Non-interactive Zero-Knowledge in Pairing-Free Groups from Weaker Assumptions

Geoffroy Couteau[1](✉), Shuichi Katsumata[2], and Bogdan Ursu[3]

[1] CNRS, IRIF, Université de Paris, Paris, France
geoffroy.couteau@irif.fr
[2] AIST, Tokyo, Japan
shuichi.katsumata@aist.go.jp
[3] ETH Zürich, Zürich, Switzerland
bogdan.ursu@inf.ethz.ch

Abstract. We provide new constructions of non-interactive zero-knowledge arguments (NIZKs) for NP from discrete-logarithm-style assumptions over cyclic groups, without relying on pairings. A previous construction from (Canetti et al., Eurocrypt'18) achieves such NIZKs under the assumption that no efficient adversary can break the key-dependent message (KDM) security of (additive) ElGamal with respect to all (even inefficient) functions over groups of size 2^λ, with probability better than $\mathsf{poly}(\lambda)/2^\lambda$. This is an extremely strong, non-falsifiable assumption. In particular, even mild (polynomial) improvements over the current best known attacks on the discrete logarithm problem would already contradict this assumption. (Canetti et al. STOC'19) describe how to improve the assumption to rely only on KDM security with respect to all efficient functions, therefore obtaining an assumption that is (in spirit) falsifiable.

Our first construction improves this state of affairs. We provide a construction of NIZKs for NP under the CDH assumption together with the assumption that no efficient adversary can break the key-dependent message one-wayness of ElGamal with respect to *efficient* functions over groups of size 2^λ, with probability better than $\mathsf{poly}(\lambda)/2^{c\lambda}$ (denoted $2^{-c\lambda}$-OW-KDM), for a constant $c = 3/4$. Unlike the previous assumption, our assumption leaves an exponential gap between the best known attack and the required security guarantee.

We also analyse whether we could build NIZKs when CDH does not hold. As a second contribution, we construct an *infinitely often* NIZK argument system for NP (where soundness and zero-knowledge are only guaranteed to hold for infinitely many security parameters), under the $2^{-c\lambda}$-OW-KDM security of ElGamal with $c = 28/29 + o(1)$, together with the existence of low-depth pseudorandom generators.

Keywords: Non-interactive zero-knowledge arguments · Pairing-free groups · KDM security

A. Canteaut and Y. Ishai (Eds.): EUROCRYPT 2020, LNCS 12107, pp. 442–471, 2020.
https://doi.org/10.1007/978-3-030-45727-3_15

1 Introduction

Zero-knowledge proof systems, introduced in [21], are a fundamental cryptographic primitive, allowing a prover to convince a verifier of the veracity of a statement, while not divulging anything beyond whether the statement is true. When the proof consists of a single message from prover to the verifier, this results in a non-interactive zero-knowledge proof system (NIZK) [5]. Due to their large number of applications in cryptography, NIZKs enjoy particular interest, ranging from efficient implementations to feasibility results.

On Building NIZKs from Concrete Assumptions. While one-way functions are known to be necessary [36] and sufficient [20] for zero-knowledge proof systems, the exact relation of NIZKs to other cryptographic assumptions and primitives is considerably less clear. NIZKs are known to exist in the plain model only for trivial languages [35]. To circumvent this issue, cryptographers design NIZKs in the common reference string (CRS) model, where a common reference string is honestly generated beforehand in a setup phase and is given to both prover and verifier. A large body of work has been dedicated to the construction of NIZKs in the CRS model from various cryptographic assumptions. As a result, NIZKs are known to exist from a wide range of assumptions, from pairing groups [22,23], factorization assumptions [5,13], and indistinguishability obfuscation [40], to circularly-secure LWE [6] and plain LWE [37]. Yet, in spite of three decades of efforts, it remains an intriguing open question whether one can construct NIZKs from discrete-logarithm-style assumptions (without relying on pairing groups), which are among the most well-established assumptions in cryptography. Here, the only known result is the recent work of [7], which constructs NIZKs under the exponential key-dependent message security of ElGamal with respect to all (even inefficient) functions. While this is a remarkable stepping stone, it remains an extremely strong and non-standard assumption. Therefore, an important question remains open:

> "Is it possible to build NIZKs from (weaker) discrete-logarithm-style assumptions?"

NIZKs from Correlation Intractability. Our work follows the blueprint of a recent line of research, which seeks to compile interactive protocols into NIZKs using the Fiat-Shamir paradigm [15], by instantiating the underlying hash function by a correlation-intractable hash function. Informally, a correlation-intractable hash function (CIH) with respect to a relation R is a hash function such that it is infeasible to find an input x satisfying $(x, H(x)) \in R$. CIH have been introduced in [8], where it was also shown that correlation-intractability for all sparse relations suffices to instantiate the Fiat-Shamir paradigm. Despite some impossibility results [4], a recent line of work has shown how to construct CIH for various sparse relations of interest [6,7,24,25,37], obtaining NIZKs from new assumptions. Out of these works, [7] relies on the exponential key-dependent messages (KDM) security for all (even inefficient) functions of an encryption scheme with universal ciphertexts, which can be instantiated over pairing-free groups with a suitable variant of ElGamal; unfortunately, this is an extremely strong assumption, which has several undesirable features. In this paper, we seek to improve the result of [7] and to construct NIZKs for NP from weaker assumptions over pairing-free groups.

On the Strong-KDM Security Assumption of [7]**.** The construction of [7] relies on the following assumption over cyclic groups: let $\mathbb{G}$ be a group of order $p \approx 2^\lambda$ with a generator g. Then, for any probabilistic polynomial time adversary $\mathcal{A}$, any (possibly inefficient) function $f : \mathbb{Z}_p \mapsto \mathbb{Z}_p$, and any superpolynomial function s, it holds that

$$\Pr\left[(a,k) \leftarrow_r \mathbb{Z}_p^2 \ : \ \mathcal{A}\left(g^a, g^{ak+f(k)}\right) = k\right] \leq \frac{s(\lambda)}{2^\lambda}.$$

While this assumption is not contradicted by known attacks on the discrete logarithm over suitably chosen elliptic curves, it is an extremely strong assumption, with several undesirable features:

- **Optimality.** Optimal security means that every PPT adversary has advantage at most $\lambda^{O(1)}/2^\lambda$.[1] The above assumption requires *optimal* security, which is equivalent to assuming that no improvement (by more than polynomial factors) to the best known existing attack will ever be found. Hence, even mild cryptanalytic improvements would already contradict the above assumption.
- **Non-falsifiablity.** The above assumption is not *falsifiable*, in the sense of [17,33], since it might not be possible to efficiently check whether an adversary breaks the assumption with respect to some specific inefficient function. However, [6] notes that it is possible to construct NIZKs even when the functions f considered in the assumption are efficient.

Insecurity with Auxiliary Inputs. In the same spirit as knowledge of exponent assumptions, which are known to become insecure (under obfuscation-style assumptions) when auxiliary inputs are allowed, unfalsifiable flavors of KDM security have been recently shown to be insecure as soon as auxiliary inputs are allowed, assuming that LWE is hard and one-way permutations exist [16]. While this does not directly contradict the unfalsifiable flavour of the assumption above, it makes it very sensitive to any side information an adversary might have access to when it is used in a higher-level application.

1.1 Our Contribution

We propose new constructions of NIZKs, improving over the NIZK of [7] in terms of the underlying assumption. As noted in [6], the assumption in [7] can be

[1] In the case of DDH groups, the best known generic PPT adversary is Pollard's rho algorithm [38], which runs in time $O(2^{\lambda/2})$ and has constant success probability. However, restricted to polynomial time, it only provides a polynomial advantage over randomly guessing the discrete logarithm. Moreover, it is known [41] that no generic algorithm with T oracle queries can have better success probability than $O\left(\frac{T^2}{2^\lambda}\right)$.

improved to consider only efficient functions and thus construct NIZKs based on a *falsifiable*-style notion of KDM-security[2]. In this work, we remove the need of relying on optimal security of the underlying assumption, while maintaining the *falsifiable* flavor of KDM security.

We note that our second construction satisfies a weaker notion of security, infinitely-often security, where soundness and zero-knowledge are only required to hold for infinitely many security parameters. For a discussion on the notion of infinitely-often security and its usage in cryptography, please refer to the full version of the paper.

In more detail, the assumption at the core of our new construction is a strong flavor of the OW-KDM security of ElGamal: given a group $\mathbb{G}$ of size $\approx 2^{\lambda}$ with generator g, the $2^{-c\lambda}$-OW-KDM assumption states that for a family of (randomized) *efficient* functions $\mathcal{F}$, any PPT adversary receiving an ElGamal ciphertext encrypting $F(k)$ (in the exponent) with the key k is unable to recover the plaintext with advantage greater than $s(\lambda)/2^{c\lambda}$, for any superpolynomial function s:

$$\Pr_{\substack{(k,a)\leftarrow_r \mathbb{Z}_q^2 \\ m\leftarrow_r F(k)}}[\mathcal{A}(g^a, g^{ak+m}) = m] \leq s(\lambda)/2^{c\lambda} \text{ for some } c \in [0,1].$$

The value c determines the strength of the assumption: $c = 1$ corresponds to assuming optimal security (as in [7]), while smaller values of c leave a gap between the success probability of the best known attacks and the success probability that can be tolerated by the assumption. In particular, a constant $c < 1$ indicates that the assumption can stand even exponential improvements in the success probability of the best known attacks.

1. Assuming the hardness of CDH and the $2^{-c\lambda}$-OW-KDM security of ElGamal with $c = 3/4$, we propose an adaptively-sound multi-theorem NIZK for all of NP. Both soundness and zero knowledge are computational, the first is implied by OW-KDM, while the second is implied by CDH.
2. Our second construction aims at analysing the complementary landscape. More precisely, we investigate the possibility of building NIZKs in groups where CDH does *not* hold, building upon the fact that this implies (using known results) the existence of a self-bilinear map. We leverage this self-bilinear map to obtain an adaptively-sound, adaptively multi-theorem zero-knowledge (infinitely often) NIZK for all of NP, under the $2^{-c\lambda}$-OW-KDM security of ElGamal with $c = 28/29 + o(1)$, together with the assumption that Goldreich's PRG [18] instantiated under the Lombardi-Vaikuntanathan predicate [29] is secure up to some (arbitrarily small) polynomial stretch.[3] Combining this result with our first construction, we obtain a construction of (infinitely-often) NIZKs for NP under the same assumptions, independently of whether CDH holds.

[2] More precisely, these assumptions are falsifiable in spirit in the sense that they can be modeled as an efficient game with a challenger, but the winning condition can occur with exponentially small probability.

[3] The security of Goldreich's PRG is a well-established and widely studied assumption, which provably resists large classes of attacks [2,3,10,32,34].

In both constructions, an important effort is devoted to obtaining the smallest possible constant c, to minimize the strength of the underlying assumption. We view it as an interesting open problem to further minimize the value of c, especially in our second construction.

1.2 Our Techniques – First Construction

Both our constructions follow a similar footprint: we start from a Σ-protocol for a carefully chosen, but limited language. We compile this Σ-protocol using a correlation-intractable (CI) hash function into a NIZK for the same limited language. Then we use different techniques to bootstrap this restricted NIZK to NIZK for all of NP, by using them to build a verifiable pseudorandom generator (VPRG) [11,26,39], which in turns leads to NIZKs for NP. Our approach is inspired by [7], their strategy is to design a correlation-intractable (CI) hash function based on a scheme with universal ciphertexts, which they use to transform an underlying sigma protocol into a NIZK. In their case, the interactive protocol is the one in [14, Section 2.1]. We diverge from this approach by applying the CI hash function to a sigma protocol for a more restricted, but still expressive enough language (which we bootstrap later to a fully-fledged NIZK through VPRGs). Looking ahead, the parameters of the KDM security assumption are intrinsically tied to the ratio between the size of the first flow of the sigma protocol and its adaptive soundness. By allowing the underlying sigma protocol to support only a more restricted language, we expand the field of potential candidates and eventually identify a protocol with a better first flow/soundness ratio. Our initial attempt is to start with the standard Σ-protocol for the Diffie-Hellman relation $\mathcal{L}_{\mathsf{DH}}$, described in Fig. 1. Choose a cyclic group $\mathbb{G}$ of prime order p, along with two generators g and h. The relation consists of all pairs of group elements of the form (g^x, h^x). To transform the sigma protocol into a NIZK for $\mathcal{L}_{\mathsf{DH}}$, the idea of the CI framework is to apply the Fiat-Shamir transform, but instead of using random oracles, the random oracle is replaced with a CI hash function.

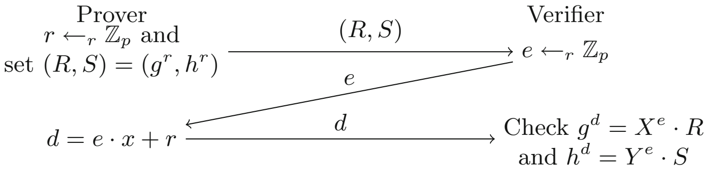
Prover
$r \leftarrow_r \mathbb{Z}_p$ and
set $(R, S) = (g^r, h^r)$
(R, S)
Verifier
$e \leftarrow_r \mathbb{Z}_p$
e
$d = e \cdot x + r$
d
Check $g^d = X^e \cdot R$
and $h^d = Y^e \cdot S$

Fig. 1. Σ-protocol for the Diffie-Hellman language for the word $(g, h, X = g^x, Y = g^y)$. This is a variant of a protocol from [1]

CI Hash Functions. A CI hash function H for a specific relation $\mathcal{R}$ is a function for which it is hard to find an input α, such that $(\alpha, \mathsf{H}(\alpha)) \in \mathcal{R}$. Consider the case where the initial relation is sparse, meaning that for every α, the number of potential β's satisfying $(\alpha, \beta) \in \mathcal{R}$ is negligible. Then, the sigma protocol can be transformed into a NIZK by asking the prover to generate the second flow himself, by running $e = \mathsf{H}(R, S)$. The verifier will only accept if the resulting transcript is accepting and also $e = \mathsf{H}(R, S)$. From the correlation intractability of H, even a malicious prover will be unable to cheat by finding a properly chosen initial flow (R, S), such that $((R, S), \mathsf{H}(R, S)) \in \mathcal{R}$ (this also holds because the sparsity of the relation $\mathcal{R}$ is bounded by the soundness error of the sigma protocol, which is negligible).

Choice of H. To construct the hash, we choose a function closely related to the one used in [7], where $\mathsf{H}(x, K)$ interprets the input x as a decryption key, and the key K as a ciphertext, end returns $\mathsf{Dec}_x(K)$. For our instantiation, we crucially rely on a specific property of the additive variant of ElGamal (which is, informally, that keys and plaintexts are "interchangeable"). Since additive ElGamal does not provide efficient decryption (the decryption procedure recovers only $\tilde{G}^m$, and we cannot guarantee that m will be small in our construction), we modify the CI hash of [7] so that it returns $\mathsf{Trunc}(\tilde{G}^m)$, where Trunc is some function that parses its input as a bitstring and truncates it appropriately. More precisely, we pick a second cyclic group $\tilde{\mathbb{G}}$ of order q, generated by $\tilde{G}$ ($\lceil \log q \rceil = 2\lceil \log p \rceil$). The CI function is keyed by key $\tilde{C} = (\tilde{C}_0, \tilde{C}_1)$, where $(\tilde{C}_0, \tilde{C}_1) \leftarrow_r \tilde{\mathbb{G}}^2$. Then, we define:

$$\mathsf{H}_{(\tilde{C}_0, \tilde{C}_1)}(\alpha) \leftarrow \text{first } \lceil \log p \rceil \text{ bits of } \tilde{C}_1 / \tilde{C}_0^{\alpha}.$$

Parameters. This protocol has $\frac{1}{p}$ soundness and the size of the first flow is $2\lceil \log p \rceil$, which translates into a $2^{-\lambda/2}$-KDM assumption for the CI hash function. Unfortunately, this Σ-protocol does not satisfy adaptive soundness (given an honestly-generated first flow and challenge, there always exist words that are not in the relation, for which there exists an accepting third flow). Adaptive soundness is a crucial requirement for bootstrapping our first NIZK to cover all NP statements. Fortunately, performing a parallel repetition of the Σ-protocol yields adaptive soundness, albeit at the cost of worse parameters in our assumption ($c = 3/4$).

Reduction to KDM for Efficient Functions. The above construction reduces to the KDM security of ElGamal, but only with respect to an inefficient function f, which maps first flows to accepting challenges. From there, we leverage the fact that an ElGamal encryption $(\tilde{G}^r, \tilde{G}^{kr+m})$ of a plaintext m with key k, with respect to a generator $\tilde{G}$, can be equivalently seen as an ElGamal encryption of k with the key m with respect to the generator $\tilde{G}^r$. Building upon this observation and the fact that f^{-1} is efficient, we show that the security of our NIZK for the DDH language can in fact be reduced to the KDM security of ElGamal with respect to the *efficient* function f^{-1}.

From $\mathsf{NIZK}_{\mathsf{DH}}$ for $\mathcal{L}_{\mathsf{DH}}$ to a NIZK for all of NP. In this step, we use an idea implicitly employed in [11,26,39]. We use the $\mathsf{NIZK}_{\mathsf{DH}}$ for the $\mathcal{L}_{\mathsf{DH}}$ relation to construct a verifiable pseudo-random generator (VPRG), which we then in turn use to instantiate the hidden bits model of [14], to obtain NIZKs for all of NP.

Intuitively, a VPRG is a pseudo-random generator with the additional property that one can compute proofs for any individual bit of the output, certifying that the bit is consistent with a commitment of the initial seed. Let $\mathbb{G}$ be a cyclic group of order p, the VPRG public parameters will consist of $m + 1$ group elements $(g, h_1, \ldots, h_m)$. Seeds are elements $\tau \leftarrow_r \mathbb{Z}_p$, and commiting to a seed is $\mathsf{Commit}(\tau) = g^\tau$. The i^{th} output bit of the VPRG is of the form $B(g^\tau, h_i^\tau)$, where B is the Goldreich-Levin hardcore bit. Now notice than we can actually certify this as a correctly computed bit, by noticing that $(g^\tau, h_i^\tau) \in \mathcal{L}_{\mathsf{DH}}$ and computing a proof using our $\mathsf{NIZK}_{\mathsf{DH}}$. (additionally, we need to output h_i^τ as well, so that the verifier can compute $B(g^\tau, h_i^\tau)$ itself). Intuitively, this VPRG satisfies the following security properties:

1. Binding: If x_i is the i^{th} output of the VPRG with respect to a seed τ, one should not be able to certify bit $1 - x_i$. This is implied in our construction by the soundness of $\mathsf{NIZK}_{\mathsf{DH}}$.
2. Hiding: An adversary should not be able to recover the i^{th} output of the VPRG, even if it received all the other output bits and proofs certifying that they are correct. In our construction, this property reduces to the CDH assumption.

NIZK for all of NP Through the Hidden-Bit Model. In this model [14], the prover and the verifier benefit from having access to a common reference string with special properties. The bits of the common reference string are initially hidden from the verifier. When proving a statement, the prover can decide to selectively reveal some bits of the common reference string, which allows the verifier to check the proof. The work of [14] has showed that NIZKs exist unconditionally in this model. The VPRG we construct allows us to simulate the hidden-bits model on the prover side. Initially, all bits are hidden from the verifier from the hiding property of the VPRG. Subsequently, the prover can decide to reveal several bits, which corresponds to computing VPRG proofs.

1.3 Our Techniques – Second Construction

The previous construction relies on the CDH assumption. In our second construction, we take the complementary road: we seek to construct NIZKs for NP (under the strong KDM security of ElGamal assuming that CDH does *not* hold. Together with our first construction, this implies a NIZK for NP that does not rely on the CDH assumption (albeit with an infinitely-often security notion). To this end, we also seek to build a VPRG.

Self-pairing. First, we notice that if CDH does not hold, there exists an efficient adversary solving it with non-negligible advantage. We use previous results by [31,41] to amplify the success probability of this adversary to obtain a self-pairing map. Since from the definition of CDH, the adversary is only guaranteed to succeed on infinitely-many security parameters, our NIZK will be secure only on infinitely-many security parameters. This self-pairing will allow us to perform homomorphic computations and to evaluate bounded integer arithmetic circuits in the exponent. Our core idea, informally, is to rely on this self-pairing to let the parties homomorphically evaluate a pseudorandom generator in the exponent: at a high level, given a (bit-by-bit) commitment c to the seed, the parties can homomorphically compute, using the self-pairing, a commitment c_i to the i-th output bit of the PRG (for all i). Then, the prover will open a given PRG value by providing a NIZK proof of correct opening.

A Commitment from Short-Exponent Discrete Logarithm. To instantiate this idea, we introduce a new commitment scheme which is perfectly binding, and which is hiding under the short-exponent discrete logarithm assumption (which states that given g^x for a random but *short* x, it is infeasible to retrieve x). This does not introduce any new assumption, as we further show that the short-exponent discrete logarithm assumption is implied by the strong OW-KDM security of ElGamal. Furthermore, we carefully design this commitment scheme so that it suffices, to convince the verifier that the opening was correct, to demonstrate that the randomness r of the commitment is *almost short*. By almost short, we mean that there exists short values (u, v) such that $v \cdot r = u \bmod p$. This turns out to be a crucial property, since the language of group elements with almost-short exponents is precisely one for which we are able to build a NIZK under the $2^{-c\lambda}$-OW-KDM security of ElGamal, for some $c < 1$.

A Σ-Protocol for Almost-Short Exponents. Let $\mathbb{G}$ be a cyclic group of p elements. We consider a simple Σ-protocol for proving that a word g^x has a short exponent, i.e. writing x as an integer yields a number $\leq 2^\ell$, for some carefully chosen $\ell < \lceil \log p \rceil$. Our protocol has a similar shape to the sigma protocol used in the previous construction, and is described in Fig. 3. However, we are unable to directly prove soundness, meaning that a malicious prover can convince the verifier of the validity of words g^x, where x is not short. Fortunately, we are able to ensure that if g^x is accepted, then $x = u \cdot v^{-1}$, where u and v are themselves short. We denote this as the language $\mathscr{L}_{\alpha,\beta}$ of (α, β)-almost-short elements:

$$\mathscr{L}_{\alpha,\beta} = \{g^x \mid x = u \cdot v^{-1} \in \mathbb{Z}_p, u \in [-2^\alpha, 2^\alpha], v \in [0, 2^\beta]\}.$$

Our Σ-protocol is somewhat atypical, in the honest run the prover must start with a word of the form g^x and a short witness x (notice that if x is short it belongs to the almost-short language). However, when proving soundness, we only safeguard membership to the larger almost-short set of words; therefore, there is a gap between the correctness requirement, and the soundness guarantees (this is similar to some lattice constructions, for example [30]).

NIZK$^{\mathsf{AS}}$ for the Language of Almost-Short Exponents. We will design another CI hash function, closely related to the one we built for the first construction, to transform the Σ-protocol above into a NIZK for the almost-short exponent language. This CI hash function will additionally employ a 2-universal hash function, which we use to reduce the security loss in our security analysis and achieve a better parameter c for the $\mathsf{OW\text{-}KDM}$ assumption. Now, equipped with our $\mathsf{NIZK}^{\mathsf{AS}}$, we only need one final tool before moving on to our VPRG.

A Low-Depth Local PRG. Equipped with the above tools, it remains to find a suitable PRG to be used in our construction. For correctness, we need to ensure that no overflow occurs during the homomorphic operations in the exponent; therefore, we must pick the group size large enough so that the homomorphic PRG evaluation does not cause an overflow. Since picking a larger group translates into a larger security loss in our reduction, we seek to rely on a PRG (with some arbitrary small polynomial stretch) that has a minimal *arithmetic degree*. Fortunately, such PRGs were recently studied in [29], which exhibits a PRG with arithmetic degree 3 which provably resists a large class of attacks for a stretch up to $1.25 - \varepsilon$. Combining this low-degree PRG with our new commitment scheme and our NIZK for the almost-short language yields a VPRG in groups where CDH does not hold, hence NIZKs for NP.

Wrapping Up. Combining our first and second construction, we get the following: assume that ElGamal is $2^{-c\lambda}$-$\mathsf{OW\text{-}KDM}$ secure with respect to efficient functions (with $c = 28/29 + o(1)$), and that the previous PRG is secure. Then either CDH holds, in which case our first construction implies a NIZK for NP, or CDH does *not* hold, in which case our second construction implies an (infinitely-often) NIZK for NP. Therefore, under a PRG assumption and the strong $\mathsf{OW\text{-}KDM}$ security of ElGamal, we prove the existence of an infinitely-often NIZK for NP (but our proof is non-constructive, in that it does not tell *which* of the two candidate constructions is actually secure; only that one is).

1.4 Organization

Section 2 introduces necessary preliminaries. Section 3 presents our first NIZK construction and Sect. 4 contains our second construction. Please consult the full version for supplementary material, on how to construct an algorithm for evaluating an arithmetic circuit in the exponent from groups where CDH is insecure, with bounds on the parameter growth when manipulating bounded-size exponents. The full version also contains all missing proofs of our theorems and a discussion on the notion of infinitely-often security.

2 Preliminaries

Notation. Throughout this paper, λ denotes the security parameter. A probabilistic polynomial time algorithm (PPT, also denoted *efficient* algorithm) runs in time polynomial in the (implicit) security parameter λ. A function f is *negligible* if for any positive polynomial p there exists a bound $B > 0$ such that, for any integer $k \geq B$, $|f(k)| \leq 1/|p(k)|$. We will write $f(\lambda) \approx 0$ to indicate that f is a negligible function of λ; we also write $f(\lambda) \approx g(\lambda)$ for $|f(\lambda) - g(\lambda)| \approx 0$. An event occurs with *overwhelming probability* p when $p \approx 1$. Given a finite set S, the notation $x \leftarrow_r S$ means a uniformly random assignment of an element of S to the variable x. For a positive integer n, m such that $n < m$, we denote by $[n]$ the set $\{1, \cdots, n\}$, by $[\pm n]$ the set $\{-n, \cdots, n\}$, and by $[n, m)$ the set $\{n, n+1, \cdots, m-1\}$. Given an element x of a set $\mathbb{Z}_p$, we denote by $\mathsf{int}(x)$ the integer $x' \in [\pm p/2]$ such that $x = x' \bmod p$. When manipulating elements (x, y) of $\mathbb{Z}_p$, we will generally abuse the notation and write $x \leq y$ for $\mathsf{int}(x) \leq \mathsf{int}(y)$.

The Computational Diffie-Hellman Assumption. Let DHGen be a deterministic algorithm that on input 1^λ returns a description $\mathcal{G} = (\mathbb{G}, p)$ where $\mathbb{G}$ is a cyclic group of prime order p. Then the computational Diffie-Hellman assumption is defined as follows.

Definition 1 (CDH Assumption). *We say that the computational Diffie-Hellman (CDH) assumption holds relative to* DHGen *if for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr\left[\mathcal{G} \leftarrow \mathsf{DHGen}(1^\lambda), g \leftarrow_r \mathbb{G}, \alpha, \beta \leftarrow_r \mathbb{Z}_p : g^{\alpha\beta} \leftarrow_r \mathcal{A}(1^\lambda, \mathcal{G}, g, g^\alpha, g^\beta)\right] \leq \mathsf{negl}(\lambda).$$

Here, note that DHGen *outputs a fixed group* $\mathbb{G}$ *per security parameter.*

2.1 Non-interactive Zero-Knowledge

A (publicly-verifiable) non-interactive zero-knowledge (NIZK) argument system for an NP relation R, with associated language $\mathscr{L}(R) = \{x \mid \exists w, (x, w) \in R\}$ is a 3-tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$, where Setup outputs a common reference string, $\mathsf{Prove}(\mathsf{crs}, x, w)$, given the crs, a word x, and a witness w, outputs a proof π, and $\mathsf{Verify}(\mathsf{crs}, x, \pi)$, on input the crs, a word x, and a proof $\boldsymbol{\pi}$, outputs a bit indicating whether the proof is accepted or not. A NIZK argument system satisfies the following: completeness, adaptive soundness, and selective single-theorem zero-knowledge properties: (we let R_λ denote the set $R \cap (\{0,1\}^\lambda \times \{0,1\}^*)$).

- A non-interactive argument system $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for an NP relation R satisfies completeness if for every $(x, w) \in R$,

$$\Pr[\mathsf{crs} \leftarrow_r \mathsf{Setup}(1^{|x|}), \boldsymbol{\pi} \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) : \mathsf{Verify}(\mathsf{crs}, x, \boldsymbol{\pi}) = 1] \approx 1.$$

- A non-interactive argument system $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for an NP relation R satisfies *adaptive soundness* if for any PPT $\mathcal{A}$,

$$\Pr\begin{bmatrix} \mathsf{crs} \leftarrow_r \mathsf{Setup}(1^\lambda), (x, \boldsymbol{\pi}) \leftarrow_r \mathcal{A}(\mathsf{crs}) : \\ \mathsf{Verify}(\mathsf{crs}, x, \boldsymbol{\pi}) = 1 \wedge x \notin \mathscr{L} \end{bmatrix} \approx 0.$$

- A non-interactive argument system $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for an NP relation R satisfies (computational, statistical) *selective single-theorem zero-knowledge* if there exists a PPT simulator Sim such that for every $(x, w) \in R$, the distribution $\{(\mathsf{crs}, \boldsymbol{\pi}) : \mathsf{crs} \leftarrow_r \mathsf{Setup}(1^\lambda), \boldsymbol{\pi} \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)\}$ and $\{(\mathsf{crs}, \boldsymbol{\pi}) : (\mathsf{crs}, \boldsymbol{\pi}) \leftarrow_r \mathsf{Sim}(x)\}$ are (computationally, statistically) indistinguishable.

Furthermore, we say that a NIZK for an NP relation R satisfies (computational, statistical) *adaptive multi-theorem zero-knowledge* if for all (computational, statistical) $\mathcal{A}$, there exists a PPT simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that if we run $\mathsf{crs} \leftarrow_r \mathsf{Setup}(1^\lambda)$ and $\overline{\mathsf{crs}} \leftarrow_r \mathsf{Sim}_1(1^\lambda)$, then we have $|\Pr[\mathcal{A}^{\mathcal{O}_0(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_1(\overline{\mathsf{crs}},\cdot,\cdot)}(\mathsf{crs}) = 1]| \approx 0$, where $\mathcal{O}_0(\mathsf{crs}, x, w)$ outputs $\mathsf{Prove}(\mathsf{crs}, x, w)$ if $(x, w) \in R$ and $\perp$ otherwise, and $\mathcal{O}_1(\overline{\mathsf{crs}}, x, w)$ outputs $\mathsf{Sim}_2(\overline{\mathsf{crs}}, x)$ if $(x, w) \in R$ and $\perp$ otherwise.

We use the following result regarding the existence of NIZKs in the hidden-bits model (HBM). Since the full definition of NIZK in the HBM will not be required in our work, we refer the readers to [13] for more details.

Theorem 2 (NIZK for all of NP in the HBM). *Let λ denote the security parameter and let $k = k(\lambda)$ be any positive integer-valued function. Then, unconditionally, there exists NIZK proof systems for any* NP *language $\mathscr{L}$ in the HBM that uses $\mathsf{hb} = k \cdot \mathsf{poly}(\lambda)$ hidden bits with soundness error $\epsilon \leq 2^{-k \cdot \lambda}$, where λ denotes the security parameter and* poly *is a function related to the* NP *language $\mathscr{L}$.*

2.2 Verifiable Pseudorandom Generators

Definition 3 (Verifiable Pseudorandom Generator). *Let $\delta(\lambda)$ and $s(\lambda)$ be positive valued polynomials. A $(\delta(\lambda), s(\lambda))$-verifiable pseudorandom generator (*VPRG*) is a four-tuple of efficient algorithms* $(\mathsf{Setup}, \mathsf{Stretch}, \mathsf{Prove}, \mathsf{Verify})$ *such that*

- $\mathsf{Setup}(1^\lambda, m)$, *on input the security parameter (in unary) and a polynomial bound $m(\lambda) \geq s(\lambda)^{1+\delta(\lambda)}$, outputs a set of public parameters* pp *(which contains 1^λ);*
- $\mathsf{Stretch}(\mathsf{pp})$, *on input the public parameters* pp, *outputs a triple* $(\mathsf{pvk}, x, \mathsf{aux})$, *where* pvk *is a public verification key of length $s(\lambda)$, x is an m-bit pseudorandom string, and* aux *is an auxiliary information;*
- $\mathsf{Prove}(\mathsf{pp}, \mathsf{aux}, i)$, *on input the public parameters* pp, *auxiliary informations* aux, *an index $i \in [m]$, outputs a proof π;*
- $\mathsf{Verify}(\mathsf{pp}, \mathsf{pvk}, i, b, \pi)$, *on input the public parameters* pp, *a public verification key* pvk, *an index $i \in [m]$, a bit b, and a proof π, outputs a bit β;*

which is in addition complete, hiding, *and* binding, *as defined below.*

Definition 4 (Completeness of a VPRG). *For any $i \in [m]$, a* complete DVPRG *scheme* $(\mathsf{Setup}, \mathsf{Stretch}, \mathsf{Prove}, \mathsf{Verify})$ *satisfies:*

$$\Pr\left[\begin{array}{l} \mathsf{pp} \leftarrow_r \mathsf{Setup}(1^\lambda, m), \\ (\mathsf{pvk}, x, \mathsf{aux}) \leftarrow_r \mathsf{Stretch}(\mathsf{pp}), : \mathsf{Verify}(\mathsf{pp}, \mathsf{pvk}, i, x_i, \pi) = 1 \\ \pi \leftarrow_r \mathsf{Prove}(\mathsf{pp}, \mathsf{aux}, i), \end{array}\right] \approx 1.$$

Note that our definition of VPRG is slightly relaxed than what is considered in [11,12,39], in that, we do not require the size of $s(\lambda)$ to be independent of $m(\lambda)$. This relaxation still allows us to construct NIZKs for NP as long as the stretch $\delta(\lambda)$ is larger than some positive constant.

Definition 5 (Binding Property of a VPRG). *Let* $(\mathsf{Setup}, \mathsf{Stretch}, \mathsf{Prove}, \mathsf{Verify})$ *be a* VPRG. *A* VPRG *is* binding *if there exists a (possibly inefficient)* extractor Ext *such that for any PPT $\mathcal{A}$, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{pp} \leftarrow_r \mathsf{Setup}(1^\lambda, m), \\ (\mathsf{pvk}, i, \pi) \leftarrow_r \mathcal{A}(\mathsf{pp}), : \mathsf{Verify}(\mathsf{pp}, \mathsf{pvk}, i, 1 - x_i, \pi) = 1 \\ x \leftarrow \mathsf{Ext}(\mathsf{pp}, \mathsf{pvk})\end{array}\right] \approx 0.$$

Note that, following [11,26,39], we consider a significantly weaker flavor of binding compared to [12], which still allows to construct NIZKs for NP.

Definition 6 (Hiding Property of a VPRG). *A* VPRG *scheme* $(\mathsf{Setup}, \mathsf{Stretch}, \mathsf{Prove}, \mathsf{Verify})$ *is* hiding *if for any $i \in [m]$ and any PPT adversary $\mathcal{A}$ that outputs bits, it holds that:*

$$\Pr\left[\begin{array}{l}\mathsf{pp} \leftarrow_r \mathsf{Setup}(1^\lambda, m), \\ (\mathsf{pvk}, x, \mathsf{aux}) \leftarrow_r \mathsf{Stretch}(\mathsf{pp}), : \mathcal{A}(\mathsf{pp}, \mathsf{pvk}, i, (x_j, \pi_j)_{j \neq i}) = x_i \\ (\pi_j \leftarrow_r \mathsf{Prove}(\mathsf{pp}, \mathsf{aux}, j))_j\end{array}\right] \approx 1/2.$$

The following shows that VPRG with a sufficient stretch is sufficient to construct NIZKs for all of NP.

Theorem 7 ((δ, s)-VPRGs ⇒ NIZKs for all of NP). *Fix an NIZK proof system for any NP language $\mathscr{L}$ in the HBM that uses $\mathsf{hb} = \mathsf{hb}(\lambda)$ hidden bits with soundness error $\epsilon \leq 2^{-\lambda}$ where $\mathsf{hb} \geq \lambda$ w.l.o.g. Suppose that a $(\delta(\lambda), s(\lambda))$-verifiable pseudorandom generator where $s(\lambda) \geq \max\{\lambda, (\mathsf{hb}^2/\lambda)^{1/\delta(\lambda)}\}$ exits. Then, there exist adaptively sound and adaptively multi-theorem zero-knowledge NIZK arguments for the NP relation $\mathscr{L}$.*

We provide a proof sketch in the full version. Since existence of an NIZK in the HBM for any NP language $\mathscr{L}$ is implied by Theorem 2, the above shows that VPRGs with some mild condition on $\delta(\lambda)$ and $s(\lambda)$ implies existence of an NIZK for any NP language $\mathscr{L}$.

2.3 Correlation-Intractable Hash Functions

We recall the definition of correlation intractability [9].

Definition 8 (Correlation Intractable Hash Function). *A collection $\mathcal{H} = \{H_\lambda : K_\lambda \times I_\lambda \mapsto O_\lambda\}_{\lambda \in \mathbb{N}}$ of (efficient) keyed hash functions is a $\mathcal{R}$-*correlation intractable hash *(CIH) family, with respect to a relation ensemble $\mathcal{R} = \{\mathcal{R}_\lambda \subseteq I_\lambda \times O_\lambda\}$, if for every (non-uniform) PPT adversary $\mathcal{A}$, it holds that*

$$\Pr_{\substack{k \leftarrow_r K_\lambda \\ x \leftarrow_r \mathcal{A}(k)}}[(x, H_\lambda(K, x)) \in \mathcal{R}_\lambda] = \mathsf{negl}(\lambda).$$

For CIH to be useful as a building block for NIZK, we require an additional property referred to as *programmability* [6].

Definition 9 (Programmability). *A collection* $\mathcal{H} = \{H_\lambda : K_\lambda \times I_\lambda \mapsto O_\lambda\}_{\lambda \in \mathbb{N}}$ *of (efficient) keyed hash functions is called* programmable *if there exists an efficient algorithm, which given* $x \in I_\lambda$ *and* $y \in O_\lambda$*, outputs a uniformly random key* k *from* K_λ*, such that* $H(k, x) = y$*.*

Finally, we define the standard notion of *sparsity.*

Definition 10 (Sparsity). *For any relation ensemble* $\mathcal{R} = \{\mathcal{R}_\lambda \subseteq I_\lambda \times O_\lambda\}$*, we say that* $\mathcal{R}$ *is* $\rho(\cdot)$-sparse *if for* $\lambda \in \mathbb{N}$ *and any* $x \in I_\lambda$*,* $\Pr_{y \leftarrow_r O_\lambda}[(x, y) \in \mathcal{R}_\lambda] \leq \rho(\lambda)$*. When* $\rho(\lambda) = \mathsf{negl}(\lambda)$*, we simply say it is* sparse*.*

2.4 Σ-Protocol

We recall the definition of Σ-protocols from [28]. A Σ-protocol is a three-move interactive proof between a prover P and a verifier V for a language $\mathscr{L}$, where the prover sends an initial message α, the verifier responds with a random $\beta \leftarrow_r S_\lambda$ for some challenge space S_λ, and the prover concludes with a message γ. Lastly, the verifier outputs 1, if it accepts and 0 otherwise. Three properties we require from a Σ-protocol are completeness, special honest-verifier zero-knowledge, and adaptive soundness.

Definition 11 (Completeness). *A* Σ*-protocol for a relation* R *with prover* P *and verifier* V *is* complete*, if* $\Pr[\mathsf{out}\langle \mathsf{P}(x, w), \mathsf{V}(x)\rangle = 1 | (x, w) \in R] = 1$*.*

Definition 12 (Special honest-verifier zero-knowledge). *A* Σ*-protocol for a relation* R *is* special honest-verifier zero-knowledge*, if there exists a polynomial-time simulator* Sim *such that the distributions* $\mathsf{Sim}(x, \beta)$ *and* $\langle \mathsf{P}(x, w), \mathsf{V}(x)\rangle$ *are statistically close for* $(x, w) \in R$*,* $\beta \in S_\lambda$*.*

Definition 13 (Adaptive soundness). *A* Σ*-protocol for a relation* R *is* $\rho(\cdot)$-adaptive sound*, if for any (possibly inefficient) cheating prover* P^* *and any first flow* α*, it holds that* $\Pr[\beta \leftarrow_r S_\lambda;\ (x, \gamma) \leftarrow_r \mathsf{P}^*(\alpha, \beta) :\ \exists x \notin \mathscr{L} \wedge V(x, \alpha, \beta, \gamma) = 1] \leq \rho(\lambda)$*. When* $\rho(\lambda) = \mathsf{negl}(\lambda)$*, we simply say it is* adaptive sound*.*

In the above notion, when the cheating P^* does not have the freedom to choose the word x, we say it is *selectively* sound. Note that a selective soundness is implied by the standard notion of special soundness of the Σ-protocol. The following lemma is due to [25], which at a high level claims that any adaptive sound Σ-protocol induces a natural sparse relation.

Lemma 14. *Let* Π *be an arbitrary* $\rho(\cdot)$*-adaptive sound* Σ*-protocol for a language* $\mathscr{L}$*. Then, the following relation induced by the* Σ*-protocol* Π *is* $\rho(\cdot)$*-parse:*

$$\mathcal{R}_{\mathsf{sparse}} = \{(\alpha, \beta) : \exists x, \gamma \text{ s.t. } x \notin \mathscr{L} \ \wedge \ V(x, \alpha, \beta, \gamma) = 1\}.$$

2.5 Secret Key Variant of ElGamal

Definition 15 (Secret Key ElGamal). *Let $\tilde{\mathbb{G}} = \{\tilde{\mathbb{G}}_\lambda\}_{\lambda \in \mathbb{N}}$ be an ensemble of groups where each group $\tilde{\mathbb{G}}_\lambda$ is of order q such that $\lceil \log q \rceil = \lambda$. The natural (secret-key) variant of additive ElGamal with message space $\mathbb{Z}_q$ consists of the following three PPT algorithms.*

- $\mathsf{Setup}(1^\lambda)$: *output public-parameter $\tilde{G} \leftarrow_r \tilde{\mathbb{G}}_\lambda$ and secret key $k \leftarrow_r \mathbb{Z}_q$.*
- $\mathsf{Enc}_{\tilde{G}}(k, m)$: *pick $\tilde{R} \leftarrow_r \tilde{\mathbb{G}}$ and output $\tilde{\mathbf{C}} = (\tilde{R}, \tilde{R}^k \cdot \tilde{G}^m)$.*
- $\mathsf{HalfDec}(k, \tilde{\mathbf{C}})$: *parse $\tilde{\mathbf{C}}$ as $(\tilde{C}_0, \tilde{C}_1)$ and output $\tilde{C}_1/\tilde{C}_0^k$.*

Throughout the paper, we omit the subscript when the meaning is clear. Note that the scheme does not allow for full decryption, but only for decryption "up to discrete logarithm": for every $(\tilde{G}, k, m)$, it holds that $\mathsf{HalfDec}(k, \mathsf{Enc}_{\tilde{G}}(k, m)) = \tilde{G}^m$. One important property of the scheme is that it enjoys the notion of *universality.* Informally, the notion claims that the ciphertexts are not associated with a specific key, but rather, could have been an output of *any* key.

Definition 16 (Universality). *For all $\lambda \in \mathbb{N}$, $\tilde{G} \in \tilde{\mathbb{G}}_\lambda$, and $k^* \in \mathbb{Z}_q$, the ciphertexts of ElGamal satisfies*

$$\{\tilde{\mathbf{C}} : (k, m) \leftarrow_r \mathbb{Z}_q^2, \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, m)\} = \{\tilde{\mathbf{C}} : m \leftarrow_r \mathbb{Z}_q, \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k^*, m)\} = \mathcal{U}_{\tilde{\mathbb{G}}^2}.$$

Definition 17 ($\mathsf{OW\text{-}KDM}$ Security). *Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be an ensemble of sets of functions where each $\mathcal{F}_\lambda = \{F_u\}_u$ is a family of (possibly randomized) efficiently-computable functions. We say that ElGamal satisfies (one-query) δ-*$\mathsf{OW\text{-}KDM}$ *security with respect to $\mathcal{F}$ if for every $F_u \in \mathcal{F}_\lambda$, every superpolynomial function s, and every (non-uniform) PPT adversary $\mathcal{A}$, it holds that*

$$\Pr_{\substack{(\tilde{G}, k) \leftarrow_r \tilde{\mathbb{G}}_\lambda \times \mathbb{Z}_q \\ m \leftarrow F_u(\tilde{G}, k) \\ \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, m)}} [\mathcal{A}(\tilde{G}, \tilde{\mathbf{C}}) = m] \leq s(\lambda) \cdot \delta(\lambda).$$

If ElGamal satisfies δ-$\mathsf{OW\text{-}KDM}$ *security with $\delta(\lambda) = 2^{-c\lambda}$ for some constant $c \in (0, 1]$, then we say it is* strong $\mathsf{OW\text{-}KDM}$ *secure.*

2.6 Low-Depth Pseudorandom Generators

Definition. A pseudorandom generator is a deterministic process that expands a short random seed into a longer sequence, so that no efficient adversary can distinguish this sequence from a uniformly random string of the same length:

Definition 18 (Pseudorandom Generator). *A $m(n)$-stretch pseudorandom generator, for a polynomial m , is a pair of PPT algorithms* $(\mathsf{PRG.Setup}, \mathsf{PRG.Eval})$ *where* $\mathsf{PRG.Setup}(1^n)$ *outputs some public parameters* pp*, which are implicitly given as input to* $\mathsf{PRG.Eval}$*, and* $\mathsf{PRG.Eval}(x)$*, on input a seed $x \in \{0, 1\}^n$, outputs a string $y \in \{0, 1\}^{m(n)}$. It satisfies the following security notion: for any probabilistic polynomial-time adversary $\mathcal{A}$ and every large enough n,*

$$\begin{aligned}&\Pr[\mathsf{pp} \leftarrow_r \mathsf{PRG.Setup}(1^n), y \leftarrow_r \{0,1\}^{m(n)} : \mathcal{A}(\mathsf{pp}, y) = 1]\\ \approx &\Pr[\mathsf{pp} \leftarrow_r \mathsf{PRG.Setup}(1^n), x \leftarrow_r \{0,1\}^n, y \leftarrow \mathsf{PRG.Eval}(x) : \mathcal{A}(\mathsf{pp}, y) = 1]\end{aligned}$$

A pseudorandom generator PRG *is d-*local *(for a function d) if for any $n \in \mathbb{N}$, every output bit of* PRG.Eval *on input a seed $x \in \{0,1\}^n$ depends on at most $d(n)$ input bits.*

Goldreich's Pseudorandom Generator. Goldreich's candidate local PRGs form a family $\mathcal{F}_{G,P}$ of local PRGs: $\mathsf{PRG}_{G,P} : \{0,1\}^n \mapsto \{0,1\}^m$, parametrized by an (n, m, d)-hypergraph $G = (\sigma^1, \ldots, \sigma^m)$ (where $m = m(n)$ is polynomial in n), and a predicate $P : \{0,1\}^d \mapsto \{0,1\}$, defined as follows: on input $x \in \{0,1\}^n$, $\mathsf{PRG}_{G,P}$ returns the m-bit string $(P(x_{\sigma^1_1}, \ldots, x_{\sigma^1_d}), \ldots, P(x_{\sigma^m_1}, \cdots, x_{\sigma^m_d}))$.

The Lombardi-Vaikuntanathan (LV) Predicate. For concreteness, we will rely on Goldreich PRG instantiated with the following predicate:

$$\mathsf{P}_{\mathsf{LV}}(x_1, x_2, x_3, x_4, x_5) = x_1 \oplus x_2 \oplus (x_1 \oplus x_3)(x_2 \oplus x_4) \oplus x_5\ .$$

This predicate leads to a PRG with locality five. This predicate was introduced and studied in [29], were it was shown that it provably resists all $\mathbb{F}_2$-linear attacks, as well as all attacks using the SDP hierarchies (such as the Lassere-Parrilo sum-of-squares hierarchy), when stretching n bits to $n^{1.25-\varepsilon}$ bits. In addition, this predicate enjoys an optimaly low arithmetic degree, since it can be computed by the following degree 3 polynomial over the integers:

$$\mathsf{P}_{\mathsf{LV}}(x_1, x_2, x_3, x_4, x_5) = x_5 + (x_1(x_4 - 1) + x_2(x_1 + x_3 - 1) - x_3 x_4) \cdot (2x_5 - 1)\ .$$

3 NIZK Based on the Security of CDH and Strong OW-KDM Security of ElGamal

In this section, we describe a construction of a NIZK from the strong OW-KDM security of ElGamal with respect to efficient functions by assuming the CDH problem is *hard* to solve. We first provide a NIZK for the specific language of the Diffie-Hellman (DH) language. This is done by constructing a CIH based on the strong OW-KDM security of ElGamal for the natural sparse relation induced by the Σ-protocol for DH languages. We then show that such a NIZK for the DH language allows us to construct a VPRG, which in return, allows us to construct a NIZK for all of NP by Theorem 7.

3.1 Σ-Protocol for the Diffie-Hellman Language

Definition 19 (Diffie-Hellman Language). *Let $\mathbb{G}$ be a group with prime order p. We define the Diffie-Hellman (DH) language $\mathscr{L}_{\mathsf{DH},t}$ parameterized by $t \in \mathbb{Z}_p^*$ as $\mathscr{L}_{\mathsf{DH},t} = \{(g, h, g^x, h^x)\ :\ g, h \in \mathbb{G}, x \in \mathbb{Z}_p, \mathsf{dlog}_g h = t\}$.*

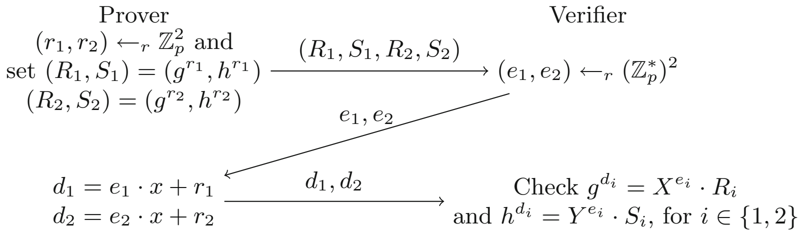
Prover
Verifier
$(r_1, r_2) \leftarrow_r \mathbb{Z}_p^2$ and
set $(R_1, S_1) = (g^{r_1}, h^{r_1})$
$(R_2, S_2) = (g^{r_2}, h^{r_2})$
(R_1, S_1, R_2, S_2)
$(e_1, e_2) \leftarrow_r (\mathbb{Z}_p^*)^2$
e_1, e_2
$d_1 = e_1 \cdot x + r_1$
$d_2 = e_2 \cdot x + r_2$
d_1, d_2
Check $g^{d_i} = X^{e_i} \cdot R_i$
and $h^{d_i} = Y^{e_i} \cdot S_i$, for $i \in \{1, 2\}$

Fig. 2. Σ-protocol for the Diffie-Hellman language for the word $(g, h, X = g^x, Y = g^y)$.

Below we recall the standard Σ-protocol for the DH relation (with parallel repetition). Here, the word is $(g, h, X, Y) \in \mathscr{L}_{\mathsf{DH},t}$ where $(X, Y) = (g^x, h^x)$.

The above Σ-protocol achieves the standard notion of correctness and special honest-verifier zero-knowledge. Adaptive soundness is covered by the following lemma, the proof is standard and provided for completeness in the full version of the paper.

Lemma 20 (Adaptive Soundness). *The Σ-protocol in Fig. 2 satisfies $\frac{1}{p-1}$-adaptive soundness.*

3.2 Correlation-Intractable Hash Function H

Let λ be a security parameter. We consider a group $\tilde{\mathbb{G}}$ of order $q(\lambda)$ with $\lceil \log q \rceil \approx \lambda$. Let $\mathsf{Trunc} : \tilde{\mathbb{G}} \mapsto \{0,1\}^{\lambda/2}$ be the function which, on input a group element $\tilde{G} \in \tilde{\mathbb{G}}$, parses it as a $\lceil \log q \rceil$-bit string and returns the first $\lambda/2$ bits of its input. We consider the following hash function $\mathsf{H} : \tilde{\mathbb{G}}^2 \times \mathbb{Z}_q \mapsto \{0,1\}^{\lambda/2}$:

- Sampling the key: pick $(\tilde{G}, k, m) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q^2$ and set $\tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, m)$. Note that the key distribution is exactly the uniform distribution over $\tilde{\mathbb{G}}^2$.
- Evaluating $\mathsf{H}(\tilde{\mathbf{C}}, \cdot) : \mathsf{H}(\tilde{\mathbf{C}}, x) = \mathsf{Trunc}(\mathsf{HalfDec}(x, \tilde{\mathbf{C}}))$.

Correlation-Intractability of H. Consider a group $\mathbb{G}$ of order $p(\lambda)$ with $\lceil \log p \rceil \approx \lambda/4$. Then the output of H can be interpreted as two elements of $\mathbb{G}$. Fix a parameter $t \in \mathbb{Z}_p^*$. Define $\mathcal{R}_{\lambda,t}$ to be the natural sparse relation associated to the language $\mathscr{L}_{\mathsf{DH},t}$ (see Lemma 14). That is,

$$\mathcal{R}_{\lambda,t} = \{(\alpha, \beta) \in \mathbb{G}^4 \times (\mathbb{Z}_p^*)^2 : \exists x, \gamma \text{ s.t. } x \notin \mathscr{L}_{\mathsf{DH},t} \ \wedge \ V(x, \alpha, \beta, \gamma) = \text{accept}\}.$$

Here, the above relation can also be described alternatively using the following (inefficient) randomized function:

$$f_t(a; z) : \begin{cases} \mathbb{G}^4 \times \mathbb{Z}_p^* \mapsto (\mathbb{Z}_p^*)^2 \\ (R_1, S_1, R_2, S_2) \times z \to (z, \log_{(R_1^t/S_1)}(R_2^t/S_2) \cdot z) \end{cases} .$$

The following is the main contribution of this section.

Theorem 21. *Assume that ElGamal satisfies* $2^{-3\lambda/4}$-OW-KDM *security with respect to efficient functions. Then the hash family* $\{\mathsf{H} : \mathsf{H} : \tilde{\mathbb{G}}^2 \times \mathbb{Z}_q \mapsto \{0,1\}^{\lambda/2}\}_\lambda$ *is correlation-intractable with respect to* $\mathcal{R}^{\mathsf{H}} := \{\mathcal{R}_\lambda := \{\mathcal{R}_{\lambda,t}\}_t\}_\lambda$.

Proof. We prove the theorem in two steps. We first show that an adversary against the correlation intractability of H can be shown to be an adversary against the OW-KDM security of ElGamal with respect to *inefficient* functions. We then show via the symmetry of messages and secret keys of ElGamal to conclude that such an adversary can indeed be used to break OW-KDM security of ElGamal with respect to *efficient* functions. The first step is summarized in the following lemma.

Lemma 22. *Let* $\mathcal{A}$ *be an adversary against the* $\mathcal{R}^{\mathsf{H}}$*-correlation intractability of* H *with (non-negligible) advantage* $\varepsilon(\lambda)$*. Then, for some* $t \in \mathbb{Z}_p^*$*, it holds that:*

$$\Pr_{\substack{(\tilde{G},a^*,m)\leftarrow_r \tilde{\mathbb{G}}\times\mathbb{Z}_q^2 \\ \tilde{\mathbf{C}}\leftarrow_r \mathsf{Enc}_{\tilde{G}}(a^*,m)}}[\mathcal{A}(\tilde{G},\tilde{\mathbf{C}}) = a^* | (a^*, \mathsf{H}(\tilde{\mathbf{C}}, a^*)) \in \mathcal{R}_{\lambda,t}] \geq \frac{\varepsilon(\lambda)}{2^{3\lambda/4}}.$$

The proof follows closely the approach of [7], but simplifies some steps of the proof and makes the exact security loss explicit. We provide it in the full version of the paper. Given Lemma 22, it remains to show that this implies a contradiction to the OW-KDM security of ElGamal for *efficient* functions. The main difficulty here is that the above can be rewritten as

$$\Pr_{\substack{(\tilde{G},a^*)\leftarrow_r \tilde{\mathbb{G}}\times\mathbb{Z}_q \\ m\leftarrow_r \alpha_t(\tilde{G},a^*) \\ \tilde{\mathbf{C}}\leftarrow_r \mathsf{Enc}_{\tilde{G}}(a^*,m)}}[\mathcal{A}(\tilde{G},\tilde{\mathbf{C}}) = a^*] \geq \frac{\varepsilon(\lambda)}{2^{3\lambda/4}}. \tag{1}$$

with $\alpha_t : \tilde{\mathbb{G}} \times \mathbb{Z}_q \times \{0,1\}^{\lambda/2} \times \mathbb{Z}_p^* \mapsto \mathbb{Z}_q$, such that $\alpha_t(\tilde{G}, a; z_1, z_2) = \mathsf{dlog}_{\tilde{G}}(f_t(a; z_2)||z_1)$. which naturally translates to an adversary against the KDM security of ElGamal where m is sampled as $\alpha_t(\tilde{G}, a^*; z_1, z_2)$, which is not an efficiently computable function. We show below how to get around this apparent issue. Define the (randomized) efficiently computable function f_t^{-1} as follows:

$$f_t^{-1}(e_1, e_2; r_1, r_2, s_1) := \begin{cases} (\mathbb{Z}_p^*)^2 \times \mathbb{G}^3 \mapsto \mathbb{G}^4 \\ (e_1, e_2; r_1, r_2, s_1) \to (g^{r_1}, g^{s_1}, g^{r_2}, g^{\frac{e_2(t\cdot r_1 - s_1)}{e_1} - t\cdot r_2}). \end{cases}$$

Furthermore, define F_t to be the following (efficient, randomized) function:

$$F_t : \begin{cases} \tilde{\mathbb{G}} \times \mathbb{Z}_q \times \{0,1\}^{\lambda/2} & \mapsto \mathbb{Z}_q \\ (\tilde{G}, m; z) & \to f_t^{-1}(\mathsf{Trunc}(\tilde{G}^m); z), \end{cases}.$$

where we assume in case the first $\lambda/4$-bits of $\mathsf{Trunc}(\tilde{G}^m)$ corresponds to $0 \in \mathbb{Z}_p$, then it outputs some fixed element in $\mathbb{Z}_q$. Consider now the distribution obtained by sampling $(\tilde{G}, a^*) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q$, $m \leftarrow_r \alpha_t(\tilde{G}, a^*)$, and outputting

$\tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(a^*, m)$. Observe that we obtain the same distribution (up to some negligible difference) by first sampling $(\tilde{G}, m) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q$, setting $k \leftarrow_r F_t(\tilde{G}, m)$, and outputting $\tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, m)$. We build upon this observation to construct, using $\mathcal{A}$, an adversary against the one-query $\mathsf{OW\text{-}KDM}$ security of ElGamal with respect to the class of (efficient, randomized) functions $\{F_t\}_t$. Let $\mathcal{A}$ be the previous adversary, which satisfies Eq. 1. By our observation above, this can be rewritten as

$$\Pr_{\substack{(\tilde{G},k) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q \\ a^* \leftarrow_r F_t(\tilde{G},k) \\ \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(a^*,k)}} [\mathcal{A}(\tilde{G}, \tilde{\mathbf{C}}) = a^*] \geq \frac{\varepsilon(\lambda)}{2^{3\lambda/4}}.$$

We build an adversary $\mathcal{B}$ against the $\mathsf{OW\text{-}KDM}$ security of ElGamal as follows: on input $(\tilde{G}, \tilde{\mathbf{C}})$, $\mathcal{B}$ parses $\tilde{\mathbf{C}}$ as $(\tilde{C_0}, \tilde{C_1})$. $\mathcal{B}$ sets $\tilde{G'} \leftarrow \tilde{C_0}$ and $\tilde{\mathbf{C}}' \leftarrow (\tilde{G}, \tilde{C_1})$. Then, $\mathcal{B}$ runs $\mathcal{A}(\tilde{G'}, \tilde{\mathbf{C}}')$ and outputs whatever $\mathcal{A}$ outputs. Observe that the distributions

$$\{(\tilde{G}, \tilde{\mathbf{C}}) \ : \ (\tilde{G}, k) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q, a^* \leftarrow_r F_t(\tilde{G}, k), \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(a^*, k)\},$$

which corresponds to the experiment in the previous probability, and

$$\{(\tilde{C_0}, (\tilde{G}, \tilde{C_1})) \ : \ (\tilde{G}, k) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q, a^* \leftarrow_r F_t(\tilde{G}, k), (\tilde{C_0}, \tilde{C_1}) \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, a^*)\}$$

are identical. Therefore,

$$\Pr_{\substack{(\tilde{G},k) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q \\ a^* \leftarrow_r F_t(\tilde{G},k) \\ \tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k,a^*)}} [\mathcal{B}(\tilde{G}, \tilde{\mathbf{C}}) = a^*] \geq \frac{\varepsilon(\lambda)}{2^{3\lambda/4}},$$

which contradicts the (one-query) $2^{-3\lambda/4}$-$\mathsf{OW\text{-}KDM}$ security of ElGamal with respect to the family of (efficient, randomized) functions $\{F_t\}_t$.

3.3 NIZK for $\mathscr{L}_{\mathsf{DH}}$ via $\mathcal{R}^{\mathsf{H}}$-Correlation-Intractability

Lemma 23. *Our $\mathcal{R}^{\mathsf{H}}$-correlation intractable hash function family is programmable.*

The proof is given in the full version.

Theorem 24 (NIZK for $\mathscr{L}_{\mathsf{DH}}$). *Assume there exists a programmable correlation intractable hash family for relation $\mathcal{R}^{\mathsf{H}}$. Then, there exists an adaptively sound and selective single-theorem zero-knowledge NIZK argument system for the Diffie-Hellman language $\mathscr{L}_{\mathsf{DH},t}$ for any $t \in \mathbb{Z}_p^*$. Moreover, our NIZK is independent of the value t and all algorithms can be run oblivious of the value t.*

The proof follows in a relatively natural way by compiling the Σ-protocol for DDH with the correlation-intractable hash function H. We provide an explicit description of the proof system and a security analysis in the full version. As stated in Theorem 24, our NIZK for $\mathscr{L}_{\mathsf{DH},t}$ is agnostic of the value of $t \in \mathbb{Z}_p^*$, since the value of t is only significant during the security proof. Therefore, whenever the meaning is clear, we will drop the subscript t and simply state it as an NIZK for $\mathscr{L}_{\mathsf{DH}}$. The important thing to keep in mind is that for each crs generated by $\mathsf{Setup}^{\mathsf{DH}}$, it is only adaptive secure for $\mathscr{L}_{\mathsf{DH},t}$ with a *fixed* t.

3.4 VPRG from NIZK for $\mathscr{L}_{\mathsf{DH}}$

Our construction relies on the CDH assumption and the NIZK argument system $(\mathsf{Setup}^{\mathsf{DH}}, \mathsf{Prove}^{\mathsf{DH}}, \mathsf{Verify}^{\mathsf{DH}})$ for $\mathscr{L}_{\mathsf{DH}}$ from the previous section. We prepare a predicate $B : \mathbb{G}^2 \mapsto \{0,1\}$ satisfying the following property: given (g^a, g^b), computing $B(g^b, g^{ab})$ should be as hard (up to polynomial factors) as computing (g^b, g^{ab}). Note that this implies that distinguishing $B(g^b, g^{ab})$ from a random bit given random tuple (g^a, g^b) is as hard as solving CDH. One way to instantiate such a predicate is to use the Goldreich-Levin hard-core predicate [19].

Construction. Let $m := m(\lambda)$ be an arbitrary polynomial. Our construction of VPRG proceeds as follows:

- $\mathsf{Setup}(1^\lambda, m)$: run $\mathcal{G} = (\mathbb{G}, p) \leftarrow_r \mathsf{DHGen}(1^\lambda)$ and sample $g \leftarrow_r \mathbb{G}$. Further, for $i = 1$ to m, pick $h_i \leftarrow_r \mathbb{G}$ and generate $\mathsf{crs}_i \leftarrow_r \mathsf{Setup}^{\mathsf{DH}}(1^\lambda)$. Finally, output $\mathsf{pp} = (g, (h_i, \mathsf{crs}_i)_{i \le m})$.
- $\mathsf{Stretch}(\mathsf{pp})$: pick $\tau \leftarrow_r \mathbb{Z}_p$, set $\mathsf{pvk} \leftarrow g^\tau$, and for $i = 1$ to m, set $x_i \leftarrow B(\mathsf{pvk}, h_i^\tau)$. Output $(\mathsf{pvk}, x = (x_i)_{i \le m}, \mathsf{aux} = \tau)$.
- $\mathsf{Prove}(\mathsf{pp}, \mathsf{aux}, i)$: set $\tau := \mathsf{aux}$ and run $\boldsymbol{\pi}_i^{\mathsf{DH}} \leftarrow_r \mathsf{Prove}^{\mathsf{DH}}(\mathsf{crs}_i, (g, h_i, \mathsf{pvk}, h_i^\tau), \tau)$. Output $\pi = (h_i^\tau, \boldsymbol{\pi}_i^{\mathsf{DH}})$.
- $\mathsf{Verify}(\mathsf{pp}, \mathsf{pvk}, i, b, \pi)$: parse $(u, \boldsymbol{\pi}^{\mathsf{DH}}) \leftarrow \pi$. If $b = B(\mathsf{pvk}, u)$, then return $\mathsf{Verify}^{\mathsf{DH}}(\mathsf{crs}_i, (g, h_i, \mathsf{pvk}, u), \boldsymbol{\pi}^{\mathsf{DH}})$. Otherwise, return 0.

Security Analysis. Correctness of the VPRG follows from the correctness of the underlying NIZK. In addition, the size of the verification key g^τ is p, and in particular, is independent of m. Hence, we can set the stretch $\delta := \delta(\lambda)$ to be an arbitrary polynomial, where we can set $m = s^{1+\delta}$ by definition.

Theorem 25. *If the CDH assumption holds relative to* DHGen *and the* NIZK *argument system for the Diffie-Hellman language* $\mathscr{L}_{\mathsf{DH}}$ *is adaptive sound and selective single-theorem, then the above construction provides a* (δ, s)-VPRG *that is binding and hiding, where* δ *is an arbitrary polynomial in the security parameter* λ *and* $s = |\mathbb{G}|$.

The binding property is shown by guessing the position where the adversary forges an opening, and showing that this implies an adversary against the adaptive soundness of the NIZK for DDH. Hiding relies on a careful modification of the CRS generation, together with the zero-knowledge property of the NIZK for DDH. We provide a complete proof in the full version of the paper. As a direct consequence of Theorems 7, 31, 33, and 38, the following is obtained.

Theorem 26. *Assume that the CDH assumption holds relative to* DHGen *and that ElGamal satisfies* $2^{-3\lambda/4}$-OW-KDM *security with respect to efficient functions, then there exists an adaptive sound and adaptive multi-theorem* NIZK *for all of* NP.

4 NIZK from Insecurity of CDH and Strong $\mathsf{OW\text{-}KDM}$ Security of ElGamal

In this section, we describe a construction of an *infinitely often* NIZK from the strong $\mathsf{OW\text{-}KDM}$ security of ElGamal with respect with efficient functions by assuming that the CDH problem is *easy* to solve. We first provide a NIZK for the specific language of the *almost-short* language. This is done by constructing a CIH based on the strong $\mathsf{OW\text{-}KDM}$ security of ElGamal for the natural sparse relation induced by the Σ-protocol for the almost-short language. We then show that such a NIZK for the almost-short language along with the short-exponent discrete-log (SEDL) assumption allows us to construct a VPRG, which in return, allows us to construct an (infinitely often) NIZK for all of NP by Theorem 7. Note that, as we will show, SEDL is not an extra assumption since it follows from the strong $\mathsf{OW\text{-}KDM}$ security of ElGamal.

4.1 Σ-Protocol for the Language of Almost-Short Elements

In this section, we introduce the language $\mathscr{L}_{\alpha,\beta}$ of elements of $\mathbb{G}$ with (α, β)-*almost-short* exponents to be the subset of $\mathbb{G}$ containing elements of the form g^x where x is *almost-short*. We say that x is (α, β)-almost-short if there exists a short value $v \leq 2^\beta$ such that vx is short as well: $vx \in [\pm 2^\alpha]$. More formally:

Definition 27 ((α, β)-Almost-Shortness). *Let $\mathbb{G}$ be a group of prime order p. We define $\mathscr{L}_{\alpha,\beta}$ over $\mathbb{G}$ with respect to the generator $g \in \mathbb{G}$ to be the language of (α, β)-almost-short elements as:*

$$\mathscr{L}_{\alpha,\beta} = \{g^x \mid x = u \cdot v^{-1} \in \mathbb{Z}_p, \mathsf{int}(u) \in [\pm 2^\alpha], \mathsf{int}(v) \in [2^\beta]\}.$$

A Σ-Protocol for the Almost-Short Language. We start by introducing a simple Σ-protocol for proving membership of an element $g^x \in \mathbb{G}$ to $\mathscr{L}_{\alpha,\beta}$. The protocol satisfies the following relaxed notion of correctness: an honest prover is guaranteed to produce an accepting proof if the input word g^x is such that $x \leq 2^\ell$ (with $\log p \gg \ell$), but soundness only guarantees that the word actually belongs to $\mathscr{L}_{\ell',c}$, where c is the challenge length, and $\ell' > c + \ell + \kappa$, for some statistical security parameter κ.[4] The protocol is represented on Fig. 3. Note that it only satisfies selective soundness.

In the full version, we prove the following lemmas:

Lemma 28 (Correctness). *If $x \in [0, 2^\ell]$, and $\ell' > \max\{c, \ell\} + \kappa$, then the Σ-protocol from Fig. 3 is correct (and the verifier accepts with probability greater than $1 - \frac{1}{2^\kappa}$).*

[4] This is similar in spirit to various Σ-protocols for lattice-based relations, where the Σ-protocol proves knowledge of a short preimage, but the protocol has some slackness, *i.e.*, a gap between the shortness needed for the honest proof to be accepted, and the shortness actually guaranteed by the soundness property; here, we have an additional "slackness" in that x is only guaranteed to be the product of a short value with the inverse of another short value.

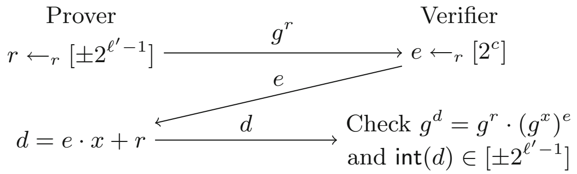
Prover
Verifier
$r \leftarrow_r [\pm 2^{\ell'-1}]$
g^r
$e \leftarrow_r [2^c]$
e
$d = e \cdot x + r$
d
Check $g^d = g^r \cdot (g^x)^e$
and $\mathsf{int}(d) \in [\pm 2^{\ell'-1}]$

Fig. 3. Σ-protocol for the almost-shortness language, for the word g^x. In a honest run, the prover posseses a short witness $x \in [0, 2^\ell]$

Lemma 29 (Selective Soundness). *If $X \notin \mathscr{L}_{\ell',c}$, then the probability that the verifier accepts is at most $\frac{1}{2^c}$.*

Lemma 30 (Honest-Verifier Zero-Knowledge). *When $\ell' > c + \ell + \kappa$ and $x \in [0, 2^\ell)$, the Σ-protocol in Fig. 3 is honest-verifier zero-knowledge for words in $x \in [0, 2^\ell]$. In particular, the statistical distance between honest transcripts and those produced by the simulator described in the full version is $\frac{1}{2^\kappa}$.*[5]

Adaptive Soundness. The above protocol only enjoys selective soundness, which does not suffice in our context. As for our previous construction, however, adaptive soundness can be obtained using sufficiently many parallel repetitions of the underlying Σ-protocol, via standard complexity leveraging: since there are p possible words g^x, if the above Σ-protocol is amplified N-times with $N \geq \lceil \log p \rceil / c$, then it is $p/2^{N \cdot c}$-adaptively sound. We denote $\Pi_N(p, \ell, \kappa, c)$ the Σ-protocol obtained by repeating N times in parallel the above Σ-protocol for $\mathscr{L}_{\ell',c}$, with $\ell' = \ell + c + \kappa + 1$. When (p, ℓ, κ, c) are clear from the context, we simply denote it Π_N.

Admissible First Flow. Given a Σ-protocol for a language $\mathscr{L}$, we say that a candidate first flow a is *(adaptively) admissible* if there exists a word $X \notin \mathscr{L}$, a challenge e, and an answer d, such that (a, e, d) form an accepting transcript for X. Note that in Π_N, there are p^N possible first flows, but only $p \cdot 2^{N(\ell'+c)}$ admissible first flows, since an admissible first flow is of the form $(g^{d_i}/(g^x)^{e_i})_{i \leq N}$, for some $d_i \in [\pm 2^{\ell'-1}]$, $e_i \in [2^c]$, and $g^x \in \mathbb{G}$.

4.2 Correlation-Intractable Hash Function

Let λ be a security parameter and fix parameters $(N(\lambda), c(\lambda), p(\lambda), \ell(\lambda), \kappa(\lambda))$. We consider a group $\tilde{\mathbb{G}}$ of order $q(\lambda)$ with $\lceil \log q \rceil \approx \lambda$, and a group $\mathbb{G}$ of order $p(\lambda)$. Let $\mathsf{Trunc}' : \tilde{\mathbb{G}} \mapsto \{0,1\}^{N \cdot c}$ be the function which, on input a group element $\tilde{G} \in \tilde{\mathbb{G}}$, parses it as a $\lceil \log q \rceil$-bit string and returns the first $N \cdot c$ bits of its input. Let $\mathsf{h} : \mathbb{G}^N \to \{0,1\}^\lambda$ be a 2-universal hash function, for a security

[5] To be precise, this does not meet the definition of our honest-verifier zero-knowledge since we only consider a small set of $\mathscr{L}_{\ell',c}$. However, this notion suffices for our application.

parameter λ which will be defined afterward. We consider the following hash function $\mathsf{H}'_\lambda : \tilde{\mathbb{G}}^2 \times \mathbb{G}^N \mapsto \{0,1\}^{N \cdot c}$:

- Sampling the key: pick $(\tilde{G}, k, m) \leftarrow_r \tilde{\mathbb{G}} \times \mathbb{Z}_q^2$ and set $\tilde{\mathbf{C}} \leftarrow_r \mathsf{Enc}_{\tilde{G}}(k, m)$. Note that the key distribution is exactly the uniform distribution over $\tilde{\mathbb{G}}^2$.
- Evaluating $\mathsf{H}'_\lambda(\tilde{\mathbf{C}}, \cdot) : \mathsf{H}'_\lambda(\tilde{\mathbf{C}}, x) = \mathsf{Trunc}'(\mathsf{HalfDec}(\mathsf{h}(x), \tilde{\mathbf{C}}))$.

Setting the Security Parameter λ. Let $\mathcal{R}_\lambda(N, c, p, \ell, \kappa) = \mathcal{R}_\lambda$ be the natural sparse relation associated to the language $\mathscr{L}_{\ell',c}$ over $\mathbb{G}$ with respect to a generator $g \in \mathbb{G}$, where $\ell' = \ell + c + \kappa$ (see Lemma 14). That is,

$$\mathcal{R}_\lambda = \{(a, b) \in \mathbb{G}^N \times \{0,1\}^{N \cdot c} : \exists X, d \text{ s.t. } X \notin \mathscr{L}_{\ell',c} \;\wedge\; V(X, a, b, c) = \text{accept}\},$$

where V is the verifier from the Σ-protocol for the language $\mathscr{L}_{\ell',c}$ in Fig. 3. The purpose of the 2-universal hash function h in our correlation-intractable hash H'_λ is to compress the size of the first flow to λ bits, without significantly decreasing the winning probability of the adversary. The core observation is that when the adversary manages to output a such that $(a, \mathsf{H}'_\lambda(\tilde{\mathbf{C}}, a)) \in \mathcal{R}_\lambda$, then a must at least be an admissible first flow. Since there are at most $p \cdot 2^{N(\ell'+c)}$ admissible first flows, we set $\lambda \leftarrow \lceil \log p \rceil + N(\ell' + c) + \kappa$, where κ is some statistical security parameter. Then, the 2-universality of h guarantees that, except with probability at most $2^{-\kappa}$ over the random choice of the hash key, all possible λ-bit strings will have at most a single admissible preimage a. In the following, we denote by Invh the (inefficient) function which, on input a λ-bit string s, outputs the unique admissible preimage of s (or $\perp$ if s has no admissible preimage).

Correlation-Intractability of H'

Theorem 31. *Fix parameters $(N(\lambda), c(\lambda), p(\lambda), \ell(\lambda), \kappa(\lambda))$. Assume that ElGamal satisfies $p^{-1} \cdot 2^{Nc-\lambda}$-$\mathsf{OW\text{-}KDM}$ security with respect to efficient functions. Then the hash family $\{\mathsf{H}'_\lambda : \tilde{\mathbb{G}}^2 \times \mathbb{G}^N \mapsto \{0,1\}^{N \cdot c}\}_\lambda$ is correlation-intractable with respect to $\mathcal{R}^{\mathsf{H}'} := \{\mathcal{R}_\lambda(N, c, p, \ell, \kappa)\}_\lambda = \{\mathcal{R}_\lambda\}_\lambda$.*

The structure of the proof is similar to the proof of Theorem 21, the core difference being that we rely on a 2-universal hash function to compress the size of the first flow, and only guess the compressed hash; then, we rely on the fact the 2-universal hash is injective with high probability over the set of admissible first flow. We provide a detailed proof in the full version.

4.3 NIZK for the Almost-Short Language via $\mathcal{R}^{\mathsf{H}'}$-Correlation-Intractability

Lemma 32. *Our $\mathcal{R}^{\mathsf{H}'}$-correlation intractable hash function family is programmable.*

The proof is essentially identical to the proof for $\mathcal{R}^{\mathsf{H}}$.

Theorem 33 (NIZK for the almost-short language $\mathscr{L}_{\ell',c}$). *Assume there exists a programmable correlation intractable hash family for the relation $\mathcal{R}^{\mathsf{H}'}$. Then, there exists an adaptive sound and selective single-theorem zero-knowledge NIZK argument system for the almost-short language $\mathscr{L}_{\ell',c}$.*

The proof of adaptive soundness and selective single-theorem zero-knowledge are essentially identical to the proof of Theorem 24. We provide an explicit description of the NIZK proof system in the full version.

4.4 A Commitment Scheme from the Short-Exponent Discrete Logarithm Assumption

Before providing our VPRG construction, we introduce one last set of tools. We first introduce the T-short-exponent discrete-logarithm (T-SEDL) assumption and then provide a simple commitment scheme based on T-SEDL.

Definition 34. *The T-SEDL assumption over an Abelian group $\mathbb{G}$ of order p with respect to the generator g states that for every PPT $\mathcal{A}$,*

$$\Pr[x \leftarrow_r [p/T], h \leftarrow g^x \ : \ \mathcal{A}(h) = x] \approx 0.$$

It is well known that under the T-SEDL assumption, it is infeasible to distinguish $\{g^x \mid x \leftarrow_r [p/T]\}$ from the uniform distribution over $\mathbb{G}$ [27].

A Commitment from T-SEDL. A commitment scheme is a pair of algorithms ($\mathsf{Commit}, \mathsf{Open}$) such that given $(c, d) \leftarrow_r \mathsf{Commit}(m)$, c hides m (more formally, no adversary can distinguish whether c was output by $\mathsf{Commit}(m)$ or $\mathsf{Commit}(m')$, for two messages (m, m') of their choice), but d binds the committer to m (more formally, no adversary can find (c, d, d', m, m') with $m \neq m'$ such that $\mathsf{Open}(c, d, m) = \mathsf{Open}(c, d', m') = 1$). We now introduce the bit commitment scheme that will underly our construction. Let $\mathbb{G}$ be a group of order p. Fix some integers (ℓ, k). $\mathsf{Commit}(b)$, on input a bit b, picks $w \leftarrow_r \{0,1\}^\ell$ and outputs $\mathsf{com} = g^{w+2^k b}$. Opening the commitment is done by revealing w. The commitment is perfectly binding, and hiding under the $p/2^\ell$-SEDL assumption.

From T-SEDL to Strong OW-KDM Security of ElGamal. In the full version of the paper, we show the following, which states that T-SEDL will be redundant with our other assumptions:

Lemma 35. *Assume that ElGamal satisfies* $(1/T)$-OW-KDM *security with respect to efficient functions. Then the T-SEDL assumption holds.*

Binding Property with Almost-Short Randomness. A useful property of the above commitment, which will play a crucial role in our construction, is that it remains computationally binding if instead of revealing w, the opener reveals b and proves (using any computationally binding argument) that $\mathsf{com} \cdot g^{-2^k b} \in \mathscr{L}_{\alpha,\beta}$, provided that $k \geq \alpha + 2$ and under some condition on the size p of the group. We elaborate below.

Lemma 36. *Let* $\mathsf{com} = g^{w+2^{\alpha+2}b}$ *be a commitment to* b*, where* $g^w \in \mathscr{L}_{\alpha,\beta}$*. Further assume that* $p > 2^{\alpha+2\beta+4}$*. Then no computationally bounded prover can produce an accepting argument that* $g^{w+2^{\alpha+2}} \in \mathscr{L}_{\alpha,\beta}$*.*

Looking ahead, we will use this lemma together with a NIZK with relaxed correctness for the language $\mathscr{L}_{\alpha,\beta}$ to guarantee correct opening of the above commitment. The relaxed correctness requirement is the same as in Sect. 4.1 and will be satisfied when the commitment is constructed honestly.

Proof. Let $\mathsf{com} \in \mathbb{G}$ be any group element. We prove that it can never simultaneously hold that $\mathsf{com} \in \mathscr{L}_{\alpha,\beta}$ and $\mathsf{com} \cdot g^{2^{\alpha+2}} \in \mathscr{L}_{\alpha,\beta}$. Assume toward contradiction that both com and $\mathsf{com} \cdot g^{2^{\alpha+2}}$ belong to $\mathscr{L}_{\alpha,\beta}$. Let $x \leftarrow \mathsf{dlog}_g(\mathsf{com})$. Then we have:

$$x = u \cdot v^{-1} \bmod p \text{ for some } u \in [\pm 2^{\alpha}], v \in [2^{\beta}],$$
$$x + 2^{\alpha+2} = u' \cdot (v')^{-1} \bmod p \text{ for some } u' \in [\pm 2^{\alpha}], v' \in [2^{\beta}].$$

Hence, $uv^{-1} + 2^{\alpha+2} = u'(v')^{-1} \bmod p$, which gives $v'(u + 2^{\alpha+2}v) = u'v \bmod p$. However, since $p > 2^{\alpha+2\beta+4}$, we have that this equation holds over the integers as well. This implies (still using the bound on p) that $v'(u + 2^{\alpha+2}v) = u'v \leq 2^{\alpha}v$. However, $u + 2^{\alpha+2}v \geq 2^{\alpha+2}v - 2^{\alpha} > 2^{\alpha}v$ (since $v \geq 1$). Therefore, we also get $2^{\alpha}v < v'(u + 2^{\alpha+2}v)$ (since $v' \geq 1$), which is a contradiction. Therefore, no bounded prover can provide an accepting argument of membership in $\mathscr{L}_{\alpha,\beta}$ (with any computationally sound argument system) for both com and $\mathsf{com} \cdot g^{2^{\alpha+2}}$.

4.5 A VPRG from NIZK for the Almost Short Language and the SEDL Assumption

With the tools we introduced, we are now ready to present our construction of a VPRG in a group where CDH is insecure.

Intuition of the Construction. Let DHGen be a deterministic algorithm that, on input 1^{λ}, returns a description $\mathcal{G} = (\mathbb{G}, p)$ where $\mathbb{G}$ is a cyclic group of prime order p. Assume that CDH does *not* hold with respect to DHGen. In the full version, we show that this means that there exists a strong CDH solver that allows to compute "self-pairings" over $(\mathbb{G}, p) = \mathsf{DHGen}(1^{\lambda})$ with negligible error probability, for infinitely many security parameters λ. We denote $(\mathsf{EvalCom}, \mathsf{EvalOpen})$ the self-pairing algorithm, which evaluates integer arithmetic circuits (IAC) in the exponent, together with the evaluation algorithm "in the clear" $\mathsf{EvalOpen}$, satisfying the following:

Theorem 37. *Let* $\{\mathcal{C}_{\lambda}\}_{\lambda \in \mathbb{N}}$ *be an ensemble of sets of IAC (with gates* $(+, \times, -)$*) where each circuit in* $\mathcal{C}_{\lambda}$ *has input length* $n = n(\lambda)$ *and size* $L = L(\lambda)$*. Let the CDH assumption relative to* DHGen *be easy. Moreover, let* $S \subset \mathbb{N}$ *be the infinite set of security parameters for which a strong CDH solver exists. Then there exists a PPT algorithm* $\mathsf{EvalCom}$ *and a deterministic polytime algorithm* $\mathsf{EvalOpen}$ *with the following properties for all* $\lambda \in S$*:*

- $\mathsf{EvalCom}(C, g_1, \cdots, g_n) \rightarrow h$: *on input an IAC* $C \in \mathcal{C}_\lambda$ *and* $(g_1, \cdots, g_n) \in \mathbb{G}$, *it outputs* $h \in \mathbb{G}$.
- $\mathsf{EvalOpen}(C, z_1, b_1, \cdots, z_n, b_n) \rightarrow z$: *on input an IAC* $C \in \mathcal{C}_\lambda$ *and* $((z_1, b_1), \cdots, (z_n, b_n)) \in (\mathbb{Z} \times \{0,1\})^n$, *it outputs* $z \in \mathbb{Z}$.
- *Let* $(\ell, t) \in \mathbb{N}^2$ *such that* $\ell + t > 2L^2$. *Further, assume* $p = |\mathbb{G}|$ *to be greater than* $L(\ell + t) \cdot \log_2 B$ *where* $B = \max_{C \in \mathcal{C}_\lambda, (b_i \in \{0,1\})_i} C(b_1, \cdots, b_n)$. *Let* $b_i \in \{0,1\}$ *and* $w_i \in [-2^\ell, 2^\ell]$ *for all* $i \in [n]$. *Then, for any* $C \in \mathcal{C}_\lambda$ *with degree* D *and* $g_i = g^{w_i + 2^{\ell+t} b_i}$, *if we run* $h \leftarrow_r \mathsf{EvalCom}(C, g_1, \cdots, g_n)$, *we have* $\mathsf{dlog}_g h = w^* + 2^{(D+1)(\ell+t)} \cdot C(b_1, \cdots, b_n)$, *where* $w^* \in [\pm(2^{D(\ell+L+t+2)})]$ *and* $\mathsf{EvalOpen}(C, (w_i, b_i)_{i \le n}) = w^*$, *except with negligible probability* $2^{-\lambda}$.

We will use this strong CDH solver to build a VPRG over DHGen, which will satisfy correctness, binding, and hiding for infinitely many security parameters. We set $(\mathsf{PRG.Setup}, \mathsf{PRG.Eval})$ to be Goldreich's PRG instantiated with the LV predicate; let PRG_i be IAC that computes, given a seed $(s_1, \cdots, s_n)$ as input, the i-th output bit of $\mathsf{PRG.Eval}(s_1, \cdots, s_n)$. Observe that PRG_i is a degree-3 integer arithmetic circuit with 9 gates (ignoring the subtractions by a constant, which are "for free"), where all intermediate values belong to $[\pm 1]$ provided that the inputs to the IAC are bits. We fix an arbitrary small positive constant $\delta_{\mathsf{PRG}} < 0.25$, such that Goldreich's PRG instantiated with the LV predicate is conjectured to be secure when stretching n bits to $m = n^{1+\delta_{\mathsf{PRG}}}$ bits.

Fix integers (l, t, κ, c). The high-level intuition of our VPRG is relatively simple. The commitment to the seed $(s_1, \cdots, s_n)$ is a bit-by-bit commitment $(\mathsf{com}_1, \cdots, \mathsf{com}_n)$, with the commitment scheme given in Sect. 4.4, which computationally hides the seed under the short-exponent discrete logarithm assumption. The pseudorandom string is simply $\mathsf{PRG.Eval}(\mathsf{pp}, (s_1, \cdots, s_n))$. Given the commitment to the seed, both parties will use the strong CDH solver, which exists since we assume that CDH does not hold over $\mathbb{G}$. In the full version, we prove a theorem that shows that the parties can both use $\mathsf{EvalCom}(\mathsf{PRG}_i, \mathsf{com}_1, \cdots, \mathsf{com}_n)$ for $i = 1$ to $m = n^{1+\delta_{\mathsf{PRG}}}$. For each such i, denoting $\mathsf{com}_i = g^{w_i + 2^{l+t} s_i}$ with $w_i \in [\pm 2^l]$ and $s_i \in \{0,1\}$, the parties get

$$\mathsf{com}_i^* \leftarrow \mathsf{EvalCom}(\mathsf{PRG}_i, (g^{w_j + 2^{l+t} s_j})_{j \le n}) = g^{w_i^* + 2^{3(l+t)} \mathsf{PRG}_i(s_1, \cdots, s_n)},$$

with $w_i^* \in [\pm(2^{3l+2t+31})]$. Let $\ell \leftarrow 3l + 2t + 31$ and $\ell' \leftarrow \ell + \kappa + c + 1$. Let $b_i \leftarrow \mathsf{PRG}_i(s_1, \cdots, s_n)$. We set $t = 34 + \kappa + c$, which guarantees that $\ell' + 2 = 3(l + t)$. Therefore, we have $\mathsf{com}_i^* = g^{w_i^* + 2^{\ell'+2} b_i}$. To provably open the i-th bit of the pseudorandom string to the bit b_i, the prover reveals b_i, and both parties homomorphically compute $g^{w_i^*}$ from com_i^*. It remains for the prover to demonstrate that he revealed the right value b_i, which he does using a NIZK to prove that $g^{w_i^*}$ belongs to $\mathscr{L}_{\ell',c}$ (which he can do since $w_i^* \in [\pm 2^\ell]$). More precisely, we will use the CIH from Sect. 4.2 to compile the Σ-protocol for the language $\mathscr{L}_{\ell',c}$ from Sect. 4.1, with challenge length c, into a NIZK. Since $\ell' + 2 = 3(l + t)$, and using Lemma 36 from Sect. 4.4, this uniquely binds the prover to b_i.

Parameters and Assumptions. To apply Lemma 36, we must pick p such that $\log p > \ell' + 2c + 4 = 3l + 5c + 3\kappa + 104$, where l is such that the $p/2^l$-SEDL assumption holds over $\mathbb{G}$, and κ is a statistical security parameter. Choosing c to be polynomially larger than $l + \kappa$, we have $\ell' = 3c + o(c)$, and we can set p such that $\log p = 5c + o(c)$. Therefore, setting the number of parallel repetitions of the Σ-protocol for $\mathscr{L}_{\ell',c}$ to $N = 6$, we get $\lambda = 5c + 6(\ell' + c) + o(c) = 29c + o(c)$. In turns, this gives $p^{-1} \cdot 2^{Nc-\lambda} = 2^{-28c-o(c)} = 2^{-(28/29+o(1))\lambda}$. Therefore, the adaptive soundness of our NIZK for $\mathscr{L}_{\ell',c}$ reduces to the $2^{-(28/29+o(1))\lambda}$-$\mathsf{OW\text{-}KDM}$ security of ElGamal (over the group $\tilde{\mathbb{G}}$ of size $q \approx 2^{\lambda}$) w.r.t. efficient functions. Observe that with this choice of parameters, it holds that $p/2^l = 2^{O(\sqrt{\log p})}$, hence the $p/2^l$-SEDL assumption is implied by the $2^{-O(\sqrt{\log p})}$-$\mathsf{OW\text{-}KDM}$ security of ElGamal over $\mathbb{G}$, which is clearly implied by the $2^{-(28/29+o(1))\log p}$-$\mathsf{OW\text{-}KDM}$ security of ElGamal over $\mathbb{G}$. Due to the large number of parameters involved in our construction, and to make it more readable, we summarize our parameters and the constraints they must satisfy in the full version.

Construction. Let $\mathsf{NIZK}^{\mathsf{AS}} = (\mathsf{Setup}^{\mathsf{AS}}, \mathsf{Prove}^{\mathsf{AS}}, \mathsf{Verify}^{\mathsf{AS}})$ be a NIZK for the almost-short language $\mathscr{L}_{\ell',c}$ over the group generator DHGen where the CDH problem is insecure. Given a security parameter n for the VPRG, we set $l(n) = \kappa(n) = n$ and $c(n) = n^2$ (so that $\kappa + l = o(c)$). We set $(\ell(n), \ell'(n), \lambda(n), p(n))$ as described previously, and $s(n) = n \cdot \lceil \log p \rceil$. Let $m = m(n)$ be $n^{1+\delta_{\mathsf{PRG}}}$. Our construction of VPRG proceeds as follows:

- $\mathsf{Setup}(1^n, m)$: run $\mathcal{G} = (\mathbb{G}, p) \leftarrow_r \mathsf{DHGen}(1^{\lambda(n)})$ and sample $g \leftarrow_r \mathbb{G}$.[6] Further, for $i = 1$ to m, generate $\mathsf{crs}_i \leftarrow_r \mathsf{Setup}^{\mathsf{AS}}(1^{\lambda(n)})$ and $\mathsf{pp}_{\mathsf{PRG}} \leftarrow_r \mathsf{PRG.Setup}(1^n)$. Finally, output $\mathsf{pp} = (g, (\mathsf{crs}_i)_{i \le m}, \mathsf{pp}_{\mathsf{PRG}})$.
- $\mathsf{Stretch}(\mathsf{pp})$: pick a seed $\mathsf{seed} = (s_1, \cdots, s_n) \leftarrow_r \{0,1\}^n$ for PRG. For $i = 1$ to n, pick $w_i \leftarrow_r [2^l]$ and compute $\mathsf{com}_i \leftarrow g^{w_i + 2^{l+t} s_i}$. Output $\mathsf{pvk} \leftarrow (\mathsf{com}_1, \cdots, \mathsf{com}_n)$, $x = \mathsf{PRG.Eval}(\mathsf{seed})$, and $\mathsf{aux} \leftarrow (\mathsf{seed}, w_1, \cdots, w_n)$.
- $\mathsf{Prove}(\mathsf{pp}, \mathsf{aux}, i)$: compute $\mathsf{com}_i^* \leftarrow \mathsf{EvalCom}(\mathsf{PRG}_i, (\mathsf{com}_1, \cdots, \mathsf{com}_n))$ and
$$w_i^* \leftarrow \mathsf{EvalOpen}(\mathsf{PRG}_i, (w_1, s_1), \cdots, (w_n, s_n)).$$
Set $x_i = \mathsf{PRG}_i(\mathsf{seed})$, $X_i = \mathsf{com}_i^* / g^{2^{l+t} x_i}$ and run $\boldsymbol{\pi}_i^{\mathsf{AS}} \leftarrow \mathsf{Prove}^{\mathsf{AS}}(\mathsf{crs}_i, X_i, w_i^*)$. Output $\pi = \boldsymbol{\pi}_i^{\mathsf{AS}}$.
- $\mathsf{Verify}(\mathsf{pp}, \mathsf{pvk}, i, b, \pi)$: compute $\mathsf{com}_i^* \leftarrow \mathsf{EvalCom}(\mathsf{PRG}_i, (\mathsf{com}_1, \cdots, \mathsf{com}_n))$ and set $X = \mathsf{com}_i^* / g^{2^{l+t} b}$. Output $\mathsf{Verify}^{\mathsf{AS}}(\mathsf{crs}_i, X, \pi)$.

Setting $(\delta(n), s(n))$ for VPRG. Before going into the security proofs, let us assess the parameter values of $\delta(n)$ and $s(n)$ of our VPRG. First, we have $m(n) = n^{1+\delta_{\mathsf{PRG}}}$ where the constant δ_{PRG} is the stretch of the underlying PRG that can be set arbitrary within $0 < \delta_{\mathsf{PRG}} < 0.25$. The size of the verification key is $s(n) := n \cdot \lceil \log p \rceil$, and in particular, $s(n) \le n^{1+\delta_{\mathsf{PRG}}/2}$ for

[6] We remark that we assume the CDH problem is insecure over the group $\mathbb{G}$ for the specific parameter $\lambda(n)$.

all sufficiently large n. Therefore, by setting $\delta(n) := \delta_{\mathsf{PRG}}/3$, we conclude $s(n)^{1+\delta(n)} \leq (n^{1+\delta_{\mathsf{PRG}}/2})^{1+\delta_{\mathsf{PRG}}/3} = n^{1+\delta_{\mathsf{PRG}}} = m(n)$. Specifically, we have a $(s(n) = n \cdot \lceil \log p \rceil, \delta(n) = \delta_{\mathsf{PRG}}/3)$-VPRG.

Theorem 38. *If the $p/2^l$-SEDL assumption holds relative to DHGen, CDH does not hold relative to DHGen, PRG is a secure pseudorandom generator stretching n bits to $n^{1+\delta_{\mathsf{PRG}}}$ bits for some arbitrarily small positive constant δ_{PRG}, and the NIZK argument system for the almost language $\mathcal{L}_{\ell'(n),c(n)}$ is adaptive sound and selective single-theorem zero-knowledge, where $\ell'(n)$ and $c(n)$ are chosen as described above, then our construction provides an $(s(n), \delta_{\mathsf{PRG}}/3)$-$\mathsf{VPRG}$ (with $s(n) = n \cdot \lceil \log p \rceil$) that is binding and hiding.*

The proof of binding is very similar to the proof of Theorem 25. For the hiding property, in a first hybrids, we first simulate all NIZK proofs, still providing correct openings. Then, we replace the commitment to the seed by random group elements, which is indistinguishable from the previous hybrids under the short-exponent discrete logarithm assumption. Eventually, we replace the PRG values by random bits, which is indistinguishable under the pseudorandomness of the PRG. In the last game, the value of all opened bits is perfectly independent of the value of the unopened bit, hence the advantage of the adversary is 0. We provide a the full version. Since the above is an $(s(n), \delta(n))$-VPRG for a constant $\delta(n) = \delta_{\mathsf{PRG}}/3$, by setting n large enough, we can satisfy the condition required in Theorem 7 for constructing NIZKs for all of NP. In particular, as a consequence of Theorems 7, 21, 24, and 25, the following is obtained.

Theorem 39. *Assume that the CDH assumption does not hold relative to DHGen, that ElGamal satisfies $2^{-(28/29+o(1))\lambda}$-OW-KDM security with respect to efficient functions, and that Goldreich's PRG instantiated with the LV predicate is secure for some (arbitrarily small) polynomial stretch. Then there exists an infinitely often adaptive sound and adaptive multi-theorem NIZK for all of NP.*

In the full version, we show that combining the results of Sect. 3 with the results of this section gives us the following theorem.

Theorem 40. *Assume that ElGamal satisfies $2^{-(28/29+o(1))\lambda}$-OW-KDM security with respect to efficient functions, and that Goldreich's PRG instantiated with the LV predicate is secure for some (arbitrarily small) polynomial stretch. Then there exists an adaptively sound and adaptive multi-theorem infinitely-often NIZK for NP, whose multi-theorem zero-knowledge property holds against uniform adversaries.*

Acknowledgements. Shuichi Katsumata was supported by JST CREST Grant Number JPMJCR19F6. Geoffroy Couteau and Bogdan Ursu were supported by ERC PREPCRYPTO Grant Agreement ID 724307. We would like to thank Dennis Hofheinz for valuable discussions and contributions to the early stages of this work. We are also grateful for the comments received from the anonymous reviewers of this paper.

References

1. Abdalla, M., Fouque, P.A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 572–590. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_34
2. Alekhnovich, M., Hirsch, E.A., Itsykson, D.: Exponential lower bounds for the running time of DPLL algorithms on satisfiable formulas. J. Autom. Reason. **35**(1–3), 51–72 (2005)
3. Applebaum, B., Lovett, S.: Algebraic attacks against random local functions and their countermeasures. In: Wichs, D., Mansour, Y. (eds.) 48th ACM STOC, pp. 1087–1100. ACM Press, New York (2016)
4. Bitansky, N., et al.: Why "Fiat-Shamir for proofs" lacks a proof. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 182–201. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_11
5. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications (extended abstract). In: 20th ACM STOC, pp. 103–112. ACM Press, May 1988
6. Canetti, R., et al.: Fiat-Shamir: from practice to theory. In: Charikar, M., Cohen, E. (eds.) 51st ACM STOC, pp. 1082–1090. ACM Press, New York (2019)
7. Canetti, R., Chen, Y., Reyzin, L., Rothblum, R.D.: Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 91–122. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_4
8. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Cryptology ePrint Archive, Report 1998/011 (1998). http://eprint.iacr.org/1998/011
9. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4), 557–594 (2004)
10. Couteau, G., Dupin, A., Méaux, P., Rossi, M., Rotella, Y.: On the concrete security of Goldreich's pseudorandom generator. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 96–124. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_4
11. Couteau, G., Hofheinz, D.: Designated-verifier pseudorandom generators, and their applications. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 562–592. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_20
12. Dwork, C., Naor, M.: Zaps and their applications. In: 41st FOCS, pp. 283–293. IEEE Computer Society Press, November 2000
13. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: 31st FOCS, pp. 308–317. IEEE Computer Society Press, October 1990
14. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM J. Comput. **29**(1), 1–28 (1999). https://doi.org/10.1137/S0097539792230010
15. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
16. Freitag, C., Komargodski, I., Pass, R.: Impossibility of strong KDM security with auxiliary input. Cryptology ePrint Archive, Report 2019/293 (2019). https://eprint.iacr.org/2019/293

17. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC, pp. 99–108. ACM Press, New York (2011)
18. Goldreich, O.: Candidate one-way functions based on expander graphs. Cryptology ePrint Archive, Report 2000/063 (2000). http://eprint.iacr.org/2000/063
19. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC, pp. 25–32. ACM Press, May 1989
20. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In: 27th FOCS, pp. 174–187. IEEE Computer Society Press, October 1986
21. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
22. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 339–358. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_21
23. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_24
24. Holmgren, J., Lombardi, A.: Cryptographic hashing from strong one-way functions (or: one-way product functions and their applications). In: Thorup, M. (ed.) 59th FOCS, pp. 850–858. IEEE Computer Society Press, October 2018
25. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 224–251. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_8
26. Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 622–651. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_22
27. Koshiba, T., Kurosawa, K.: Short exponent Diffie-Hellman problems. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 173–186. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24632-9_13
28. Lindell, Y.: An efficient transform from sigma protocols to NIZK with a CRS and non-programmable random oracle. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 93–109. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_5
29. Lombardi, A., Vaikuntanathan, V.: Minimizing the complexity of Goldreich's pseudorandom generator. Cryptology ePrint Archive, Report 2017/277 (2017). http://eprint.iacr.org/2017/277
30. Lyubashevsky, V., Neven, G.: One-shot verifiable encryption from lattices. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 293–323. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56620-7_11
31. Maurer, U.M., Wolf, S.: Diffie-Hellman oracles. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 268–282. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_21
32. Mossel, E., Shpilka, A., Trevisan, L.: On e-biased generators in NC0. In: 44th FOCS, pp. 136–145. IEEE Computer Society Press, October 2003
33. Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_6

34. ODonnell, R., Witmer, D.: Goldreich's PRG: evidence for near-optimal polynomial stretch. In: 2014 IEEE 29th Conference on Computational Complexity (CCC), pp. 1–12. IEEE (2014)
35. Oren, Y.: On the cunning power of cheating verifiers: some observations about zero knowledge proofs (extended abstract). In: 28th FOCS, pp. 462–471. IEEE Computer Society Press, October 1987
36. Ostrovsky, R., Wigderson, A.: One-way functions are essential for non-trivial zero-knowledge. In: Proceedings of the 2nd Israel Symposium on the Theory and Computing Systems, pp. 3–17. IEEE (1993)
37. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 89–114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_4
38. Pollard, J.M.: A Monte Carlo method for factorization. BIT Numer. Math. **15**(3), 331–334 (1975). https://doi.org/10.1007/BF01933667
39. Quach, W., Rothblum, R.D., Wichs, D.: Reusable designated-verifier NIZKs for all NP from CDH. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 593–621. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_21
40. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 475–484. ACM Press, New York (2014)
41. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_18