.
57. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
58. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, Aug. 2012.
59. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. *J. ACM*, 62(6):45:1–45:33, 2015.
60. S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In R. A. Servedio and R. Rubinfeld, editors, *47th ACM STOC*, pages 469–477. ACM Press, June 2015.

61. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, Oct. / Nov. 2006. Available as Cryptology ePrint Archive Report 2006/309.
62. J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, Dec. 2010.
63. J. Groth, R. Ostrovsky, and A. Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.
64. J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012.
65. J. Holmgren and A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In M. Thorup, editor, *59th FOCS*, pages 850–858. IEEE Computer Society Press, Oct. 2018.
66. Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM J. Comput.*, 39(3):1121–1152, 2009.
67. R. Johnson, D. Molnar, D. X. Song, and D. Wagner. Homomorphic signature schemes. In B. Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 244–262. Springer, Heidelberg, Feb. 2002.
68. Y. T. Kalai, G. N. Rothblum, and R. D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 224–251. Springer, Heidelberg, Aug. 2017.
69. S. Katsumata. On the untapped potential of encoding predicates by arithmetic circuits and their applications. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 95–125. Springer, Heidelberg, Dec. 2017.
70. J. Kilian, S. Micali, and R. Ostrovsky. Minimum resource zero-knowledge proofs (extended abstract). In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 545–546. Springer, Heidelberg, Aug. 1990.
71. S. Kim and D. J. Wu. Multi-theorem preprocessing NIZKs from lattices. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 733–765. Springer, Heidelberg, Aug. 2018.
72. S. Kim and D. J. Wu. Multi-theorem preprocessing nizks from lattices. Cryptology ePrint Archive, Report 2018/272, 2018. https://eprint.iacr.org/2018/272.pdf, Version 20180606:204702. Preliminary version appeared in CRYPTO 2018.
73. D. Lapidot and A. Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In A. J. Menezes and S. A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 353–365. Springer, Heidelberg, Aug. 1991.
74. H. Lipmaa. Optimally sound sigma protocols under DCRA. In A. Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 182–203. Springer, Heidelberg, Apr. 2017.
75. M. Nandi and T. Pandit. On the power of pair encodings: Frameworks for predicate cryptographic primitives. Cryptology ePrint Archive, Report 2015/955, 2015. http://eprint.iacr.org/2015/955.
76. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.
77. R. Pass, a. shelat, and V. Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 271–289. Springer, Heidelberg, Aug. 2006.
78. C. Peikert and S. Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. *IACR Cryptology ePrint Archive*, 2019:158, 2019.
79. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

80. W. Quach, R. Rothblum, and D. Wichs. Reusable designated-verifier nizks for all np from cdh. In *Eurocrypt*, pages ???–??? Springer, 2019.
81. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Heidelberg, Dec. 2001.
82. R. D. Rothblum, A. Sealfon, and K. Sotiraki. Towards non-interactive zero-knowledge for NP from LWE. Cryptology ePrint Archive, Report 2018/240, 2018. https://eprint.iacr.org/2018/240.
83. Y. Rouselakis and B. Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 2013*, pages 463–474. ACM Press, Nov. 2013.
84. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, Oct. 1999.
85. A. Sahai and H. Seyalioglu. Worry-free encryption: functional encryption with public keys. In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *ACM CCS 2010*, pages 463–472. ACM Press, Oct. 2010.
86. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In D. B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.
87. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.
88. H. Shacham and B. Waters. Compact proofs of retrievability. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 90–107. Springer, Heidelberg, Dec. 2008.
89. I. Teranishi, J. Furukawa, and K. Sako. k-Times anonymous authentication (extended abstract). In P. J. Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 308–322. Springer, Heidelberg, Dec. 2004.
90. R. Tsabary. An equivalence between attribute-based signatures and homomorphic signatures, and new constructions for both. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 489–518. Springer, Heidelberg, Nov. 2017.
91. C. Ventre and I. Visconti. Co-sound zero-knowledge with public keys. In B. Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 287–304. Springer, Heidelberg, June 2009.
92. B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Heidelberg, Mar. 2011.