

New Pseudorandom Correlation Functions and Exponent VRFs from DCR

Archisman Dutta¹[0009-0002-8793-6049], Lisa Kohl²[0009-0007-1870-2181], Lawrence Roy³[0009-0003-8436-0029], and Peter Scholl¹[0000-0002-7937-8422]

¹ Aarhus University, Aarhus, Denmark

archisman@cs.au.dk, peter.scholl@cs.au.dk

² Cryptology Group, CWI Amsterdam, Amsterdam, The Netherlands

lisa.kohl@cwi.nl

³ IBM Research, Zurich, Switzerland

ldr709@gmail.com

Abstract. Pseudorandom correlation functions (PCFs) allow two parties to expand short correlated keys into an unbounded, locally evaluable source of correlated randomness. In this paper, we give the first constructions of PCFs for correlated oblivious transfer (COT) and vector oblivious linear evaluation (VOLE) correlations over $\mathbb{Z}_M$ for an arbitrary modulus M , with reasonable concrete efficiency. Our construction comes with a lightweight, two-round distributed setup protocol, and relies solely on the decisional composite residuosity (DCR) assumption.

As part of our security analysis, we introduce a new family of *extended power DDH* problems over subgroup indistinguishability groups, and give a novel master theorem showing that this is implied by standard DCR. This suffices to prove security of our PCF based on DCR, and also implies that prior constructions relying on related power-DDH style assumptions in composite groups can be based on DCR alone.

We further extend our construction to obtain a PCF for oblivious linear evaluation (OLE) and general degree-two correlations over $\mathbb{F}_2$.

As an application of our PCF for VOLE, we derive a designated-verifier exponent-verifiable random function (eVRF), where the publicly verifiable output is the VRF output in the exponent. As an independent contribution, we additionally give a publicly verifiable eVRF from an offline/online variant of homomorphic secret sharing. Both of these constructions have much shorter proof sizes than prior work.

Keywords: Pseudorandom correlation functions · Correlated oblivious transfer · Vector oblivious linear evaluation · Decisional composite residuosity · Exponent verifiable random functions

Table of Contents

New Pseudorandom Correlation Functions and Exponent VRFs from DCR	1
<i>Archisman Dutta, Lisa Kohl, Lawrence Roy, and Peter Scholl</i>	
1 Introduction.....	3
1.1 Our Contributions	3
1.2 Technical Overview	4
2 Preliminaries	8
2.1 Notation	8
2.2 Pseudorandom Functions.....	8
2.3 Decisional Composite Residuosity and DDLog	9
2.4 Sparse Power DCR	9
2.5 Pseudorandom Correlation Functions	9
2.6 Offline/Online HSS	10
2.7 Exponent Verifiable Random Functions	10
3 Extended Power DDH from Subgroup Indistinguishability	12
4 PCF for Correlated OT	15
4.1 From Correlated OT to $\mathbb{Z}_N$ -VOLE	15
4.2 Distributed Setup	17
5 PCF for OLE and Multiplication Triples	17
6 Exponent-VRFs	18
6.1 Designated-Verifier eVRF from VOLE-PCF	18
6.2 Publicly Verifiable eVRF from Offline/Online HSS	18
6.3 Applications to Threshold Schnorr and ECDSA	19
7 Experimental Setup	20
7.1 Benchmark Results	20
7.2 Comparison with Prior Work	21
A Additional Related Work	23
B Additional Preliminaries.....	24
B.1 Pseudorandom Functions.....	24
B.2 Pseudorandom Correlation Functions and Reverse-Sampleable Correlations	25
B.3 Fiat–Shamir forking	27
C Distributed Setup	29
D Subgroup Indistinguishability $\Rightarrow$ Extended Power-DDH	31
D.1 Proof of Theorem 1	31
E Proofs for the PCF for COT	34
E.1 PCF for COT	34
E.2 PCF for OLE	38
F From Correlated OT to $\mathbb{Z}_N$ -VOLE	41
F.1 Local aggregation to $\mathbb{Z}_N$ -VOLE	41
F.2 Towards a PCF for VOLE	42
F.3 Using a random oracle for external labels	43
F.4 Powers-of-two aggregation and statistical uniformity	44
F.5 Integer lifting and prime-field reduction	45
G Details on Exponent-VRF Constructions	46
G.1 Designated-Verifier eVRF.....	46
G.2 Publicly Verifiable eVRF	49
H Details on Experimental Setup	51

1 Introduction

Cryptographic protocols for secure multi-party computation (MPC) are typically organised into an input-independent preprocessing phase and a lightweight online phase. The preprocessing phase supplies correlated randomness such as random oblivious transfer (OT), correlated OT, vector oblivious linear evaluation (VOLE), and multiplication (Beaver) triples. In many classic MPC protocols, this preprocessing dominates the concrete cost: generating it requires communication and computation that scale linearly with the number of correlations required by the online protocol.

To circumvent this, a recent line of work has proposed to use *pseudorandom correlation generators* (PCGs) [BCGI18, BCG⁺19] and *pseudorandom correlation functions* (PCFs) [BCG⁺20a], which compress correlated randomness into short, correlated seeds. A PCG produces a fixed, polynomially long batch of correlated samples from a pair of short keys, while a PCF is a stronger primitive: each party receives a key k_σ such that, for every input x , the locally-evaluated outputs $y_\sigma \leftarrow \text{PCF.Eval}(\sigma, k_\sigma, x)$ are jointly distributed as a sample from the target correlation $\mathcal{Y}$. In particular, PCFs offer an unbounded (and incrementally evaluable) source of correlated randomness, which makes them attractive for protocols that require large quantities of correlations or that benefit from on-demand evaluation.

The Challenge of Building PCFs. Despite their conceptual appeal, PCFs have proven significantly harder to construct than PCGs. The state of the art falls into two main families.

The first family is based on variants of the learning parity with noise (LPN) assumption. Building on the template of [BCG⁺20a], constructions use variable-density LPN [BCG⁺20a], expand-accumulate LPN [BCG⁺22] or sparse LPN [BCM⁺25] to obtain PCFs for a range of correlations including (correlated) OT, VOLE, and (with much worse efficiency) multiplication triples. While their evaluation algorithm is in some cases concretely efficient, these constructions suffer from heavy, non-black-box key-generation procedures and rely on less standard variants of LPN.

The second family instead uses more traditional “public-key” cryptographic techniques such as groups [OSY21, BCM⁺24] or lattices [HRK25]. The PCFs of [OSY21] build on ideas from homomorphic secret sharing to obtain a simple, Paillier-based construction for generating VOLE correlations modulo an RSA modulus N , or a construction for OT correlations based on Goldwasser-Micali, that has a significantly more expensive evaluation phase. Subsequently, [BCM⁺24] took a different route via *constrained* pseudorandom functions based on the Naor–Reingold PRF [NR97], leading to a surprisingly efficient PCF for random OT correlations, by building a constrained PRF for a constraint class that includes a weak PRF. Both of these works allow for simple, *public-key* setup procedures, where the PCF keys can be securely generated in just one round of interaction. Unfortunately, their construction is limited to generating random OT correlations and has not been extended to support, for instance, OLE or VOLE. Finally, the work of [HRK25] presents a PCF for multiplication triples (and other correlations) based on fully homomorphic encryption, but suffers from slow performance and large key sizes due to bootstrapping.

To summarise, for OT correlations, the only PCFs with good concrete efficiency are the recent sparse LPN candidate of [BCM⁺25], with the drawback of an expensive, non-black-box distributed key generation protocol to set up distributed point functions, and the Naor-Reingold based candidate of [BCM⁺24], which supports a simple public-key setup.

For VOLE correlations over an integer ring, we also have few options: the PCF for VOLE over $\mathbb{Z}_N$ from [OSY21] is very simple and reasonably efficient, but has limited applicability due to the RSA modulus N . A recent construction by Kondi [Kon25] based on SoftSpokenOT [Roy22] makes only black-box use of a pseudorandom generator and is very practical, but can only generate VOLE modulo M for smooth moduli M with small prime-power factors.

1.1 Our Contributions

In this work, we give the first concretely efficient construction of a PCF for correlated oblivious transfer (COT) over $\mathbb{Z}_N$, where N is an RSA modulus, and explore several applications.

PCF for COT and VOLE over $\mathbb{Z}_M$. Our main PCF construction is inspired by the constrained Naor–Reingold paradigm from [BCM⁺24], but adapts it to work over the Paillier group, taking advantage of distributed discrete log operations from [OSY21, RS21]. This change allows us to obtain a PCF that produces *correlated OT correlations* over $\mathbb{Z}_N$, rather than the random OT correlations of [BCM⁺24]. In COT, each sample of the correlation gives (μ, m_μ) to the receiver and (m_0, m_1) to the sender, where $\mu \in \{0, 1\}$, $m_0 \in \mathbb{Z}_N$ and $m_1 = m_0 + \Delta$

for some fixed, secret integer offset Δ . While correlated OT can be easily converted into random OT using a correlation-robust hash [IKNP03], it is also useful on its own. Our correlations over $\mathbb{Z}_N$ can be locally lifted into integer correlations and then combined together to create VOLE correlations over $\mathbb{Z}_M$, for *any* modulus M . Since the underlying correlation lives over the integers, it can moreover be reduced with respect to several moduli at once. In particular, we obtain PCFs for *mixed-modulus* VOLE correlations, where the same value is multiplied with respect to two different moduli; we are not aware of any prior construction achieving this without evaluating a PRF inside homomorphic secret sharing or fully homomorphic encryption.

Extended Power DDH Problem. Our construction is proven secure based on a new *Sparse-Power DCR* assumption, which generalizes previous power-DDH style assumptions in Paillier groups [ARS24]. Instead of relying on this new assumption, we can in fact show that this is a special case of a more general class of problems which we call *extended power DDH*, which we prove is implied by the standard DCR assumption. More generally, we prove this result for subgroup indistinguishability assumption, which includes DCR, as well as other common assumptions over unknown order groups. This result can be viewed as a master-theorem for algebraic indistinguishability problems over DCR, which we believe may be of independent interest. The result offers a framework for simplifying assumptions used in prior work, including the power-DDH assumption over DCR groups used in [ARS24, ILL25].⁴

PCF for OLE and Degree-2 Correlations. We show how to upgrade our PCF for correlated OT to a PCF for OLE and general degree-2 correlations (such as matrix multiplication triples), by combining it with obliviously sampled ciphertexts from the Goldwasser-Micali cryptosystem.

Exponent Verifiable Random Functions with Short Proofs. As an application of our PCF for VOLE, we obtain an exponent-verifiable random function (eVRF) in the designated-verifier setting, with extremely short proofs of just one group element (32 bytes). Independently, we additionally construct a *publicly verifiable* eVRF using techniques from homomorphic secret sharing, which is less practical but still has fairly small proofs of around 128 bytes, more than 4x smaller than the recent work of [BCK25].

An exponent VRF [BHLS25] is a form of VRF where the pseudorandom output of the function lies in the exponent of a group generator. They have various applications to threshold cryptography, including two-party threshold ECDSA and threshold deterministic Schnorr signing.

Implementation. We implement our PCF for COT/VOLE and evaluate its performance. The sender can evaluate around 650 COTs/s on a single core, and this can be trivially parallelized to obtain significantly higher throughput. The receiver’s computation, however, is much slower since it cannot know the factorization of N , and takes around half a second per OT. For the designated-verifier eVRF, this translates into highly succinct proofs with fast verification, and the tradeoff of a slower prover time compared with e.g. [BCK25]. We give a detailed comparison with the most closely related prior PCFs, covering setup model, key sizes, and evaluation cost, in Table 3.

1.2 Technical Overview

Prior Work: PCF for OT. We start by recalling the approach for achieving PCFs for oblivious transfer via the constrained Naor-Reingold approach [BCM⁺24]. Here, we focus on the PCF with setup rather than the public-key version (which we will discuss below). More precisely, a PCF for OT consists of two algorithms *Setup* and *Eval*, where *Setup*(1^λ) returns a pair of short keys k_0, k_1 , each of which individually reveals nothing about the other party’s output beyond what can be inferred from the correlation itself, and where evaluating both keys on the same input $x \in \{0, 1\}^n$ yields a pair of correlated outputs $\text{Eval}(k_0, x) = (b, m_b)$ and $\text{Eval}(k_1, x) = (m_0, m_1)$, for a random bit b and random messages $m_0, m_1 \in \{0, 1\}$ subject to the oblivious transfer correlation.

On a high level, the idea of [BCM⁺24] is to start with an “outer” PRF $F_K : \{0, 1\}^n \rightarrow \{0, 1\}$. Using this, the setup algorithm samples PRF keys K_0, K_1 and outputs $k_1 := (K_0, K_1)$ to party P_1 (the sender). The corresponding evaluation is defined as $\text{Eval}(1, k_1, x) := (F_{K_0}(x), F_{K_1}(x))$. In other words, the sender uses keys K_0 and K_1 to generate messages m_0 and m_1 , respectively.

To generate the keys for party P_0 (the receiver), another PRF is used, the “inner” PRF $f_z : \{0, 1\}^n \rightarrow \{0, 1\}$. The purpose of f_z is to determine the output bit b of the receiver. Moreover, the receiver is given constrained keys K_0^* and K_1^* , where K_b^* allows the evaluation of $F_{K_b}(x)$ exactly on those x with $f_z(x) = b$. With $k_0 :=$

⁴ Note that our framework relies on subgroup indistinguishability problems such as DCR, so cannot be used to prove security of power-DDH over prime-order groups (as used in [BCM⁺24]) under standard DDH.

(z, K_0^*, K_1^*) , evaluation is defined as $\text{Eval}(0, k_0, x) := (f_z(x), F_{K_{f_z(x)}}^*(x))$, meaning the receiver's output consists of the bit $b = f_z(x)$ and the message $m_b = F_{K_{f_z(x)}}^*(x)$.

To instantiate the outer PRF F_K , the paper uses the Naor-Reingold PRF [NR97] with keys $K = (a_0, \dots, a_n) \in \mathbb{Z}_p^{n+1}$, where

$$F_K(x) = g^{a_0 \prod_{i: x_i=1} a_i}.$$

To instantiate f_z , the paper uses an inner product membership PRF, such as the BIPSW weak PRF [BIP⁺18], where the key is a bit string $z \in \{0, 1\}^n$ and the function is defined as $f_z(x) = \chi_S(\langle z, x \rangle)$, with $S \subseteq \{0, 1, \dots, n\}$ a public set and χ_S its indicator function. (For BIPSW, $S := \{t \in [n] : t \bmod 6 \in \{3, 4, 5\}\}$.) Throughout, we write $s_x := \langle z, x \rangle = \sum_{i: x_i=1} z_i$.

To constrain keys of the outer Naor-Reingold PRF relative to f_z , [BCM⁺24] uses the following observation: setting $\alpha_i := a_i \cdot r^{-z_i}$ for a freshly sampled r , and defining

$$h_s := g^{a_0 r^s} \quad \text{and} \quad e_x := \prod_{i: x_i=1} \alpha_i,$$

results in (recalling $s_x = \langle z, x \rangle = \sum_{i: x_i=1} z_i$):

$$F_K(x) = g^{a_0 \prod_{i: x_i=1} a_i} = g^{a_0 \prod_{i: x_i=1} (\alpha_i r^{z_i})} = g^{a_0 r^{s_x} \prod_{i: x_i=1} \alpha_i} = h_{s_x}^{e_x}.$$

Hence, to constrain K_0 relative to f_z so that the receiver can compute $F_{K_0}(x)$ whenever $\chi_S(s_x) = 0$, it suffices to give the receiver $\{\alpha_i\}_{i \in [n]}$ together with the “public table” $\{h_s\}_{s: \chi_S(s)=0}$. This further hides $F_{K_0}(x)$ whenever $\chi_S(s_x) = 1$, as required. Analogous material is given for K_1 on the inputs where $f_z(x) = 1$.

Note that this approach is very different from the one used in LPN-based PCFs [BCG⁺20a, BCG⁺22]. Namely, LPN-based PCFs exploit the linearity of variants of LPN to construct a PCF for subfield-VOLE (or, equivalently, correlated OT), and then use the IKNP transformation [IKNP03] to convert subfield-VOLE to OT. In contrast, for public-key PCFs following the Naor-Reingold paradigm, the “easy” case is directly constructing OT, whereas obtaining correlated OT and VOLE is more challenging, as it requires the original keys K_0, K_1 to be correlated, rather than independent.

Our Approach. Our goal is to construct a PCF for *correlated* OT, where instead of outputting independent messages m_0 and m_1 , the sender outputs messages satisfying $m_1 = m_0 + \Delta$ for a fixed offset Δ (known to the sender). The main difficulty in obtaining correlated OT via the constrained Naor-Reingold approach is that we need to be able to constrain *correlated* keys.

Moving to the Paillier group. Our first observation is that moving from a prime-order DDH group to the Paillier group $\mathbb{Z}_{N^2}^*$, where N is an RSA modulus, naturally supports adding correlated offsets. Every element $U \in \mathbb{Z}_{N^2}^*$ has a unique decomposition

$$U = U_0(1 + N)^\Delta, \quad U_0 \in \text{NR}_N, \quad \Delta \in \mathbb{Z}_N,$$

where $\text{NR}_N := \{u^N \bmod N^2 : u \in \mathbb{Z}_{N^2}^*\}$ denotes the subgroup of N -th residues. The factor $(1 + N)^\Delta$ can be used to embed an additive offset Δ in the group element.

Distributed discrete logarithm (DDLog). To extract this offset, we use the DDLog_N function on the Paillier group [OSY21, RS21], which satisfies the property that for any $U \in \mathbb{Z}_{N^2}^*$ and $\Delta \in \mathbb{Z}_N$,

$$\text{DDLog}_N(U(1 + N)^\Delta) = \text{DDLog}_N(U) + \Delta \pmod{N}.$$

First attempt. Our idea is to replace the two independent keys K_0 and K_1 by a single key $K = (a_0, \dots, a_n)$ and instead introduce a secret shift Δ (known to the sender) at the level of the PRF. Concretely, choosing $g \in \text{NR}_N$, we define two correlated PRFs:

$$F_K(x) = g^{a_0 \prod_{i: x_i=1} a_i}, \quad F_{K, \Delta}(x) = F_K(x) \cdot (1 + N)^\Delta.$$

By the shift property of DDLog_N , outputting $m_0 := \text{DDLog}_N(F_K(x))$ and $m_1 := \text{DDLog}_N(F_{K, \Delta}(x))$ yields a correlated OT sample with offset Δ .

For the receiver's constrained key, we proceed analogously to the OT construction. A natural first attempt is to give the receiver $\{\alpha_i\}_{i \in [n]}$ together with the public table

$$h_s := \begin{cases} g^{a_0 r^s} & \text{if } \chi_S(s) = 0, \\ g^{a_0 r^s} (1 + N)^\Delta & \text{if } \chi_S(s) = 1, \end{cases}$$

for $s \in \{0, \dots, n\}$. The receiver then raises h_{s_x} to e_x , hoping to obtain $g^{A_x}(1+N)^{\chi_S(s_x)\Delta}$ where $A_x := a_0 \prod_{i: x_i=1} a_i$.

The issue. However, raising h_{s_x} to e_x also scales the $(1+N)^\Delta$ factor: we obtain

$$h_{s_x}^{e_x} = g^{A_x}(1+N)^{\chi_S(s_x)e_x\Delta},$$

and the offset is scaled by e_x , breaking correctness.

The fix. To resolve this, we sample a_i from the set $\mathcal{E}_N$ of exponents in $\mathbb{Z}_{N\phi(N)}$ that are congruent to 1 modulo N (so the offset is preserved), and r from the subset $\mathcal{E}_N^*$ of those that are in addition invertible modulo $\phi(N)$ (so $\alpha_i = a_i r^{-z_i}$ is well-defined). With this choice, $e_x \equiv 1 \pmod{N}$, and so $(1+N)^{\chi_S(s_x)e_x\Delta} = (1+N)^{\chi_S(s_x)\Delta}$, giving

$$h_{s_x}^{e_x} = g^{A_x}(1+N)^{\chi_S(s_x)\Delta}$$

as desired. (Although it might seem that restricting a_i to $\mathcal{E}_N$ changes the underlying Naor-Reingold PRF, it does not: assuming $\gcd(N, \phi(N)) = 1$, the map $a_i \mapsto a_i \bmod \phi(N)$ is a bijection from $\mathcal{E}_N$ to $\mathbb{Z}_{\phi(N)}$, so the resulting PRF is identically distributed to the original.)

Sparse-Power DCR. It remains to identify the assumption under which Δ is hidden by the receiver's constrained key. To gain intuition, consider a simpler variant where *every* table entry is shifted by $(1+N)^\Delta$, i.e., the receiver is given $\{G^{r^s}(1+N)^\Delta\}_{s \in \{0, \dots, n\}}$ (where $G = g^{a_0}$ is a random N -th residue). Hiding Δ then follows directly from standard DCR: given a DCR challenge C , the tuple $(C^r)_s$ preserves the same Paillier message Δ in every coordinate, since $r \equiv 1 \pmod{N}$ implies $r^s \equiv 1 \pmod{N}$.

Our construction is more delicate: it shifts *only* the entries at positions $s \in S$. This selective programming is what the receiver key must hide, and is not captured by the all-shift argument above. We isolate this as a separate *Sparse-Power DCR* assumption that makes the support set S explicit. Somewhat surprisingly, we further show that Sparse-Power DCR is implied by standard DCR, so our construction ultimately rests only on DCR (together with an inner product membership PRF). Sparse-Power DCR captures previous Power-DDH style assumptions over DCR groups that have been used in homomorphic secret sharing [ARS24] and algebraic homomorphic MACs [ILL25]. Our result implies that these constructions can be proven under standard DCR.

Setting up the keys. To go from the standard PCF to a public-key PCF, [BCM⁺24] leverages the fact that the values $\alpha_i := a_i \cdot r^{-z_i}$ can be set up non-interactively over $\mathbb{Z}_p$: they are a multiplicative secret sharing of r^{-z_i} , where the sender holds r and the receiver holds z . The sender can also publish the public h_s values directly, since they do not depend on z . While at first glance this seems to carry over to our construction, there is a subtle issue: the shares $\alpha_i = a_i r^{-z_i}$ would need to be set up modulo $N\phi(N)$, where $\phi(N)$ is not known to the receiver. In [BCM⁺24], working over $\mathbb{Z}_p$, one can first create an additively shared correlation and then lift it to the exponent of an element of $\mathbb{Z}_p^*$ to convert to multiplicative shares. We do not see a way to do this in one parallel round when working over $\mathbb{Z}_{N\phi(N)}$ where the group order is secret. Instead, we give a simple 2-round setup protocol based on OT.

From Correlated OT to VOLE. Given ℓ -many correlated OT samples

$$(b_j, \mu_j), \quad (m_{0,j}, m_{1,j}), \quad m_{1,j} = m_{0,j} + \Delta, \quad \mu_j = m_{b_j, j},$$

together with public coefficients c_j , the parties can locally aggregate them into a $\mathbb{Z}_N$ -VOLE sample by setting

$$u = \sum_j c_j b_j, \quad v = \sum_j c_j m_{0,j}, \quad w = \sum_j c_j \mu_j,$$

which satisfies $w = v + \Delta u$ over $\mathbb{Z}_N$. Choosing the coefficients c_j as powers of two yields an almost-uniform u over $\mathbb{Z}_N$ (up to a statistical loss we quantify in the body), and a suitable lifting condition lets us reinterpret a $\mathbb{Z}_N$ sample over the integers and reduce it to a prime field $\mathbb{Z}_q$.

We also show that the additional algebraic structure of our correlated-OT PCF can be leveraged to obtain a PCF for oblivious linear evaluation (OLE) and more general degree-two correlations over $\mathbb{F}_2$. Following a similar approach that was used to build PCGs for OLE and triples in [OSY21], we combine correlated OT over a larger modulus $\mathbb{Z}_{N'}$ with Goldwasser–Micali ciphertexts and distributed decryption, so that one party learns a bit a coming from the PCF for correlated-OT, while the other learns a bit b coming from an obviously sampled Goldwasser–Micali ciphertext for which only one party knows the key. This allows the two parties obtain additive shares of $a \cdot b$. This gives an OLE correlation directly, and further allows for obtaining shares of products $a_i \cdot b_j$ for arbitrary PCF output indices i, j , which leads to a PCF for general degree-two correlations. Importantly, this approach to building a PCF has a useful *programmability* property, which allows it to be generalized to the multi-party setting [BCG⁺20b].

A Designated-Verifier Exponent-VRF. As an application of the additional structure provided by working modulo N , we construct a *designated-verifier exponent-verifiable random function* (eVRF). An eVRF, as introduced in [BHLS25], is intuitively a verifiable random function whose output lives “in the exponent” of a group: on input x , the prover obtains a secret value r and a public group element $Y = g^r$ together with a proof π that can be publicly verified. Such primitives have recently emerged as a useful tool in threshold signatures and other distributed protocols, where one wants verifiable, deterministic per-message nonces of the form g^r without revealing r . In this work we focus on a *designated-verifier* variant of eVRFs, where verification can only be performed by a designated verifier holding a secret verification key. This notion was already introduced in [KOR23] under the name *discrete-log PCFs*; we adopt the terminology designated-verifier eVRF to emphasize the connection to eVRFs.

Following the observation from [KOR23], a PCF for VOLE can directly be used to obtain a designated-verifier eVRF. Recall that a PCF for VOLE produces, on input x , shares (u_x, w_x) for P_0 and (Δ, v_x) for P_1 , satisfying the VOLE relation

$$w_x = v_x + \Delta \cdot u_x.$$

To turn this into an eVRF, we let the prover hold the party-0 PCF key and the designated verifier hold the party-1 PCF key. On input x , the prover evaluates its key to obtain (u_x, w_x) and outputs the eVRF value $Y = g^{u_x}$ together with the proof $\pi = g^{w_x}$. The verifier evaluates its own key to obtain (Δ, v_x) , and accepts if

$$\pi = g^{v_x} \cdot Y^\Delta,$$

which by the VOLE relation is equivalent to $g^{w_x} = g^{v_x + \Delta \cdot u_x}$.

Importantly, although our PCF outputs correlated OT/VOLE modulo N , we are *not* restricted to working with groups of order N ; indeed, by “lifting” shares of small secrets modulo N to shares over the integers (as in e.g. [BKS19]), we can obtain VOLE correlations modulo p for any modulus p . This means that we can avoid the costly modulus conversion and range proofs needed in [KOR23], significantly reducing the proof size. Soundness follows from the fact that Δ is hidden from the prover: to forge a proof for some $Y' \neq g^{u_x}$, a malicious prover would have to produce π' satisfying $\pi' = g^{v_x} \cdot Y'^\Delta$.

The proof size in our DV-eVRF is just a single group element, significantly smaller than the state-of-the-art, e.g. 29 group elements in [BCK25].

Publicly Verifiable eVRF from Offline/Online HSS. For public verification, we cannot use the VOLE-based template directly, since that requires the verifier to hold the secret VOLE multiplier Δ . We therefore present a separate construction that does not directly use a PCF, but rather, uses a form of *homomorphic secret sharing* (HSS) scheme to homomorphically evaluate and publicly verify the computation of a PRF, leading to a publicly verifiable eVRF. This construction is independent of our Paillier PCF.

Recall that a (two-party) homomorphic secret sharing scheme allows a key K to be shared between two parties P_0, P_1 such that, given an input x , each party can locally compute an additive share W_b of $F_K(x)$, i.e., $W_1 - W_0 = F_K(x)$. Note that here one can use any PRF to instantiate F_K , but for efficiency it is favorable to choose a PRF which has a small circuit size suitable for HSS evaluation.)

At a high level, we want the prover to convince the verifier that the eVRF output is $Y = g^{F_K(x)}$ for the prover’s secret K , without revealing K . The idea is to let the prover (as P_0) and the verifier (as P_1) jointly evaluate $F_K(x)$ via an HSS scheme. Before giving our construction, we first describe a first and second attempt to give some intuition.

First attempt. Suppose the parties jointly evaluate $F_K(x)$ via the HSS, so that each obtains an additive share with $W_1 - W_0 = F_K(x)$. For an honest prover, $Y = g^{F_K(x)} = g^{W_1 - W_0}$. To make this verifiable, the prover sends $T := g^{-W_0}$, and the verifier computes W_1 from their share and accepts if $Y = g^{W_1} \cdot T$.

This, however, allows a malicious prover to prove arbitrary outputs, as it can choose T adversarially. Namely, for any value Y' , setting $\tilde{Y} := Y \cdot Y'$ and $\tilde{T} := T \cdot Y'$ gives $g^{W_1} \cdot \tilde{T} = g^{W_1} \cdot T \cdot Y' = Y \cdot Y' = \tilde{Y}$, so the verifier accepts $\tilde{Y}$ as the eVRF output instead of Y .

Second attempt. To fix this, the idea is to introduce a random challenge δ , and let the parties evaluate $\delta \cdot F_K(x)$ instead of $F_K(x)$. The verification equation is now

$$Y^\delta = g^{W_1} \cdot T,$$

which for an honest prover equals $g^{\delta F_K(x)} = g^{W_1 - W_0}$ as required. The key observation is that in an *offline/online* HSS [CMPR23, MORS25], shares (θ_0, θ_1) of some value δ that is input to the computation satisfy $\vec{\theta}_1 = \vec{\alpha} \cdot \delta + \vec{\theta}_0$, where α is a secret key used in the HSS. Due to the simple linear structure, $\vec{\theta}_0$ can be sampled uniformly at

random in the offline phase, before δ is known. Hence the prover can determine its share W_0 (and thus $T = g^{-W_0}$) in the offline phase, and commit to T before seeing δ .

Now, the prover can no longer mount the previous attack: T is fixed before the prover learns which δ they need to satisfy. Intuitively, this gives soundness via a rewinding argument: two accepting transcripts with the same T but different challenges δ, δ' should combine to extract $\log_g Y = F_K(x)$, since by HSS correctness $W_1 - W_0 = \delta F_K(x)$ and $W'_1 - W_0 = \delta' F_K(x)$, so $W_1 - W'_1 = (\delta - \delta') F_K(x)$.

However, this argument implicitly assumes that the prover uses *the same* HSS material (in particular, the same secret key $\vec{\alpha}$ and the same offline mask $\vec{\theta}_0$) across both runs, otherwise the HSS correctness equations in the two transcripts would be about different functions $F_K, F_{K'}$ and would not combine. *Third attempt.* To enforce consistency, we additionally include a commitment of α via a Pedersen-style homomorphic commitment $C_\alpha = \text{Com}(\vec{\alpha}; \rho_\alpha)$ in the public key.

The protocol then proceeds as follows. In the commit phase, the prover sends $T := g^{-W_0}$ together with a commitment $C_\theta := \text{Com}(\vec{\theta}_0; \rho_\theta)$ to the offline share. After the challenge δ is derived (via Fiat–Shamir from the commit phase), the prover sends $\vec{\theta}_1$ together with the opening of the randomness $\rho_1 := \rho_\alpha \delta + \rho_\theta$. The verifier checks that $C_\alpha^\delta \cdot C_\theta = \text{Com}(\vec{\theta}_1; \rho_1)$; by the homomorphism of Com , this binds $\vec{\alpha}$ and $\vec{\theta}_0$. The verifier then computes W_1 from $\vec{\theta}_1$ and accepts if $Y^\delta = g^{W_1} \cdot T$. Soundness now follows by the argument in the second attempt, using the binding to ensure consistency when rewinding.

2 Preliminaries

2.1 Notation

We write λ for the security parameter, $[n] := \{1, \dots, n\}$, and $\text{negl}(\lambda)$ for a negligible function of λ . All algorithms are probabilistic polynomial time unless explicitly stated otherwise. For a finite set S , the notation $x \xleftarrow{\$} S$ denotes uniform sampling from S . For a distribution D , the notation $x \leftarrow D$ denotes sampling according to D . For a modulus M , the ring $\mathbb{Z}_M$ is written additively unless the ambient notation explicitly concerns a multiplicative subgroup.

For an RSA modulus $N = pq$, let $1 + N\mathbb{Z}_N := \{1 + Nt \bmod N^2 : t \in \mathbb{Z}_N\} \subseteq \mathbb{Z}_{N^2}^*$, the subgroup of order N generated by $1 + N$, and let $\text{NR}_N := \{u^N \bmod N^2 : u \in \mathbb{Z}_{N^2}^*\}$ denote the subgroup of N -th residues. The map $\text{NR}_N \times (1 + N\mathbb{Z}_N) \rightarrow \mathbb{Z}_{N^2}^*$, $(R, (1 + N)^a) \mapsto R(1 + N)^a$, is a bijection, so every $U \in \mathbb{Z}_{N^2}^*$ has a unique Paillier decomposition

$$U = U_0(1 + N)^a, \quad U_0 \in \text{NR}_N, \quad a \in \mathbb{Z}_N.$$

For a set S and an element t , write $\chi_S(t) := 1$ if $t \in S$ and $\chi_S(t) := 0$ otherwise; for a predicate P , we also write $\chi[P] \in \{0, 1\}$ for its truth value.

2.2 Pseudorandom Functions

We consider the class of inner-product membership (weak) PRFs, which captures the “BIPSW” candidate [BIP⁺18] as well as the Applebaum-Raykov weak PRF [AR16] based on Goldreich’s one-way function. See Section B.1 for the standard definitions of a (weak) PRF.

Definition 1 (Binary inner-product membership weak PRF). Let $n = n(\lambda)$ be polynomially bounded and let $S \subseteq \{0, \dots, n\}$ be a public set. A binary inner-product membership weak PRF (IPM-wPRF) for S is a family

$$f_z : \{0, 1\}^n \rightarrow \{0, 1\}, \quad f_z(\mathbf{x}) = \chi_S \left(\sum_{i=1}^n z_i x_i \right),$$

where $\mathbf{z} \xleftarrow{\$} \{0, 1\}^n$, such that f_z is a weak PRF on uniformly random binary inputs.

Assumption 1 (BIPSW [BIP⁺18]) For $k, x \in \{0, 1\}^n$, define

$$F_k^{\text{BIPSW}}(x) := \left(\sum_{i=1}^n k_i x_i \bmod 2 \right) + \left(\sum_{i=1}^n k_i x_i \bmod 3 \right) \pmod{2}.$$

The BIPSW assumption states that the family $\{F_k^{\text{BIPSW}}\}_{k \in \{0, 1\}^n}$ is a weak PRF.

The BIPSW candidate is an instance of Definition 1. Indeed, writing $s = \sum_i k_i x_i$, we have $(s \bmod 2) + (s \bmod 3) \equiv 1 \pmod{2}$ if and only if $s \bmod 6 \in \{3, 4, 5\}$, so that $F_k^{\text{BIPSW}} = \chi_{S_{\text{BIPSW}}}(\sum_{i=1}^n k_i x_i)$ for the support set $S_{\text{BIPSW}} := [0, n] \cap (6\mathbb{Z} + \{3, 4, 5\})$. In particular, $0 \notin S_{\text{BIPSW}}$, a fact used when formulating our sparse Power-DCR assumption below.

2.3 Decisional Composite Residuosity and DDLog

Assumption 2 (DCR [Pai99, DJ01]) Let $\text{ModGen}(1^\lambda)$ output (N, p, q) where $N = pq$ is an RSA modulus. Let $\zeta > 0$. The decisional composite residuosity assumption states that

$$\begin{aligned} & \{ (N, u^{N^\zeta} \bmod N^{\zeta+1}) \mid (N, p, q) \xleftarrow{\$} \text{ModGen}(1^\lambda), u \xleftarrow{\$} \mathbb{Z}_{N^{\zeta+1}}^* \} \\ & \stackrel{\text{comp}}{\equiv} \{ (N, v) \mid (N, p, q) \xleftarrow{\$} \text{ModGen}(1^\lambda), v \xleftarrow{\$} \mathbb{Z}_{N^{\zeta+1}}^* \}. \end{aligned}$$

Definition 2 (DDLog [OSY21, RS21]). For $U \in \mathbb{Z}_{N^{\zeta+1}}^*$, write uniquely $U = h + h'N$ with $h \in \{0, \dots, N-1\}$; since U is invertible modulo $N^{\zeta+1}$, the residue h lies in $\mathbb{Z}_{N^{\zeta+1}}^*$. Define $\text{DDLog}_N(U) := \log_{1+N}(U \cdot h^{-1}) \bmod N^\zeta$, which can be computed in polynomial time using Hensel lifting [DJ01].

Lemma 1 (DDLog correctness [OSY21, RS21]). For every $U \in \mathbb{Z}_{N^{\zeta+1}}^*$ and every $x \in \mathbb{Z}_{N^\zeta}$, we have $\text{DDLog}_N(U(1+N)^x) = \text{DDLog}_N(U) + x \bmod N^\zeta$.

2.4 Sparse Power DCR

Our construction uses a sparse analogue of Power DCR, where only the table positions in a public support set are shifted by a secret offset. We define the assumption as follows.

Assumption 3 (Sparse-Power DCR) Let $T = T_\lambda \subseteq \mathbb{Z}_{\geq 0}$ be a public polynomial-size index set, and let $B = B_\lambda \subseteq T_\lambda$ be a public support set. If $0 \in T_\lambda$, require $0 \notin B_\lambda$. Let $\mathcal{D}_E = \{\mathcal{D}_{E,N}\}$ be an exponent distribution over the integers (concretely, see Definition 7), and let $\mathcal{D}_\Delta = \{\mathcal{D}_{\Delta,N}\}$ be an efficiently samplable offset distribution over $\mathbb{Z}_N$.

Sample $(N, p, q) \xleftarrow{\$} \text{ModGen}(1^\lambda)$, $g \xleftarrow{\$} \text{NR}_N$, $r \leftarrow \mathcal{D}_{E,N}$, and $\Delta \leftarrow \mathcal{D}_{\Delta,N}$. Define

$$\mathcal{D}_0 := (N, g, (g^{r^s})_{s \in T}), \quad \mathcal{D}_1 := (N, g, (g^{r^s(1+N)^{\chi_B(s)\Delta}})_{s \in T}).$$

The Sparse-Power DCR assumption for $(T, B, \mathcal{D}_E, \mathcal{D}_\Delta)$ states that $\mathcal{D}_0 \stackrel{\text{comp}}{\equiv} \mathcal{D}_1$.

The condition $0 \notin B$ is necessary for the formulation that includes both g and the coordinate indexed by 0. If $0 \in B$, then the shifted distribution contains $u_0 = g(1+N)^\Delta$, while the unshifted distribution contains $u_0 = g$; the quotient u_0/g distinguishes except with probability $\Pr[\Delta = 0]$.

2.5 Pseudorandom Correlation Functions

For $\sigma \in \{0, 1\}$, let $\mathcal{R}_{\sigma,\lambda}$ denote the set, or efficiently samplable distribution, of admissible setup-programming strings for the σ -th party. If a party has no setup input, we take $\mathcal{R}_{\sigma,\lambda}$ to be singleton.

A PCF is used to generate a large number of samples from a given pair of correlated distributions. In the case of VOLE, each sample is derived with respect to a fixed set of setup parameters that may contain private information; we denote the parties' setup strings by ρ_0, ρ_1 . For our case of correlated OT or VOLE over $\mathbb{Z}_N$, the string ρ_0 is empty, while ρ_1 contains the factorisation (p, q) as well as the sender's secret offset Δ .

For VOLE over $\mathbb{Z}_N$ with respect to a distribution $\mathcal{D}_N$ over $\mathbb{Z}_N$, the correlation sampler $\mathcal{Y}_{\text{VOLE}}(\rho_0, \rho_1)$ samples $u \xleftarrow{\$} \mathcal{D}_N, v \xleftarrow{\$} \mathbb{Z}_N$ defines $w = v + \Delta u \bmod N$. It then outputs the sample (y_0, y_1) where $y_0 = (u, w) \in \mathbb{Z}_N^2$ and $y_1 = (\Delta, v) \in \mathbb{Z}_N^2$.

Note that, for the special case where $\mathcal{D}_N$ is the uniform distribution over $\{0, 1\}$, this is equivalent to (fixed-offset) correlated OT, since $w = v$ if $u = 0$ and $w = v + \Delta$ otherwise. We refer to this correlation as $\mathcal{Y}_{\text{COT}}$.

A PCF for such a correlation is then described by a pair of algorithms $\text{PCF} = (\text{PCF.Gen}, \text{PCF.Eval})$ with the syntax:

- $(k_0, k_1) \leftarrow \text{PCF.Gen}(1^\lambda, \rho_0, \rho_1)$ samples a pair of PCF keys.
- $\text{PCF.Eval}(\sigma, k_\sigma, x)$, on input a key k_σ , and an input $x \in \mathcal{X}_\lambda$, produces an output of the correlation for party σ .

A PCF should satisfy two main properties: the $\mathcal{Y}$ -pseudorandomness property says that the set of outputs from Eval of both parties should be computationally indistinguishable from a real set of outputs from the correlation $\mathcal{Y}$. Meanwhile, the security property requires that if an adversary is given one of the keys k_σ together with a set of outputs for party $1 - \sigma$, it cannot distinguish whether the honest party's outputs are generated using $k_{1-\sigma}$ or sampled as fresh outputs from the distribution of $\mathcal{Y}$ conditioned on party σ 's outputs being fixed.

We defer the formal definitions to Section B.

2.6 Offline/Online HSS

Recall that a homomorphic secret sharing scheme [BGI16] lets a client secret-share an input, lets two evaluators locally evaluate a function on their respective shares, and lets an output client reconstruct the function value from the two output shares. We use subtractive reconstruction throughout this paper: output shares (Y_0, Y_1) reconstruct to $Y_1 - Y_0$.

The public eVRF template in Section 6.2 uses the offline/online HSS notion of Meyer–Orlandi–Roy–Scholl [MORS25].

Definition 3 (Offline/online HSS). Let $\mathbb{F} = \mathbb{Z}_q$ be a prime field, let $\mathcal{K}$ be a private-key space, and let $\mathcal{C} = \{C : \mathcal{K} \times \mathbb{F} \rightarrow \mathbb{F}\}$ be a class of circuits with one private input and one semi-private field input. An offline/online HSS scheme for $\mathcal{C}$ is a tuple of PPT algorithms

$$\text{OHSS} = (\text{Setup}, \text{ShareKey}, \text{Share}_0^{\text{sp}}, \text{Share}_1^{\text{sp}}, \text{AuthEval}_0, \text{AuthEval}_1)$$

with the following syntax.

- $\text{Setup}(1^\lambda)$ outputs public parameters pp_H , a secret HSS key sk_H , evaluation keys (ek_0, ek_1) , and a dimension $d = d(\lambda)$. There is an efficiently computable deterministic map

$$f_H : \text{Supp}(\text{sk}_H) \rightarrow \mathbb{F}^d, \quad \vec{\alpha} = f_H(\text{sk}_H),$$

called the authentication-vector map.

- $\text{ShareKey}(\text{sk}_H, K)$, for $K \in \mathcal{K}$, outputs private-input shares (κ_0, κ_1) .
- $\text{Share}_0^{\text{sp}}(\text{pp}_H)$ outputs a uniformly random offline share $\vec{\theta}_0 \xleftarrow{\$} \mathbb{F}^d$, independent of the future semi-private input.
- $\text{Share}_1^{\text{sp}}(\text{sk}_H, \delta, \vec{\theta}_0)$, for $\delta \in \mathbb{F}$, outputs the online party-1 share

$$(\delta, \vec{\theta}_1), \quad \vec{\theta}_1 := \vec{\alpha}\delta + \vec{\theta}_0.$$

Equivalently, the semi-private sharing of δ is the pair $\vec{\theta}_0$ and $(\delta, \vec{\theta}_1)$.

- $\text{AuthEval}_0(ek_0, \kappa_0, \vec{\theta}_0, C)$ outputs a scalar $W_0 \in \mathbb{F}$.
- $\text{AuthEval}_1(ek_1, \kappa_1, (\delta, \vec{\theta}_1), C)$ outputs a scalar $W_1 \in \mathbb{F}$.

The scheme must satisfy the following properties.

1. **Strong correctness.** For every sufficiently large λ , every $C \in \mathcal{C}$, every $K \in \mathcal{K}$, every $\delta \in \mathbb{F}$, and every offline share $\vec{\theta}_0 \in \mathbb{F}^d$, let

$$(\delta, \vec{\theta}_1) := \text{Share}_1^{\text{sp}}(\text{sk}_H, \delta, \vec{\theta}_0),$$

and define W_0, W_1 by the two authenticated-evaluation algorithms. Then

$$W_1 - W_0 = C(K, \delta) \quad \text{in } \mathbb{F},$$

except with negligible probability over setup and private-key sharing randomness.

2. **Public-view simulatability.** The party-1 public view is simulatable without the private input K . More explicitly, for every PPT adversary that adaptively chooses circuits $C_j \in \mathcal{C}$ and semi-private shares $(\delta_j, \vec{\theta}_{1,j})$, the joint distribution of

$$\left(\text{pp}_H, ek_1, \kappa_1, \text{AuthEval}_1(ek_1, \kappa_1, (\delta_j, \vec{\theta}_{1,j}), C_j) \right)_j$$

is computationally indistinguishable from the output of a simulator given only 1^λ , the circuit descriptions, and the semi-private shares. In particular, the simulator is not given K .

2.7 Exponent Verifiable Random Functions

We use a single definition for both publicly verifiable and designated-verifier exponent VRFs. An exponent VRF differs from an ordinary VRF in that the hidden VRF value is an exponent r , while the visible output is the group element $Y = g^r$. The two variants differ only in who holds the verification key and in the leakage allowed in the verifiability experiment.

Let $\text{GGen}(1^\lambda)$ output a cyclic group $G = \langle g \rangle$ of known order $Q = Q(\lambda)$; in the prime-order case we write $Q = q$. Let $\mathcal{X} = \{\mathcal{X}_\lambda\}$ be the input-domain ensemble, and let $\mathcal{D}_R = \{\mathcal{D}_{R,\lambda}\}$ be an efficiently sampleable distribution supported on $\mathbb{Z}_Q$. Unless stated otherwise, the full-range case is $\mathcal{D}_{R,\lambda} = \mathbb{Z}_Q$.

Definition 4 (Exponent-VRF: public and designated-verifier). Fix a variant $\tau \in \{\text{pub}, \text{dv}\}$. A τ -exponent VRF over $(G, g, \mathcal{X}, D_R)$ is a tuple

$$\Pi_\tau = (\text{KeyGen}, \text{Eval}, \text{Verify}, \text{Sim})$$

with the following syntax.

- $(\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda)$ outputs an evaluation key ek and a verification key vk . In public variant, $\text{ek} = \text{sk}$ is secret and $\text{vk} = \text{pk}$ is public. In designated-verifier variant, $\text{ek} = k_P$ is held by the evaluator/prover and vk is held only by the designated verifier.
- $\text{Eval}(\text{ek}, x)$ is written as

$$\text{Eval}(\text{ek}, x) = (\text{Eval}_1(\text{ek}, x), \text{Eval}_2(\text{ek}, x), \text{Prove}(\text{ek}, x)) = (r, Y, \pi),$$

where Eval_1 and Eval_2 are deterministic, $r \in \mathbb{Z}_Q$, $Y = g^r \in G$, and π is a proof. The proof algorithm Prove may be randomized.

- $\text{Verify}(\text{vk}, x, Y, \pi) \in \{0, 1\}$ is deterministic.
- $\text{Sim}(\text{vk}, x, Y)$ is a proof simulator used in the security experiments. In the public variant, if working in the programmable random oracle model, Sim may be an oracle-aided simulator.

The value r is the secret exponent, Y is the exponent output, and (Y, π) is the verifier-facing transcript. We say that Π_τ is secure if the following properties hold:

1. **Correctness.** For every PPT adversary $\mathcal{A}$,

$$\Pr \left[x \in \mathcal{X}_\lambda \wedge (Y \neq g^r \vee \text{Verify}(\text{vk}, x, Y, \pi) = 0) : \begin{array}{l} (\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda), \\ x \leftarrow \mathcal{A}(1^\lambda, \text{vk}), \\ (r, Y, \pi) \leftarrow \text{Eval}(\text{ek}, x) \end{array} \right] \leq \text{negl}(\lambda).$$

2. **Verifiability.** Let

$$\text{leak}_{\text{ver}}^\tau(\text{ek}, \text{vk}) = \begin{cases} (\text{vk}, \text{ek}) & \tau = \text{pub}, \\ \text{ek}, & \tau = \text{dv}. \end{cases}$$

For honestly generated keys define two validation oracles:

$$\mathcal{V}_{\text{real}}^{\text{vk}}(x, Y, \pi) := \text{Verify}(\text{vk}, x, Y, \pi),$$

and

$$\mathcal{V}_{\text{ideal}}^{\text{ek}, \text{vk}}(x, Y, \pi) := \begin{cases} \text{Verify}(\text{vk}, x, Y, \pi), & \text{if } Y = \text{Eval}_2(\text{ek}, x), \\ 0, & \text{otherwise.} \end{cases}$$

For every PPT adversary $\mathcal{A}$,

$$\left| \Pr \left[\mathcal{A}^{\mathcal{V}_{\text{real}}^{\text{vk}}}(1^\lambda, \text{leak}_{\text{ver}}^\tau(\text{ek}, \text{vk})) = 1 : (\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda) \right] - \Pr \left[\mathcal{A}^{\mathcal{V}_{\text{ideal}}^{\text{ek}, \text{vk}}}(1^\lambda, \text{leak}_{\text{ver}}^\tau(\text{ek}, \text{vk})) = 1 : (\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda) \right] \right| \leq \text{negl}(\lambda).$$

In public variant this says that, for an honestly generated public key, no adversary—even one given the evaluation key in the verifiability experiment—can make the public verifier accept an exponent output different from the unique value $\text{Eval}_2(\text{ek}, x)$, except with negligible probability. (Note that in this case, the adversary's verification oracle is useless.) In designated-verifier variant this says that no adversary given the prover/evaluation key, but not vk , can make the hidden verifier accept an incorrect exponent output, except with negligible probability.

3. **Pseudorandomness with simulated proofs.** Define, for $b \in \{0, 1\}$, a stateful oracle $\mathcal{Q}_b^{\text{ek}, \text{vk}}$ that maintains a table $\mathcal{U}$ of exponent outputs. On input $x \in \mathcal{X}_\lambda$:

if $\mathcal{U}[x]$ is undefined and $b = 0$, set $\mathcal{U}[x] := \text{Eval}_2(\text{ek}, x)$;
 if $\mathcal{U}[x]$ is undefined and $b = 1$, sample $r_x \xleftarrow{\$} D_{R, \lambda}$ and set $\mathcal{U}[x] := g^{r_x}$;
 $Y_x := \mathcal{U}[x]$;
 if $b = 0$, compute $(r_x, Y_x, \pi) \leftarrow \text{Eval}(\text{ek}, x)$;
 if $b = 1$, compute $\pi \leftarrow \text{Sim}(\text{vk}, x, Y_x)$;
 return (Y_x, π) .

For every PPT distinguisher $\mathcal{A}$,

$$\left| \Pr \left[\mathcal{A}^{\mathcal{Q}_0^{\text{ek}, \text{vk}}} (1^\lambda, \text{vk}) = 1 : (\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda) \right] - \Pr \left[\mathcal{A}^{\mathcal{Q}_1^{\text{ek}, \text{vk}}} (1^\lambda, \text{vk}) = 1 : (\text{ek}, \text{vk}) \xleftarrow{\$} \text{KeyGen}(1^\lambda) \right] \right| \leq \text{negl}(\lambda).$$

3 Extended Power DDH from Subgroup Indistinguishability

In this section, we show that the assumption of Sparse-Power DCR is implied by DCR. In fact, we take a more general perspective.

Namely, DCR can be viewed as a subgroup indistinguishability problem: $\mathbb{Z}_{N^2}^*$ decomposes as a direct product of a hard subgroup NR_N of N -th residues and a message subgroup $\langle 1 + N \rangle \cong \mathbb{Z}_N$, and the DCR assumption states that one cannot efficiently detect the presence of a $\langle 1 + N \rangle$ -component in a group element, i.e., $g \cdot (1 + N)^\Delta$ is computationally indistinguishable from g for $g \leftarrow \text{NR}_N$ and $\Delta \leftarrow \mathbb{Z}_N$ sampled at random.

Sparse-Power DCR can be viewed as an instance of this principle applied to structured tuples rather than single elements, where the tuple has the form $(g^{r^i})_{i \in T}$ and the shift $(1 + N)^\Delta \in \mathbb{Z}_N$ is applied to some subset $B \subseteq T$ of entries.

Rather than analyze each such assumption in isolation, we introduce a single umbrella problem, the Extended Power DDH problem, that implies a broad family of them. We define this assumption in Definition 6, prove in Theorem 1 that it is implied by subgroup indistinguishability problems, including DCR, and derive Sparse-Power DCR as a special case (Corollary 3).

A natural first attempt at generalizing the subgroup indistinguishability principle to structured tuples is to ask: given a structured tuple of hard-subgroup elements $(g_1, \dots, g_k)$, can each entry be independently blinded by a fresh uniform element of the message subgroup without the adversary noticing? For unstructured tuples this is essentially DCR applied entry-by-entry. But for tuples generated by an algebraic process, such as the Sparse-Power DCR tuple $(g^{r^i})_{i \in T}$, or more generally $(g^{f(\vec{x})})_f$ for a family of polynomials f in secret exponents $\vec{x}$, this naive generalization is too strong to be true. Namely, the real-world tuple satisfies algebraic relations among its entries: if $f_3 = f_1 + f_2$, then $g^{f_3(\vec{x})} = g^{f_1(\vec{x})} \cdot g^{f_2(\vec{x})}$. A tuple blinded by independent uniform message-subgroup elements would not satisfy this relation, and the adversary could trivially distinguish such a tuple from random.

The fix is to blind in a way that respects the algebraic structure of the tuple. Concretely: the blinding factors themselves should satisfy the same linear relations as the underlying polynomials. If $f_3 = f_1 + f_2$, the blinding factor on the third entry should be the product of the blinding factors on the first two; and more generally, the assignment from polynomials to blinding factors should respect $\mathbb{Z}$ -linear combinations. This is exactly the structure of a $\mathbb{Z}$ -linear map.

This motivates our formulation of Extended Power DDH below. We consider a linear space $\mathcal{F}$ of polynomials over the secret exponents, and posit that the oracle $f \mapsto g^{f(\vec{x})}$ is computationally indistinguishable from the oracle $f \mapsto g^{f(\vec{x})} \cdot r(f)$, where r is a uniformly random $\mathbb{Z}$ -linear map from $\mathcal{F}$ to the message subgroup. For linearly independent queries the blinding factors $r(f)$ are jointly independent and uniform, recovering the naive “independent blinding” picture in the regime where it is consistent with the algebraic structure; for linearly dependent queries the blindings inherit exactly the same dependencies as the queries themselves, preserving the relations the real world satisfies.

Definition 5 (Subgroup Indistinguishability [BG10]). A subgroup indistinguishability group sampler is an algorithm $(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda)$ that outputs abelian groups $\mathbb{G} \equiv \mathbb{G}_M \times \mathbb{G}_L$, where $\mathbb{G}_M$ is a cyclic group of order M coprime to $|\mathbb{G}_L|$.

$$\begin{aligned} & \{ (g, \mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \mid (\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda), g \xleftarrow{\$} \mathbb{G} \} \\ \stackrel{\text{comp}}{=} & \{ (g', \mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \mid (\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda), g' \xleftarrow{\$} \mathbb{G}_L \} \end{aligned}$$

Definition 6 (Extended Power DDH). Let $(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda)$ be a subgroup indistinguishability group sampler (Definition 5), and let $(x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})$ be a sampler for a vector of exponents over $(x_1, \dots,$

$x_n) \in \mathbb{Z}^n$. We say that $(\mathbb{G}_{\text{Gen}}, \mathcal{D})$ satisfies Extended Power DDH if

$$\left| \begin{aligned} & \Pr_{\substack{(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda) \\ g \xleftarrow{\$} \mathbb{G}_L \\ (x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})}} [\mathcal{A}^{f \mapsto g^{f(x_1, \dots, x_n)}}(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) = 1] \\ & - \Pr_{\substack{(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda) \\ g \xleftarrow{\$} \mathbb{G}_L \\ (x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})}} [\mathcal{A}^{f \mapsto g^{f(x_1, \dots, x_n) \cdot r(f)}}(\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) = 1] \end{aligned} \right|,$$

where $\mathcal{A}$ must query its oracle on efficiently evaluable polynomials $f_1, \dots, f_Q \in \mathbb{Z}[x_1, \dots, x_n]$ satisfying that, for all prime factors p of M , $\Pr_{\vec{y} \leftarrow \mathbb{Z}_p^n} [f(\vec{y}) \equiv 0 \pmod{p}] \leq \frac{p-1}{p}$,⁵ and $r : \mathbb{Z}[x_1, \dots, x_n] \rightarrow \mathbb{G}_M$ is a random $\mathbb{Z}$ -linear map into the message group.⁶

We prove the following in Section D.

Theorem 1. Let $\mathbb{G}_{\text{Gen}}$ satisfy Definition 5 and $\mathcal{D}(\mathbb{G})$ be an efficiently samplable distribution on $\mathbb{Z}^n$ as in Definition 6. Let L be such that $g^L = 1$ for all $g \in \mathbb{G}_L$. If

$$\begin{aligned} & \left\{ \left(\begin{array}{c} x_1 \bmod L, \dots, x_n \bmod L, \\ x_1 \bmod M, \dots, x_n \bmod M \end{array} \right) \middle| \begin{array}{c} (\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda) \\ (x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G}) \end{array} \right\} \\ \equiv & \left\{ \left(\begin{array}{c} x_1 \bmod L, \dots, x_n \bmod L, \\ y_1, \dots, y_n \end{array} \right) \middle| \begin{array}{c} (\mathbb{G}, \mathbb{G}_M, \mathbb{G}_L, M) \xleftarrow{\$} \mathbb{G}_{\text{Gen}}(1^\lambda) \\ (x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G}) \\ (y_1, \dots, y_n) \xleftarrow{\$} \mathbb{Z}_M^n \end{array} \right\}, \end{aligned}$$

then $(\mathbb{G}_{\text{Gen}}, \mathcal{D})$ satisfies Extended Power DDH.

Corollary 1. Let $\mathbb{G}_{\text{Gen}}$ satisfy Definition 5, let L be such that $g^L = 1$ for all $g \in \mathbb{G}_L$, and let $(x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})$ be the uniform distribution $[0, L]^n$. Then $(\mathbb{G}_{\text{Gen}}, \mathcal{D})$ satisfies Extended Power DDH.

Proof. Let $G \leq 2^{\text{poly}(\lambda)}$ be an efficiently computable upper bound on the order of $\mathbb{G}$ (e.g., based on the bit sizes of elements), and let $\mathcal{D}'(\mathbb{G})$ be the uniform distribution on $[0, 2^\lambda G]^n$. We argue (1) that the distributions $\mathcal{D}$ and $\mathcal{D}'$ lead to statistically indistinguishable Extended Power DDH distributions, and (2) that $\mathcal{D}'$ satisfies the conditions for Theorem 1. For (1), notice that $(x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})$ is only used in the exponent of $g \in \mathbb{G}_L$, and $g^{x_1^d} = g^{x_1^d \bmod L} = g^{(x_1 \bmod L)^d \bmod L}$, so $\mathcal{D}(\mathbb{G})$ only matters modulo L . By assumption $\mathcal{D}(\mathbb{G})$ is uniform modulo L , while $\mathcal{D}'(\mathbb{G})$ modulo L has statistical distance at most

$$\frac{(2^\lambda G) \bmod L}{2^\lambda G} \leq \frac{L-1}{2^\lambda G} < 2^{-\lambda} \frac{L}{G} \leq 2^{-\lambda}$$

from uniform.

For (2), first we have that $\mathcal{D}'(\mathbb{G})$ is efficient to sample from, since it's uniform over an efficiently computable range. Second, since L is coprime to M , we have that the condition for Theorem 1 holds if $\mathcal{D}'(\mathbb{G})$ is uniform modulo $L \cdot M$. It has statistical distance from uniform at most

$$\frac{(2^\lambda G) \bmod (L \cdot M)}{2^\lambda G} \leq \frac{L \cdot M - 1}{2^\lambda G} < 2^{-\lambda} \frac{L \cdot M}{G} \leq 2^{-\lambda}.$$

Corollary 2. Let $\mathbb{G}_{\text{Gen}}$ satisfy Definition 5, let L be such that $g^L = 1$ for all $g \in \mathbb{G}_L$, and let $\mathcal{E}(\mathbb{G}_L)$ be the uniform distribution on all integers in $[1, L)$ that are coprime to L . If $(x_1, \dots, x_n) \xleftarrow{\$} \mathcal{D}(\mathbb{G})$ is the uniform distribution $\mathcal{E}(\mathbb{G}_L)^n$, if all prime factors of M are superpolynomial, and if the Generalized Riemann hypothesis holds, then $(\mathbb{G}_{\text{Gen}}, \mathcal{D})$ satisfies Extended Power DDH.

⁵ This is satisfied if, e.g., the total degree of all f_i less than p .

⁶ Note that while r is infinite-dimensional, it is still well-defined as it is only evaluated on polynomials, which have a finite number of nonzero monomials. We do not specify how to efficiently sample from the random distribution here, but for our application it will be easy to sample from an equivalent distribution. In particular, if for all prime factors p of M , $f_1, \dots, f_Q$ are linearly independent when reduced mod p , then $(r(f_1), \dots, r(f_Q))$ is equivalent to the uniform distribution on $\mathbb{Z}_M^Q$.

Proof. The proof will follow a similar structure to that of Corollary 1. Let $G \leq 2^{\text{poly}(\lambda)}$ be an efficiently computable superpolynomial upper bound on the order of $\mathbb{G}$, let $\mathcal{P}(G^3)$ be the uniform distribution on all primes less than G^3 , and let $\mathcal{D}'(G) = \mathcal{P}(G^3)^n$. We argue (1) that the distributions $\mathcal{D}$ and $\mathcal{D}'$ lead to statistically indistinguishable Extended Power DDH distributions, and (2) that $\mathcal{D}'$ satisfies the conditions for Theorem 1. Similarly to Corollary 1, both statements will be implied by showing that $x \leftarrow \mathcal{P}(G^3)$ satisfies $(x \bmod L, x \bmod M)$ is statistically close to (x', x'') where $x' \leftarrow \mathcal{E}(\mathbb{G}_L)$ is uniform coprime to L , and x'' is uniform modulo M .

Let

$$\pi(z) = \sum_{\substack{p \leq z \\ p \text{ prime}}} 1 \quad \text{and} \quad \pi(z, a, d) = \sum_{\substack{p \leq z \\ p \equiv a \pmod{d} \\ p \text{ prime}}} 1.$$

The chance that $x \leftarrow \mathcal{P}(G^3)$ satisfies $x \leq G$ is at most $\pi(G)/\pi(G^3) = O(G^{-2})$ by the prime number theorem, which is negligible. If prime $x > G \geq L \cdot M$, then it must be coprime to L , so $\mathcal{P}(G^3)$ has negligible probability of sampling a number that is not coprime to L . Next, we argue that $\pi(G^3, a, d)$ is almost independent of a , which implies that $\mathcal{P}(G^3)$ is uniform over all integers coprime to $L \cdot M$. If we show this then we are done, as since M has only superpolynomial prime factors, a random number modulo M has negligible probability of not being coprime, so sampling uniformly coprime to M is statistically close to sampling uniformly modulo M .

To complete the proof, we use a strong bound on the arithmetic progressions of primes implied by the Generalized Riemann hypothesis:

$$\pi(z, a, d) = \frac{1}{\phi(d)} \int_2^z \frac{dt}{\ln t} + \tilde{\pi}(z, a, d) \quad \text{where} \quad \tilde{\pi}(z, a, d) = O(z^{3/5}).$$

See, e.g., [Dav80, pp. 125] for the corresponding bound on

$$\psi(z, a, d) = \sum_{\substack{p \leq z \\ p \equiv a \pmod{d} \\ p \text{ prime}}} \ln p,$$

which can be converted into a bound on $\pi(z, a, d)$ using partial summation. Let $\mathbb{Z}_d^*$ be the set of integers up to d that are coprime to d . Using this bound, we have that the statistical distance from uniform (over the numbers coprime to d) of sampling a random prime up to z , then reducing it modulo d , is at most

$$\begin{aligned} \sum_{a \in \mathbb{Z}_d^*} \left| \frac{\pi(z, a, d)}{\sum_{a' \in \mathbb{Z}_d^*} \pi(z, a', d)} - \frac{1}{\phi(d)} \right| &= \frac{\sum_{a \in \mathbb{Z}_d^*} \left| \pi(z, a, d) - \frac{1}{\phi(d)} \sum_{a' \in \mathbb{Z}_d^*} \pi(z, a', d) \right|}{\sum_{a' \in \mathbb{Z}_d^*} \pi(z, a', d)} \\ &= \frac{\sum_{a \in \mathbb{Z}_d^*} \left| \tilde{\pi}(z, a, d) - \frac{1}{\phi(d)} \sum_{a' \in \mathbb{Z}_d^*} \tilde{\pi}(z, a', d) \right|}{\int_2^z \frac{dt}{\ln t} + \sum_{a' \in \mathbb{Z}_d^*} \tilde{\pi}(z, a', d)} \\ &\leq \frac{O(dz^{3/5})}{z/\ln(z) - O(z/\ln^2(z)) - O(dz^{3/5})} \\ &= \frac{O(d \ln(z) z^{-2/5})}{1 - O(1/\ln(z)) - O(d \ln(z) z^{-2/5})} \end{aligned}$$

Setting $z = G^3$ and $d = L \cdot M \leq G$ gives an upper bound of $O(\ln(G) \cdot G^{-1/5})$, which is negligible.

Application to Sparse Power DCR. We first fix the exponent distribution $\mathcal{D}_E$ for Sparse-Power DCR (Assumption 3) used in our PCF construction in Section 4.

Definition 7 (Exponent distribution $\mathcal{D}_E$). For an RSA modulus N , let

$$\mathcal{E}_N^* := \{ e \in \{0, \dots, N\varphi(N) - 1\} : e \equiv 1 \pmod{N} \text{ and } \gcd(e, N\varphi(N)) = 1 \},$$

and let $\mathcal{D}_{E,N}$ be the uniform distribution over $\mathcal{E}_N^*$. We write $\mathcal{D}_E = \{\mathcal{D}_{E,N}\}_N$. To sample from $\mathcal{D}_{E,N}$, given $\varphi(N)$, sample $e = 1 + Nt \bmod N\varphi(N)$ with $t \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}$, rejecting unless $\gcd(e, N\varphi(N)) = 1$.

Corollary 3 (DCR $\Rightarrow$ Sparse-Power DCR). Let ModGen output $N = pq$ where p, q are superpolynomial in λ and $\gcd(N, \varphi(N)) = 1$. Assume DCR and the Generalized Riemann hypothesis. Then Sparse-Power DCR (Assumption 3) holds for $(T, B, \mathcal{D}_E, \mathcal{D}_\Delta)$ with $\mathcal{D}_E$ as in Definition 7, for every index set T with $\max(T) = \text{poly}(\lambda)$, every support set $B \subseteq T$ (with $0 \notin B$ if $0 \in T$), and every efficiently samplable offset distribution $\mathcal{D}_\Delta$.

Proof. Instantiate subgroup indistinguishability with $G = \mathbb{Z}_{N^2}^*$, $G_L = \mathbb{NR}_N$, $G_M = \langle 1 + N \rangle$, and $L = \varphi(N)$. By the Chinese remainder theorem, $r \leftarrow \mathcal{D}_{E,N}$ is uniform on the units modulo $\varphi(N)$. Since the hard-subgroup powers depend only on exponents modulo $\varphi(N)$, Corollary 2 implies Extended Power DDH for these tuples.

Consider Extended Power DDH with queries $f = 1$ and $f = X^s$ for $s \in T \setminus \{0\}$, and use the answer to $f = 1$ both as g and as the coordinate indexed by 0 if $0 \in T$. In the real world this gives $(g, (g^{r^s})_{s \in T})$, which together with N is exactly distributed as $\mathcal{D}_0$. Distinct monomials are linearly independent modulo p and modulo q , so in the random world we get $(g(1 + N)^{u_0}, (g^{r^s}(1 + N)^{u_s})_{s \in T})$ where u_0 and $(u_s)_{s \in T \setminus \{0\}}$ are independent and uniform over $\mathbb{Z}_N$; call this distribution $\mathcal{R}$. Thus $\mathcal{D}_0 \stackrel{\text{comp}}{=} \mathcal{R}$. Now let Φ be the map that samples $\Delta \leftarrow \mathcal{D}_{\Delta,N}$ and multiplies the coordinate indexed by s by $(1 + N)^{\Delta \chi_B(s)}$, leaving g untouched. By definition $\Phi(\mathcal{D}_0) = \mathcal{D}_1$, and since $0 \notin B$ and $u_s + \Delta \chi_B(s)$ is uniform and independent of everything else for every $s \in B$, we have $\Phi(\mathcal{R}) = \mathcal{R}$. Therefore $\mathcal{D}_1 = \Phi(\mathcal{D}_0) \stackrel{\text{comp}}{=} \Phi(\mathcal{R}) = \mathcal{R} \stackrel{\text{comp}}{=} \mathcal{D}_0$.

4 PCF for Correlated OT

We now present our construction of a PCF for correlated OT over $\mathbb{Z}_N$.

Let $n = n(\lambda)$ be polynomially bounded, and let

$$F^{\text{IPM}} = \{F_z^{\text{IPM}} : \{0, 1\}^n \rightarrow \{0, 1\}\}_{z \in \{0, 1\}^n}$$

be a binary inner-product-membership weak PRF for a public support set $S \subseteq \{0, \dots, n\}$, as in Definition 1. We assume throughout this section that $0 \notin S$, which is satisfied by the BIPSW support $S_{\text{BIPSW}} = [0, n] \cap (6\mathbb{Z} + \{3, 4, 5\})$. Further, let

$$F = \{F_k : \{0, 1\}^n \rightarrow \mathbb{Z}_N\}_{k \in \mathcal{K}}$$

be a weak PRF with range $\mathbb{Z}_N$. The construction realizes the fixed-offset correlated-OT correlation over $\mathbb{Z}_N$ Definition 11, where the setup string ρ_1 for party 1 contains the modulus information (N, p, q) and offset Δ .

Exponent Sampling Convention. To ensure correctness, we need to sample exponents as residues modulo $N\varphi(N)$ that are congruent to 1 modulo N . Define

$$\mathcal{E}_N := \{e \in \mathbb{Z}_{N\varphi(N)} : e \equiv 1 \pmod{N}\} = \{1 + Nt \pmod{N\varphi(N)} : t \in \mathbb{Z}_{\varphi(N)}\} \quad (1)$$

Let $\mathcal{E}_N^*$ be the subset of residues in $\mathcal{E}_N$ that are invertible modulo $N\varphi(N)$; sampling uniformly from $\mathcal{E}_N^*$ corresponds to sampling from the distribution $\mathcal{D}_E$ from Definition 7. Since our key generation algorithm knows $\varphi(N)$ (from ρ_1), it can sample uniformly from $\mathcal{E}_N$ by sampling $t \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}$ and outputting $1 + Nt \pmod{N\varphi(N)}$; it can sample uniformly from $\mathcal{E}_N^*$ by rejection sampling.

Whenever an element of $\mathcal{E}_N$ is used as an ordinary integer exponent, we use its canonical representative in $\{0, \dots, N\varphi(N) - 1\}$, which is congruent to 1 modulo N . Hence, for every $e \in \mathcal{E}_N$ and every $a \in \mathbb{Z}_N$,

$$(1 + N)^{ae} \equiv (1 + N)^a \pmod{N^2}.$$

Note that the receiver's PCF key does not contain $\varphi(N)$, and all exponents in this key are represented by their canonical integer representatives.

Construction. Define

$$\text{PCF}_{\text{COT}} = (\text{PCF.Gen}, \text{PCF.Eval})$$

as in Figure 1.

For intuition behind the correctness and security, we refer to the overview in Section 1.2. We present the full proofs in Section E.1.

4.1 From Correlated OT to $\mathbb{Z}_N$ -VOLE

Given a PCF for correlated OT over $\mathbb{Z}_N$, we can easily transform it into a PCF for VOLE over $\mathbb{Z}_M$ for any modulus M , as follows. Write the OT outputs $(b_x, \mu_x), (m_{0,x}, m_{0,x} + \Delta)$ as $w = \mu_x$, $u = b_x$ and $v = m_{0,x}$. First, if Δ is chosen to be at least a factor 2^λ smaller than N , then except with probability $2^{-\lambda}$, the relation $w = u\Delta + v$ holds over the integers, not just modulo N . Then, we can take a batch of ℓ such outputs, convert them to integers and sum them up to obtain $w = \sum_i 2^i w_i$ etc. and reduce the results modulo the target M . If M is sufficiently close to a large power of two and $\ell = \lceil \log M \rceil$ then w is statistically close to uniform in $\mathbb{Z}_M$. Otherwise, for arbitrary M we can set $\ell = \lceil \log M \rceil + \kappa$ to bound statistical distance to $2^{-\kappa}$. The result is that a single output of a PCF for VOLE modulo M can be obtained using ℓ outputs of the PCF for correlated OT. We present the formal details in Section F.

PCF.Gen($1^\lambda, \rho_0, \rho_1$): Parse $\rho_0 = \perp, \rho_1 = (N, p, q, \Delta)$.

1. Sample $g \xleftarrow{\$} \text{NR}_N$

2. Sample

$$a_0, \alpha_1, \dots, \alpha_n \xleftarrow{\$} \mathcal{E}_N,$$

independently.

3. Sample $r \xleftarrow{\$} \mathcal{E}_N^*$.

4. Sample $z = (z_1, \dots, z_n) \xleftarrow{\$} \{0, 1\}^n$.

5. For every $i \in [n]$, define the sender-side exponent residue

$$a_i := \alpha_i r^{z_i} \pmod{N\varphi(N)}.$$

Since $\alpha_i \equiv 1 \pmod{N}$ and $r \equiv 1 \pmod{N}$, every a_i lies in $\mathcal{E}_N$.

6. For every $s \in \{0, \dots, n\}$, define

$$h_s := g^{a_0 r^s} (1 + N)^{\chi_S(s)\Delta} \pmod{N^2}.$$

7. Sample a weak-PRF key $k \xleftarrow{\$} \mathcal{K}$ for F .

8. Output

$$k_0 = (N, g, k, z, (\alpha_i)_{i \in [n]}, (h_s)_{s=0}^n),$$

$$k_1 = (N, p, q, g, k, \Delta, a_0, (a_i)_{i \in [n]}).$$

PCF.Eval(σ, k_σ, x): Let $x = (x_1, \dots, x_n) \in \{0, 1\}^n$.

1. If $\sigma = 0$, parse

$$k_0 = (N, g, k, z, (\alpha_i)_{i \in [n]}, (h_s)_{s=0}^n).$$

Compute

$$s_x := \sum_{i=1}^n z_i x_i, \quad b_x := \chi_S(s_x) = F_z^{\text{IPM}}(x), \quad e_x := \prod_{i: x_i=1} \alpha_i \in \mathbb{Z}.$$

Set

$$\mu_x := \text{DDLog}_N(h_{s_x}^{e_x}) + F_k(x) \pmod{N},$$

and output

$$y_0(x) := (b_x, \mu_x).$$

2. If $\sigma = 1$, parse

$$k_1 = (N, p, q, g, k, \Delta, a_0, (a_i)_{i \in [n]}).$$

Compute

$$A_x := a_0 \prod_{i: x_i=1} a_i \pmod{N\varphi(N)}.$$

Set

$$m_{0,x} := \text{DDLog}_N(g^{A_x}) + F_k(x) \pmod{N},$$

$$m_{1,x} := \text{DDLog}_N(g^{A_x}(1 + N)^\Delta) + F_k(x) \pmod{N},$$

and output

$$y_1(x) := (m_{0,x}, m_{1,x}).$$

Fig. 1. PCF for COT over $\mathbb{Z}_N$. The exponent sets $\mathcal{E}_N, \mathcal{E}_N^*$ are defined in (1).

4.2 Distributed Setup

The construction in Figure 1 is a setup-programmed PCF: a single key-generation algorithm samples both correlated evaluation keys. We now make precise how the same setup can be realized by the two parties using oblivious transfer. This is a useful OT setup protocol, though it does not qualify as a public-key PCF in the sense of [OSY21, BCM⁺24]. In that stronger syntax, each party publishes a public key once, stores a secret key, and any pair can later derive PCF keys locally from (sk_i, pk_j) and (sk_j, pk_i) without any interaction (i.e. using just one simultaneous round of communication).

To give some intuition about why we require more than one round of communication, observe that the only non-local part of Figure 1 is the component-wise exponent relation

$$a_i = \alpha_i r^{z_i} \pmod{N\varphi(N)}, \quad a_i, \alpha_i \in \mathcal{E}_N, \quad r \in \mathcal{E}_N^*. \quad (2)$$

Party 0 holds z_i and α_i , while party 1 holds the factorization of N , the multiplier r , and a_i . Party 0 must obtain α_i as an actual numerical exponent representative, not merely as a group encoding, because later it computes

$$e_x = \prod_{i: x_i=1} \alpha_i$$

and uses e_x as an exponent in $\mathbb{Z}_{N^2}^*$. This is the precise point at which the known-order public-key lift of constrained Naor–Reingold PCFs stops applying directly. Over a known-order group, an additive correlation in the exponent can be converted into a multiplicative key relation by exponentiating a public generator. Here the relevant exponent ring is $\mathbb{Z}_{N\varphi(N)}$, and the non-factorizing party cannot compute in this ring except on explicitly provided integer representatives.

Consequently, any direct setup must implement the following one-out-of-two transfer for every coordinate:

$$\text{party 1 holds } (a_i, a_i r^{-1}), \quad \text{party 0 obtains } a_i r^{-z_i}.$$

A receiver-public-key OT can make the sender’s response non-interactive once party 0’s OT receiver keys have been posted. This gives a useful one-response OT setup. It is still not a public-key PCF, because party 1 must send pair-specific ciphertexts depending on party 0’s receiver public keys. A simultaneous one-message setup from independently generated public keys would require a non-interactive hidden-order exponent-transfer primitive for (2); we do not know how to instantiate such a primitive from the Paillier/DDLog machinery used here without reintroducing interaction or heavy general-purpose tools.

We formally state the results in Section C as a realization of the key-generation functionality in the OT-hybrid model.

5 PCF for OLE and Multiplication Triples

Using our PCF for correlated OT over $\mathbb{Z}_N$, we can additionally build a PCF for oblivious linear evaluation (OLE) over $\mathbb{F}_2$. Since two OLE correlations can be locally converted into a multiplication triple, this also immediately yields a PCF for multiplication triples over $\mathbb{F}_2$. Although the PCF for random OT from [BCM⁺24] also implies OLE and triples (since random OT is equivalent to OLE over $\mathbb{F}_2$), our approach additionally satisfies a *programmable* feature, meaning that a portion of the PCF keys can be reused to produce another pair of keys for related OLE correlations that share some of the same components. This can be used to build a multi-party PCF for multiplication triples, and also for circuit-dependent preprocessing with low communication online phase, as described in [BCG⁺19, BCG⁺20a].

The idea, which comes from [OSY21], is that we will use the PCF to generate a batch of correlated OTs over $\mathbb{Z}_{N'}$, for a larger modulus $N' \gg N$. These can be seen as a batch of *authenticated bits* $b_i \in \{0, 1\}$, where one party holds the bits together with an information-theoretic MAC on each bit, under keys held by the other party. Since the b_i ’s are bits, and as long as the MAC key Δ is much smaller than the modulus N' , these MACs can be viewed as MACs over the integers. Then, we exploit the fact that using a distributed decryption procedure used in homomorphic secret sharing [OSY21], such authenticated bits can be *multiplied* by ciphertexts under the Goldwasser-Micali cryptosystem, provided we set Δ to equal the secret key of the Goldwasser-Micali scheme, which lies over a different ring $\mathbb{Z}_N$. This leads to an OLE correlation over $\mathbb{F}_2$: one party learns the ciphertext bit a_i , the other learns the authenticated bit b_i , and the two parties obtain additive shares of the product $a_i \cdot b_i$.

In order for Δ to be small compared to the modulus of the underlying correlated-OT PCF, we instantiate the latter with a (much) larger RSA modulus N' , separate from the Goldwasser–Micali modulus N that defines

Δ . Concretely, we pick N' such that $\Delta \ll N'$, so the MAC relation $\mu_i = m_{0,i} + b_i \cdot \Delta$ holds over the integers (with overwhelming probability), enabling the integer-MAC interpretation needed for the distributed decryption.

To produce an unbounded quantity of OLE correlations with our PCF for correlated OT, we need a way to generate an unbounded number of Goldwasser-Micali ciphertexts of unknown random messages. Fortunately, since every element of $\mathbb{Z}_N^*$ with Jacobi symbol 1 is a valid ciphertext, we can just sample them locally from a random oracle on input a public nonce.

The complete construction is presented and proven secure in Section E.2.

Remark 1 (PCF for Degree-2 Correlations). The same key generation procedure for the PCF for OLE can actually be used to build a PCF for *arbitrary* degree-two correlations, that is, the parties can obtain additive shares of $a_i \cdot b_j$ for the i -th and j -th outputs, for any i, j . This is straightforward to see, since the Goldwasser-Micali ciphertexts are generated independently from the correlated OT PCF outputs, so the two values can be “mixed and matched” in an arbitrary manner. For simplicity, we present and analyze the construction solely for the case of OLE correlations.

6 Exponent-VRFs

6.1 Designated-Verifier eVRF from VOLE-PCF

Given a PCF for VOLE over $\mathbb{Z}_q$, we can build an exponent VRF over any cyclic group $\mathbb{G} = \langle g \rangle$ of order q . The construction follows the approach of [KOR23], except we can omit the complex range proof machinery needed in their construction, since we work directly with a PCF over $\mathbb{Z}_q$ rather than $\mathbb{Z}_N$.

The idea is that the prover’s key will contain a VOLE PCF key k_0 , which allows outputting samples (u_i, w_i) . A output of the eVRF will be the value $R = g^{u_i}$, and the proof is simply $\pi = g^{w_i}$. The verifier, given the PCF key k_1 , can compute its VOLE output Δ, v_i where $w_i = v_i + \Delta u_i$. On input the proof π and a purported output R' , it simply checks whether $(R')^\Delta \cdot g^{v_i} = \pi$.

We present the complete construction and its security proof in Section G.1. Note that we obtain an extremely succinct proof size of just one group element, typically 32 bytes when working over a 256-bit elliptic curve group. In contrast, the construction of [KOR23] has proofs of 451 bytes, and the (publicly verifiable) construction of [BCK25] has a proof containing 29 group elements, or 928 bytes over the same curve. (Note that [BHLS25] does not analyze the concrete proof sizes of their eVRFs, and we expect them to be larger.)

6.2 Publicly Verifiable eVRF from Offline/Online HSS

We now separately propose a publicly verifiable eVRF template based on offline/online HSS. Note that this is independent of the Paillier PCF in Section 4; we make precise what is needed from an HSS scheme for attaining a publicly verifiable eVRF: an authenticated semi-private evaluation of the map

$$(K, \delta) \mapsto \delta F_K(x)$$

with public verification of consistency. The template follows the offline/online HSS paradigm of [MORS25, KOR23].

Let $\mathbb{F} = \mathbb{Z}_q$ for a prime q , let $\mathbb{G} = \langle g \rangle$ be a group of order q , and let $F = \{F_K : \mathcal{X} \rightarrow \mathbb{F}\}$ be a PRF. Let

$$\text{Com}(\vec{m}; \rho) = h_C^\rho \prod_{i=1}^d g_{C,i}^{m_i}$$

be a perfectly hiding and computationally binding Pedersen commitment to vectors in $\mathbb{F}^d$. Let $H_{\text{FS}} : \{0, 1\}^* \rightarrow \mathbb{F}^*$ be a programmable random oracle implementing Fiat–Shamir.

Offline/Online HSS Scheme. We use a offline/online HSS scheme OHSS in the sense of Definition 3, instantiated for the circuit family

$$C_x(K, \delta) := \delta F_K(x), \quad x \in \mathcal{X}.$$

For keys generated by OHSS.Setup, write

$$\vec{\alpha} := f_H(\text{sk}_H) \in \mathbb{F}^d$$

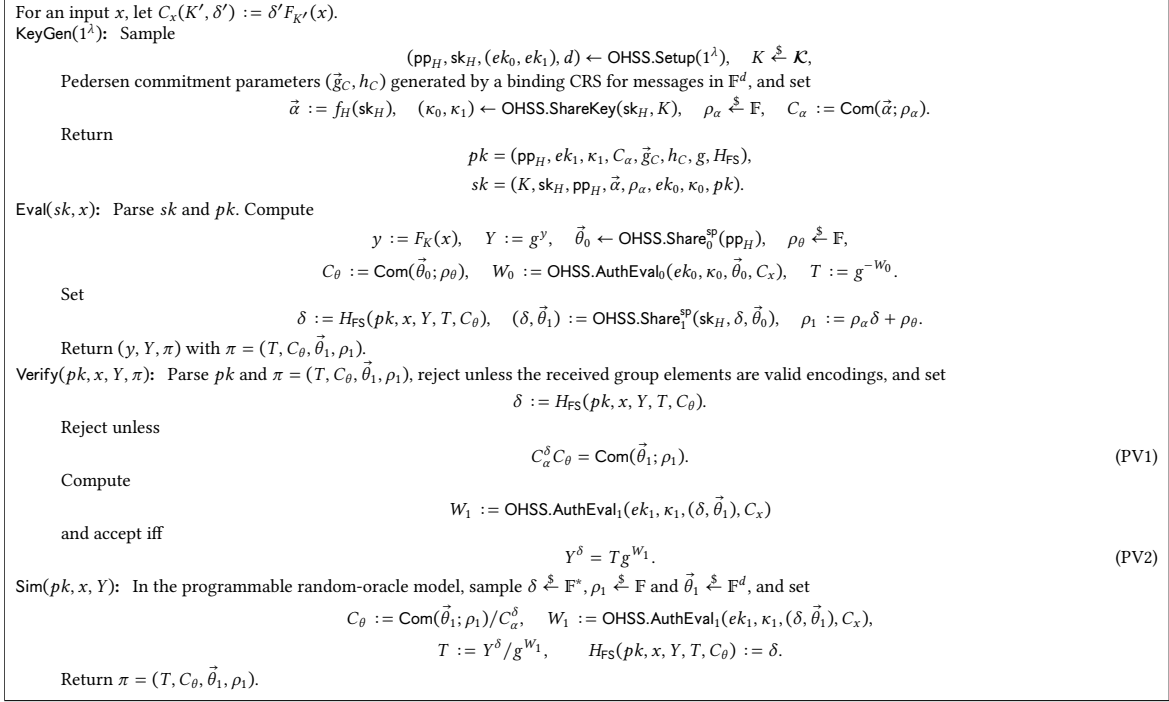


Fig. 2. Publicly verifiable eVRF template based on offline/online HSS.

for the authentication vector. Thus, for every K , every challenge $\delta \in \mathbb{F}$, and every valid offline/online sharing

$$\vec{\theta}_0 \leftarrow \text{OHSS.Share}_0^{\text{sp}}(\text{pp}_H), \quad (\delta, \vec{\theta}_1) \leftarrow \text{OHSS.Share}_1^{\text{sp}}(\text{sk}_H, \delta, \vec{\theta}_0),$$

the scalar authenticated evaluations satisfy

$$W_1 - W_0 = C_x(K, \delta) = \delta F_K(x), \tag{*}$$

where

$$W_0 := \text{OHSS.AuthEval}_0(ek_0, \kappa_0, \vec{\theta}_0, C_x),$$

$$W_1 := \text{OHSS.AuthEval}_1(ek_1, \kappa_1, (\delta, \vec{\theta}_1), C_x).$$

The privacy assumption used to argue the security is the public-view simulatability of Definition 3: the public verifier-side HSS view $(\text{pp}_H, ek_1, \kappa_1)$ and its local authenticated-evaluation outputs are simulatable without the PRF key K .

Construction. The public eVRF $\Pi_{\text{pub}}[F, \text{OHSS}]$ is defined as in Figure 2, together with the simulator used in the security proof. We defer the correctness and security proofs to Section G.

Note that the proof π consists of the four elements $(T, C_\theta, \vec{\theta}_1, \rho_1)$, where T is a group element that authenticates the output of the homomorphic computation of $C_x(K, \delta)$, C_θ is a Pedersen commitment to the HSS share θ_0 of the challenge δ , while θ_1 is the other share derived after computing the challenge, and finally ρ_1 is the commitment opening used to verify correctness of θ_1 . Thus the proof contains one eVRF-group element, one commitment-group element, and $d + 1$ field elements; if $G_C = \mathbb{G}$, this is 2 group elements and $d + 1$ scalars where d is the dimension of the authentication vector. In the common 256-bit-curve encoding, the proof is $32d + 96$ bytes; using the offline/online HSS of [MORS25], key length is $d = 1$, giving a proof size of 128 bytes, or 160 bytes including the exponent output Y .

6.3 Applications to Threshold Schnorr and ECDSA

Exponent VRFs are a useful tool for reliably obtaining exponent-nonces for threshold cryptographic protocols, as shown in [KOR23, BHLS25]. They allow the prover to deterministically produce a hidden exponent

$r = F_K(x)$, release only $R = g^r$, and prove to the relevant verifier that R is the correct group encoding of the deterministic exponent. This was used for stateless deterministic two-party Schnorr signing [KOR23] (using designated-verifier eVRFs), while [BHLS25] derived threshold Schnorr, two-party ECDSA, auditable threshold signing, and verifiable hierarchical deterministic (HD) key derivation from publicly verifiable eVRFs. Our eVRF constructions can be plugged into all of these applications, with the result that the communication will be significantly reduced due to the small proof sizes of the eVRF.

Two-Party Threshold Schnorr. For Schnorr, the signing equation is linear in the nonce and secret-key shares. The transformation from [KOR23] requires a stateless deterministic way to generate the nonce commitment and enough PCF-generated arithmetic correlation to check or combine the parties' local signing equations. Instantiating the PCF layer with Theorems 3 and 5 gives the same deterministic threshold Schnorr structure after a short setup phase, while the eVRF formulation gives an explicit proof object for the hidden nonce exponent. The setup can be realized by the OT-based protocol of Theorem 2; after this setup, the parties expand arbitrarily many signing nonces and arithmetic checks locally.

Two-Party Threshold ECDSA. ECDSA is more demanding because signing requires multiplicative relations involving the nonce inverse and the secret key. The eVRF compiler of Boneh–Haitner–Lindell–Segev uses the eVRF as a deterministic, verifiable nonce source and combines it with standard arithmetic preprocessing for the remaining multiplication/inversion checks [BHLS25]. Our construction supplies the two required ingredients separately: the eVRF produces $R = g^r$ without revealing r , and the COT-to-VOLE and OLE/triple transformations supply silent arithmetic correlations over $\mathbb{Z}_N$, over lifted integers, and over a prime signing field. The prime-field instantiation must include the statistical reduction terms of the integer-lifting step (Corollary 8); these terms are the only additional loss beyond the compiler's own ECDSA proof and the PCF/eVRF assumptions.

Public Auditability and HD Derivation. The public eVRF template of Theorem 13 gives public uniqueness of the exponent output. Observe that this matches the auditability applications of [BHLS25]: a signer can later prove that a published nonce commitment or derived public key was computed from the committed hidden exponent, while the exponent itself remains hidden. In the HD-derivation application, the input x is a derivation path or context string, and the output $Y = g^{F_K(x)}$ is combined with the base public key in the usual additive or multiplicative derivation rule. Public uniqueness prevents equivocation across auditors, while proof simulation preserves pseudorandomness of unseen derivation paths.

7 Experimental Setup

We implement and benchmark the Paillier PCF-COT in Rust at the 2048-bit RSA modulus level, i.e., with exponentiation in the Paillier group modulo N^2 where $|N^2| \approx 4096$ bits on a Ubuntu 22.04 LTS server running on an Intel Xeon Silver 4314 CPU @ 2.40 GHz.

The benchmark uses Rust release mode with native code generation, fat LTO, one codegen unit, AVX-512 IFMA enabled by the crate feature `avx512`, and Rayon parallelism over the 32 hardware threads. Unless otherwise stated, the parameters are $|N| = 2048$, input length $n = 128$, batch size $B = 256$, subset-product window $w_p = 8$, fixed-base window $w = 4$, Criterion sample size 20, and Criterion measurement time 10 s. We port the residue number system (RNS) representation of the vectorized algorithms from [LHD26] to Rust in order to reduce the latency of modular arithmetic within the scope of our implementation.

We provide further details in Section H.

7.1 Benchmark Results

The timing tables should be interpreted asymmetrically. Sender evaluation has access to $\varphi(N)$, so can reduce the large integer exponents before performing exponentiation, while the receiver evaluation cannot use $\varphi(N)$, leading to a substantial slowdown over the sender. Batch rows report both aggregate throughput and amortised latency per underlying PCF-COT instance.

We observe that the sender side is already competitive within the Paillier regime. In the 32-thread batch benchmark, sender evaluation reaches almost 13,600 COT/s, or around $74 \mu\text{s}$ amortized per COT. The receiver side, on the other hand, reaches around 32 COT/s in batch, or roughly 31 ms amortized per COT. The powers-of-two VOLE aggregation also adds essentially no measurable overhead relative to generating the underlying

Table 1. RNS/AVX-512 PCF-COT timings for $|N| = 2048$, $n = 128$, and batch size $B = 256$.

Operation	Mean time	Throughput	Amortised latency
Key generation	483.40 ms	2.07 keys/s	–
Receiver single evaluation	522.80 ms	1.91 COT/s	522.80 ms
Sender single evaluation	1.53 ms	654.55 COT/s	1.53 ms
Receiver batch, Rayon	7.94 s	32.24 COT/s	31.02 ms
Sender batch, Rayon	18.91 ms	13,537.84 COT/s	73.87 μ s

Table 2. RNS/AVX-512 powers-of-two $\mathbb{Z}_N$ -VOLE aggregation timings. Each block aggregates $B = 256$ PCF-COT samples.

Operation	Mean time/block	Throughput
Receiver VOLE agg., Rayon	7.93 s	0.13 blocks/s
Sender VOLE agg., Rayon	18.92 ms	52.84 blocks/s

$B \approx \log p$ COT samples, since it consists only of local doublings and additions in $\mathbb{Z}_N$: aggregating a block of $B = 256$ PCF-COT samples costs 18.92 ms on the sender side and 7.93 s on the receiver side, matching the corresponding batch rows of Table 1.

Finally, the PCF key sizes are fairly compact, with a total key pair size of 195 KiB for $|N| = 2048$ and $n = 128$.

7.2 Comparison with Prior Work

Table 3 compares our construction with the three most closely related, concretely efficient PCFs: the constrained Naor–Reingold PCF of [BCM⁺24], the Paillier-based PCF of [OSY21], and the sparse-LPN PCF of [BCM⁺25]. Evaluation times are single-core and per correlation; the corresponding batched, amortized times for our construction are 74 μ s (sender) and 31 ms (receiver).

Table 3. Comparison with prior concretely efficient PCFs. “PK” denotes a non-interactive (public-key) setup, and “DKG” a distributed key-generation protocol making non-black-box use of a PRG. Key sizes are sender key / receiver key, and evaluation times are single-core per correlation. Our $\mathbb{Z}_N$ -COT is locally aggregated into VOLE over an arbitrary modulus (Section 4.1).

Scheme	Correlation	Assumption	Setup	Keys (k_1/k_0)	Eval. (ms)	
					S	R
[BCM ⁺ 24]	random OT	power-DDH	2 rds., OT	16 B / 30 KiB	0.1	0.1
[OSY21]	$\mathbb{F}_2^\lambda$ -COT	DCR, QR	1 rd., PK	≈ 50 KiB (each)	105	105
[BCM ⁺ 25]	COT, VOLE	sparse LPN	DKG	≈ 3.4 MiB (pair)	$\ll 1$	$\ll 1$
This work	$\mathbb{Z}_N$ -COT	DCR	2 rds., OT	65.8 KiB / 129.3 KiB	1.53	523

For [BCM⁺24], the table reports the precomputable variant. That work also gives a non-interactive public-key variant, with reported sender/receiver public-key sizes of 32.8 kB / 38.4 kB for its balanced XOR–MAJ instantiation. The comparison with [BCM⁺24] is only partial, since that work directly builds *random* OT rather than correlated OT with a fixed integer offset. Converting random OT into our relation requires the sender to publish one correction value per output, so the evaluation is no longer silent. Thus, while [BCM⁺24] is much faster, it does not produce the correlation that we target.

The Paillier-based PCF of [OSY21] needs around 128 exponentiations in $\mathbb{Z}_N^*$ per 128-bit COT; using our RNS-optimized implementation of the underlying arithmetic, we estimate around 105 ms per COT on either side, with keys of around 50 KiB each. Our sender is therefore roughly 69 $\times$ faster and our receiver roughly 5 $\times$ slower, with a key pair roughly 2 $\times$ larger. This is a tradeoff rather than a strict improvement, with the further differences that we obtain integer-correlated OT rather than XOR-correlated OT, while [OSY21] has a true public-key setup.

Compared with the sparse-LPN PCF of [BCM⁺25], our key pair is about 18 $\times$ smaller, while their evaluation is several orders of magnitude faster. On the other hand, our setup is a simple two-round pairwise

protocol based on parallel OT, and avoids the expensive, non-black-box distributed key-generation machinery required to set up the distributed point functions of their sparse-LPN construction.

Acknowledgments. A. Dutta and P. Scholl were supported by Cyberagentur under the project “Multi-Party Computation in the Confidential Cloud (MPCC)”. L. Kohl was supported by a NWO (Dutch Research Council) grant with number VI.Veni.222.348 and by NWO Gravitation project QSC. L. Roy was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement number 101124977 (DECRYSIS), and by the Swiss National Science Foundation (SNSF) under the Starting grant “Low-Entropy Cryptography for the Quantum Era”. Part of this work was conducted while L. Kohl, L. Roy, and P. Scholl were visiting the Simons Institute for the Theory of Computing.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

- AR16. B. Applebaum and P. Raykov. On the relationship between statistical zero-knowledge and statistical randomized encodings. In *CRYPTO 2016, Part III, LNCS 9816*, pages 449–477. Springer, Berlin, Heidelberg, August 2016.
- ARS24. D. Abram, L. Roy, and P. Scholl. Succinct homomorphic secret sharing. In *EUROCRYPT 2024, Part VI, LNCS 14656*, pages 301–330. Springer, Cham, May 2024.
- BCG⁺19. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In *CRYPTO 2019, Part III, LNCS 11694*, pages 489–518. Springer, Cham, August 2019.
- BCG⁺20a. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Correlated pseudorandom functions from variable-density LPN. In *61st FOCS*, pages 1069–1080. IEEE Computer Society Press, November 2020.
- BCG⁺20b. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, and P. Scholl. Efficient pseudorandom correlation generators from ring-LPN. In *CRYPTO 2020, Part II, LNCS 12171*, pages 387–416. Springer, Cham, August 2020.
- BCG⁺22. E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, N. Resch, and P. Scholl. Correlated pseudorandomness from expand-accumulate codes. In *CRYPTO 2022, Part II, LNCS 13508*, pages 603–633. Springer, Cham, August 2022.
- BCGI18. E. Boyle, G. Couteau, N. Gilboa, and Y. Ishai. Compressing vector OLE. In *ACM CCS 2018*, pages 896–912. ACM Press, October 2018.
- BCK25. B. Bünz, K. Choi, and C. Komlo. Golden: Lightweight non-interactive distributed key generation. Cryptology ePrint Archive, Report 2025/1924, 2025.
- BCM⁺24. D. Bui, G. Couteau, P. Meyer, A. Passelègue, and M. Riahinia. Fast public-key silent OT and more from constrained Naor-Reingold. In *EUROCRYPT 2024, Part VI, LNCS 14656*, pages 88–118. Springer, Cham, May 2024.
- BCM⁺25. L. Braun, G. Couteau, K. Melissaris, M. Riahinia, and E. Sadeghi. Fast pseudorandom correlation functions from sparse LPN. In *ASIACRYPT*, 2025.
- BG10. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO 2010, LNCS 6223*, pages 1–20. Springer, Berlin, Heidelberg, August 2010.
- BGI16. E. Boyle, N. Gilboa, and Y. Ishai. Breaking the circuit size barrier for secure computation under DDH. In *CRYPTO 2016, Part I, LNCS 9814*, pages 509–539. Springer, Berlin, Heidelberg, August 2016.
- BHLS25. D. Boneh, I. Haitner, Y. Lindell, and G. Segev. Exponent-VRFs and their applications. In *EUROCRYPT 2025, Part VII, LNCS 15607*, pages 195–224. Springer, Cham, May 2025.
- BIP⁺18. D. Boneh, Y. Ishai, A. Passelègue, A. Sahai, and D. J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In *TCC 2018, Part II, LNCS 11240*, pages 699–729. Springer, Cham, November 2018.
- BKS19. E. Boyle, L. Kohl, and P. Scholl. Homomorphic secret sharing from lattices without FHE. In *EUROCRYPT 2019, Part II, LNCS 11477*, pages 3–33. Springer, Cham, May 2019.
- BN06. M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *ACM CCS 2006*, pages 390–399. ACM Press, October / November 2006.
- CMR23. G. Couteau, P. Meyer, A. Passelègue, and M. Riahinia. Constrained pseudorandom functions from homomorphic secret sharing. In *EUROCRYPT 2023, Part III, LNCS 14006*, pages 194–224. Springer, Cham, April 2023.
- Dav80. H. Davenport. *Multiplicative Number Theory, Graduate Texts in Mathematics 74*. Springer-Verlag, second edition, 1980.
- DJ01. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier’s probabilistic public-key system. In *PKC 2001, LNCS 1992*, pages 119–136. Springer, Berlin, Heidelberg, February 2001.

- HR25. S. Hasler and P. Reiser. Pseudorandom correlation functions for multiparty beaver triples from sparse LPN. Cryptology ePrint Archive, Report 2025/2002, 2025.
- HRK25. S. Hasler, P. Reiser, and R. Küsters. Pseudorandom correlation functions from ring-LWR. In *ASIACRYPT*, 2025.
- IKNP03. Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In *CRYPTO 2003, LNCS 2729*, pages 145–161. Springer, Berlin, Heidelberg, August 2003.
- ILL25. Y. Ishai, H. Li, and H. Lin. Succinct homomorphic MACs from groups and applications. In *STOC*, 2025.
- Kon25. Y. Kondi. Two-party ECDSA signing at constant communication overhead. Cryptology ePrint Archive, Report 2025/1813, 2025.
- KOR23. Y. Kondi, C. Orlandi, and L. Roy. Two-round stateless deterministic two-party Schnorr signatures from pseudorandom correlation functions. In *CRYPTO 2023, Part I, LNCS 14081*, pages 646–677. Springer, Cham, August 2023.
- LHD26. S. Langowski, K. He, and S. Devadas. VROOM: Accelerating (almost all) number-theoretic cryptography using vectorization and the residue number system. Cryptology ePrint Archive, Paper 2026/393, 2026.
- MMST25. P. Miao, A. Murphy, A. Srinivasan, and M. Tromanhauser. Pseudorandom correlation generators for multiparty beaver triples over F_2 . Cryptology ePrint Archive, Report 2025/1182, 2025.
- MORS25. P. Meyer, C. Orlandi, L. Roy, and P. Scholl. Silent circuit relinearisation: Sublinear-size (boolean and arithmetic) garbled circuits from DCR. In *CRYPTO 2025, Part IV, LNCS 16003*, pages 426–458. Springer, Cham, August 2025.
- NR97. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.
- OSY21. C. Orlandi, P. Scholl, and S. Yakubov. The rise of paillier: Homomorphic secret sharing and public-key silent OT. In *EUROCRYPT 2021, Part I, LNCS 12696*, pages 678–708. Springer, Cham, October 2021.
- Pai99. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT’99, LNCS 1592*, pages 223–238. Springer, Berlin, Heidelberg, May 1999.
- PS00. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.
- Roy22. L. Roy. SoftSpokenOT: Quieter OT extension from small-field silent VOLE in the minicrypt model. In *CRYPTO 2022, Part I, LNCS 13507*, pages 657–687. Springer, Cham, August 2022.
- RS21. L. Roy and J. Singh. Large message homomorphic secret sharing from DCR and applications. In *CRYPTO 2021, Part III, LNCS 12827*, pages 687–717, Virtual Event, August 2021. Springer, Cham.

A Additional Related Work

PCGs, PCFs, and LPN/LWR-based correlated pseudorandomness. Pseudorandom correlation generators and functions were introduced to remove the linear communication cost of preprocessing in secure computation [BCG⁺19, BCG⁺20a]. The original PCF constructions are based on structured variants of LPN and show how to obtain on-demand OT/VOLE-type correlations from short correlated keys. The expand-accumulate line improves the concrete cost of PCGs and PCFs by using sparse expansion followed by an accumulation step [BCG⁺22]. More recent work gives PCFs from sparse LPN, PCFs from ring-LWR, and PCFs for OLE and multiparty Beaver triples over finite fields [BCM⁺25, HRK25, HR25]. These works are closest to ours in the range of supported arithmetic correlations. The main difference is structural: they rely on noisy-linear assumptions and LPN/LWR-style decoding phenomena, whereas our construction is number-theoretic and follows a constrained Naor–Reingold evaluation pattern inside the Paillier group. Another difference is that LPN-based PCFs typically rely on distributed point functions with an exponentially large domain size, and creating a distributed key generation protocol for these is challenging, requiring non-black-box evaluation of pseudorandom generators inside 2-PC.

Constrained Naor–Reingold PCFs. Bui, Couteau, Meyer, Passelègue, and Riahinia construct efficient PCFs for OT from constrained PRFs, instantiated by variants of the Naor–Reingold PRF together with low-complexity weak PRFs such as BIPSW and GAR [BCM⁺24]. Their construction is highly optimized for OT and reports public-key PCF throughput in the range of tens of thousands of OT correlations per second. Our construction deliberately gives up this prime-order OT specialization. The contribution is orthogonal: we keep the constrained-Naor–Reingold cancellation mechanism, but move it to $\mathbb{Z}_{N^2}^*$ so that a selected table entry can additionally carry an element of the easy-discrete-log subgroup $\langle 1 + N \rangle$. This subgroup element is what turns a random-OT-style selected group element into a fixed arithmetic offset, and is what makes the subsequent COT-to-VOLE aggregation possible.

Weak PRFs and inner-product membership. The constrained-Naor–Reingold approach needs a predicate that is weakly pseudorandom on random inputs and has a compact representation compatible with the exponent table. We use the BIPSW candidate of Boneh, Ishai, Passelègue, Sahai, and Wu [BIP⁺18], written as an inner-product-membership predicate

$$x \mapsto \chi_s(\langle z, x \rangle)$$

for the support $S = [0, n] \cap (6\mathbb{Z} + \{3, 4, 5\})$. The construction is modular in this point: any binary inner-product-membership weak PRF with an explicitly described support set can replace BIPSW. What is not modular is the sparse table-hiding assumption, since the support set of the weak PRF is exactly the support on which the Paillier table is shifted.

Paillier DDLog, HSS, and public-key silent OT/VOLE. The closest number-theoretic predecessor is the Paillier DDLog technique of Orlandi, Scholl, and Yakoubov [OSY21], building on the HSS/DDLog tradition of Boyle, Gilboa, and Ishai [BGI16] and subsequent DCR-based HSS work [RS21]. Orlandi–Scholl–Yakoubov show how to convert multiplicative sharing in Paillier groups into additive sharing modulo N and use this to obtain HSS, PCFs for OT and VOLE, and public-key silent OT/VOLE from QR/DCR. We therefore do not claim that Paillier/DDLog first yields VOLE here. The distinction is more specific: our construction embeds the Paillier DDLog shift into a constrained Naor–Reingold table, so that a weak-PRF predicate selects whether the DDLog output contains the hidden offset Δ . This gives a different route to fixed-offset COT and arithmetic VOLE, with a different key structure and a different assumption profile.

OLE, triples, and programmability. Correlated OT, OLE, and multiplication triples are interreducible in several preprocessing models, but the reductions do not automatically preserve the PCF property, programmability, or efficient key reuse. LPN/LWR-based PCFs and PCGs already support OLE, authenticated triples, and more general degree-two correlations in various settings [BCG⁺22, HR25, MMST25]. Our Goldwasser–Micali/DDLog section can be viewed in this context. We construct it as a Paillier-based template showing how the fixed-offset COT keys can be interpreted as authenticated bits and combined with distributed Goldwasser–Micali decryption.

Exponent VRFs and discrete-log PCFs. Exponent VRFs output a group element g^r together with evidence that the hidden exponent r is the deterministic VRF value. They are motivated in part by threshold signing, where deterministic and verifiable per-message nonce generation is useful but the nonce exponent itself must remain hidden [KOR23, BHLS25]. Kondi, Orlandi, and Roy study closely related discrete-log PCFs and use VOLE-style relations to verify exponent outputs [KOR23]. Our designated-verifier eVRF follows the same algebraic identity

$$w = v + \Delta u \implies g^w = g^v(g^u)^\Delta,$$

but obtains the underlying VOLE from the Paillier PCF. The technical point is that the PCF is weak, so adversarially chosen eVRF inputs must first be hashed into random PCF inputs in the random-oracle model. Once this is done, the designated-verifier security directly follows from VOLE-PCF pseudorandomness and insider security.

B Additional Preliminaries

B.1 Pseudorandom Functions

We give the standard definition of a PRF and weak PRF.

Definition 8 (PRF and weak PRF). Let $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$, $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, and $\mathcal{K} = \{\mathcal{K}_\lambda\}_{\lambda \in \mathbb{N}}$ be efficiently samplable ensembles. A keyed family

$$F = \{F_k : \mathcal{X}_\lambda \rightarrow \mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}, k \in \mathcal{K}_\lambda}$$

is a pseudorandom function if for every PPT distinguisher $\mathcal{A}$,

$$\left| \Pr[\mathcal{A}^{F_k(\cdot)}(1^\lambda) = 1 : k \xleftarrow{\$} \mathcal{K}_\lambda] - \Pr[\mathcal{A}^{R(\cdot)}(1^\lambda) = 1 : R \xleftarrow{\$} (\mathcal{X}_\lambda \rightarrow \mathcal{Y}_\lambda)] \right| \leq \text{negl}(\lambda).$$

It is a weak pseudorandom function if for every PPT distinguisher $\mathcal{A}$ and every polynomially bounded $m = m(\lambda)$,

$$\left\{ ((x_i, F_k(x_i)))_{i \in [m]} : k \xleftarrow{\$} \mathcal{K}_\lambda, x_i \xleftarrow{\$} \mathcal{X}_\lambda \right\} \stackrel{\text{comp}}{\equiv} \left\{ ((x_i, y_i))_{i \in [m]} : x_i \xleftarrow{\$} \mathcal{X}_\lambda, y_i \xleftarrow{\$} \mathcal{Y}_\lambda \right\}.$$

B.2 Pseudorandom Correlation Functions and Reverse-Sampleable Correlations

Before defining PCFs, we present the necessary definition of a reverse-sampleable correlation. We use the variant with a *setup* that encodes information that is fixed for all samples of the correlation; in our case, this is the RSA modulus and the correlation offset Δ .

Definition 9 (Reverse-sampleable correlation with setup). Let $\ell_0, \ell_1 \in \text{poly}(\lambda)$, and let $\mathcal{R}_{\sigma, \lambda}$, for $\sigma \in \{0, 1\}$ be a pair of sets or efficiently samplable distributions. A correlation with setup consists of a probabilistic polynomial-time algorithm $\mathcal{Y}$ that, on input the setup strings (ρ_0, ρ_1) , samples a pair:

$$(y_0, y_1) \in \{0, 1\}^{\ell_0(\lambda)} \times \{0, 1\}^{\ell_1(\lambda)}.$$

The correlation $\mathcal{Y}$ is reverse-sampleable if there exists a PPT algorithm $\mathcal{Y}.\text{RSample}$ such that for every ρ_0, ρ_1 in the support of $\mathcal{R}_{0, \lambda} \times \mathcal{R}_{1, \lambda}$, and every $\sigma \in \{0, 1\}$, we require

$$\left\{ (y_0, y_1) : (y_0, y_1) \leftarrow \mathcal{Y}(1^\lambda, \rho_0, \rho_1) \right\}^{\text{stat}} \equiv \left\{ \begin{array}{l} (y_0, y_1) \leftarrow \mathcal{Y}(1^\lambda, \rho_0, \rho_1), \\ (y'_0, y'_1) : y'_\sigma := y_\sigma, \\ y'_{1-\sigma} \leftarrow \mathcal{Y}.\text{RSample}(1^\lambda, \sigma, \rho_0, \rho_1, y'_\sigma) \end{array} \right\}.$$

Definition 10 (Weak pseudorandom correlation function). Let $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ be an input ensemble, and let $\mathcal{Y}$ be a reverse-sampleable correlation with setup as in Definition 9. For $\sigma \in \{0, 1\}$, let $\mathcal{R}_{\sigma, \lambda}$ denote the set, or efficiently samplable distribution, of admissible setup-programming strings for the σ -th party. If a party has no setup input, we take $\mathcal{R}_{\sigma, \lambda}$ to be singleton.

A weak PCF for $\mathcal{Y}$ consists of algorithms

$$\text{PCF} = (\text{PCF.Gen}, \text{PCF.Eval})$$

with the following syntax.

- $(k_0, k_1) \leftarrow \text{PCF.Gen}(1^\lambda, \rho_0, \rho_1)$ is a PPT key-generation algorithm, where $\rho_\sigma \in \mathcal{R}_{\sigma, \lambda}$.
- $\text{PCF.Eval}(\sigma, k_\sigma, x)$ is deterministic, takes $\sigma \in \{0, 1\}$, a key k_σ , and an input $x \in \mathcal{X}_\lambda$, and outputs an element in the σ -th output space of the correlation.

We require the following properties.

1. **$\mathcal{Y}$ -pseudorandomness.** For a PPT distinguisher $\mathcal{A}$, a polynomially bounded sample count $m = m(\lambda)$, and a bit $b \in \{0, 1\}$, define the experiment

$$\text{Exp}_{\text{PCF}, \mathcal{A}, m, b}^{\text{pr}}(1^\lambda, \rho_0, \rho_1)$$

as follows.

$\text{Exp}_{\text{PCF}, \mathcal{A}, m, 0}^{\text{pr}}(1^\lambda)$	$\text{Exp}_{\text{PCF}, \mathcal{A}, m, 1}^{\text{pr}}(1^\lambda)$
$(\rho_0, \rho_1) \leftarrow \mathcal{R}_{0, \lambda} \times \mathcal{R}_{1, \lambda}$ $T \leftarrow \emptyset$ for $i = 1, \dots, m$ do $x_i \xleftarrow{\$} \mathcal{X}_\lambda$ if $T[x_i]$ is undefined then $(y_0, y_1) \leftarrow \mathcal{Y}(1^\lambda, \rho_0, \rho_1)$ $T[x_i] \leftarrow (y_0, y_1)$ $(y_0^{(i)}, y_1^{(i)}) \leftarrow T[x_i]$ $b' \leftarrow \mathcal{A}(1^\lambda, (x_i, y_0^{(i)}, y_1^{(i)})_{i \in [m]})$ output b'	$(\rho_0, \rho_1) \leftarrow \mathcal{R}_{0, \lambda} \times \mathcal{R}_{1, \lambda}$ $(k_0, k_1) \leftarrow \text{PCF.Gen}(1^\lambda, \rho_0, \rho_1)$ for $i = 1, \dots, m$ do $x_i \xleftarrow{\$} \mathcal{X}_\lambda$ $y_0^{(i)} \leftarrow \text{PCF.Eval}(0, k_0, x_i)$ $y_1^{(i)} \leftarrow \text{PCF.Eval}(1, k_1, x_i)$ $b' \leftarrow \mathcal{A}(1^\lambda, (x_i, y_0^{(i)}, y_1^{(i)})_{i \in [m]})$ output b'

We require that for every PPT distinguisher $\mathcal{A}$ and every polynomially bounded $m = m(\lambda)$,

$$\left| \Pr[\text{Exp}_{\text{PCF}, \mathcal{A}, m, 0}^{\text{pr}}(1^\lambda, \rho_0^{(\lambda)}, \rho_1^{(\lambda)}) = 1] - \Pr[\text{Exp}_{\text{PCF}, \mathcal{A}, m, 1}^{\text{pr}}(1^\lambda, \rho_0^{(\lambda)}, \rho_1^{(\lambda)}) = 1] \right| \leq \text{negl}(\lambda).$$

2. **Weak insider security.** For a PPT distinguisher $\mathcal{A}$, a polynomially bounded sample count $m = m(\lambda)$, a party index $\sigma \in \{0, 1\}$, and a bit $b \in \{0, 1\}$, define the experiment

$$\text{Exp}_{\text{PCF}, \mathcal{A}, m, \sigma, b}^{\text{sec}}(1^\lambda, \rho_0, \rho_1)$$

as follows.

$\text{Exp}_{\text{PCF}, \mathcal{A}, m, \sigma, 0}^{\text{sec}}(1^\lambda, \rho_\sigma)$	$\text{Exp}_{\text{PCF}, \mathcal{A}, m, \sigma, 1}^{\text{sec}}(1^\lambda, \rho_\sigma)$
$\rho_{1-\sigma} \xleftarrow{\$} \mathcal{R}_{1-\sigma, \lambda}$ $(k_0, k_1) \leftarrow \text{PCF.Gen}(1^\lambda, \rho_0, \rho_1)$ $T \leftarrow \emptyset$ for $i = 1, \dots, m$ do $x_i \xleftarrow{\$} \mathcal{X}_\lambda$ if $T[x_i]$ is undefined then $T[x_i] \leftarrow \text{PCF.Eval}(1 - \sigma, k_{1-\sigma}, x_i)$ $y_{1-\sigma}^{(i)} \leftarrow T[x_i]$ $b' \leftarrow \mathcal{A}(1^\lambda, \sigma, k_\sigma, (x_i, y_{1-\sigma}^{(i)})_{i \in [m]})$ output b'	$\rho_{1-\sigma} \xleftarrow{\$} \mathcal{R}_{1-\sigma, \lambda}$ $\tilde{\rho}_{1-\sigma} \xleftarrow{\$} \mathcal{R}_{1-\sigma, \lambda}$ $\rho_\sigma^* \leftarrow \rho_\sigma, \rho_{1-\sigma}^* \leftarrow \tilde{\rho}_{1-\sigma}$ $(k_0, k_1) \leftarrow \text{PCF.Gen}(1^\lambda, \rho_\sigma^*, \rho_{1-\sigma}^*)$ $T \leftarrow \emptyset$ for $i = 1, \dots, m$ do $x_i \xleftarrow{\$} \mathcal{X}_\lambda$ if $T[x_i]$ is undefined then $y_\sigma \leftarrow \text{PCF.Eval}(\sigma, k_\sigma, x_i)$ $y_{1-\sigma} \leftarrow \mathcal{Y}.\text{RSample}(1^\lambda, \sigma, \rho_\sigma, \rho_{1-\sigma}, y_\sigma)$ $T[x_i] \leftarrow y_{1-\sigma}$ $y_{1-\sigma}^{(i)} \leftarrow T[x_i]$ $b' \leftarrow \mathcal{A}(1^\lambda, \sigma, k_\sigma, (x_i, y_{1-\sigma}^{(i)})_{i \in [m]})$ output b'

We require that for every PPT distinguisher $\mathcal{A}$, every polynomially bounded $m = m(\lambda)$, every $\sigma \in \{0, 1\}$, and every ρ_σ in the support of $\mathcal{R}_{\sigma, \lambda}$,

$$\left| \Pr[\text{Exp}_{\text{PCF}, \mathcal{A}, m, \sigma, 0}^{\text{sec}}(1^\lambda, \rho_0, \rho_1) = 1] - \Pr[\text{Exp}_{\text{PCF}, \mathcal{A}, m, \sigma, 1}^{\text{sec}}(1^\lambda, \rho_0, \rho_1) = 1] \right| \leq \text{negl}(\lambda).$$

Definition 11 (Fixed-offset correlated OT over $\mathbb{Z}_N$). Let $\mathcal{D}_\Delta = \{\mathcal{D}_{\Delta, N}\}$ be an offset distribution over $\mathbb{Z}_N$. The setup string of party 0 is the dummy value $\rho_0 = \perp$. The setup string of party 1 is

$$\rho_1 = (N, p, q, \Delta),$$

where $(N, p, q) \xleftarrow{\$} \text{ModGen}(1^\lambda)$ and $\Delta \leftarrow \mathcal{D}_{\Delta, N}$.

Given $\rho_1 = (N, p, q, \Delta)$, the sampler $\mathcal{Y}_{\text{COT}}$ samples

$$b \xleftarrow{\$} \{0, 1\}, \quad m_0 \xleftarrow{\$} \mathbb{Z}_N,$$

sets

$$m_1 := m_0 + \Delta \pmod{N}, \quad \mu := m_b,$$

and outputs

$$y_0 = (b, \mu) \in \{0, 1\} \times \mathbb{Z}_N, \quad y_1 = (m_0, m_1) \in \mathbb{Z}_N^2.$$

Lemma 2 (Reverse sampling for fixed-offset correlated OT). The correlation in Definition 11 is reverse-sampleable.

Proof. Fix $\rho_1 = (N, p, q, \Delta)$. If $\sigma = 0$ and $y_0 = (b, \mu)$, define

$$\mathcal{Y}_{\text{COT}}.\text{RSample}(1^\lambda, 0, \perp, \rho_1, (b, \mu)) := (\mu - b\Delta, \mu + (1 - b)\Delta) \pmod{N}.$$

This is the unique pair (m_0, m_1) satisfying $m_1 = m_0 + \Delta$ and $\mu = m_b$. If $\sigma = 1$ and $y_1 = (m_0, m_1)$, define

$$\mathcal{Y}_{\text{COT}}.\text{RSample}(1^\lambda, 1, \perp, \rho_1, (m_0, m_1)) := (b, m_b) \quad \text{for } b \xleftarrow{\$} \{0, 1\}.$$

In the first case reverse sampling exactly reconstructs the sender output compatible with the fixed receiver output. In the second case it samples the receiver choice bit with the same distribution as the real sampler and sets μ accordingly. Therefore the joint distribution is identical to the real COT sampler.

Definition 12 ($\mathbb{Z}_M$ -VOLE with fixed multiplier and input distribution). Let $M = M(\lambda)$ be a modulus, let $\mathcal{D}_\Delta$ be an efficiently samplable distribution over $\mathbb{Z}_M$, and let $\mathcal{D}_U$ be an efficiently samplable distribution over $\mathbb{Z}_M$. The setup string of party 0 is the dummy value $\rho_0 = \perp$. The setup string of party 1 is

$$\rho_1 = (M, \Delta), \quad \Delta \leftarrow \mathcal{D}_\Delta.$$

Given $\rho_1 = (M, \Delta)$, the sampler $\mathcal{Y}_{\text{VOLE}, M}$ samples

$$u \leftarrow \mathcal{D}_U, \quad v \xleftarrow{\$} \mathbb{Z}_M,$$

sets

$$w := v + \Delta u \pmod{M},$$

and outputs

$$y_0 = (u, w) \in \mathbb{Z}_M^2, \quad y_1 = (\Delta, v) \in \mathbb{Z}_M^2.$$

When $\mathcal{D}_U$ is uniform over $\mathbb{Z}_M$, we call this the ideal fixed-multiplier $\mathbb{Z}_M$ -VOLE correlation. In the prime-field eVRF application, $M = q$ is prime and both $\mathcal{D}_\Delta$ and $\mathcal{D}_U$ are the uniform distribution over $\mathbb{Z}_q$.

Lemma 3 (Reverse sampling for VOLE). The fixed-multiplier VOLE correlation of Definition 12 is reverse-samplable for every efficiently samplable input distribution $\mathcal{D}_U$.

Proof. Fix $\rho_1 = (M, \Delta)$. Given party-0 output (u, w) , reverse sampling for party 1 returns

$$(\Delta, w - \Delta u) \pmod{M}.$$

Given party-1 output (Δ, v) , reverse sampling for party 0 samples $u \leftarrow \mathcal{D}_U$ and returns

$$(u, v + \Delta u) \pmod{M}.$$

In the first case, the value $v = w - \Delta u$ is the unique verifier-side value compatible with (u, w) and Δ . In the second case, the sampler draws u from exactly the same distribution as the real VOLE sampler and then sets w by the defining VOLE equation. Hence both procedures reproduce the corresponding conditional distributions exactly.

B.3 Fiat–Shamir forking

We use the general forking lemma of Bellare and Neven [BN06, Lemma 1], which abstracts the probabilistic rewinding argument away from any particular signature scheme. This general form is itself a refinement of the original forking lemma of Pointcheval and Stern [PS00, Theorem 1].

Lemma 4 (General Forking Lemma [BN06, Lemma 1]). Let $q \geq 1$, and let $\mathcal{H}$ be a finite set of size $h \geq 2$. Let IG be a randomized input generator, and let $\mathcal{A}$ be a randomized algorithm which, on input

$$x, h_1, \dots, h_q \quad \text{with } x \leftarrow \text{IG and } h_1, \dots, h_q \in \mathcal{H},$$

outputs a pair (J, σ) , where $J \in \{0, 1, \dots, q\}$. Define the accepting probability

$$\text{acc} := \Pr [J \geq 1 : x \leftarrow \text{IG}, h_1, \dots, h_q \xleftarrow{\$} \mathcal{H}, (J, \sigma) \leftarrow \mathcal{A}(x, h_1, \dots, h_q)].$$

The associated forking algorithm $\text{FA}^{\mathcal{A}}$ is defined as follows. On input x , it samples random coins ρ for $\mathcal{A}$, samples $h_1, \dots, h_q \xleftarrow{\$} \mathcal{H}$, and runs

$$(I, \sigma) \leftarrow \mathcal{A}(x, h_1, \dots, h_q; \rho).$$

If $I = 0$, it aborts. Otherwise it samples

$$h'_I, h'_{I+1}, \dots, h'_q \xleftarrow{\$} \mathcal{H}$$

and reruns $\mathcal{A}$ with the same coins ρ on the related input

$$x, h_1, \dots, h_{I-1}, h'_I, h'_{I+1}, \dots, h'_q.$$

Let

$$(I', \sigma') \leftarrow \mathcal{A}(x, h_1, \dots, h_{I-1}, h'_I, h'_{I+1}, \dots, h'_q; \rho).$$

The forking algorithm succeeds if $I' = I$ and $h'_I \neq h_I$.

Let

$$\text{frk} := \Pr [\text{FA}^A(x) \text{ succeeds} : x \leftarrow \text{IG}].$$

Then

$$\text{frk} \geq \text{acc} \left(\frac{\text{acc}}{q} - \frac{1}{h} \right).$$

We shall use Lemma 4 in the following Fiat–Shamir form.

Corollary 4 (Forking for Fiat–Shamir transcript predicates). *Let C be a finite challenge space, let $H : \{0, 1\}^* \rightarrow C$ be a random oracle, and let*

$$\text{Acc} \subseteq \{0, 1\}^* \times C \times \{0, 1\}^*$$

be an efficiently decidable transcript predicate. We interpret $u \in \{0, 1\}^*$ as a Fiat–Shamir transcript prefix, $c \in C$ as the challenge, and $\rho \in \{0, 1\}^*$ as the response. A transcript (u, ρ) is accepting with respect to H if

$$\text{Acc}(u, H(u), \rho) = 1.$$

Let $\mathcal{A}^H$ be a randomized oracle algorithm making at most Q_H queries to H . Suppose that $\mathcal{A}^H$ outputs an accepting transcript (u, ρ) and has queried u before outputting with probability ε_q . Then there is a rewinding algorithm Fork^A , running $\mathcal{A}$ twice with the same random tape and with a reprogrammed random-oracle answer at the first query to u , which outputs two accepting transcripts

$$(u, c, \rho), \quad (u, c', \rho')$$

with the same prefix u and distinct challenges $c \neq c'$ with probability at least

$$\Pr[\text{Fork}^A \text{ succeeds}] \geq \varepsilon_q \left(\frac{\varepsilon_q}{Q_H} - \frac{1}{|C|} \right).$$

Equivalently, if ε is the overall accepting probability and ν is the probability that $\mathcal{A}$ outputs an accepting transcript without first querying the corresponding prefix u , then

$$\Pr[\text{Fork}^A \text{ succeeds}] \geq (\varepsilon - \nu) \left(\frac{\varepsilon - \nu}{Q_H} - \frac{1}{|C|} \right).$$

Proof. Construct an algorithm $\mathcal{B}$ to which Lemma 4 will be applied. The input x to $\mathcal{B}$ contains all public parameters and the random tape of the outer experiment, except for the random-oracle answers. The values

$$h_1, \dots, h_{Q_H} \in C$$

are used as the random-oracle answers to the first, second, ..., Q_H -th distinct oracle queries made by $\mathcal{A}$, with dummy answers added if fewer than Q_H queries are made.

The algorithm $\mathcal{B}$ runs $\mathcal{A}$ with this simulated random oracle. If $\mathcal{A}$ outputs an accepting transcript (u, ρ) and the first query equal to u was made at position j , then $\mathcal{B}$ outputs

$$(J, \sigma) := (j, (u, \rho)).$$

Otherwise it outputs $J = 0$.

By construction, the accepting probability of $\mathcal{B}$ in Lemma 4 is exactly ε_q . Applying Lemma 4 with

$$q = Q_H, \quad \mathcal{H} = C, \quad h = |C|,$$

gives a fork with probability at least

$$\varepsilon_q \left(\frac{\varepsilon_q}{Q_H} - \frac{1}{|C|} \right).$$

When the fork succeeds, the two executions have the same transcript prefix u , because the executions agree up to the first query equal to u , and they have distinct challenges because the rewind execution receives an independently resampled answer at that query. Both transcripts are accepting by the definition of the accepting index J . This proves the first bound. The second bound follows from $\varepsilon_q = \varepsilon - \nu$.

For the publicly verifiable eVRF, we instantiate the transcript prefix and challenge space as

$$u = (pk, x, Y, T, C_\theta), \quad C = \mathbb{F}^*.$$

Thus, except with probability bounded by the forking loss and by the probability of an accepting proof whose prefix was never queried to the random oracle, any adversary producing an accepting proof can be rewound to produce two accepting proofs

$$(T, C_\theta, \delta, \vec{\theta}_1, \rho_1), \quad (T, C_\theta, \delta', \vec{\theta}'_1, \rho'_1)$$

for the same (pk, x, Y) and the same first message (T, C_θ) , with $\delta \neq \delta'$.

C Distributed Setup

This appendix expands on the setup discussion in Section 4.2. It is included to make explicit where the known public-key constrained Naor–Reingold techniques cease to be black-box applicable to the Paillier exponent domain used in this paper, and formalizes the workaround that we instead adopt.

Definition 13 (PCF key-generation functionality). Let $\mathcal{F}_{\text{KGen}}^{\text{COT}}$ be the following two-party randomized functionality, parameterized by n , the support set $S \subseteq \{0, \dots, n\}$, the weak-PRF key space, and the offset distribution $\mathcal{D}_{\Delta, N}$. Note that this is essentially PCF.Gen from Figure 1.

$$(N, p, q) \leftarrow \text{ModGen}(1^\lambda), \quad g \xleftarrow{\$} \text{NR}_N, \quad k \xleftarrow{\$} \mathcal{K}, \quad \Delta \leftarrow \mathcal{D}_{\Delta, N}, \\ a_0, a_1, \dots, a_n \xleftarrow{\$} \mathcal{E}_N, \quad r \xleftarrow{\$} \mathcal{E}_N^*.$$

The functionality also samples $z \xleftarrow{\$} \{0, 1\}^n$. It then computes, for every $i \in [n]$,

$$\alpha_i := a_i r^{-z_i} \pmod{N\phi(N)},$$

and, for every $s \in \{0, \dots, n\}$,

$$h_s := g^{a_0 r^s} (1 + N)^{\chi_S(s)\Delta} \pmod{N^2}.$$

It sends

$$k_0 = (N, g, k, z, (\alpha_i)_{i \in [n]}, (h_s)_{s=0}^n)$$

to party 0 and

$$k_1 = (N, p, q, g, k, \Delta, a_0, (a_i)_{i \in [n]})$$

to party 1.

Theorem 2 (Realizing PCF key generation from OT). In the static semi-honest ideal-OT hybrid model, the protocol of Figure 3 perfectly realizes $\mathcal{F}_{\text{KGen}}^{\text{COT}}$ from Definition 13. Consequently, if the ideal OT invocations are replaced by a parallel OT protocol Π_{OT} that is $\epsilon_{\text{OT}}(\lambda)$ -secure for messages in $\mathcal{E}_N$, then the resulting setup protocol realizes $\mathcal{F}_{\text{KGen}}^{\text{COT}}$ with distinguishing advantage at most $\epsilon_{\text{OT}}(\lambda)$, up to the standard parallel-composition loss of the OT protocol.

Proof. We give simulators for the two possible static corruptions. The case of no corruption is immediate, and the case in which both parties are corrupted is vacuous because the joint view can be generated by running the real protocol internally.

Corrupt party P_0 . In the ideal execution, $\mathcal{F}_{\text{KGen}}^{\text{COT}}$ sends

$$k_0 = (N, g, k, z, (\alpha_i)_i, (h_s)_s)$$

to the corrupted party. The simulator gives the adversary the local random tape z , the ideal-OT receiver inputs $(z_i)_i$, the ideal-OT receiver outputs $(\alpha_i)_i$, the public message $\tau = (N, g, k, (h_s)_s)$, and the final output k_0 . This is exactly the real ideal-OT-hybrid view of a semi-honest receiver: ideal OT reveals only the selected messages, and every other value received by P_0 is contained in k_0 . Hence the simulated and real views are identical.

Corrupt party P_1 . Let the corrupted party's honest random tape be

$$\eta_1 = (N, p, q, g, k, \Delta, a_0, a_1, \dots, a_n, r).$$

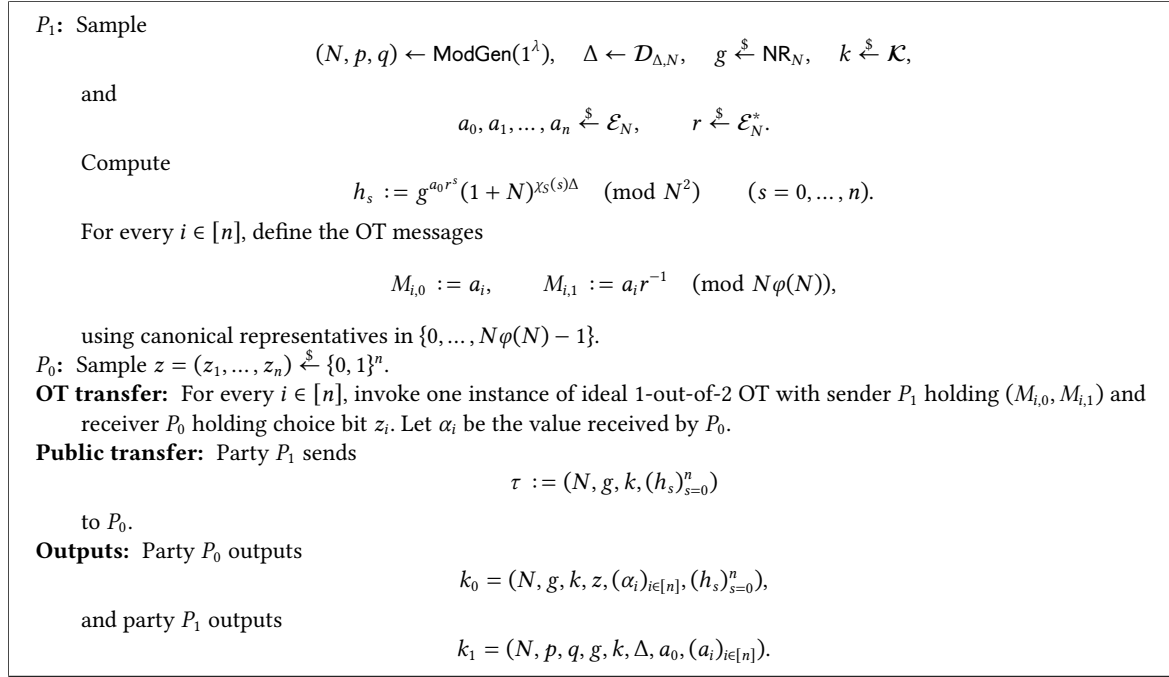


Fig. 3. OT-based realization of the PCF key-generation functionality.

The simulator sends this value as party 1's input to $\mathcal{F}_{\text{KGen}}^{\text{COT}}$. The functionality samples the honest receiver's z and computes $\alpha_i = a_i r^{-z_i}$ and the table $(h_s)_s$ using the same η_1 . The simulator gives the adversary the random tape η_1 , the public transfer

$$\tau = (N, g, k, (h_s)_{s=0}^n),$$

the ideal-OT sender inputs

$$(M_{i,0}, M_{i,1}) = (a_i, a_i r^{-1}) \pmod{N\varphi(N)} \quad (i \in [n]),$$

and the final output $k_1 = (N, p, q, g, k, \Delta, a_0, (a_i)_i)$. In the ideal-OT hybrid, the sender receives no information about the receiver's choices or outputs. Therefore the corrupted sender's real view is a deterministic function of exactly η_1 , and the simulated view is identical. Moreover, because the ideal functionality used the same η_1 , the honest receiver's ideal output is jointly consistent with the simulated sender view.

The two simulations are perfect. This proves perfect realization in the ideal-OT hybrid. The final statement follows by the sequential/parallel composition theorem for semi-honest OT: replacing the ideal OT calls by Π_{OT} changes the environment's view by at most the composed OT distinguishing advantage.

Corollary 5 (Setup-realized PCF). *Run the protocol of Figure 3 and then use the local evaluation algorithms of Figure 1. In the ideal-OT hybrid model this realizes the weak PCF functionality of Theorem 3 with no additional loss. With a real parallel OT protocol Π_{OT} , the distinguishing advantage is bounded by the advantage in Theorem 3 plus the composed security loss of Π_{OT} .*

Proof. By Theorem 2, the setup phase realizes exactly the key-generation functionality whose outputs are the keys of Figure 1. The post-setup algorithms are local and deterministic. Applying the standard composition theorem and then Theorem 3 gives the claim.

Public-key PCF discussion. In a public-key PCF, there is a public setup algorithm, local key generation algorithms

$$(\text{sk}_\sigma, \text{pk}_\sigma) \leftarrow \text{Gen}_\sigma(1^\lambda), \quad \sigma \in \{0, 1\},$$

and deterministic derivation algorithms

$$k_0 := \text{KeyDer}_0(\text{sk}_0, \text{pk}_1), \quad k_1 := \text{KeyDer}_1(\text{sk}_1, \text{pk}_0).$$

The public keys may be posted in one simultaneous round, or stored in a PKI, but there is no subsequent message whose contents depend on the other party's public key. A protocol that runs ordinary OT after the public keys are known is therefore an interactive key-generation protocol for a PCF, not a public-key PCF. This distinction is essential in network settings: public-key PCFs allow m users to post m public keys and then locally derive all $\binom{m}{2}$ pairwise PCF keys, whereas an OT-based setup still requires a pairwise interactive execution for every pair.

A one-round public-key realization of our Paillier COT PCF would need to realize the following distribution. Party 0 samples $z \xleftarrow{\$} \{0, 1\}^n$. Party 1 samples $(N, p, q) \leftarrow \text{ModGen}(1^\lambda)$, $r \xleftarrow{\$} \mathcal{E}_N^*$, and $a_1, \dots, a_n \xleftarrow{\$} \mathcal{E}_N$. The derived keys must contain values α_i for party 0 satisfying

$$\alpha_i = a_i r^{-z_i} \pmod{N\varphi(N)}.$$

The values α_i must be represented as canonical integer residues, because party 0 later computes products of them and uses these products as exponents in $\mathbb{Z}_{N^2}^*$. In particular, an encryption of α_i , a commitment to α_i , or a group element G^{α_i} in another group is not a valid substitute unless the party can efficiently recover the residue itself.

The Paillier–ElGamal public-key lift of the known-order constrained Naor–Reingold construction produces additive correlations and then maps them into multiplicative shares by exponentiating with a public generator of a known-order cyclic group. The correctness of that step uses the public homomorphism

$$t \mapsto G^t$$

from an explicitly known exponent ring to the group generated by G . Our construction needs the exponent ring $\mathbb{Z}_{N\varphi(N)}$, and only party 1 knows $\varphi(N)$. The public subgroup $\langle 1 + N \rangle \subseteq \mathbb{Z}_{N^2}^*$ has order N , not $N\varphi(N)$, so it only preserves the Paillier message component; it loses the N -th-residue exponent component on which the Naor–Reingold cancellation depends. Conversely, using a separate known-order group gives group encodings of the desired residues, but the receiver cannot multiply these encodings and then use the product as the integer exponent e_x required by Figure 1.

One might try to avoid the hidden modulus by sampling small integer exponents and maintaining the relation $a_i = \alpha_i r^{z_i}$ over the integers. This changes the construction. The current proof uses that the exponent residues are close to uniform in $\mathcal{E}_N$ and that multiplication by $r \in \mathcal{E}_N^*$ is a permutation of that set. Small integer exponents do not have this distribution, and powers r^s appearing in the table have bitlength growing with $s \log r$. Such a variant may be a useful heuristic optimization or a separate construction, but it would require a new table-hiding assumption and a new correctness-and-statistical analysis. Hence, our Paillier construction gives a setup-programmed weak PCF and admits an efficient pairwise OT-based key-generation protocol. A true one-round public-key version would require a new non-interactive hidden-order exponent-transfer primitive, or a redesign in which the constrained Naor–Reingold cancellation takes place in a publicly known exponent group while the Paillier offset remains extractable by DDLog.

D Subgroup Indistinguishability $\Rightarrow$ Extended Power-DDH

D.1 Proof of Theorem 1

Proof. We use the following hybrid proof, going from real to random. In particular, we define the following pair of hybrids for $0 \leq i \leq T$, where T is a parameter to be chosen later. We will make $\text{Hyb}_{0,1}$ be the real distribution, and end by analyzing the statistical distance between $\text{Hyb}_{T,1}$ and the random distribution.

$\text{Hyb}_{i,1}$: We only write the oracle as the adversary's view in each hybrid, as the rest of its view remains unchanged throughout the hybrid proof. Clearly, when $i = 0$ this is exactly the real distribution.

$$\left\{ f \mapsto g^{f(x_1, \dots, x_n)} \cdot \prod_{j=1}^i z_j^{f(y_{j1}, \dots, y_{jn})} \mid \begin{array}{l} (G, G_M, G_L, M) \xleftarrow{\$} \text{GGen}(1^\lambda) \\ g \xleftarrow{\$} G_L \\ (x_1, \dots, x_n) \xleftarrow{\$} D(G) \\ (y_{11}, \dots, y_{1n}) \xleftarrow{\$} \mathbb{Z}_M^n \\ \vdots \\ (y_{i1}, \dots, y_{in}) \xleftarrow{\$} \mathbb{Z}_M^n \\ (z_1, \dots, z_i) \xleftarrow{\$} G_M^i \end{array} \right\}$$

Hyb_{i,2}: This change is a direct reduction to subgroup indistinguishability (Definition 5):

$$\left\{ f \mapsto g^{f(x_1, \dots, x_n)} \cdot \prod_{j=1}^i z_j^{f(y_{j1}, \dots, y_{jn})} \right\} \left\{ \begin{array}{l} (G, G_M, G_L, M) \xleftarrow{\$} G_{\text{Gen}}(1^\lambda) \\ g \xleftarrow{\$} G \\ (x_1, \dots, x_n) \xleftarrow{\$} D(G) \\ (y_{11}, \dots, y_{1n}) \xleftarrow{\$} \mathbb{Z}_M^n \\ \vdots \\ (y_{i1}, \dots, y_{in}) \xleftarrow{\$} \mathbb{Z}_M^n \\ (z_1, \dots, z_i) \xleftarrow{\$} G_M^i \end{array} \right\}$$

Hyb_{i,3}: This change follows from G decomposing as $G \equiv G_M \times G_L$ (Definition 5):

$$\left\{ f \mapsto (g \cdot z_{i+1})^{f(x_1, \dots, x_n)} \cdot \prod_{j=1}^i z_j^{f(y_{j1}, \dots, y_{jn})} \right\} \left\{ \begin{array}{l} (G, G_M, G_L, M) \xleftarrow{\$} G_{\text{Gen}}(1^\lambda) \\ g \xleftarrow{\$} G_L \\ (x_1, \dots, x_n) \xleftarrow{\$} D(G) \\ (y_{11}, \dots, y_{1n}) \xleftarrow{\$} \mathbb{Z}_M^n \\ \vdots \\ (y_{i1}, \dots, y_{in}) \xleftarrow{\$} \mathbb{Z}_M^n \\ (z_1, \dots, z_i, z_{i+1}) \xleftarrow{\$} G_M^{i+1} \end{array} \right\}$$

Hyb_{i,4}: This change follows from the condition on $D(G)$:

$$\left\{ f \mapsto g^{f(x_1, \dots, x_n)} \cdot \prod_{j=1}^{i+1} z_j^{f(y_{j1}, \dots, y_{jn})} \right\} \left\{ \begin{array}{l} (G, G_M, G_L, M) \xleftarrow{\$} G_{\text{Gen}}(1^\lambda) \\ g \xleftarrow{\$} G_L \\ (x_1, \dots, x_n) \xleftarrow{\$} D(G) \\ (y_{11}, \dots, y_{1n}) \xleftarrow{\$} \mathbb{Z}_M^n \\ \vdots \\ (y_{i1}, \dots, y_{in}) \xleftarrow{\$} \mathbb{Z}_M^n \\ (y_{(i+1)1}, \dots, y_{(i+1)n}) \xleftarrow{\$} \mathbb{Z}_M^n \\ (z_1, \dots, z_i, z_{i+1}) \xleftarrow{\$} G_M^{i+1} \end{array} \right\}$$

That is, g has order dividing L and z_{i+1} has order dividing M , so

$$(g \cdot z_{i+1})^{f(x_1, \dots, x_n)} = g^{f(x_1 \bmod L, \dots, x_n \bmod L)} \cdot z_{i+1}^{f(x_1 \bmod M, \dots, x_n \bmod M)},$$

and by the condition on $D(G)$, $(x_1 \bmod M, \dots, x_n \bmod M)$ can be replaced with uniform.

By inspection, this hybrid is equivalent to Hyb_{i+1,1}.

Finally, we must show that Hyb_{T,1} is statistically close to the random distribution, for suitable choice of T . We follow an identical-until-bad approach, where we define a sequence of bad events $\{\text{Bad}_i\}_{i \in [Q]}$ and show that Hyb_{T,1} behaves statistically identically to the random distribution for the first i queries if none of the first i bad events occur. Then the standard identical-until-bad argument implies that the statistical distance is at most the probability that some Bad_i fails in the random distribution, and we present a bound on this probability.

Let $f_1, \dots, f_Q$ be the polynomials queried by the adversary, and consider the matrices representing the first i queries:

$$F_i = \begin{bmatrix} \vec{f}_1^T \\ \vdots \\ \vec{f}_Q^T \end{bmatrix} \quad F_{i,y} = \begin{bmatrix} f_1(y_{11}, \dots, y_{1n}) & \cdots & f_1(y_{T1}, \dots, y_{Tn}) \\ \vdots & & \vdots \\ f_i(y_{11}, \dots, y_{1n}) & \cdots & f_i(y_{T1}, \dots, y_{Tn}) \end{bmatrix},$$

where $\vec{f}_i$ is (possibly exponentially long) coefficient vector for polynomial f_i . We let Bad_i be the event that the column span of F_i is not the same as the column span of $F_{i,y}$. The motivation for this bad event is as follows. Let g_M be a generator of G_M , let $g_M^{z'_j} = z_j$, and let $\vec{z}'$ be the vector of all z'_j . Similarly, let $\vec{r}'$ be vector of discrete

logs of coefficients of $r : \mathbb{Z}[x_1, \dots, x_n] \rightarrow \mathbb{G}_M$, so that $r(f) = g_M^{\vec{f}^T \vec{r}'}$. Then in $\text{Hyb}_{T,1}$ the $\mathbb{G}_M$ part of the i oracle responses are sampled as $g_M^{F_{i,y} \vec{z}'}$, and in the random world they are sampled as $g_M^{F_i \vec{r}'}$.⁷ Clearly, the column span of $F_{i,y}$ must be the same as F_i if these distributions are to be indistinguishable.

Now we give the identical-until-bad argument. Consider the adversary making its i th oracle query, and let $\mathbb{G}_M$ part of the first $i - 1$ oracle responses be $g_M^{\vec{u}_{i-1}}$. We show that the conditional probability distributions for $\text{Hyb}_{T,1}$ and the random distribution are identical, conditioning on $\vec{u}_{i-1}$ and on Bad_i not occurring. In $\text{Hyb}_{T,1}$, we have $\vec{u}_{i-1} = F_{i-1,y} \vec{z}'$, while in the random distribution we have $\vec{u}_{i-1} = F_{i-1} \vec{r}'$. In their respective conditional distributions, $\vec{z}'$ and $\vec{r}'$ are sampled uniformly subject to the respective constraint. Equivalently, there exist particular solution $\vec{z}'_0$ and $\vec{r}'_0$ to these constraints, and the distributions can be written as

$$\vec{z}' \leftarrow \vec{z}'_0 + \ker(F_{i-1,y}) \quad \vec{r}' \leftarrow \vec{r}'_0 + \ker(F_{i-1}).$$

Now, consider the adversary's view up to and including the i th oracle response. The $\mathbb{G}_M$ part of their view is $\vec{u}_i = F_{i,y} \vec{z}'$ in $\text{Hyb}_{T,1}$, and $\vec{u}_i = F_i \vec{r}'$ in the random distribution. Let Z_i be the set of length i vectors which are 0 in their first $i - 1$ components. The two distributions are then

$$\begin{aligned} & F_{i,y}(\vec{z}'_0 + \ker(F_{i-1,y})) & F_i(\vec{r}'_0 + \ker(F_{i-1})) \\ &= F_{i,y} \vec{z}'_0 + F_{i,y} \ker(F_{i-1,y}) &= F_i \vec{r}'_0 + F_i \ker(F_{i-1}) \\ &= F_{i,y} \vec{z}'_0 + \text{Span}(F_{i,y}) \cap Z_i &= F_i \vec{r}'_0 + \text{Span}(F_i) \cap Z_i \end{aligned}$$

since $F_{i,y}$ is $F_{i-1,y}$ with a row added at the bottom (and similarly for F_i). We assumed that Bad_i does not occur, so $\text{Span}(F_{i,y}) = \text{Span}(F_i)$ and so the second terms are the same. Finally, we must argue that $F_{i,y} \vec{z}'_0 - F_i \vec{r}'_0$ is in $\text{Span}(F_i) \cap Z_i$. We have that $F_{i,y} \vec{z}'_0 - F_i \vec{r}'_0 \in \mathbb{Z}_i$, since both cases were conditioned to agree with $\vec{u}_{i-1}$. We also have $F_{i,y} \vec{z}'_0 \in \text{Span}(F_{i,y}) = \text{Span}(F_i)$ and $F_i \vec{r}'_0 \in \text{Span}(F_i)$, so $F_{i,y} \vec{z}'_0 - F_i \vec{r}'_0 \in \text{Span}(F_i)$. Therefore, when Bad_i does not occur, the conditional distribution of the i th oracle response is identical between $\text{Hyb}_{T,1}$ and the random distribution.

Now, we just need to bound the bad event probability in the random distribution. Note that any bad event Bad_i occurring implies Bad_Q , because each F_i is just the first i th rows of F_Q . Therefore, we need only bound the probability of Bad_Q .

Note that the columns of $F_{Q,y}$ are in the span of the columns of F_Q , since evaluating a polynomial at $(y_1, \dots, y_n)$ is a linear function of the polynomial's coefficients. Therefore, we have $\text{Span}(F_{Q,y}) \subseteq \text{Span}(F_Q)$, and we need only show that $\text{Span}(F_{Q,y}) \supseteq \text{Span}(F_Q)$. Equivalently, we must show that $F_{Q,y}$ has full rank when considered as a map into the column space of F_Q . By CRT, it suffices to show this modulo each prime factor p of M .⁸

If Bad_Q occurs, then there must be some prime factor p of M and some vector $\vec{v} \in \mathbb{Z}_p$ such that $\vec{v}^T F_{Q,y} = 0$ yet $\vec{v}^T F_Q \neq 0$. Consider the polynomial $f_{\vec{v}}$ defined by $F_Q^T \vec{v}$, i.e., $f_{\vec{v}} = \sum_i v_i \cdot f_i$. This polynomial is nonzero because $\vec{v}^T F_Q \neq 0$, yet its evaluations at all y_j are 0 because $\vec{v}^T F_{Q,y} = 0$. We now bound the probability this occurring for any fixed $\vec{v}$, then use a union bound to argue that no such $\vec{v}$ can exist. Let d be the maximum total degree across all f_i ; we have $d \leq \text{poly}(\lambda)$ because all f_i are efficiently evaluable (running f_i on appropriate k -bit integer inputs will, for sufficiently large k , lead to numbers at least $d \cdot k$ bits long). Consider sampling a random input $(y_1, \dots, y_n) \leftarrow \mathbb{Z}_p^n$, and substituting each y_i into f , one at a time. By assumption about $\mathcal{F}$ and by the Schwartz–Zippel lemma, we have

$$\Pr_{(y_1, \dots, y_n) \leftarrow \mathbb{Z}_p^n} [f_{\vec{v}}(y_1, \dots, y_n) = 0] \leq \min\left(\frac{p-1}{p}, \frac{d}{p}\right).$$

Therefore, the probability that $\vec{v}^T F_{Q,y} = 0$, i.e., that all T random evaluations result in 0, is at most

$$\min\left(\frac{p-1}{p}, \frac{d}{p}\right)^T$$

⁷ Ignoring that the queries f_i can be chosen adaptively based on previous oracle queries.

⁸ Prime vs prime-powers moduli do not affect the ranks of matrices, as for a submatrix to be considered nonsingular it must have a determinant that is a unit in the ring, i.e., nonzero modulo the prime.

Next, there are at most p^Q choices of $\vec{v}$, so a union bound shows that

$$\begin{aligned}
& \Pr_{\substack{(y_{11}, \dots, y_{1n}) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^n \\ \vdots \\ (y_{T1}, \dots, y_{Tn}) \stackrel{\$}{\leftarrow} \mathbb{Z}_p^n}} \left[\exists \vec{v} \in \mathbb{Z}_p^Q. \vec{v}^T F_Q \neq 0 \wedge \vec{v}^T F_{Q,y} = 0 \right] \\
& \leq p^Q \min\left(\frac{p-1}{p}, \frac{d}{p}\right)^T \\
& \leq d^Q \min\left(\frac{p-1}{p}, \frac{d}{p}\right)^{T-Q} \\
& \leq d^Q \min\left(\frac{p-1}{p}, \frac{d}{d+1}\right)^{T-Q} \\
& \leq d^Q \left(1 - \frac{1}{d+1}\right)^{T-Q} \\
& \leq d^Q e^{-\frac{(T-Q)}{d+1}}.
\end{aligned}$$

This probability is bounded by $e^{-\lambda}$ if we set $T = Q + (d+1)(Q \ln d + \lambda) = \text{poly}(\lambda)$.

E Proofs for the PCF for COT

E.1 PCF for COT

Before proving the pseudorandomness and security properties, we first show that the outputs form a consistent OT relation.

Lemma 5 (Correctness). *For every honestly generated key pair (k_0, k_1) and every $x \in \{0, 1\}^n$, let*

$$(b_x, \mu_x) := \text{PCF.Eval}(0, k_0, x), \quad (m_{0,x}, m_{1,x}) := \text{PCF.Eval}(1, k_1, x).$$

Then

$$m_{1,x} = m_{0,x} + \Delta \pmod{N}, \quad \mu_x = m_{b_x, x}.$$

Proof. Let

$$s_x := \sum_{i=1}^n z_i x_i, \quad b_x := \chi_S(s_x), \quad e_x := \prod_{i: x_i=1} \alpha_i.$$

By construction, for every $i \in [n]$,

$$a_i \equiv \alpha_i r^{z_i} \pmod{N\varphi(N)}.$$

Therefore

$$\begin{aligned}
A_x &= a_0 \prod_{i: x_i=1} a_i \\
&\equiv a_0 \prod_{i: x_i=1} \alpha_i r^{z_i} \\
&= a_0 e_x r^{\sum_{i: x_i=1} z_i} \\
&= a_0 e_x r^{s_x} \pmod{N\varphi(N)}.
\end{aligned}$$

Consequently, in the N -th-residue component,

$$(g^{a_0 r^{s_x}})^{e_x} = g^{A_x} \pmod{N^2}.$$

Moreover, each α_i is congruent to 1 modulo N , so $e_x \equiv 1 \pmod{N}$. Hence

$$((1+N)^{b_x \Delta})^{e_x} = (1+N)^{b_x \Delta} \pmod{N^2}.$$

Combining the two equalities gives

$$h_{s_x}^{e_x} = g^{A_x} (1+N)^{b_x \Delta} \pmod{N^2}.$$

By Lemma 1,

$$\text{DDLog}_N(g^{A_x}(1+N)^\Delta) = \text{DDLog}_N(g^{A_x}) + \Delta \pmod{N},$$

and therefore

$$m_{1,x} = m_{0,x} + \Delta \pmod{N}.$$

Applying the same shift identity with exponent $b_x\Delta$ gives

$$\begin{aligned} \mu_x &= \text{DDLog}_N(g^{A_x}(1+N)^{b_x\Delta}) + F_k(x) \\ &= \text{DDLog}_N(g^{A_x}) + b_x\Delta + F_k(x) \\ &= m_{b_x,x} \pmod{N}. \end{aligned}$$

Lemma 6 (Pseudorandom correlated outputs). *Assume that F is a weak PRF with range $\mathbb{Z}_N$ and that F^{IPM} is an IPM-wPRF for S . Then the pseudorandom-output experiments of Definition 10 for PCF_{COT} are computationally indistinguishable.*

Proof. Fix a polynomially bounded sample count $m = m(\lambda)$ and random inputs $x_1, \dots, x_m \xleftarrow{\$} \{0, 1\}^n$. For each $i \in [m]$, define

$$b_i := F_z^{\text{IPM}}(x_i), \quad A_i := a_0 \prod_{j: x_{i,j}=1} a_j \pmod{N\varphi(N)}, \quad T_i := \text{DDLog}_N(g^{A_i}).$$

We use the following hybrids.

Hybrid H_0 . This is the real PCF-output experiment. It samples keys honestly and returns, for each i ,

$$(x_i, (b_i, \mu_i), (m_{0,i}, m_{1,i})) := (x_i, \text{PCF.Eval}(0, k_0, x_i), \text{PCF.Eval}(1, k_1, x_i)).$$

Hybrid H_1 . Rewrite the outputs using Lemma 5. This is an identical distribution, and the returned tuple is

$$(x_i, (b_i, T_i + b_i\Delta + F_k(x_i)), (T_i + F_k(x_i), T_i + \Delta + F_k(x_i)))_{i \in [m]}.$$

Hybrid H_2 . Replace the values $F_k(x_i)$ by independent uniform masks $R_i \xleftarrow{\$} \mathbb{Z}_N$, using a table for repeated inputs to answer all repeated queries consistently. Since the inputs are random, this transition is justified by the weak-PRF security of F . The hybrid returns

$$(x_i, (b_i, T_i + b_i\Delta + R_i), (T_i + R_i, T_i + \Delta + R_i))_{i \in [m]}.$$

Let $U_i := T_i + R_i \pmod{N}$. Conditioned on all other variables, each U_i is uniform in $\mathbb{Z}_N$ and independent of b_i and Δ . Thus H_2 is statistically identical to

$$(x_i, (b_i, U_i + b_i\Delta), (U_i, U_i + \Delta))_{i \in [m]}.$$

Hybrid H_3 . Replace $b_i = F_z^{\text{IPM}}(x_i)$ by independent uniform bits $c_i \xleftarrow{\$} \{0, 1\}$, again using a table for repeated inputs answer all repeated queries consistently. The only computational difference between H_2 and H_3 is the replacement of the IPM-wPRF outputs on random inputs by random bits, and hence $H_2 \stackrel{\text{comp}}{\equiv} H_3$ by Definition 1. The resulting distribution is

$$(x_i, (c_i, U_i + c_i\Delta), (U_i, U_i + \Delta))_{i \in [m]}.$$

This is exactly the ideal fixed-offset COT distribution from Definition 11. Therefore the real PCF-output experiment and the ideal correlation experiment are computationally indistinguishable.

Towards proving the PCF security property, we show that the receiver's key computationally hides the sender's offset Δ .

Lemma 7 (Receiver-key offset hiding). *Fix an efficiently samplable offset distribution $\mathcal{D}_\Delta$ over $\mathbb{Z}_N$. Assume Sparse-Power DCR for*

$$T = \{0, \dots, n\}, \quad B = S,$$

with the same exponent distribution for r as in key generation and with offset distribution $\mathcal{D}_\Delta$. Then a receiver key generated with $\Delta \leftarrow \mathcal{D}_\Delta$ is computationally indistinguishable from a receiver key where each h_s is replaced with the unshifted value

$$\tilde{h}_s := g^{a_0 r^s} \quad (s = 0, \dots, n).$$

Consequently, if Sparse-Power DCR holds for two offset distributions $\mathcal{D}_{\Delta,0}$ and $\mathcal{D}_{\Delta,1}$, then the corresponding receiver-key distributions are computationally indistinguishable. The same statement applies to any two fixed offsets $\Delta_0, \Delta_1 \in \mathbb{Z}_N$ provided the assumption is instantiated with the two degenerate offset distributions.

Proof. It is enough to prove that, for any admissible offset distribution, a receiver key generated with a shifted table is indistinguishable from a receiver key generated with the unshifted table

$$\tilde{h}_s := g^{a_0 r^s} \quad (s = 0, \dots, n),$$

because the statement for two offset distributions then follows by the triangle inequality.

We build a distinguisher for Sparse-Power DCR. Given a challenge

$$(N, g, (u_s)_{s=0}^n),$$

where either

$$u_s = g^{r^s} \quad \text{for all } s,$$

or

$$u_s = g^{r^s} (1 + N)^{\chi(s)\Delta} \quad \text{for all } s,$$

the reduction samples

$$k \xleftarrow{\$} \mathcal{K}, \quad z \xleftarrow{\$} \{0, 1\}^n, \quad \alpha_1, \dots, \alpha_n \xleftarrow{\$} \tilde{\mathcal{E}}_N, \quad a_0 \xleftarrow{\$} \tilde{\mathcal{E}}_N,$$

where $\tilde{\mathcal{E}}_N := \{1 + Nt : t \in \{0, \dots, N-1\}\}$. The reduction does not know $\varphi(N)$, so it cannot sample uniformly from $\mathcal{E}_N$ as key generation does, and draws from $\tilde{\mathcal{E}}_N$ instead. A uniform element of $\mathcal{E}_N$ is $1 + Nt$ for t uniform in $\mathbb{Z}_{\varphi(N)}$, and $1 + Nt < N\varphi(N)$ for every such t , so the two distributions differ only in the range from which t is drawn and are at statistical distance

$$\frac{N - \varphi(N)}{N} = \frac{p + q - 1}{N},$$

which is negligible for an RSA modulus $N = pq$. Every element of $\tilde{\mathcal{E}}_N$ is congruent to 1 modulo N , and this is the only property of these exponents used below.

The reduction sets

$$h_s := u_s^{a_0} \pmod{N^2} \quad (s = 0, \dots, n),$$

and outputs the simulated receiver key

$$k_0 = (N, g, k, z, (\alpha_i)_{i \in [n]}, (h_s)_{s=0}^n).$$

All components other than the table are independent of the Sparse-Power DCR challenge, and are within statistical distance $(n+1)(p+q-1)/N$ of their distribution in a real receiver key.

If the challenge is unshifted, then

$$h_s = (g^{r^s})^{a_0} = g^{a_0 r^s},$$

which is exactly the unshifted receiver-key table. If the challenge is shifted, then

$$\begin{aligned} h_s &= (g^{r^s} (1 + N)^{\chi(s)\Delta})^{a_0} \\ &= g^{a_0 r^s} (1 + N)^{\chi(s)a_0 \Delta}. \end{aligned}$$

Since $a_0 \equiv 1 \pmod{N}$, we have

$$(1 + N)^{\chi(s)a_0 \Delta} = (1 + N)^{\chi(s)\Delta} \pmod{N^2}.$$

Thus the shifted challenge gives exactly the real receiver-key table. Any distinguisher between the shifted and unshifted receiver-key distributions therefore distinguishes the two Sparse-Power DCR distributions, up to the exponent-sampling loss above: advantage ε against the receiver key gives advantage at least $\varepsilon - (n+1)(p+q-1)/N$ against Sparse-Power DCR.

Lemma 8 (Sender-key independence from the IPM key). *Conditioned on $(N, p, q, g, k, \Delta, a_0)$ and on the sampled invertible residue $r \in \mathcal{E}_N^*$, the sender-side residues $(a_i)_{i \in [n]}$ are distributed uniformly over $\mathcal{E}_N^n$ and are independent of z .*

Proof. For every $i \in [n]$, the correction exponent α_i is uniform over $\mathcal{E}_N$. Since $r \in \mathcal{E}_N^*$, multiplication by r^{z_i} is a permutation of $\mathcal{E}_N$, both when $z_i = 0$ and when $z_i = 1$. Therefore

$$a_i = \alpha_i r^{z_i} \pmod{N\varphi(N)}$$

is uniform over $\mathcal{E}_N$ and independent of z_i . The values $\alpha_1, \dots, \alpha_n$ are sampled independently, so the vector $(a_i)_{i \in [n]}$ is uniform over $\mathcal{E}_N^n$ and independent of z .

We are now ready to prove PCF security.

Lemma 9 (Insider security for corrupt sender). *Assume that F^{IPM} is an IPM-wPRF for S . Then the weak-insider-security experiments of Definition 10 for $\sigma = 1$ are computationally indistinguishable.*

Proof. The adversary receives the sender key

$$k_1 = (N, p, q, g, k, \Delta, a_0, (a_i)_{i \in [n]})$$

and receiver-side samples on random inputs. We define hybrids.

Hybrid H_0 . This is the real insider experiment for $\sigma = 1$.

Hybrid H_1 . Resample the sender-side residues $(a_i)_{i \in [n]}$ from the identical distribution described in Lemma 8, making explicit that this distribution is independent of z . All receiver-side samples are computed consistently with the resampled residues through the values

$$A_i := a_0 \prod_{j: x_{i,j}=1} a_j \pmod{N\varphi(N)}, \quad T_i := \text{DDLog}_N(g^{A_i}).$$

The transition from H_0 to H_1 is distribution-preserving by Lemma 8. In H_1 , conditioned on the adversary's key and on the inputs, the only remaining dependence on z is through

$$b_i = F_z^{\text{IPM}}(x_i).$$

The receiver-side samples have the form

$$(b_i, T_i + b_i \Delta + F_k(x_i)).$$

Hybrid H_2 . Replace the bits $b_i = F_z^{\text{IPM}}(x_i)$ by independent uniform bits $c_i \xleftarrow{\$} \{0, 1\}$, using a table for repeated inputs. Since the inputs x_i are uniform and the conditioning information in H_1 is independent of z , this transition is justified by the IPM-wPRF security of F^{IPM} . The samples are now

$$(c_i, T_i + c_i \Delta + F_k(x_i)).$$

Hybrid H_2 is exactly the ideal insider experiment for $\sigma = 1$: given the sender output $(m_{0,i}, m_{1,i}) = (T_i + F_k(x_i), T_i + \Delta + F_k(x_i))$, reverse sampling for COT chooses an independent bit c_i and returns $(c_i, m_{c_i,i})$. Hence $H_0 \stackrel{\text{comp}}{=} H_2$.

Lemma 10 (Insider security for corrupt receiver). *Assume receiver-key offset hiding as in Lemma 7. Then the weak-insider-security experiments of Definition 10 for $\sigma = 0$ are computationally indistinguishable.*

Proof. Fix the real offset Δ in the real programming string $\rho_1 = (N, p, q, \Delta)$. We define four hybrids.

Hybrid H_0 . This is the real insider experiment for party 0. The adversary receives k_0 generated with offset Δ , and for each random input x_i it receives the honest sender output

$$y_{1,i} = \text{PCF.Eval}(1, k_1, x_i).$$

Hybrid H_1 . Replace each honest sender output by the reverse-sampled value

$$\mathcal{J}_{\text{COT}}.\text{RSample}(1^\lambda, 0, \perp, (N, p, q, \Delta), y_{0,i}), \quad y_{0,i} := \text{PCF.Eval}(0, k_0, x_i).$$

By Lemma 5 and Lemma 2, this replacement is distribution-preserving. Hence H_0 and H_1 are identical.

Hybrid H_2 . Replace the receiver key k_0 generated with offset Δ by an unshifted receiver key in which the table entries are $g^{a_0 r^s}$. The sender outputs continue to be generated by reverse sampling from the party-0 outputs using the real offset Δ . The transition $H_1 \stackrel{\text{comp}}{\equiv} H_2$ follows from Lemma 7 and closure of computational indistinguishability under efficient postprocessing.

Hybrid H_3 . Replace the unshifted receiver key by a receiver key generated with an independently refreshed offset $\tilde{\Delta} \leftarrow \mathcal{D}_{\Delta, N}$, while still reverse sampling the missing sender outputs using the real offset Δ . By another application of Lemma 7, $H_2 \stackrel{\text{comp}}{\equiv} H_3$.

Hybrid H_3 is exactly the ideal insider experiment for $\sigma = 0$: the adversary's key is generated with the refreshed party-1 programming string, whereas the missing party's outputs are sampled from the real correlation by applying the COT reverse sampler to the party-0 output and the real offset. Therefore $H_0 \stackrel{\text{comp}}{\equiv} H_3$.

Theorem 3 (Weak-PCF security for correlated OT). *Assume that F^{IPM} is a binary inner-product-membership weak PRF for S , that F is a secure weak PRF with range $\mathbb{Z}_N$, and that Sparse-Power DCR holds for $T = \{0, \dots, n\}$ and support S with the exponent and offset distributions used above. Then PCF_{COT} is a secure weak PCF for the correlated-OT correlation of Definition 11 in the sense of Definition 10.*

Proof. Correctness is Lemma 5. Pseudorandom correlated outputs are Lemma 6. Weak insider security for party 1 is Lemma 9, and weak insider security for party 0 is Lemma 10. These are exactly the requirements of Definition 10.

E.2 PCF for OLE

We now present the detailed construction of the PCF for OLE correlations, based on the high-level overview from Section 5. We begin with some background on the Goldwasser-Micali cryptosystem.

The target correlation. A bit-OLE sample is

$$((b, u_0), (a, u_1)) \in (\mathbb{F}_2^2) \times (\mathbb{F}_2^2)$$

where $a, b, u_0 \xleftarrow{\$} \mathbb{F}_2$ and

$$u_1 = u_0 + ab.$$

Equivalently, the parties hold additive shares u_0, u_1 of the product ab over $\mathbb{F}_2$, with party 0 holding b and party 1 holding a .

Goldwasser–Micali distributed decryption. Let $N = pq$ be a Blum integer and let

$$J_N^{+1} := \{c \in \mathbb{Z}_N^* : (c/N) = +1\}$$

be the Jacobi-+1 subgroup. Set

$$\Delta_N := \text{lcm}(p-1, q-1)/2.$$

For $c \in J_N^{+1}$ define its Goldwasser–Micali plaintext bit by

$$\text{gm}_N(c) = 0 \iff c \in QR_N, \quad \text{gm}_N(c) = 1 \iff c \in J_N^{+1} \setminus QR_N.$$

Then

$$c^{\Delta_N} = (-1)^{\text{gm}_N(c)} \pmod{N}.$$

We use the following share-conversion procedure from [OSY21]. For $u \in \mathbb{Z}_N^*$, let $\langle u \rangle_N \in \{1, \dots, N-1\}$ denote its canonical representative and define

$$\text{DDec}_N(u) := \chi[\langle u \rangle_N > N/2] \in \mathbb{F}_2.$$

Since N is odd, exactly one of u and $-u$ has representative larger than $N/2$, and hence

$$\text{DDec}_N(-u) = \text{DDec}_N(u) + 1 \quad \text{in } \mathbb{F}_2.$$

Lemma 11 (Goldwasser–Micali distributed decryption). *For every $c \in J_N^{+1}$, every $b \in \mathbb{F}_2$, and all integers s_0, s_1 satisfying*

$$s_0 - s_1 = b \Delta_N,$$

we have

$$\text{DDec}_N(c^{s_0}) + \text{DDec}_N(c^{s_1}) = b \cdot \text{gm}_N(c) \quad \text{in } \mathbb{F}_2.$$

Moreover, if the distribution of c^{s_1} is within statistical distance $\varepsilon_{\text{DDec}}$ of a distribution invariant under multiplication by -1 , then each decryption share is within $\varepsilon_{\text{DDec}}$ of uniform over $\mathbb{F}_2$.

Proof. If $b = 0$, then $s_0 = s_1$ and the two shares are identical, so their sum is zero. If $b = 1$, then

$$c^{s_0} = c^{s_1} c^{\Delta_N} = c^{s_1} (-1)^{\text{gm}_N(c)}.$$

When $\text{gm}_N(c) = 0$, the two inputs to DDec_N are identical and the sum is zero. When $\text{gm}_N(c) = 1$, the two inputs differ by a factor -1 , so the displayed sign relation gives a sum of one. This proves correctness. For the marginal statement, the preimage of 0 under DDec_N is mapped bijectively to the preimage of 1 by multiplication by -1 . Therefore any negation-invariant input distribution gives a uniform output bit, and statistical distance increases by at most the distance to such a distribution.

PCF Construction. Our construction PCF^{OLE} is given in Figure 4.

Lemma 12 (Correctness). *Conditioned on the no-wrap event*

$$\mu_x = m_{0,x} + b_x \Delta \quad \text{over } \mathbb{Z}$$

for the canonical representatives of $\mu_x, m_{0,x} \in \mathbb{Z}_{N'}$, the outputs of PCF^{OLE} form a correct bit-OLE sample.

Proof. The correlated-OT correctness equation gives

$$m_{1,x} = m_{0,x} + \Delta \pmod{N'}, \quad \mu_x = m_{b_x, x} \pmod{N'}.$$

Under the no-wrap event this is the integer identity $\mu_x = m_{0,x} + b_x \Delta$. Therefore

$$T_{0,x}/T_{1,x} = c_x^{\mu_x - m_{0,x}} = c_x^{b_x \Delta} = (-1)^{b_x a_x} \pmod{N},$$

where $a_x = \text{gm}_N(c_x)$. Applying Lemma 11 with $s_0 = \mu_x$ and $s_1 = m_{0,x}$ gives

$$u_{0,x} + u_{1,x} = a_x b_x \quad \text{in } \mathbb{F}_2.$$

Thus the two local outputs are a correct OLE sample.

Theorem 4 (Security). *Assume that PCF^{COT} is a secure weak PCF for fixed-offset COT over $\mathbb{Z}_{N'}$, that H_{GM} is modeled as a random oracle into J_N^{+1} , and that the marginal negation-invariance error in Lemma 11 is at most $\varepsilon_{\text{DDec}}$. For any adversary making at most m evaluation samples and Q_H random-oracle queries, PCF^{OLE} is a weak PCF for bit-OLE with distinguishing advantage bounded by*

$$\text{Adv}_{\text{PCF}^{\text{COT}}}^{\text{wpcf}}(m) + \varepsilon_{\text{wrap}} + \varepsilon_{\text{DDec}} + \binom{Q_H}{2} / |J_N^{+1}|.$$

The same bound, with the corresponding insider advantage of PCF^{COT} , holds for each weak-insider-security experiment.

Proof. We give the pseudorandom-output proof; the insider proofs are the same sequence with the corrupted party's OLE key fixed and the missing COT outputs replaced using the corresponding insider-security experiment of the underlying COT PCF.

Hybrid H_0 is the real pseudorandom-output experiment for the OLE PCF on random inputs $x_1, \dots, x_m$.

Hybrid H_1 aborts if the random oracle H_{GM} has a collision among all points queried by the experiment and the adversary. Since H_{GM} is a random oracle into J_N^{+1} , the transition costs at most $\binom{Q_H}{2} / |J_N^{+1}|$.

Hybrid H_2 replaces the underlying COT-PCF outputs on $x_1, \dots, x_m$ by ideal fixed-offset COT samples with the same offset Δ . Thus, for each i , it samples

$$b_i \xleftarrow{\$} \mathbb{F}_2, \quad m_{0,i} \xleftarrow{\$} \mathbb{Z}_{N'}, \quad m_{1,i} := m_{0,i} + \Delta \pmod{N'}, \quad \mu_i := m_{b_i, i}.$$

Let $\text{PCF}^{\text{OT}} = (\text{PCF}^{\text{OT}}.\text{Gen}, \text{PCF}^{\text{OT}}.\text{Eval})$ be the PCF for correlated OT over $\mathbb{Z}_{N'}$ from the previous construction. Let $H : \{0, 1\}^* \rightarrow J_N^{+1}$ be a random oracle that outputs Jacobi-symbol-+1 elements modulo N , possibly by rejection sampling i.e. sample $c \xleftarrow{\$} \mathbb{Z}_N^*$ until $(c/N) = 1$. This requires no knowledge of factors of N .

$\text{Gen}(1^\lambda)$:

1. Sample a Blum integer $N = pq$ (i.e. distinct primes with $p \equiv q \equiv 3 \pmod{4}$); this is the Goldwasser–Micali modulus.
2. Set $\Delta := \text{lcm}(p-1, q-1)/2 \in \mathbb{Z}$, the Goldwasser–Micali decryption key. For any $c \in J_N^{+1}$,

$$c^\Delta \equiv \begin{cases} +1 \pmod{N} & \text{if } c \in QR_N, \\ -1 \pmod{N} & \text{if } c \in J_N^{+1} \setminus QR_N. \end{cases}$$

3. Sample an RSA modulus N' for the underlying correlated-OT PCF, with $N' \geq 2^\lambda \cdot \Delta$ so that $\Delta \ll N'$. (This guarantees that the MAC relation produced by PCF^{OT} holds also over $\mathbb{Z}$, with overwhelming probability over the random choice of $\mu^x, m_0^x \in \mathbb{Z}_{N'}$.)
4. Run $(k_0^{\text{OT}}, k_1^{\text{OT}}) \leftarrow \text{PCF}^{\text{OT}}.\text{Gen}(1^\lambda, \rho_0, \rho_1)$ on modulus N' where $\rho_0 := \perp$, $\rho_1 := (N', p', q', \Delta)$
5. Output the keys

$$k_0 := (N, N', k_0^{\text{OT}}), \quad k_1 := (N, N', \Delta, k_1^{\text{OT}}).$$

$\text{Eval}(\sigma, k_\sigma, x)$ where $x \in \{0, 1\}^n$:

1. Compute the public Goldwasser–Micali ciphertext $c^x := H(x) \in J_N^{+1}$, which encrypts an (unknown) bit $a^x \in \{0, 1\}$ defined by $a^x = 0$ iff $c^x \in QR_N$.
2. If $\sigma = 0$, parse $k_0 = (N, N', k_0^{\text{OT}})$:
 - (a) Compute $(b^x, \mu^x) \leftarrow \text{PCF}^{\text{OT}}.\text{Eval}(0, k_0^{\text{OT}}, x)$, $b^x \in \{0, 1\}$ and $\mu^x \in \mathbb{Z}_{N'}$.
 - (b) Compute the partial decryption $T_0^x := (c^x)^{\mu^x} \pmod{N}$, treating μ^x as a non-negative integer less than N' , and extract $u_0^x := \text{DDLog}_{\text{GM}}(T_0^x) \in \{0, 1\}$ (where DDLog_{GM} is the GM DDLog procedure from [OSY21]).
 - (c) Output the OLE share (b^x, u_0^x) .
3. If $\sigma = 1$, parse $k_1 = (N, N', \Delta, k_1^{\text{OT}})$:
 - (a) Compute $(m_0^x, m_1^x) \leftarrow \text{PCF}^{\text{OT}}.\text{Eval}(1, k_1^{\text{OT}}, x)$, with $m_0^x, m_1^x \in \mathbb{Z}_{N'}$.
 - (b) Recover $a^x \in \{0, 1\}$ by decrypting c^x with the GM secret key Δ : set $a^x := 0$ if $(c^x)^\Delta \equiv +1 \pmod{N}$, and $a^x := 1$ otherwise.
 - (c) Compute the partial decryption $T_1^x := (c^x)^{m_0^x} \pmod{N}$, treating m_0^x as a non-negative integer less than N' , and extract $u_1^x := \text{DDec}(T_1^x) \in \{0, 1\}$.
 - (d) Output the OLE share (a^x, u_1^x) .

Correctness. The correlated OT correlation gives $\mu^x \equiv m_0^x + b^x \cdot \Delta \pmod{N'}$. Since $\Delta \ll N'$ and $\mu^x, m_0^x \in \mathbb{Z}_{N'}$, this identity holds over $\mathbb{Z}$ except with negligible probability. Treating the exponents as integers and reducing modulo N ,

$$\frac{T_0^x}{T_1^x} = (c^x)^{\mu^x - m_0^x} = (c^x)^{b^x \cdot \Delta} \pmod{N},$$

which equals 1 when $b^x = 0$ and $(c^x)^\Delta \in \{+1, -1\}$ when $b^x = 1$. Since $(c^x)^\Delta = (-1)^{a^x}$ by the choice of Δ as the GM decryption key, in all cases $T_0^x/T_1^x \in \{+1, -1\}$ and equals -1 iff $a^x \cdot b^x = 1$. The Orlandi–Scholl–Yakoubov rounding DDec then yields

$$u_0^x \oplus u_1^x = a^x \cdot b^x \pmod{2},$$

and therefore the output is a valid OLE correlation over $\mathbb{F}_2$, where party 0 receives (b^x, u_0^x) and party 1 receives (a^x, u_1^x) :

$$a^x \cdot b^x = u_0^x \oplus u_1^x.$$

Fig. 4. PCF for OLE over $\mathbb{F}_2$ based on Goldwasser–Micali

The transition is bounded by the pseudorandom-output advantage of PCF^{COT} .

Hybrid H_3 conditions on the no-wrap event that, for every i , the canonical representatives satisfy

$$\mu_i = m_{0,i} + b_i \Delta \quad \text{over } \mathbb{Z}.$$

For one sample the event fails only when $b_i = 1$ and $m_{0,i} > N' - 1 - \Delta$, which has probability at most Δ/N' . By the parameter choice and a union bound over m samples, the loss is at most $\varepsilon_{\text{wrap}}$.

Hybrid H_4 uses the random-oracle table to sample independent uniform ciphertexts $c_i \xleftarrow{\$} J_N^{+1}$ for the distinct inputs x_i . This is a syntactic rewrite of H_3 after conditioning on no collisions. For uniform $c_i \in J_N^{+1}$, the bit

$$a_i := \text{gm}_N(c_i)$$

is uniform in $\mathbb{F}_2$ because QR_N has index two in J_N^{+1} , and it is independent of b_i and $m_{0,i}$.

In H_4 , the OLE outputs are

$$(b_i, u_{0,i}), \quad (a_i, u_{1,i}),$$

where

$$u_{0,i} = \text{DDec}_N(c_i^{m_{0,i} + b_i \Delta}), \quad u_{1,i} = \text{DDec}_N(c_i^{m_{0,i}}).$$

By Lemma 11, the shares satisfy

$$u_{0,i} + u_{1,i} = a_i b_i \quad \text{in } \mathbb{F}_2,$$

and the marginal share distribution is within $\varepsilon_{\text{DDec}}$ of uniform. Therefore the joint distribution of each tuple is statistically close to the ideal bit-OLE distribution. Independence across distinct inputs follows from the independent COT samples and independent random-oracle ciphertexts. Combining the hybrid distances proves the pseudorandom-output claim.

For insider security, fix $\sigma \in \{0, 1\}$. The real insider experiment gives the adversary the OLE key k_σ , which contains the corresponding COT key k_σ^{COT} and only public GM information, except that for $\sigma = 1$ it also contains the GM trapdoor Δ and hence the party-1 OLE input bit a_i . Replace the missing party's COT outputs by reverse-sampled ideal COT outputs using the weak-insider security of PCF^{COT} . The random-oracle and no-wrap hybrids above are independent of the corrupted key. Finally, Lemma 11 converts the ideal COT relation into ideal OLE shares. This gives the same bound with the COT insider advantage in place of the COT pseudorandom-output advantage.

Corollary 6 (Multiplication triples over $\mathbb{F}_2$). *The bit-OLE PCF yields a weak PCF for multiplication triples over $\mathbb{F}_2$ by the standard local conversion: sample a bit-OLE tuple $((b, u_0), (a, u_1))$, let party 0 output (b, u_0) and party 1 output (a, u_1) , and interpret u_0, u_1 as additive shares of ab . Equivalently, two independent bit-OLE samples can be locally rearranged into the common symmetric triple convention in which both multiplicands are additively shared.*

F From Correlated OT to $\mathbb{Z}_N$ -VOLE

F.1 Local aggregation to $\mathbb{Z}_N$ -VOLE

The correlated-OT PCF of Section 4 produces binary-choice correlations with a common hidden offset. Now we discuss how local aggregation of such correlated OT samples results in $\mathbb{Z}_N$ -VOLE. Note that this transformation is purely algebraic; once the parties hold fixed-offset COT samples with the same offset, no further cryptographic assumption is used.

Fix a modulus N , an offset $\Delta \in \mathbb{Z}_N$, an integer $\ell = \ell(\lambda)$, and public coefficients

$$c_0, c_1, \dots, c_{\ell-1} \in \mathbb{Z}_N.$$

Suppose that the two parties hold ℓ independently sampled fixed-offset COT correlations over $\mathbb{Z}_N$ with common offset Δ . Thus, for each $j \in \{0, \dots, \ell-1\}$, party 0 holds

$$(b_j, \mu_j) \in \{0, 1\} \times \mathbb{Z}_N,$$

party 1 holds

$$(m_{0,j}, m_{1,j}) \in \mathbb{Z}_N^2,$$

and the values satisfy

$$m_{1,j} = m_{0,j} + \Delta \pmod{N}, \quad \mu_j = m_{b_j, j}.$$

Equivalently,

$$\mu_j = m_{0,j} + b_j \Delta \pmod{N}.$$

Define the local aggregation map $\mathbf{Agg}_{\vec{c}}$ by

$$\begin{aligned} u &:= \sum_{j=0}^{\ell-1} c_j b_j \pmod{N}, \\ v &:= \sum_{j=0}^{\ell-1} c_j m_{0,j} \pmod{N}, \\ w &:= \sum_{j=0}^{\ell-1} c_j \mu_j \pmod{N}. \end{aligned}$$

Party 0 outputs (u, w) , while party 1 outputs (Δ, v) .

Lemma 13 (Local COT-to-VOLE aggregation). *For every fixed coefficient vector $\vec{c} = (c_0, \dots, c_{\ell-1}) \in \mathbb{Z}_N^\ell$, the map $\mathbf{Agg}_{\vec{c}}$ sends ℓ fixed-offset COT samples with common offset Δ to a valid $\mathbb{Z}_N$ -VOLE sample with multiplier Δ . More explicitly, the aggregated values satisfy*

$$w = v + \Delta u \pmod{N}.$$

Furthermore, if the underlying COT samples are distributed as the ideal fixed-offset COT correlation, then the aggregated distribution is the distribution obtained by applying the deterministic map $\mathbf{Agg}_{\vec{c}}$ to that ideal correlation. If at least one coefficient c_j is a unit in $\mathbb{Z}_N$, then $v = \sum_j c_j m_{0,j}$ is uniform over $\mathbb{Z}_N$ even conditioned on u and Δ ; in particular this holds for the powers-of-two coefficients because $c_0 = 1$. Consequently, computational indistinguishability of real and ideal COT samples is preserved under aggregation.

Proof. Using the COT identity $\mu_j = m_{0,j} + b_j \Delta$ for every j , we compute in $\mathbb{Z}_N$:

$$\begin{aligned} w &= \sum_{j=0}^{\ell-1} c_j \mu_j \\ &= \sum_{j=0}^{\ell-1} c_j (m_{0,j} + b_j \Delta) \\ &= \sum_{j=0}^{\ell-1} c_j m_{0,j} + \Delta \sum_{j=0}^{\ell-1} c_j b_j \\ &= v + \Delta u. \end{aligned}$$

The distributional statement is immediate because $\mathbf{Agg}_{\vec{c}}$ is a deterministic polynomial-time map applied locally to the outputs of the COT correlation. If some c_j is a unit modulo N , then for fixed values of all variables except $m_{0,j}$, the map $m_{0,j} \mapsto \sum_i c_i m_{0,i}$ is a bijection of $\mathbb{Z}_N$; hence v is uniform, including after conditioning on (u, Δ) . Finally, if two ensembles of COT output tuples are computationally indistinguishable, then their images under any deterministic polynomial-time map are computationally indistinguishable.

F.2 Towards a PCF for VOLE

Let

$$\text{PCF}_{\text{COT}} = (\text{PCF}_{\text{COT}}.\text{Gen}, \text{PCF}_{\text{COT}}.\text{Eval})$$

be the weak PCF for fixed-offset correlated OT from Section 4, with input domain $\{0, 1\}^n$. The induced VOLE-PCF has input domain

$$\mathcal{X}_{\text{VOLE}} := (\{0, 1\}^n)^\ell.$$

For an input

$$X = (x_0, \dots, x_{\ell-1}) \in \mathcal{X}_{\text{VOLE}},$$

party 0 evaluates the COT-PCF on every component x_j , obtains outputs (b_j, μ_j) , and returns

$$(u, w) = \left(\sum_{j=0}^{\ell-1} c_j b_j, \sum_{j=0}^{\ell-1} c_j \mu_j \right) \pmod{N}.$$

Party 1 evaluates the COT-PCF on every component x_j , obtains outputs $(m_{0,j}, m_{1,j})$, and returns

$$(\Delta, v) = \left(\Delta, \sum_{j=0}^{\ell-1} c_j m_{0,j} \right) \pmod{N}.$$

The public parameters and the key-generation algorithm are inherited from PCF_{COT} ; only evaluation is changed by applying the local aggregation map.

Theorem 5 (Weak VOLE-PCF from weak COT-PCF). *Assume that PCF_{COT} is a secure weak PCF for fixed-offset correlated OT over $\mathbb{Z}_N$ in the sense of Definition 10. Then the construction above is a secure weak PCF for the fixed-multiplier $\mathbb{Z}_N$ -VOLE correlation, with respect to the aggregated distribution induced by the coefficient vector $\vec{c}$.*

Proof. We give the proof through explicit hybrids. Fix a polynomially bounded number $m = m(\lambda)$ of VOLE-PCF inputs, and let

$$X_i = (x_{i,0}, \dots, x_{i,\ell-1}) \xleftarrow{\$} (\{0, 1\}^n)^\ell \quad (i \in [m])$$

be the random inputs sampled by the weak-PCF experiment. The total number of underlying COT inputs is $m\ell$, which is polynomially bounded.

Hybrid Hyb_0 is the real VOLE-PCF experiment. It generates the COT-PCF keys once and, for every pair (i, j) , computes the real COT-PCF outputs on input $x_{i,j}$. It then applies $\text{Agg}_{\vec{c}}$ separately for each i .

Hybrid Hyb_1 replaces all underlying real COT-PCF output pairs on the inputs

$$(x_{i,j})_{i \in [m], 0 \leq j < \ell}$$

by samples from the ideal fixed-offset COT correlation with the same public setup and the same offset. This transition is justified directly by the pseudorandom-output security of PCF_{COT} , because the multiset of all underlying inputs is a polynomial-size list of uniformly random COT-PCF inputs. Repetitions are handled exactly as in Definition 10, by using a table so that equal inputs receive equal correlation samples in both experiments.

Hybrid Hyb_2 applies the deterministic aggregation map $\text{Agg}_{\vec{c}}$ to the ideal COT samples. This is not a distributional change from Hyb_1 ; it is only a rewriting of the sampled values. By Lemma 13, each aggregated tuple satisfies

$$w_i = v_i + \Delta u_i \pmod{N}.$$

Thus Hyb_2 is exactly the ideal weak-PCF experiment for the induced $\mathbb{Z}_N$ -VOLE correlation. Therefore the real and ideal VOLE-PCF experiments are computationally indistinguishable.

The insider-security experiments are handled in the same way. For a fixed party σ , the adversary receives that party's key and the opposite party's outputs on polynomially many random VOLE inputs. Expanding each VOLE input into its ℓ COT inputs gives a polynomial-size weak-insider experiment for the underlying COT-PCF. We first replace the opposite party's underlying COT outputs by the corresponding reverse-sampled ideal COT outputs, using the insider security of PCF_{COT} . We then apply the same deterministic aggregation map. Since reverse sampling for VOLE is exactly the image of reverse sampling for the underlying COT samples under $\text{Agg}_{\vec{c}}$, the resulting distribution is the ideal VOLE insider experiment. This proves weak-PCF security for the aggregated construction.

F.3 Using a random oracle for external labels

Weak-PCF security is stated for uniformly random inputs. In applications such as the exponent-VRF of Section 6, the external input is adversarially chosen. We therefore separate the external label space from the PCF input space by a domain-separated hash

$$H_{\text{VOLE}} : \{0, 1\}^* \rightarrow (\{0, 1\}^n)^\ell,$$

modeled as a random oracle. On an external label x , the VOLE-PCF is evaluated at

$$X_x := H_{\text{VOLE}}(x).$$

For security as a weak PCF it is enough that each tuple X_x be distributed as a uniform element of $(\{0, 1\}^n)^\ell$. For applications that require the aggregated choice value to be close to uniform, however, one also wants

the underlying COT labels inside all queried tuples to be distinct. We therefore use the following explicit collision event. For an external label x , write

$$H_{\text{VOLE}}(x) = (H_0(x), \dots, H_{\ell-1}(x)), \quad H_j(x) \in \{0, 1\}^n.$$

For at most $Q_H = Q_H(\lambda)$ distinct random-oracle queries, define

$$\text{Coll}_{\text{COT}} := \left\{ \exists(x, j) \neq (x', j') : H_j(x) = H_{j'}(x') \right\}.$$

Since the random oracle samples each $H_j(x)$ uniformly in $\{0, 1\}^n$, the union bound gives

$$\Pr[\text{Coll}_{\text{COT}}] \leq \binom{Q_H \ell}{2} 2^{-n}.$$

Conditioned on the complement of Coll_{COT} , every underlying COT-PCF input used by the adversary is a fresh uniform input. The weak-PCF proof therefore applies verbatim to the list of all underlying COT inputs, and the powers-of-two aggregation analysed below has independent ideal choice bits. Thus the random-oracle loses at most the collision probability above, in addition to the weak-PCF advantage.

F.4 Powers-of-two aggregation and statistical uniformity

The most useful coefficient vector for typical applications is the powers-of-two vector

$$c_j = 2^j \quad (0 \leq j < \ell).$$

For independent uniform bits $b_0, \dots, b_{\ell-1}$, the integer

$$U := \sum_{j=0}^{\ell-1} 2^j b_j$$

is exactly uniform over the interval $\{0, 1, \dots, 2^\ell - 1\}$. In the PCF, these bits are independent in the ideal experiment whenever the underlying COT inputs used for the ℓ positions are distinct; the random oracle usage above enforces this condition except with probability at most $\binom{Q_H \ell}{2} 2^{-n}$ over Q_H external queries. The aggregated VOLE choice value is $u := U \bmod M$ for the chosen output modulus M .

Lemma 14 (Reduction of a binary integer modulo M). *Let $M \geq 2$ be an integer, let $U \xleftarrow{\$} \{0, 1, \dots, 2^\ell - 1\}$, and let $u := U \bmod M$. If U_M denotes the uniform distribution over $\mathbb{Z}_M$, then*

$$\Delta_{\text{stat}}(u, U_M) \leq \frac{M}{2^\ell}.$$

In particular, if $\ell \geq \lceil \log_2 M \rceil + \kappa$, then u is within statistical distance at most $2^{-\kappa}$ of uniform over $\mathbb{Z}_M$.

Proof. Write

$$2^\ell = tM + r, \quad 0 \leq r < M.$$

Exactly r residue classes modulo M have $t + 1$ preimages in $\{0, \dots, 2^\ell - 1\}$, and the remaining $M - r$ residue classes have t preimages. Therefore, for every $a \in \mathbb{Z}_M$,

$$\Pr[u = a] \in \left\{ \frac{t}{2^\ell}, \frac{t+1}{2^\ell} \right\}.$$

The deviation of any point probability from $1/M$ is at most $1/2^\ell$. Hence

$$\begin{aligned} \Delta_{\text{stat}}(u, U_M) &= \frac{1}{2} \sum_{a \in \mathbb{Z}_M} \left| \Pr[u = a] - \frac{1}{M} \right| \\ &\leq \frac{1}{2} \cdot M \cdot \frac{1}{2^\ell} \leq \frac{M}{2^\ell}. \end{aligned}$$

The stated parameter consequence follows immediately from $2^\ell \geq 2^\kappa M$.

For exact uniformity one can instead use rejection sampling. Namely, sample $U \xleftarrow{\$} \{0, \dots, 2^\ell - 1\}$, accept only if $U < tM$ for $t = \lfloor 2^\ell / M \rfloor$, and output $U \bmod M$. This produces the exact uniform distribution over $\mathbb{Z}_M$ and rejects with probability $r/2^\ell < M/2^\ell$. In a PCF, however, rejection must be implemented through deterministic domain separation, because both parties must evaluate the same accepted index set without communication. For this reason the applications below use the simpler statistically close powers-of-two sampler.

F.5 Integer lifting and prime-field reduction

The Paillier PCF natively gives correlations modulo the RSA modulus N . Some applications, including the exponent-VRF in a group of prime order q , require VOLE over $\mathbb{Z}_q$. The safe route is to first interpret the $\mathbb{Z}_N$ correlation as an integer relation, and then reduce the integer equality modulo q . This step is valid only when the product term is small relative to N , as in the DDLog-based lifting arguments of [OSY21].

Lemma 15 (Integer lift for a fixed product). *Represent elements of $\mathbb{Z}_N$ by their canonical representatives in $\{0, \dots, N-1\}$. Let*

$$((u, w), (\Delta, v)) \in (\mathbb{Z}_N^2) \times (\mathbb{Z}_N^2)$$

satisfy

$$w \equiv v + \Delta u \pmod{N}.$$

Assume that the canonical representatives of Δ and u are nonnegative integers and set

$$t := \Delta u \in \mathbb{Z}.$$

If $0 \leq t < N$ and v is uniform over $\mathbb{Z}_N$ even after conditioning on u and Δ , then

$$\Pr[w - v = t \text{ over } \mathbb{Z} \mid u, \Delta] = 1 - \frac{t}{N}.$$

In particular, if $t \leq N2^{-\kappa}$, then the canonical representatives satisfy

$$w = v + \Delta u \text{ over } \mathbb{Z}$$

except with probability at most $2^{-\kappa}$.

Proof. Fix the canonical representatives of u and Δ , and hence fix $t = \Delta u$. Since $w \equiv v + t \pmod{N}$ and $0 \leq t < N$, the canonical representative $w \in \{0, \dots, N-1\}$ is equal to $v + t$ over the integers if and only if

$$v + t < N.$$

Equivalently, lifting fails if and only if

$$v \in \{N - t, N - t + 1, \dots, N - 1\}.$$

This set has exactly t elements. Because v is uniform over $\mathbb{Z}_N$ conditioned on u and Δ , the failure probability is t/N , and the success probability is $1 - t/N$. The final statement follows from $t \leq N2^{-\kappa}$.

Corollary 7 (Prime-field VOLE from a lifted Paillier VOLE). *Let q be prime. Suppose that a $\mathbb{Z}_N$ -VOLE sample satisfies the hypotheses of Lemma 15, and suppose that the lifting failure probability is at most ϵ_{lift} . Reducing the canonical representatives of u, w, Δ, v modulo q gives values satisfying*

$$w_q = v_q + \Delta_q u_q \pmod{q}$$

except with probability at most ϵ_{lift} . If v is uniform over $\mathbb{Z}_N$, then $v_q = v \bmod q$ is within statistical distance at most q/N of uniform over $\mathbb{Z}_q$. Thus the reduced sample is within distance at most $\epsilon_{\text{lift}} + q/N$ from the ideal $\mathbb{Z}_q$ -VOLE distribution, plus any statistical error in the distributions of u_q and Δ_q .

Proof. On the complement of the lifting failure event, the canonical representatives satisfy the equality

$$w = v + \Delta u$$

over the integers. Applying the canonical ring homomorphism $\mathbb{Z} \rightarrow \mathbb{Z}_q$ gives

$$w \bmod q = (v \bmod q) + (\Delta \bmod q)(u \bmod q) \pmod{q}.$$

This proves the algebraic relation outside the lifting failure event. The additional statistical term comes from reducing a uniform residue modulo N to modulo q : by Lemma 14, applied with interval length N and modulus q , the distribution of $v \bmod q$ is at distance at most q/N from uniform over $\mathbb{Z}_q$.

Corollary 8 (Concrete parameter condition for prime-order targets). *Let q be a prime target modulus, let κ_{stat} , κ_{lift} , and κ_{red} be statistical security parameters, and use powers-of-two aggregation with $c_j = 2^j$. Suppose*

$$\ell \geq \lceil \log_2 q \rceil + \kappa_{\text{stat}}$$

and the Paillier modulus N is chosen so that

$$(q-1)(2^\ell - 1) \leq N2^{-\kappa_{\text{lift}}} \quad \text{and} \quad q/N \leq 2^{-\kappa_{\text{red}}}.$$

If the offset is sampled as an integer $\Delta \in \{0, \dots, q-1\}$ and embedded into $\mathbb{Z}_N$, then the induced output, reduced modulo q , is within statistical distance at most

$$2^{-\kappa_{\text{stat}}} + 2^{-\kappa_{\text{lift}}} + 2^{-\kappa_{\text{red}}}$$

from an ideal $\mathbb{Z}_q$ -VOLE sample for a single fresh VOLE label, apart from the computational error of the underlying PCF and apart from any collision event among the underlying COT labels. When the random oracle is used for at most Q_H distinct external labels, one adds the term $\binom{Q_H \ell}{2} 2^{-n}$.

Proof. The first inequality and Lemma 14 imply that $U \bmod q$ is within distance at most $2^{-\kappa_{\text{stat}}}$ of uniform over $\mathbb{Z}_q$. The second inequality implies that, for every possible powers-of-two integer $U \in \{0, \dots, 2^\ell - 1\}$ and every possible offset $\Delta \in \{0, \dots, q-1\}$,

$$\Delta U \leq (q-1)(2^\ell - 1) \leq N2^{-\kappa_{\text{lift}}}.$$

Thus Lemma 15 gives lifting failure probability at most $2^{-\kappa_{\text{lift}}}$. Finally, Corollary 7 transports the lifted integer equality to $\mathbb{Z}_q$ and accounts for the residual non-uniformity of $v \bmod q$, bounded by $q/N \leq 2^{-\kappa_{\text{red}}}$. The three statistical losses are combined by the triangle inequality.

G Details on Exponent-VRF Constructions

G.1 Designated-Verifier eVRF

Let

$$\text{PCF}_{\text{VOLE},q} = (\text{PCF.Gen}, \text{PCF.Eval})$$

be a weak PCF for $C_{\text{VOLE},q}$. Such a PCF is obtained from Section 4 by applying the aggregation of Section F.2 and then the integer-lifting and prime-field reduction of Section F.5. The concrete parameter condition in Corollary 8 is sufficient for the induced $\mathbb{Z}_q$ -VOLE distribution to be statistically close to ideal.

Because the PCF is weak, we evaluate it on random-oracle images of the external eVRF inputs. Let

$$H_{\text{VOLE}} : \{0, 1\}^* \rightarrow \mathcal{X}_{\text{VOLE}}$$

be a domain-separated random oracle into the VOLE-PCF input domain. For an external input x , write

$$X_x := H_{\text{VOLE}}(x).$$

For any execution that evaluates at most Q distinct external labels under H_{VOLE} , let $\epsilon_{\text{RO}}(Q)$ denote the probability that the random-oracle images fail the freshness condition required by the weak-PCF proof. For an abstract VOLE-PCF with input domain $\mathcal{X}_{\text{VOLE}}$, one may take

$$\epsilon_{\text{RO}}(Q) \leq \binom{Q}{2} / |\mathcal{X}_{\text{VOLE}}|.$$

For the concrete COT-aggregation instantiation of Section F.3, where $\mathcal{X}_{\text{VOLE}} = (\{0, 1\}^n)^\ell$, the stronger freshness event is that all underlying COT labels are distinct, and the bound is

$$\epsilon_{\text{RO}}(Q) \leq \binom{Q\ell}{2} 2^{-n}.$$

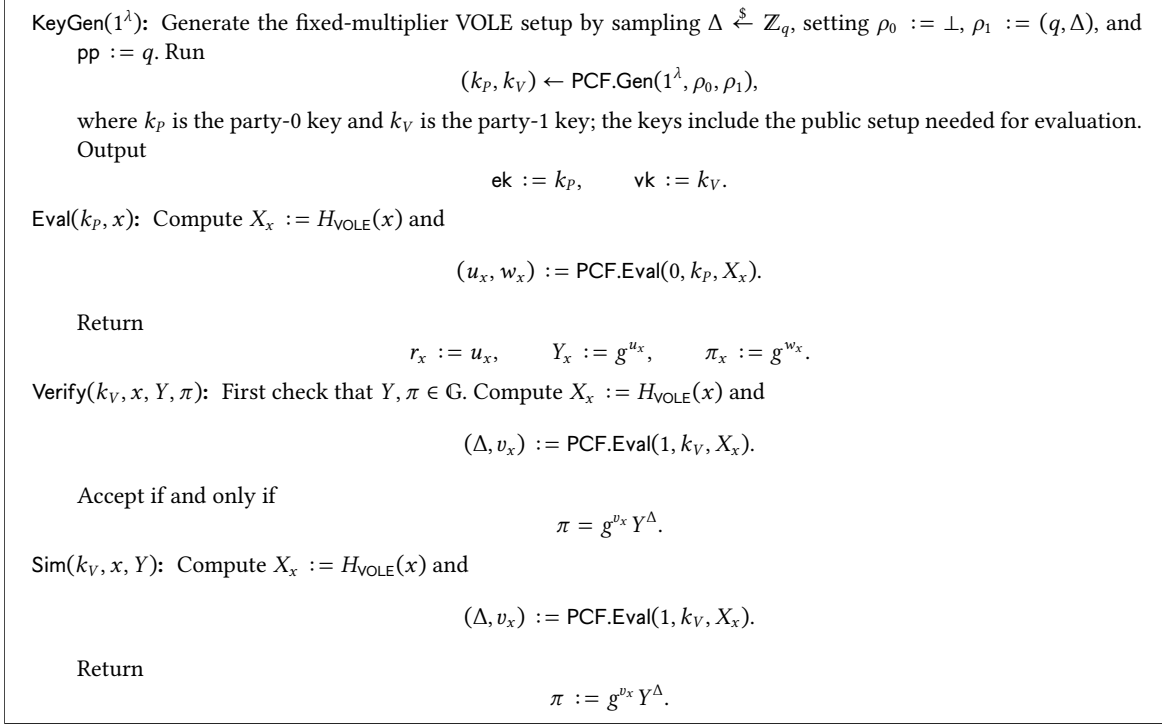


Fig. 5. Designated-verifier eVRF based on VOLE-PCF

Construction. The designated-verifier eVRF

$$\Pi_{\text{dv}}[\text{PCF}_{\text{VOLE},q}] = (\text{KeyGen}, \text{Eval}, \text{Verify}, \text{Sim})$$

is defined as in Figure 5.

Theorem 6 (Correctness). *If the underlying VOLE-PCF is correct, then $\Pi_{\text{dv}}[\text{PCF}_{\text{VOLE},q}]$ satisfies correctness in Definition 4.*

Proof. Fix honestly generated keys and an input x . Let

$$(u_x, w_x) = \text{PCF.Eval}(0, k_P, X_x), \quad (\Delta, v_x) = \text{PCF.Eval}(1, k_V, X_x).$$

VOLE correctness gives

$$w_x = v_x + \Delta u_x \pmod{q}.$$

Therefore

$$\pi_x = g^{w_x} = g^{v_x + \Delta u_x} = g^{v_x} (g^{u_x})^\Delta = g^{v_x} Y_x^\Delta.$$

The verifier accepts, and $Y_x = g^{r_x}$ by the definition $r_x = u_x$.

Lemma 16 (Exact proof simulation for fixed keys). *For every honestly generated verifier key k_V , every input x , and every group element $Y \in \mathbb{G}$, the simulator output*

$$\text{Sim}(k_V, x, Y) = g^{v_x} Y^\Delta, \quad (\Delta, v_x) = \text{PCF.Eval}(1, k_V, H_{\text{VOLE}}(x)),$$

is accepted by $\text{Verify}(k_V, x, Y, \cdot)$. In particular, for an honestly generated eVRF output $Y_x = g^{u_x}$, the simulated proof is identical to the honestly generated proof.

Proof. The first statement follows by substituting the simulator output into the verifier equation. For the second statement, correctness gives

$$g^{w_x} = g^{v_x} Y_x^\Delta,$$

so the honest proof g^{w_x} is exactly the simulator output.

Theorem 7 (Designated-verifier verifiability). *Assume that the underlying VOLE-PCF satisfies weak insider security against party 0 and that the real multiplier Δ is uniform over $\mathbb{Z}_q$. Let an adversary in the designated-verifier verifiability experiment make at most Q_V verification queries and at most Q_H direct queries to H_{VOLE} , and set $Q := Q_H + Q_V$ to also count oracle evaluations performed internally by the verifier. Then its advantage is at most*

$$\text{Adv}_{\text{PCF}}^{\text{ins}}(Q) + \frac{Q_V}{q} + \varepsilon_{\text{RO}}(Q),$$

where $\text{Adv}_{\text{PCF}}^{\text{ins}}$ denotes the corresponding weak-insider advantage of the VOLE-PCF. In particular, the error is negligible when q is exponential in λ .

Proof. We spell out the hybrids. In the designated-verifier verifiability experiment the adversary is given the prover key k_P and oracle access to a verifier that uses the hidden verifier key k_V . For a query (x, Y^*, π^*) , let $X_x = H_{\text{VOLE}}(x)$, let

$$(u_x, w_x) := \text{PCF.Eval}(0, k_P, X_x), \quad Y_x := g^{u_x}, \quad \pi_x := g^{w_x},$$

and call the query wrong if $Y^* \neq Y_x$.

Hybrid Hyb_0 is the real verifiability experiment.

Hybrid Hyb_1 conditions on the random-oracle freshness event for the at most $Q = Q_H + Q_V$ labels used in the experiment, defining $\varepsilon_{\text{RO}}(Q)$. The loss is at most $\varepsilon_{\text{RO}}(Q)$.

Hybrid Hyb_2 replaces the verifier-side outputs on all queried PCF inputs by ideal reverse-sampled VOLE outputs conditioned on the prover-side values. That is, the hybrid samples the fixed multiplier $\Delta \xleftarrow{\$} \mathbb{Z}_q$ once, as in the VOLE setup, and for every input X_x , after computing (u_x, w_x) from the prover key, sets

$$v_x := w_x - \Delta u_x \pmod{q}.$$

The verification oracle then accepts exactly when

$$\pi^* = g^{v_x}(Y^*)^\Delta.$$

The transition $\text{Hyb}_1 \approx_c \text{Hyb}_2$ is the weak insider security of the VOLE-PCF for $\sigma = 0$, because the adversary holds the party-0 key and sees only party-1 outputs on random PCF inputs, possibly through the deterministic verification predicate. The random-oracle conditioning ensures that those PCF inputs are fresh random inputs.

We now bound the probability that a wrong query accepts in Hyb_2 . Fix the adversary's coins and the random-oracle table, and consider any verification query before the verifier's answer is returned. If $Y^* \neq Y_x$, then because G has prime order there exists a unique nonzero $d \in \mathbb{Z}_q^*$ such that

$$Y^*/Y_x = g^d.$$

Using $\pi_x = g^{w_x} = g^{v_x} Y_x^\Delta$, the acceptance equation for the wrong output is equivalent to

$$\frac{\pi^*}{\pi_x} = \left(\frac{Y^*}{Y_x} \right)^\Delta = g^{d\Delta}.$$

For fixed Y^*, π^*, Y_x, π_x and nonzero d , this equation has exactly one solution for $\Delta \in \mathbb{Z}_q$. Therefore a fixed wrong query accepts with probability at most $1/q$ over the uniform multiplier. Taking a union bound over the at most Q_V verification queries gives Q_V/q . Combining this with the insider-security loss and the random-oracle collision loss proves the theorem.

Theorem 8 (Pseudorandomness with simulated proofs). *Assume that the underlying VOLE-PCF satisfies weak insider security against party 1. For a pseudorandomness adversary making at most Q_E eVRF-oracle queries and Q_H direct queries to H_{VOLE} , set $Q := Q_E + Q_H$. Then its distinguishing advantage is at most*

$$\text{Adv}_{\text{PCF}}^{\text{ins}}(Q_E) + \varepsilon_{\text{RO}}(Q),$$

where $\text{Adv}_{\text{PCF}}^{\text{ins}}(Q_E)$ is the corresponding weak-insider advantage for party 1 on Q_E random PCF inputs. If the concrete Paillier-to-prime-field instantiation of Corollary 8 is used, the statistical approximation error from that corollary is added to the bound.

Proof. Let $\mathcal{A}$ be a distinguisher in the pseudorandomness experiment of Definition 4. The adversary is given the verifier key k_V and oracle access to either the real eVRF oracle or the ideal oracle that returns random group elements with simulated proofs.

Hybrid Hyb_0 is the real eVRF oracle. On input x , it computes $X_x = H_{\text{VOLE}}(x)$, evaluates

$$(u_x, w_x) = \text{PCF.Eval}(0, k_P, X_x),$$

and returns

$$(Y_x, \pi_x) = (g^{u_x}, g^{w_x}).$$

Hybrid Hyb_1 conditions on the random-oracle freshness event for the at most $Q = Q_E + Q_H$ labels used in the experiment. The loss is at most $\varepsilon_{\text{RO}}(Q)$.

Hybrid Hyb_2 replaces the prover-side VOLE-PCF outputs by ideal reverse-sampled outputs conditioned on the verifier-side output. More precisely, for each fresh PCF input X_x , compute

$$(\Delta, v_x) := \text{PCF.Eval}(1, k_V, X_x)$$

from the verifier key, then sample

$$u_x \xleftarrow{\$} \mathbb{Z}_q, \quad w_x := v_x + \Delta u_x \pmod{q}.$$

Return (g^{u_x}, g^{w_x}) . The transition from Hyb_1 to Hyb_2 is exactly weak insider security of the VOLE-PCF for $\sigma = 1$: the distinguisher holds the party-1 key and sees party-0 outputs on uniformly random PCF inputs. This transition contributes $\text{Adv}_{\text{PCF}}^{\text{ins}}(Q_E)$.

In Hyb_2 , the group element $Y_x = g^{u_x}$ is uniform in $\mathbb{G}$, since u_x is uniform in $\mathbb{Z}_q$. Moreover

$$g^{w_x} = g^{v_x + \Delta u_x} = g^{v_x} (g^{u_x})^\Delta = \text{Sim}(k_V, x, Y_x).$$

Thus Hyb_2 is exactly the ideal pseudorandomness oracle of Definition 4. This proves the theorem.

Theorem 9 (Security of the designated-verifier construction). *Under the assumptions of Theorem 5, the statistical conditions of Corollary 8, and the random-oracle model for H_{VOLE} , the construction $\Pi_{\text{dv}}[\text{PCF}_{\text{VOLE},q}]$ is a designated-verifier exponent VRF in the sense of Definition 4. The verifiability error for Q_V verification queries is at most Q_V/q plus the VOLE-PCF insider-security advantage, the prime-field statistical approximation error from Corollary 8, and the random-oracle freshness term $\varepsilon_{\text{RO}}(Q)$.*

Proof. Correctness is Theorem 6. The simulator required by the pseudorandomness experiment is the exact simulator of Lemma 16. Verifiability is Theorem 7, and pseudorandomness with simulated proofs is Theorem 8.

G.2 Publicly Verifiable eVRF

Theorem 10 (Correctness). *If the offline/online HSS scheme satisfies the strong correctness equation $(\star)$, then honest public eVRF evaluations verify.*

Proof. For an honest proof, the commitment check holds because

$$C_\alpha^\delta C_\theta = \text{Com}(\vec{\alpha}; \rho_\alpha)^\delta \text{Com}(\vec{\theta}_0; \rho_\theta) = \text{Com}(\vec{\alpha}\delta + \vec{\theta}_0; \rho_\alpha\delta + \rho_\theta) = \text{Com}(\vec{\theta}_1; \rho_1).$$

By the HSS correctness equation,

$$W_1 - W_0 = \delta F_K(x) = \delta y.$$

Since $T = g^{-W_0}$ and $Y = g^y$, we have

$$T g^{W_1} = g^{-W_0} g^{W_1} = g^{\delta y} = Y^\delta.$$

Thus both verifier checks accept.

Theorem 11 (Public verifiability). *Assume that the Pedersen commitment is computationally binding, that the HSS correctness equation $(\star)$ holds for all valid offline/online shares, and that H_{FS} is modeled as a random oracle. Then, for an honestly generated key pair, every PPT adversary given (pk, sk) and oracle access to H_{FS} outputs an accepting transcript (x, Y, π) with*

$$Y \neq g^{F_K(x)}$$

only with negligible probability, up to the commitment-binding error, the random-oracle guessing probability, and the standard Fiat–Shamir forking loss. Consequently, there is at most one valid exponent output for each input.

Proof. Let $\pi = (T, C_\theta, \vec{\theta}_1, \rho_1)$ be an accepting proof for challenge

$$\delta = H_{FS}(pk, x, Y, T, C_\theta).$$

If the adversary did not query the random oracle on (pk, x, Y, T, C_θ) before outputting the proof, then the challenge is independent of its view and the event is covered by the usual $1/q - 1$ guessing loss. Otherwise, the forking lemma gives, from any non-negligible successful adversary, two accepting transcripts with the same (pk, x, Y, T, C_θ) and distinct challenges $\delta \neq \delta'$, except for the standard forking loss. Write the second response as $(\vec{\theta}'_1, \rho'_1)$ and let the corresponding verifier-side HSS outputs be W_1 and W'_1 .

The honestly generated commitment C_α has opening $(\vec{\alpha}, \rho_\alpha)$. The two commitment checks imply

$$C_\theta = \text{Com}(\vec{\theta}_1 - \vec{\alpha}\delta; \rho_1 - \rho_\alpha\delta) = \text{Com}(\vec{\theta}'_1 - \vec{\alpha}\delta'; \rho'_1 - \rho_\alpha\delta').$$

By binding of Com, except with negligible probability, these two openings define the same offline share

$$\vec{\theta}_0 := \vec{\theta}_1 - \vec{\alpha}\delta = \vec{\theta}'_1 - \vec{\alpha}\delta'.$$

Hence both transcripts correspond to valid semi-private HSS sharings generated from the same offline share $\vec{\theta}_0$. Define

$$W_0 := \text{OHSS.AuthEval}_0(ek_0, \kappa_0, \vec{\theta}_0, C_x).$$

Strong HSS correctness gives

$$W_1 - W_0 = \delta F_K(x), \quad W'_1 - W_0 = \delta' F_K(x).$$

Subtracting,

$$W_1 - W'_1 = (\delta - \delta') F_K(x).$$

The two final verification equations give

$$Y^\delta = T g^{W_1}, \quad Y^{\delta'} = T g^{W'_1}.$$

Dividing them yields

$$Y^{\delta - \delta'} = g^{W_1 - W'_1} = g^{(\delta - \delta') F_K(x)}.$$

Since q is prime and $\delta - \delta' \neq 0$, this implies

$$Y = g^{F_K(x)}.$$

Thus every accepting proof is bound to the unique output $g^{F_K(x)}$, except for the losses stated above. This is precisely the public-verification condition in Definition 4, even against an adversary given the public-variant leakage (pk, sk) .

Lemma 17 (Proof simulatability). *The simulator in the construction produces accepting proofs for arbitrary $Y \in \mathbb{G}$. Moreover, when $Y = g^{F_K(x)}$, the simulated proof distribution is identical to the honest proof distribution, up to the standard random-oracle programming step.*

Proof. The commitment equation holds by construction:

$$C_\alpha^\delta C_\theta = C_\alpha^\delta \cdot \text{Com}(\vec{\theta}_1; \rho_1) / C_\alpha^\delta = \text{Com}(\vec{\theta}_1; \rho_1).$$

The final verification equation holds because the simulator defines $T = Y^\delta / g^{W_1}$.

Assume now that $Y = g^{F_K(x)}$. In an honest proof, before the Fiat–Shamir challenge is fixed, the values $\vec{\theta}_0$ and ρ_θ are uniform. After conditioning on a challenge δ , the response values

$$\vec{\theta}_1 = \vec{\alpha}\delta + \vec{\theta}_0, \quad \rho_1 = \rho_\alpha\delta + \rho_\theta$$

are uniform over $\mathbb{F}^d$ and $\mathbb{F}$, respectively, and

$$C_\theta = \text{Com}(\vec{\theta}_1; \rho_1) / C_\alpha^\delta.$$

HSS correctness gives $T = g^{-W_0} = Y^\delta / g^{W_1}$. These are exactly the simulator’s assignments, except that the simulator chooses δ first and then programs the random oracle.

Theorem 12 (Pseudorandomness). *Assume that F is a secure PRF, that the HSS public view is simulatable without the PRF key for the verifier-side authenticated-evaluation queries arising in simulated proofs, that Pedersen commitments are perfectly hiding, and that H_{FS} is programmable. Then the public eVRF satisfies pseudorandomness with simulated proofs in the sense of Definition 4.*

Proof. Hybrid Hyb_0 is the real pseudorandomness oracle, returning honest pairs (Y_x, π_x) with

$$Y_x = g^{F_K(x)}.$$

Hybrid Hyb_1 replaces every honest proof by a simulated proof for the same group element Y_x . By Lemma 17, this changes the distribution only by the random-oracle programming loss.

Hybrid Hyb_2 replaces the HSS public view $(\text{pp}_H, ek_1, \kappa_1)$ by the simulator guaranteed by public-view privacy. The commitment C_α is perfectly hiding, so it can be simulated as a commitment to zero. The adversary’s view is computationally indistinguishable by the public-view privacy assumption.

Hybrid Hyb_3 replaces the PRF outputs $F_K(x)$ by independent uniform elements $r_x \in \mathbb{Z}_q$, using PRF security. Consequently the group elements become

$$Y_x = g^{r_x},$$

which are independent uniform elements of G . The proofs are already produced by the simulator. This is exactly the ideal pseudorandomness oracle of Definition 4.

Theorem 13 (Security of the public eVRF). *Under the hypotheses of Theorems 10 to 12, the construction $\Pi_{\text{pub}}[F, \text{OHSS}]$ is a publicly verifiable exponent VRF in the sense of Definition 4, in the programmable random-oracle model.*

Proof. Correctness is Theorem 10. Public verifiability is Theorem 11. Pseudorandomness with simulated proofs is Theorem 12. These are the three properties required by Definition 4 for $\tau = \text{pub}$.

H Details on Experimental Setup

The code implements the final asymmetric key format. The sender key contains $\varphi(N)$ and uses it only for sender-side trapdoor reduction of exponents of the public N -th residue g . The receiver key never contains $\varphi(N)$ or $N\varphi(N)$. Receiver evaluation combines public values $\alpha_i = 1 + Nq_i$ by exact quotient products

$$\prod_{i: x_i=1} \alpha_i = 1 + NQ_x$$

and evaluates

$$h_{s_x}^{1+NQ_x} = h_{s_x} (h_{s_x}^N)^{Q_x} \bmod N^2.$$

Thus the receiver follows the construction’s public evaluation rule without being given the trapdoor. Receiver-side precomputation is restricted to public material: $h_s^N \bmod N^2$, shallow RNS window tables for these bases, and an RNS-encoded copy of h_s for the final multiplication. The sender, in contrast, uses a positional RNS fixed-base table for the single hot base g , because its exponents are reduced modulo $\varphi(N)$.

Table 4. Fixed-width wire sizes for the RNS-only PCF-COT artifact with $|N| = 2048$, $n = 128$, and batch size $B = 256$. Local RNS and subset-product precomputation tables are deterministic accelerators and are not transmitted.

Artifact	KiB
Receiver key k_0	129.30
Sender key k_1	65.78
Key pair	195.08
Receiver COT output	0.25
Sender COT output	0.50
COT output pair	0.75
Receiver COT outputs, B samples	64.25
Sender COT outputs, B samples	128.00
COT output pairs, B samples	192.25
Receiver $\mathbb{Z}_N$ -VOLE output	0.50
Sender $\mathbb{Z}_N$ -VOLE output	0.50
$\mathbb{Z}_N$ -VOLE output pair	1.00

Implementation optimisations. The implementation uses RNS Montgomery multiplication throughout the PCF exponentiation path, with AVX-512 IFMA 52-bit multiply-add instructions when the `avx512` feature is enabled. Key generation parallelises the independent construction of the $(h_s)_{s=0}^n$ table and the derived public bases $(h_s^N)_{s=0}^n$. Online batch evaluation parallelises independent exponentiations across Rayon workers and then applies a single batch inversion for all DDLog_N computations in the batch. Sender evaluation uses a windowed subset-product table for the reduced product $a_0 \prod_{i: x_i=1} a_i \bmod \varphi(N)$ and a positional fixed-base RNS table for g . Receiver evaluation uses an exact quotient subset-product table for the public product of the α_i 's, shallow RNS window tables for h_s^N , and keeps the terminal multiplication by h_s inside the Montgomery domain. The powers-of-two $\mathbb{Z}_N$ -VOLE aggregation is evaluated locally by Horner form, so aggregation after the underlying PCF-COT samples consists only of modular doublings and additions.

We assert that the implementation is not intended as a replacement for the fastest constrained-Naor-Reingold OT-only PCFs. Those constructions operate in prime-order groups and are optimized for random OT; their reported public-key PCF throughput is substantially higher for raw OT evaluation [BCM⁺24]. Our construction instead targets Paillier/DDLog arithmetic correlations: a hidden additive offset in the subgroup $\langle 1 + N \rangle$ is selected by the constrained Naor-Reingold table and then extracted by DDLog. We argue that the right comparison is therefore not “fastest OT”, but “number-theoretic PCF for fixed-offset COT and arithmetic VOLE”.

LPN/LWR-based PCFs have different asymptotic and concrete tradeoffs: they can be much faster in symmetric-key evaluation and support broader classes of correlations, but their concrete key-generation and key-size profiles depend heavily on the chosen LPN/LWR parameters and on the target correlation class [BCG⁺22, BCM⁺25, HRK25, HR25].