Beyond Quadratic: Unlocking Pseudorandomness with Quartic Character*

Mriganka Dey[1,2], Sampa Dey[3], Sampurna Pal[4], Subhabrata Samajder[2], Rana Barua[2]

[1] Ramakrishna Mission Vivekananda Educational and Research Institute, India
mriganka.dey.86@tcgcrest.org
[2] Institute for Advancing Intelligence (IAI), TCG CREST, India
subhabrata.samajder@tcgcrest.org, rana.barua@tcgcrest.org
[3] Statistic and Mathematics Unit, Indian Statistical Institute, India
sampa.math@gmail.com
[4] Department of Mathematics and Statistics, IIT Kanpur, India
sampu.andul@gmail.com

Abstract. We study pseudorandomness arising from quartic Dirichlet characters and obtain results that connect analytic and cryptographic perspectives. From an analytic perspective, and following the framework of Mauduit and Sárközy, we define a Boolean function ψ_π from the quartic character χ_π modulo a Gaussian prime π and analyze sequence $E_{p-1} = (\psi_\pi(1), \ldots, \psi_\pi(p-1))$ for $p = \pi\bar{\pi} \equiv 1 \bmod 4$. Using classical character-sum bounds (Pólya-Vinogradov and refinements of Mauduit-Sárközy and Oon), we show that E_{p-1} satisfies $W(E_{p-1}) = O(\sqrt{p}\log p)$ and $C_\mu(E_{p-1}) \leq 2^{\frac{\mu}{2}+1}\mu\sqrt{p}\log p$, which imply strong pseudorandomness for small μ. From the cryptographic side, we resolve an open question posed by Damgård by proving that quartic characters yield the secure pseudorandom generators and weak pseudorandom functions (wPRFs). Adapting similar techniques of Corrigan-Gibbs and Wu, we have shown that distinguishing quartic wPRF implies solving quadratic residuosity, via a chain of polynomial-time reductions. Our results show that under the Quadratic Residuosity Assumption, the quartic character also yields cryptographically secure wPRFs whose one-wayness was assumed in the construction of Quartapus signature scheme by Brier et al. and the post-quantum secure signature scheme PorcRoast$_4$ by Beullens et al. that are more efficient and secure than legacy schemes based on the Legendre character.

Keywords: Quadratic Residuosity · Dirichlet characters · Pseudorandom measures · Pseudorandom generators · Weak pseudorandom functions

1 Introduction

Solomonov and Kolmogorov initiated the systematic study of computational randomness in the 1960s [37,59,60]. It was rediscovered by Chaitin in the early

* This article is the full version of the paper that appeared at CRYPTO 2026.

1970s [16]. They defined *Kolmogorov complexity* as a measure of randomness in a given string. Informally, given a string s, the Kolmogorov complexity is defined as the length of the shortest computer program that produces the string s as output. Knuth popularized the study of pseudorandom sequences in his book *The Art of Computer Programming: Volume 2* [36], where he introduced a broad range of statistical tests (e.g., equidistribution, serial correlations, spectral tests) for evaluating the quality of pseudorandom generators. While Knuth's treatment was largely algorithmic and experimental, he stressed that pseudorandomness is an inherently multifaceted notion that *cannot be captured by a single universal definition*. Building on this insight, Mauduit and Sárközy [41] initiated a rigorous number-theoretic framework for pseudorandom binary sequences, introducing analytic measures such as the *well-distribution measure* and the *correlation measure of order* μ. Their results demonstrated that sequences derived from the Legendre symbol possess strong pseudorandom properties. Subsequent work extended these ideas to more general Dirichlet characters [50]. A natural question that arises is whether there exist functions beyond Dirichlet characters that can generate pseudorandom sequences in the rigorous sense defined by Mauduit and Sárközy. We answer this question in the affirmative by introducing a function ψ_π derived from quartic characters and show that it induces pseudorandom sequences in the sense of Mauduit and Sárközy, by bounding both the well-distribution and correlation measures of order μ.

It is important to note that the above approach of Kolmogorov and others is "ontological" in nature and it can be shown that the Kolmogorov complexity is uncomputable. Thereby making it inherently dependent on heuristic approaches. This approach considers "pseudo-random generators" as programs that produce the bit-sequences and "passes" *some specific* statistical tests. Thereby making it quite arbitrary and application specific. Consequently, an alternative approach to the study of pseudo-randomness was initiated in the early 1980s by a series of works starting with Blum and Micali [7], Yao [62], Goldreich et al [27] and Levin [40]. Their approach, in contrast, was "behavioral" in nature as it considers the ability of the observer to distinguish the given sequence from a uniformly random chosen string of the same length. This approach is more rigorous in nature as it models the observer as a probabilistic polynomial time (PPT) Turing machine in the length of the given string s. The main result in this regard states that one-way function exists if and only if pseudo-random generators (PRGs) exist [26,31] which is a culmination of a series of work starting with Yao [62], Levin [40] and Goldreich et al. [28]. But the constructions of provable PRGs (or one-way functions) are still an open problem.

In the cryptographic context, in 1988 Damgård [20] put forward an influential conjecture that short blocks of consecutive Legendre symbols, computed modulo a hidden prime, exhibit pseudorandom behavior. It remained unresolved for more than three decades, until recently, Corrigan-Gibbs and Wu [18] settled the problem by proving that a natural generalization of Damgård's pseudo-random generator is secure under the Quadratic Residuosity assumption. They further showed that this idea behind the generalized PRG also produces a

weak pseudorandom function (wPRF). In the same work, Damgård [20] further emphasized the broader open problem of establishing pseudorandomness for character sequences extending beyond the Legendre symbol from a cryptographic viewpoint. In this paper, we resolve this question by showing that a quartic character, which is also a Dirichlet character, yields a secure pseudorandom generator and a weak pseudorandom function under the Quadratic Residuosity assumption.

In the context of cryptographic applications, Landrock [39] observed that high power residue characters provide a "higher" level of security and using them, he constructed a verifiable computational delegation scheme. Later, this observation of Landrock was used by Monnerat and Vaudenay to construct undeniable signature schemes called MOVA [47,46]. Recently, Beullens et al. [6] gave two post-quantum signature schemes called LegRoast and $\mathsf{ProcRoast}_k$ based on the quadratic characters and the kth higher power residue character, respectively. They argued that $\mathsf{PorcRoast}_k$ is more efficient than LegRoast as the underlying PRF, which generates $\log_2 k$-bits at a time, is more efficient than the underlying PRF of LegRoast. The authors also observed that both these constructions are more efficient than those relying on the security of AES and the less scrutinized block cipher LowMC [55]. Earlier, Grassi et al. [30] had observed that PRF constructions based on quadratic (or high power residue) symbols are multiparty computation (MPC) friendly as they have low circuit size, are highly efficient, and parallelizable. Therefore, investigating the pseudorandomness behaviour of quadric characters not only has theoretical but also has significant practical importance.

Our results thus highlight two distinct aspects of pseudorandomness. On the one hand, the function ψ_π induces sequences that are pseudorandom in the statistical sense of Mauduit and Sárközy, as witnessed by small well-distribution and correlation measures of order μ. On the other hand, the quartic character itself gives rise to pseudorandom generators and weak pseudorandom functions that are secure in the cryptographic sense of Damgård and Corrigan-Gibbs & Wu, under the Quadratic Residuosity Assumption. Unlike the quadratic character, which outputs a single bit per evaluation, the quartic character produces two bits while preserving unpredictability under the same assumption. This higher output entropy allows constructions to use fewer PRF evaluations to reach a given security level. As a result, signature schemes built like $\mathsf{PorcRoast}_4$ [6] and Quartapus [8], which are based on quartic character achieves improved concrete security and efficiency over legacy quadratic character based schemes.

1.1 Related Works

In this article, we study the Quartic-symbol-based cryptographic constructions where the underlying prime modulus p (therefore its corresponding factorization) is *secret*. This is a generalization of the Legendre symbol based construction to Dirichlet characters. Over the years, the study of the *statistical properties* of Legendre sequences has received quite a lot of attention, especially from number theorists [1,11,21,32,51]. Davenport in [21] established that for any prime p and

any binary sequence $\boldsymbol{y} \in \{-1,1\}^\ell$ of length $\ell \leq 9$, the number of solutions $x \in \mathbb{Z}_p$ such that the vector $\boldsymbol{y}$ equals to the ℓ consecutive Lengendre symbols starting from $x+1,\ldots,x+\ell$, i.e., $\boldsymbol{y} = \left(\left(\frac{x+1}{p}\right),\ldots,\left(\frac{x+\ell}{p}\right)\right)$, lies in the range $\frac{p}{2^\ell} \pm \ell \cdot O(p^\epsilon)$ for some $\epsilon \geq 1/2$. Subsequently, Peralta [51], using the Weil bound [61], extended this result to arbitrary ℓ and showed that the number of such x lies within the range $\frac{p}{2^\ell} \pm \ell \cdot O(\sqrt{p})$.

Building on the works of Davenport [21] and Peralta [51], it is known that for every fixed $\ell \in \mathbb{N}$, the distribution of ℓ consecutive Legendre symbols converges in distribution to the uniform measure on $\{-1,1\}^\ell$ as the prime modulus $p \to \infty$. Thus providing evidence that Legendre sequences are uniformly distributed. Inspired by the works of [52,11,23,9,2,3] which provided evidence of pseudorandom properties of Legendre symbols, Damgård in 1988 [20] initiated the study of cryptographic pseudorandom generators based on sequences constructed from characters. He conjectured that, given a *secret* random prime p, a secret random element $x \in \mathbb{Z}_p^\star$, and any sequence length $\ell = \mathsf{poly}(\log p)$, the sequence $\left((\frac{x+1}{p}),\ldots,(\frac{x+\ell}{p})\right)$ is pseudorandom. This remained an open problem for more than three decades. In addition to this question, he also asked the following couple of questions. First, is the complexity of predicting Legendre sequences related to other number-theoretic problems? Second, is it possible to use characters other than the quadratic one to construct pseudorandom generators? Recently, among these three questions, the first two were answered affirmatively by Corrigan-Gibbs and Wu [18], showing that a natural generalization of the Damgård's generator is *secure* under the Quadratic Residuosity Assumption.

In their variant, the generator outputs the sequence $\left((\frac{x+a_1}{p}),\ldots,(\frac{x+a_\ell}{p})\right)$, where the public offsets $\boldsymbol{a} = (a_1,\ldots,a_\ell)$ are chosen uniformly at random with bit-length $O(\log p)$. They further showed that the keyed function $L((p,x),a) := (\frac{x+a}{p})$ is a wPRF, where (p,x) denotes the secret key and $a \in \{1,\ldots,2^{3\lambda}\}$ such that λ is the security parameter. However, to the best of our knowledge, to date Damgård's final question of constructing pseudorandom generators using characters other than the quadratic one remains an open problem.

Landrock in [39] proposed a verifiable computational delegation scheme based on his observation that high power residue provides a "higher" security level compared to the legacy quadratic residue symbols. Monnerat and Vaudenay [47,46] later used this idea of Landrock to propose an undeniable signature scheme called MOVA based on the quartic residue character. One of the attractive features of the Legendre function L_p is its small arithmetic circuit size over $\mathbb{Z}_p$ due to the *Euler's criterion*: $L_p(x,a) = \left(\frac{x+a}{p}\right) \equiv (x+a)^{\frac{p-1}{2}} \bmod p$. Thereby making it concretely efficient to evaluate inside an MPC [4,6,24,30] or as zero-knowledge proofs [47,46,6,8]. Recently, Beullens et al. [6] proposed a post-quantum secure signature called Legroast based on the one-wayness of the Legendre symbols and a generalized family of post-quantum secure signatures called PorcRoast$_k$ based on the one-wayness of the kth high power residues. Thus, as a consequence of the result Corrigan-Gibbs and Wu [18] security of LegRoast is now based on the

standard quadratic residuosity assumption. However, formalising the security notion of $\mathsf{PorcRoast}_4$ remains an open problem. It is important to mention here that Beullens et al. had actually shown that PorcRoast is secure under the assumption that computing an approximate witness of the public key is *hard*. The evidence of which was provided using the Weil Bound [61] for (Dirichlet) character sums. It would be interesting to build cryptographic schemes based on non-Dirichlet characters and prove their security based on similar bounds.

Most recently, a series of works [4,5,6,24,30,34,57] has looked into the question whether these Legendre-symbol-based constructions remain secure even when the prime modulus p is *public*. Specifically, these works conjecture that, for a public prime p, a secret key $x \in \mathbb{Z}_p^\star$ and an input $a \in \mathbb{Z}_p^\star$, the function $L_p(x,a) := \left(\frac{x+a}{p}\right)$ is a (strong) pseudorandom function, where $\left(\frac{\cdot}{p}\right)$ denotes the Legendre symbol with prime modulus p. It is therefore a worthwhile future direction to prove pseudorandomness of L_p under a standard number-theoretic assumption, where p is public.

In [35], Khovratovich presented an $O(\sqrt{p})$ key-recovery attack on the public-prime function L_p, contradicting the previously held belief that the best possible attack requires $O(p)$ time, i.e., the same complexity as that of the brute force attack. Subsequently, key-recovery attacks of Legendre PRFs in the quantum setting have also received attention of late. In [25], Frixons and Schrottenloher gave two key-recovery attacks on shifted Legendre symbols with unknown shifts with complexity better that the Grover's algorithm, whose time complexity is $O(\sqrt{p})$, query complexity is $\mathsf{poly}(\log p)$ and required $\mathsf{poly}(\log p)$ many (logical) qubits. Their first algorithm is a variant of the table-based collision algorithm with a time complexity of $O(p^{3/7})$, query complexity $O(p^{3/14})$, and requires $p^{3/14}$ classical bits plus $\mathsf{poly}(\log p)$ (logical) qubits. The second attack is an offline variant of Kuperberg's abelian hidden shift algorithm [38] and has a time complexity of $O\left(\left(2^{\frac{4}{3}\sqrt{2\log_2 3}\cdot\sqrt{\log_2 p^{1/3}}}\right)p^{1/3}\right)$, query complexity $O(p^{1/3})$ and requires $O\left(2^{\sqrt{2\log_2 3}\cdot\sqrt{\log_2 p^{1/3}}}\right)$ (logical) qubits. Note that under the assumption that the adversary can make superposition queries to the pseudorandom function, there is also a quantum polynomial-time attack [19]. However, in the classical setting, there are still no known polynomial-time (or even sub-exponential-time) attacks against the Legendre pseudorandom function.

On the other hand Aladov and other [1,11,21,32,51] have investigated the statistical properties of Legendre symbols. Mauduit et al. in a series of works [41,42,12,13,43,44,14] extended the study to various pseudorandomness measures for binary sequences and established rigorous tests of pseudorandomness for several natural constructions. Their approach is based on the principle that a sequence E_N can be regarded as a *good* pseudorandom sequence provided its *well-distribution measure* and its *correlation measure of order* μ are sufficiently small, at least for small values of μ. For instance, Mauduit and Sárközy [41] introduced the *combined pseudorandomness measure*, which merges the equidistribution measure with correlation measure of order μ along arithmetic progressions. They proved (see [41, Theorem 1]) that sequences generated by the Legendre

symbol satisfy strong pseudorandomness guarantees in this framework, showing that the combined measure is upper bounded by $O(\mu\sqrt{N}\log N)$. Later, Oon [50] generalized these results to arbitrary Dirichlet characters. However, to the best of our knowledge, the question of the existence of non-Dirichlet characters exhibiting a strong pseudorandomness guarantee is still an open problem.

1.2 Our Contribution

The contribution of this paper is twofold. Motivated by the works of Casssigne et al. [41,14], in the first part of this work, we introduce a new function $\psi_\pi : \mathbb{Z} \to \{-1, 1\}$ defined from the quartic Dirichlet character $\chi_\pi(\cdot)$ modulo a Gaussian prime $\pi \in \mathbb{Z}[i]$ (which is neither a divisor of 2 nor a rational prime). Specifically, we set $\psi_\pi(n) := \mathfrak{Re}(\chi_\pi(n)) + \mathfrak{Im}(\chi_\pi(n))$, where $\mathfrak{Re}(\chi_\pi(n))$ and $\mathfrak{Im}(\chi_\pi(n))$ denote the real and imaginary parts of $\chi_\pi(n)$, respectively. It is important to note here that the function $\psi_\pi(n)$ is *not* a Dirichlet character as it destroys the algebraic structure that is present in the quartic Dirichlet character $\chi_\pi(n)$. Next, we consider the sequence $E_{p-1} = (\psi_\pi(1), \psi_\pi(2), \ldots, \psi_\pi(p-1))$ and prove that it satisfies the pseudorandomness criteria of Mauduit and Sárközy by showing that both the well-distribution measure $W(E_{p-1})$ and the correlation measures $C_\mu(E_{p-1})$ are small, where p is an integer prime such that $p = \pi\bar{\pi}$. In this sense, the function ψ_π *induces pseudorandomness* for the associated binary sequence.

In the second part of this work, we extend the cryptographic results of Corrigan-Gibbs and Wu [18] from quadratic to quartic characters. Concretely, we show that sequences generated directly from the quartic character $\chi_\pi(\cdot)$ are *cryptographically pseudorandom* in the sense of Damgård [20], and Corrigan-Gibbs and Wu [18]. In particular, we consider the generator that outputs $F(\chi_\pi(x+a_1)),\ F(\chi_\pi(x+a_2)),\ \ldots,\ F(\chi_\pi(x+a_\ell))$, where the encoding function $F : \{\pm 1, \pm i\} \to \{0,1\}^2$ is defined by $1 \mapsto 00, i \mapsto 01, -1 \mapsto 10$ and $-i \mapsto 11$. Note that the construction does not depend on the choice of the encoding function F, and any of the possible 24 bijective functions can be taken. We prove that if the offsets $\boldsymbol{a} = (a_1, a_2, \ldots, a_\ell)$ are chosen uniformly at random with bit-length $O(\log p)$, then no *efficient* adversary can distinguish the 2ℓ-bit string outputs of this generator from those of a truly random 2ℓ-bits string under the Quadratic Residuosity Assumption. Moreover, we show that the keyed function $Q((p,x),a) := F(\chi_\pi(x+a))$, is a *weak pseudorandom function*, where the secret key consists of a secret prime $p \in \mathbb{Z}$, a Gaussian prime $\pi \in \mathbb{Z}[i]$ such that $p = \pi\bar{\pi}$ with $\bar{\pi}$ denoting the complex conjugate of π.

To summarize, our contributions establish two different perspectives on the pseudorandomness. First, we show that the function ψ_π induces pseudorandom sequences in the analytic sense of Mauduit and Sárközy, namely, by proving that the associated well-distribution and correlation measures are small. Second, we demonstrate that the quartic character itself gives rise to cryptographically secure pseudorandom generators, in the standard indistinguishability-based sense.

1.3 Why Quadratic Character Is Not Enough?

In this section, we highlight the practical significance of the results presented in this paper. We provide evidence that building cryptographic constructions based on the quartic residuosity assumption, as opposed to the legacy quadratic residuosity assumption gives both improved security and efficiency.

Quadratic pseudorandomness modulo an integer $N = pq$ (where $p \equiv q \equiv 1 \bmod 4$) provides security only as long as the factorization of N remains hidden. Landrock in [39] observed that, if factorization in $\mathbb{Z}[i]$ is computationally hard, then at least one of the following problems must also be hard: (i) given N, finding a Fermat pair of N, or (ii) given one Fermat pair of N, finding a non-equivalent Fermat pair. According to the authors, this provides a higher level of security as any owner having the knowledge of p and q can delegate signing capability to two executives, E_1 and E_2 by computing the two *reduced Fermat pairs* [39] (a, b) and (c, d) with $a^2+b^2 = c^2+d^2 = N$ and distribute (a, b) to E_1 and (c, d) to E_2. This ensures that the Owner, knowing p and q, can impersonate either executive if necessary, whereas neither executive can impersonate the Owner. Moreover, since the two pairs are non-equivalent, one executive cannot impersonate the other. However, if E_1 and E_2 colludes then they can recover the factorization (p, q) of N. This fundamental observation forms the core idea behind constructing a verifiable computational delegation scheme [39] that achieves a genuine two-level security guarantee that is inherently unattainable using only quadratic characters. The idea was later exploited in signature schemes like MOVA [47,46].

In 2020, Beullens et al. [6] proposed LegRoast and PorcRoast$_k$, two post-quantum signature schemes whose security relies on the one-wayness of the Legendre PRF and the quartic power residue PRF, respectively. Both schemes build upon the MPC-in-the-head paradigm to construct efficient zero-knowledge proofs, which are then transformed into signature schemes via the Fiat–Shamir transform. By extending the Legendre PRF to quartic characters, PorcRoast$_4$ achieves noticeably smaller signatures and faster signing while retaining tight security reductions in the classical random oracle model. This is because PorcRoast$_4$ requires roughly half as many PRF outputs to achieve the same entropy in the public key compared to LegRoast = PorcRoast$_2$. Thereby, directly lowering the number of symbols involved in the underlying zero-knowledge proof and as a result allows the verifier to check fewer residue symbols per execution and to use fewer parallel repetitions of the identification protocol, leading to a noticeable reduction in signature size compared to LegRoast.

Both LegRoast and PorcRoast$_4$ benefit from the relatively cheap computation of the corresponding residue symbols. This is due to Euler's criterion in case of the Legendre symbols and a similar result for the quartic symbol $\chi_\pi(x)$, which is $\chi_\pi(x) \equiv x^{(p-1)/4} \bmod \pi$. Also, these computations can be highly parallelized, as the computation for one PRF evaluation is independent of the other. However, in PorcRoast$_4$ computations one needs to perform the modular exponentiation over $\mathbb{Z}[i]$ modulo π and not $\mathbb{Z}_p$. However, using the fact that $x^{(p-1)/4} \bmod \pi \equiv (x^{(p-1)/4} \bmod p) \bmod \pi$ one reduces the number of modular multiplications over $\mathbb{Z}[i]$ from around $\log_2\{(p-1)/4\}$ to $\log_2\{(p-1)/4\}$ many

modular integer multiplications plus one modular multiplication over $\mathbb{Z}[i]$ for every two bits of output. These observations make $\mathsf{PorcRoast}_4$ more efficient compare to LegRoast. Details of this can be found in Appendix B.

1.4 Technical Overview

We first analyze the *statistical* pseudorandomness of the sequence $E_{p-1} = (\psi_\pi(1), \ldots, \psi_\pi(p-1))$, by establishing upper bounds on its *well-distribution measure* $W(E_{p-1})$ and *correlation measures* $C_\mu(E_{p-1})$. Our analysis relies on two classical tools: the *Pólya–Vinogradov inequality* [33] for character sums and a refined character sum estimate of Mauduit–Sárközy [41], later strengthened by Oon [50] The details of our construction are presented in Section 3.

Bounding the well-distribution measure. The key step is to express $\psi_\pi(n)$ as a linear combination of $\chi_\pi(n)$ and its conjugate $\bar{\chi}_\pi(n)$:

$$\psi_\pi(n) \;=\; \left(\tfrac{1-i}{2}\right) \cdot \chi_\pi(n) + \left(\tfrac{1+i}{2}\right) \cdot \bar{\chi}_\pi(n).$$

This reduces the analysis of $W(E_{p-1})$ to bounding short character sums of the form $\sum_{j=0}^{t-1} \chi_\pi(d + jb)$. By exploiting multiplicative invariance and applying the Pólya–Vinogradov inequality, we obtain the following bound

$$W(E_{p-1}) = O(\sqrt{p} \log p).$$

Bounding the correlation measure. The correlation measure involves products of shifted values of ψ_π, which expand into bilinear forms involving χ_π and $\bar{\chi}_\pi$. Using the fact that $\bar{\chi}_\pi = \chi_\pi^3$ (since χ_π has order 4), these products reduce to character sums of polynomials with controlled degree. Applying Oon's modification of the Mauduit–Sárközy estimate yields a nontrivial bound for each such sum. Careful bookkeeping of all subsets in the expansion shows that

$$C_\mu(E_{p-1}) \;\le\; 2^{\frac{\mu}{2}+1} \cdot \mu \cdot \sqrt{p} \log p.$$

Together, these results demonstrate that the sequence induced by ψ_π achieves strong statistical pseudorandomness guarantees in Mauduit–Sárközy framework, particularly when the correlation order μ is small (e.g., $\mu \ll \log\log p$ as suggested in prior works [14]). This analytic pseudorandomness serves as the foundation for the first part of our contributions. The details are given in Section 3.

In the second part of our work, we look at the cryptographic side of the problem. Note that in the first part, no assumption is made on the computational power of the adversary. But in the cryptographic setting we assume that the adversary has PPT (in $\log_2 p$) computation power. We establish the security of our quartic wPRF. At a high level, we prove that if there exists a PPT adversary that can distinguish, with non-negligible probability, between outputs of our quartic pseudorandom function and a truly random function, then one can efficiently solve the quadratic residuosity problem. Since the quadratic residuosity problem

is widely believed to be computationally hard, this implies the security of our construction.

Formally, we provide a polynomial-time reduction from the quadratic residuosity problem $\mathsf{QR}_{\mathsf{RSA}}$ to the security game $\mathcal{Q}^{\mathsf{wPRF}}$. In other words, we show $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$, where $\mathsf{QR}_{\mathsf{RSA}}$ is the quadratic residuosity problem with respect to an RSA modulus $N = pq$ (with $p \equiv q \equiv 5 \bmod 8$), and $\mathcal{Q}^{\mathsf{wPRF}}$ denotes the wPRF security game with respect to our quartic pseudorandom function keyed by such a modulus. Our proof strategy adapts the techniques of Corrigan-Gibbs and Wu [18], but departs from their setting of Blum primes ($p \equiv q \equiv 3 \bmod 4$) to the quartic setting. Our reduction can be broadly divided into the following three steps.

Step 1: From quadratic residuosity for RSA modulus to quadratic residuosity for Quartic modulus. We first show that $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{Q}}$, where QR_{Q} is the *quadratic residuosity problem* for Quartic modulus $N = pq$ with $p \equiv q \equiv 5 \bmod 8$. Corrigan-Gibbs and Wu [18] established this reduction for Blum integers ($p \equiv q \equiv 3 \bmod 4$); we extend and adapt their proof to our setting. Note that this part of the reduction actually holds for $p \equiv q \equiv 1 \bmod 4$. However, for reasons to be explained below, the restriction of $p \equiv q \equiv 5 \bmod 8$ will be needed later. The details of this part of the reduction are presented in Section 4.1.

Step 2: From quadratic residuosity for Quartic modulus to decisional quartic residuosity. Next, we show that QR_{Q} reduces to the decisional *quartic* residuosity problem $\mathsf{QuarticRD}_{\mathsf{Q}}$, i.e., $\mathsf{QR}_{\mathsf{Q}} \leq_p \mathsf{QuarticRD}_{\mathsf{Q}}$. Like Step 1 above, the reduction also holds for $p \equiv q \equiv 1 \bmod 4$. The details of this part of the reduction is presented in Section 4.2.

Step 3: From decisional quartic residuosity to wPRFs. Finally, we establish that the decisional quartic residuosity problem reduces to breaking our quartic wPRF, i.e., $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$, where $\mathcal{Q}^{\mathsf{wPRF}}$ denotes the family of wPRFs defined by the quartic residuosity symbol. The reduction from $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$ actually goes through the intermediate $\mathsf{Quartic}_\ell$ game, i.e., we show that $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathsf{Quartic}_\ell \leq_p \mathcal{Q}^{\mathsf{wPRF}}$. Now, the proof of $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathsf{Quartic}_\ell$ involves a resampling algorithm which on input an x sampled uniformly at random from $\mathbb{Z}_N^\star$ and it's corresponding quartic character, outputs an r (different from x) sampled uniformly at random from $\mathbb{Z}_N^\star$ and the value of it's corresponding quartic character. This resampling algorithm requires sampling uniformly from $\mathbb{J}_N \setminus \mathbb{QR}_N$ which in turn requires the knowledge of some $\alpha \in \mathbb{J}_N \setminus \mathbb{QR}_N$. For $p \equiv q \equiv 5 \bmod 8$ it is known that $\alpha = 2 \in \mathbb{J}_N \setminus \mathbb{QR}_N$. Since no appropriate α is known for the case $p \equiv q \equiv 1 \bmod 8$, our reduction necessarily focuses on primes satisfying $p \equiv q \equiv 5 \bmod 8$. We also note that this is the only step that requires this restriction. As such, if one can find such an $\alpha \in \mathbb{J}_N \setminus \mathbb{QR}_N$ for $p \equiv q \equiv 1 \bmod 8$, then this restriction can be removed. The details of this part of the reduction is presented in Section 4.3.

Overall reduction. Combining the above steps yields the following chain of reductions:

$$\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{SP}} \leq_p \mathsf{QR}_{\mathsf{Q}} \leq_p \mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathcal{Q}^{\mathsf{wPRF}}.$$

This establishes that any efficient (or PPT) adversary against our quartic wPRF would imply an efficient algorithm for solving the quadratic residuosity problem for the RSA modulus. Thereby proving the security of our construction under the quadratic residuosity assumption.

Additionally, given a prime $p \equiv 5 \bmod 8$, we define a function $f : \mathbb{Z}_p^\star \to \{-1,1\}^2$ as $f(n) = (\psi_\pi(n), \psi_{\bar{\pi}}(n))$, where $p = \pi\bar{\pi}$ and $\psi_\pi : \mathbb{Z}_p^\star \to \{-1,1\}$ with $\psi_\pi(n) = \mathfrak{Re}(\chi_\pi(n)) + \mathfrak{Im}(\chi_\pi(n))$. Subsequently, we prove that $\mathsf{QuarticRD}_{\mathsf{Q}}$ is polynomially equivalent to the problem of distinguishing this function f from random, i.e., $\mathsf{QuarticRD}_{\mathsf{Q}} \equiv_p \mathsf{fD}$. The details of this part of the reduction is presented at the end of Section 4.3.

2 Preliminaries

We use standard notation for sets of numbers: $\mathbb{N}$ denotes the natural numbers, $\mathbb{Z}$ the integers, $\mathbb{Q}$ the rationals, $\mathbb{R}$ the reals, and $\mathbb{C}$ the complex numbers. Standard notation $f(x) = O(g(x))$ is used to imply that $|f(x)| \leq C|g(x)|$ for some absolute constant C and large values of x.We also use $f(x) = o(g(x))$ to denote that the ratio $\frac{f(x)}{g(x)}$ goes to 0 as x goes to ∞. For a finite set S, we write $x \xleftarrow{\$} S$ to denote that x is sampled uniformly at random from S, and $\mathcal{U}(S)$ for the uniform distribution over S. An algorithm is called *efficient* if it runs in probabilistic polynomial time (PPT) in the length of its input. $\mathcal{A}(x) = y$ denotes that on an input x algorithm $\mathcal{A}$ outputs y. We use the notation $x := a$ to denote that x is defined to be a, whereas $x \leftarrow a$ denotes assignment. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is called *negligible* if $\mathsf{negl}(\lambda) = o(\lambda^{-c})$ for every constant $c > 0$. For $n \in \mathbb{N}$, $[n]$ denote the set $\{1, 2, \ldots, n\}$. For $m, n \in \mathbb{Z}$ with $m < n$, $[m, n]$ denote the set $\{m, m+1, \ldots, n\}$.

2.1 Basic Number-Theoretic Notation and Results

For a positive integer N, $\mathbb{Z}_N^\star$ denotes the multiplicative group of integers modulo N that are coprime to N. Define $\mathbb{J}_N := \{x \in \mathbb{Z}_N^\star \mid \left(\frac{x}{N}\right) = 1\}$ and $\mathbb{QR}_N := \{x^2 \mid x \in \mathbb{Z}_N^\star\} \subseteq \mathbb{J}_N$, where $\left(\frac{\cdot}{N}\right)$ denotes the Jacobi symbol. We write $d \mid N$ to denote d divides N, $a \equiv b \bmod N$ for modular congruences, and $\gcd(a, b)$ for greatest common divisors. Recall that a ***Dirichlet Character*** of modulus N is a group homomorphism $\chi : \mathbb{Z}_N^\star \to \mathbb{C}^\star$, i.e. for any $n_1, n_2 \in \mathbb{Z}_N^\star$, $\chi(n_1 n_2) = \chi(n_1)\chi(n_2)$. A character is called *principal* if it takes the value 1 on every element of $\mathbb{Z}_N^\star$. Characters are extended to functions $\chi : \mathbb{Z} \to \mathbb{C}$ by defining $\chi(a) = 0$ whenever $\gcd(a, N) \neq 1$. By $\mathbb{F}_p$ we denote a finite *field* of p many elements.

Next, we recall the following result due to Shoup [58], which gives the density of primes less than x and which are congruent to a mod d for some arbitrary a and d.

Theorem 1 ([58, Theorem 5.21]). *Let $a, d \in \mathbb{Z}$ such that $d > 0$, $\gcd(a, d) = 1$. Then*

$$\pi(x; d, a) \approx \frac{x}{\varphi(d) \log x} \quad \text{as } x \to \infty,$$

where $\pi(x; d, a)$ denotes the number of primes $p \leq x$ such that $p \equiv a \bmod d$, and φ is Euler's totient function.

Theorem 2 (Generalized Riemannian Hypothesis (GRH)). *For every Dirichlet character χ, all zeros of its associated L-function $L(\chi, s)$ in the critical strip $0 < \mathfrak{Re}(s) < 1$ satisfy $\mathfrak{Re}(s) = 1/2$, where $L(\chi, s) := \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$.*

2.2 Gaussian Integers

Denote by $\mathbb{Q}(i) := \{a + ib : a, b \in \mathbb{Q}\}$ the quadratic number field extension of the set of rational numbers $\mathbb{Q}$ using the irreducible polynomial $x^2 + 1$ over $\mathbb{Q}$, where $i = \sqrt{-1}$ is the imaginary number. Then the ring of integers in this quadratic number field $\mathbb{Q}(i)$ denoted by $\mathbb{Z}[i]$ is called the *Gaussian ring* or *Gaussian integers*. It is easy to check that the units of $\mathbb{Z}[i]$ are $\{1, -1, i, -i\}$. Let, $\|\cdot\|$ denote the norm of an element in $\mathbb{Z}[i]$, i.e., $\|a+ib\| := a^2+b^2$. It is also well known that the prime elements of the Gaussian integers $\mathbb{Z}[i]$ fall into three classes. First, up to multiplication by units, the element $1+i$ is the unique prime dividing 2. Second, for any integer prime $p \equiv 1 \bmod 4$, there exist integers a, b such that $p = a^2 + b^2$, and the elements $\pi = a + ib$ and its complex conjugate $\bar{\pi} = a-ib$ are Gaussian primes with $p = \pi\bar{\pi}$ upto multiplication by units and they are called *Gaussian primes*. Finally, every integer prime $q \equiv 3 \bmod 4$ remains prime in $\mathbb{Z}[i]$ and these primes are called *rational primes*.

Now, $\mathbb{Z}[i]$ is an *Euclidean Domain*, so it is also a *Unique Factorization Domain* (UFD). The algorithms we use for division and for computing the greatest common divisor (gcd) in $\mathbb{Z}[i]$ can be easily obtained (see Algorithms 3 and 4 in Appendix A.1 and A.2 respectively) from the classical treatments of Dummit and Foote [22, page 270-272] and Caviness and Collins [15]. It is also straightforward to use Algorithms 3 and 4 to develop an efficient modular exponentiation algorithm for complex numbers similar to the well-known Modular Square and Multiply algorithm [49].

For any Gaussian prime π, we consider the map, $\left(\frac{\alpha}{\pi}\right) : \mathbb{Z}[i] \to \{1, -1, i, -i\}$, satisfying $\left(\frac{\alpha}{\pi}\right) \equiv \alpha^{\frac{p-1}{4}} \bmod \pi$, where $p = \pi\bar{\pi}$. This function, denoted by $\chi_\pi(\cdot)$, is called the *quartic residue symbol*. By convention, if $\pi \mid \alpha$, then $\left(\frac{\alpha}{\pi}\right) = 0$. For any given Gaussian prime π corresponding to some integer prime $p \equiv 1 \bmod 4$ such that $\pi\bar{\pi} = p$ we restrict the function χ_π (and similarly $\chi_{\bar{\pi}}$) on $\mathbb{Z}$ to get a Dirichlet character of order 4.

Next, we state some facts which will be used later to define our constructions and prove our main results. The proofs of these facts are quite straightforward and they have been deferred to Appendix C.

Fact 1 *For an integer prime $p \equiv 1 \bmod 4$, the Dirichlet character of order 4, denoted by χ_π, is well defined, i.e., it remains invariant under different choices of the factorization of p in $\mathbb{Z}[i]$.*

Fact 2 *For $p \equiv 1 \bmod 4$ and $\chi_\pi(n)$ for some given $n \in \mathbb{Z}_p^\star$, $\left(\frac{n}{p}\right) = \chi_\pi(n)^2$, where $n \in \mathbb{Z}_p^\star$ and $p = \pi\bar{\pi}$.*

Fact 3 *Let $\bar{\chi}_\pi(\cdot)$ denote the complex conjugate of the quartic residue symbol modulo π, i.e., χ_π, then for every $n \in \mathbb{Z}$ we have $\bar{\chi}_\pi(n) = \chi_{\bar{\pi}}(n) = \left(\frac{n}{\bar{\pi}}\right)$, where $\pi \cdot \bar{\pi} = p$ and $p \equiv 1 \bmod 4$.*

Fact 4 *For an integer prime $p \equiv 1 \bmod 4$ and $n \in \mathbb{Z}$ we have the following $\left(\frac{n}{\bar{\pi}}\right) = \left(\frac{m}{\pi}\right)$, where m is the multiplicative inverse of n in $\mathbb{Z}_p$ and $p = \pi\bar{\pi}$.*

2.3 Measures of Statistical Pseudorandomness

Given a binary sequence $E_N = (a_1, a_2, \cdots, a_N) \in \{-1, 1\}^N$, to measure the statistical pseudorandomness, Mauduit and Sárközy [41] proposed the following Well distribution measure and correlation measure of order μ.

Definition 1. *(Well-distribution measure [41, page 370], [44, page 282]) For a binary sequence $E_N = (a_1, a_2, \cdots, a_N) \in \{-1, 1\}^N$, the **Well-distribution measure** of E_N is given by*

$$W(E_N) := \max_{d,b,t} \left| \sum_{j=0}^{t-1} a_{d+jb} \right|,$$

where maximum is taken over all possible sub-sequences whose indices are in an arithmetic progression of length t, starting index d, and common difference b such that $1 \leq d \leq d + (t-1)b \leq N$.

Definition 2. *(Correlation measure of order μ [41, page 370], [44, page 282]) For a binary sequence $E_N = (a_1, a_2, \cdots, a_N) \in \{-1, 1\}^N$, the **correlation measure of order** μ of E_N is given by*

$$C_\mu(E_N) := \max_{M,D} \left| \sum_{j=1}^{M} a_{j+d_1} a_{j+d_2} \ldots a_{j+d_\mu} \right|$$

for all $D = (d_1, d_2, ..., d_\mu) \in \mathbb{N}^\mu$, and $0 \leq d_1 < d_2 < ... < d_\mu \leq N - M$.

2.4 Standard Definitions of Cryptographic Pseudorandomness

In this section, we recall standard notions of cryptographic pseudorandomness. Since we will be working throughout with Quartic residue symbols, it will be convenient for us to define standard pseudorandomness notions with respect to values in $\{0, 1\}^2$.

Definition 3 (Pseudorandom Generator (PRG)). *Let $\mathcal{G} = \{\mathcal{G}_\lambda \mid \lambda \in \mathbb{N}\}$ be a family of collections of functions. For each security parameter $\lambda \in \mathbb{N}$, the set $\mathcal{G}_\lambda$ consists of efficiently computable, length-increasing functions $G : \mathcal{S}_G \to \left(\{0,1\}^2\right)^{\ell(\lambda)}$, where $\mathcal{S}_G$ denotes the seed space of G and $\ell(\lambda) > \lambda$. We assume that $\mathcal{S}_G$ is determined by G, and that there exists an efficient sampling algorithm that, given λ, outputs a seed $s \xleftarrow{\$} \mathcal{S}_G$. Let $\mathcal{A}$ denote an arbitrary probabilistic polynomial-time (PPT) algorithm.*

For $b \in \{0,1\}$, define

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(G,\sigma_b) = 1 : G \leftarrow \mathcal{G}_\lambda,\, r \leftarrow \mathcal{S}_G,\, \sigma_0 \leftarrow G(r),\, \sigma_1 \leftarrow \{\{0,1\}^2\}^{\ell(\lambda)}\right].$$

The advantage of $\mathcal{A}$ in distinguishing $\mathcal{G}$ from a uniformly random string of length $2\ell(\lambda)$ is defined as follows.

$$\mathsf{PRGAdv}[\mathcal{A},\mathcal{G}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

*We call the PRG $\mathcal{G}$ is **indistinguishable from random** or **secure** if for all PPT adversary $\mathcal{A}$ and for sufficiently large values of λ*

$$\mathsf{PRGAdv}[\mathcal{A},\mathcal{G}](\lambda) \;\leq\; \mathsf{negl}(\lambda).$$

Definition 4 (Weak Pseudorandom Function (wPRF)). *Let $\mathcal{F} = \{\mathcal{F}_\lambda \mid \lambda \in \mathbb{N}\}$ be a family of efficiently computable functions $F_\lambda : \mathcal{K}_\lambda \times \mathcal{D}_\lambda \to \{0,1\}^2$, where $\mathcal{K}_\lambda$ and $\mathcal{D}_\lambda$ denote the key space and domain space, respectively. Both $\mathcal{K}_\lambda$ and $\mathcal{D}_\lambda$ are assumed to be efficiently sampleable given λ.*

Let $\mathcal{A}$ be a PPT algorithm. For $b \in \{0,1\}$, define

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}^{\mathcal{O}_b}(1^\lambda) = 1 : k \xleftarrow{\$} \mathcal{K}_\lambda,\, f \xleftarrow{\$} \mathsf{Func}[\mathcal{D}_\lambda, \{0,1\}^2]\right],$$

where $\mathcal{A}^{\mathcal{O}_b}$ means that the adversary $\mathcal{A}$ is given an access to the oracle $\mathcal{O}_b$, $\mathsf{Func}[\mathsf{Dom},\mathsf{Range}]$ denotes the set of all functions from the domain Dom to the range Range, and the oracles $\mathcal{O}_0$ and $\mathcal{O}_1$ behave as follows.

- $\mathcal{O}_0$**:** *Sample $x \xleftarrow{\$} \mathcal{D}_\lambda$, return $(x, F_\lambda(k,x))$*
- $\mathcal{O}_1$**:** *Sample $x \xleftarrow{\$} \mathcal{D}_\lambda$, return $(x, f(x))$*

The wPRF advantage of $\mathcal{A}$ is then defined as follows.

$$\mathsf{wPRFAdv}[\mathcal{A},\mathcal{F}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

*We call the wPRF $\mathcal{F}$ is **indistinguihable from random** or **secure** if for all PPT adversary $\mathcal{A}$ and for sufficiently large values of λ*

$$\mathsf{wPRFAdv}[\mathcal{A},\mathcal{F}](\lambda) \;\leq\; \mathsf{negl}(\lambda).$$

2.5 Sampling Quadratic Residues and Non-Residues without Knowing the Factorization of N

In this section, we present how to efficiently sample from $\mathbb{J}_N, \mathbb{QR}_N$ and $\mathbb{J}_N \backslash \mathbb{QR}_N$ uniformly at random without knowing the factorization of N.

Sampling from $\mathbb{QR}_N$: To obtain a uniformly random quadratic residue modulo N, one samples $y \xleftarrow{\$} \mathbb{Z}_N^\star$ and outputs $x \leftarrow y^2 \bmod N$. By construction, $x \in \mathbb{QR}_N$, and the distribution of x is uniform over $\mathbb{QR}_N$.

Quartic integers and sampling from $\mathbb{J}_N \setminus \mathbb{QR}_N$: To sample an element $x \in \mathbb{J}_N \setminus \in \mathbb{QR}_N$, where $N = pq$ with $p \equiv q \equiv 5 \bmod 8$ (without knowing the value of p, q), we proceed as follows. First note that $2 \in \mathbb{QNR}_p$ and $2 \in \mathbb{QNR}_q$ since $p \equiv q \equiv 5 \bmod 8$. This implies that $2 \in \mathbb{J}_N \setminus \mathbb{QR}_N$ as $\left(\frac{2}{N}\right) = 1$. Then, to sample $y \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$, choose an $x \xleftarrow{\$} \mathbb{Z}_N^\star$ and output $y \leftarrow 2 \cdot x^2 \bmod N$. It is easy to check $y \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$. Note that for $p \equiv q \equiv 5 \bmod 8$, it is known that $2 \in \mathbb{QNR}_p$ and $2 \in \mathbb{QNR}_q$. However, no such general result is known for $p \equiv q \equiv 1 \bmod 8$, i.e., to the best of our knowledge, there exists no known element satisfying $a \in \mathbb{Z}_p$ such that $\left(\frac{a}{p}\right) = -1$ for $p \equiv 1 \bmod 8$. Since the above sampling depends on knowledge of an a such that $a \in \mathbb{QNR}_p$ and $a \in \mathbb{QNR}_p$, as of now, the above sampling procedure cannot be generalised to handle the case when $p \equiv q \equiv 1 \bmod 8$. We point out that this is the main reason for restricting the results of this paper to the case when $p \equiv q \equiv 5 \bmod 8$ instead of the more general case of $p \equiv q \equiv 1 \bmod 4$.

Sampling from $\mathbb{J}_N$ (resp. $\mathbb{Z}_N^\star \setminus \mathbb{J}_N$): The Jacobi symbol modulo N can be computed efficiently without knowledge of the factorization of N [58]. Since approximately half of the elements in $\mathbb{Z}_N^\star$ belong to $\mathbb{J}_N$ (resp. $\mathbb{Z}_N^\star \setminus \mathbb{J}_N$), rejection sampling [54] provides an efficient method to obtain uniform samples. One repeatedly samples $x \xleftarrow{\$} \mathbb{Z}_N^\star$ and accepts if $\left(\frac{x}{N}\right) = 1$ (resp. $\left(\frac{x}{N}\right) = -1$). Each iteration succeeds with probability $1/2$, hence repeating the process independently λ times yields a failure probability bounded by $2^{-\lambda}$.

3 Statistical Pseudorandomness of $\psi_{\pi}(n)$

We begin by recalling the following two important results from the theory of analytic number theory. These results will be used to prove the main results of this section. The first result is the Pólya-Vinogradov inequality.

Lemma 1 (Pólya-Vinogradov inequality [33, Theorem 12.5]). *Let χ be a non-principal character modulo p. Then, for $M, N \in \mathbb{N}$, we have*

$$\left| \sum_{M < n \leq M+N} \chi(n) \right| \leq 6\sqrt{p} \log p.$$

It is worth noting that, under the assumption of the GRH (see Theorem 2), Montgomery and Vaughan showed around 1977 [48, Theorem 2] that the above sum satisfies the bound $\ll \sqrt{p} \log \log p$, which is best possible up to determination of the implicit constant. Moreover, one may obtain an unconditional and sharper

estimate in the short range of the sum relative to p by using the Burgess bound. Specifically, in a series of papers (starting in 1962 [10]) Burgess showed that, if p is sufficiently large and $N > p^{1/4+\epsilon}$, for any $\epsilon > 0$, then there exists a constant $\delta = \delta(\epsilon) > 0$, such that the above sum is bounded by $< Np^{-\delta}$ uniformly for all M. Furthermore, under the assumption of the GRH, a similar bound $< Np^{-\delta}$ holds in the wider range $N > p^{\varepsilon}$ for any $\varepsilon > 0$.

Next we recall the following result due to Mauduit and Sárközy [41, Theorem 2], which was later modified by Sárközy [56, Lemma 2] and Oon [50, Lemma 1]. Lemma 2 gives the modified version by Oon.

Lemma 2 ([50, Lemma 1]). *Let p be a prime and χ be any non-principal character modulo p of order d such that $d|p-1$. Let $\overline{\mathbb{F}}_p$ denote the algebraic closer of $\mathbb{F}_p$. Suppose $f(x) \in \mathbb{F}_p$ has a factorization*

$$f(x) \;=\; b(x-x_1)^{d_1}...(x-x_s)^{d_s}$$

such that $x_i \neq x_j$ for $i \neq j$ in $\overline{\mathbb{F}}_p$ with $\gcd(d, d_1, \ldots, d_s) = 1$. Let M, N be real numbers with $0 < N \leq p$. Then

$$\left| \sum_{M<n\leq M+N} \chi(f(n)) \right| \;\leq\; 2s\sqrt{p}\log p.$$

3.1 Construction of $\psi_\pi(n)$

Given a Gaussian prime π and some integer n, the following algorithm outputs the sum of the real and imaginary parts of the corresponding quartic symbol $\chi_\pi(n)$. The algorithm begins by computing $p = \pi\bar{\pi}$ and then checks if n is divisible by π. If so, the algorithm returns 0. Otherwise, it computes the quartic symbol by computing $\chi_\pi(n) = n^{(p-1)/4} \bmod \pi$. The algorithms finally output the sum of the real and imaginary parts of $\chi_\pi(n)$. Further algorithmic details regarding the construction of $\psi_\pi(n) = \mathfrak{Re}(\chi_\pi(n)) + \mathfrak{Im}(\chi_\pi(n))$ are provided in Algorithm 5 of Appendix A.3.

Next, we show that our pseudorandom sequence $\psi_\pi(n)$ exhibits a strong statistical pseudorandomness in the rigorous sense of Mauduit and Sárközy. Using Lemma 1 and 2 we give upper bounds for the well-distribution measure (see Theorem 3) and the correlation measure (see Theorem 4) of our pseudorandom sequence $\psi_\pi(n)$.

Theorem 3 (Well-distribution measure). *For integer prime $p \equiv 1 \bmod 4$ and a Gaussian prime π such that $p = \pi\bar{\pi}$, let $E_{p-1} = (\psi_\pi(1), \psi_\pi(2), \cdots, \psi_\pi(p-1))$. Then the Well-distribution measure of E_{p-1} is upper bounded by*

$$W(E_{p-1}) \;\leq\; 6\sqrt{2}\cdot\sqrt{p}\log p.$$

Proof (Theorem 3). Since for $z \in \mathbb{C}$, we have $\mathfrak{Re}(z) = \frac{z+\bar{z}}{2}$ and $\mathfrak{Im}(z) = \frac{z-\bar{z}}{2i}$, which implies that

$$\psi_\pi(n) \;=\; \frac{(1-i)}{2}\chi_\pi(n) + \frac{(1+i)}{2}\bar{\chi}_\pi(n). \tag{1}$$

Then using (1) in Definition 1 we get

$$W(E_{p-1}) = \max_{d,b,t} \left| \frac{(1-i)}{2} \sum_{j=0}^{t-1} \chi_\pi(d+jb) + \frac{(1+i)}{2} \sum_{j=0}^{t-1} \bar{\chi}_\pi(d+jb) \right|$$
$$\leq \sqrt{2} \max_{d,b,t} \left| \sum_{j=0}^{t-1} \chi_\pi(d+jb) \right|,$$

where the last inequality follows from the triangle inequality and the symmetry with respect to the complex conjugation. Now, since p is prime and $b \neq 0$, b has a multiplicative inverse b' in the field $\mathbb{F}_p$. Therefore for any possible values of d, b and t we have

$$\left| \sum_{j=0}^{t-1} \chi_\pi(d+jb) \right| = \left| \sum_{j=0}^{t-1} \chi_\pi(db'+j)\chi_\pi(b) \right| = \left| \sum_{j=0}^{t-1} \chi_\pi(db'+j) \right|.$$

We bound this character sum by the Pólya-Vinogradov inequality (see Lemma 1) by setting $M = db'$ and $N = t$, which gives

$$W(E_{p-1}) \leq \sqrt{2} \max_{d,b,t} \left| \sum_{j=0}^{t-1} \chi_\pi(db'+j) \right| \leq 6\sqrt{2}\sqrt{p}\log p.$$

□

Theorem 4 (Correlation measure). *For an integer prime $p \equiv 1 \bmod 4$ and a Gaussian prime π such that $p = \pi\bar{\pi}$, let $E_{p-1} = (\psi_\pi(1), \psi_\pi(2), \cdots, \psi_\pi(p-1))$. Then the correlation measure of E_{p-1} of order μ is bounded above by*

$$C_\mu(E_{p-1}) \leq 2^{\frac{\mu}{2}+1} \mu \sqrt{p} \log p.$$

Remark 1. From Theorem 4 observe that E_{p-1} is "good" pseudorandom sequence when $\mu \ll \log\log p$ as suggested in [14, Theorem 2]. Under the assumption of the GRH (see Theorem 2), the corresponding upper bound can be improved to $2^{\frac{\mu}{2}+1}\mu\sqrt{p}\log\log p$. Importantly, without relying on GRH (see Theorem 2) "a notably strong assumption", our result attains the best currently known upper bound on the correlation measure of E_{p-1}.

Proof (Theorem 4). From Definition 2 we get

$$C_\mu(E_{p-1}) = \max_{M,D} \left| \sum_{n=1}^{M} \psi_\pi(n+d_1)\psi_\pi(n+d_2)\cdots\psi_\pi(n+d_\mu) \right|$$

for all $D = (d_1, d_2, ..., d_\mu) \in \mathbb{N}^\mu$, and $0 \leq d_1 < d_2 < ... < d_\mu \leq p-1-M$. Let $\mathcal{L} = \{1, 2, \cdots, \mu\}$, $\mathcal{I} \subseteq \mathcal{L}$ and let $\mathcal{I}^c = \mathcal{L} \setminus \mathcal{I}$. Also, let $f_\mathcal{I}(n) := \prod_{i\in\mathcal{I}}(n+d_i)$

and $f_{\mathcal{I}^c}(n) := \prod_{i\in\mathcal{I}^c}(n+d_i)$. Then, using (1) one can write $C_\mu(E_{p-1})$ as

$$C_\mu(E_{p-1}) = \max_{M,D}\left|\sum_{n=1}^{M}\prod_{j\in\mathcal{L}}\left[\frac{(1-i)}{2}\cdot\chi_\pi(n+d_j)+\frac{(1+i)}{2}\cdot\bar{\chi}_\pi(n+d_j)\right]\right|$$
$$= \max_{M,D}\left|\sum_{\mathcal{I}\subset\mathcal{L}}\frac{(1-i)^{|\mathcal{I}|}(1+i)^{|\mathcal{I}^c|}}{2^\mu}\sum_{n=1}^{M}\chi_\pi(f_{\mathcal{I}}(n))\chi_{\bar{\pi}}(f_{\mathcal{I}^c}(n))\right|, \quad (2)$$

by the multiplicative property of χ_π. As $\chi_\pi(n)$ is a Dirichlet character of order 4 modulo p, we have $\chi_\pi(n^4)=1$ for $\gcd(n,p)=1$. Hence, $\chi_\pi((n+b_j)^4)=1$ for $n\neq p-b_j$. Without loss of generality we can take $|\mathcal{I}^c|\leq\mu/2$ (otherwise $|\mathcal{I}|\leq\mu/2$), then using the multiplicative property of $\chi_\pi(\cdot)$ we get $\chi_\pi(f_{\mathcal{I}^c}(n)^3)\cdot\chi_\pi(f_{\mathcal{I}^c}(n))=\chi_\pi(f_{\mathcal{I}^c}(n)^4)=1$ which implies that

$$\chi_\pi(f_{\mathcal{I}^c}(n)^3) \;=\; \chi_{\bar{\pi}}(f_{\mathcal{I}^c}(n)), \quad (3)$$

by Fact 3 and the multiplicative property of $\chi_\pi(\cdot)$. Then using (3) in the right hand summation of (2) and from the multiplicative property of $\chi_\pi(\cdot)$, we obtain

$$\sum_{n=1}^{M}\chi_\pi(f_{\mathcal{I}}(n))\chi_{\bar{\pi}}(f_{\mathcal{I}^c}(n)) \;=\; \sum_{n=1}^{M}\chi_\pi(f_{\mathcal{I}}(n))\chi_\pi^3(f_{\mathcal{I}^c}(n)) \;=\; \sum_{n=1}^{M}\chi_\pi(f_{\mathcal{I}}(n)\cdot f_{\mathcal{I}^c}(n)^3).$$

Applying Lemma 2 in the case $p\equiv 1 \bmod 4$, and considering the non-principal quartic character $\chi_\pi(\cdot)$ of order 4 (so that $4\mid(p-1)$ and $\gcd(4,1,3)=1$), we observe that $d_j\in\mathbb{F}_p$ for all $j\in\mathcal{I}$ or $j\in\mathcal{I}^c$. Moreover, the number of distinct linear factors of the polynomial $f_{\mathcal{I}}(n)\cdot f_{\mathcal{I}^c}(n)^3$ is μ. Consequently, we obtain

$$\left|\sum_{n=1}^{M}\chi_\pi(f_{\mathcal{I}}(n)\cdot f_{\mathcal{I}^c}(n)^3)\right| \;\leq\; 2\mu\sqrt{p}\log p \quad (4)$$

Then applying triangle inequality in using (2) and then using (4), we get

$$C_\mu(E_{p-1}) \;\leq\; \max_{M,D}\sum_{\mathcal{I}\subset\mathcal{L}}\frac{\sqrt{2}^{|\mathcal{I}|+|\mathcal{I}^c|}}{2^\mu}\cdot 2\mu\sqrt{p}\log p \;=\; \max_{M,D}\sum_{\mathcal{I}\subset\mathcal{L}}\frac{2^{\frac{\mu}{2}+1}}{2^\mu}\cdot\mu\sqrt{p}\log p$$
$$= 2^\mu\frac{2^{\frac{\mu}{2}+1}}{2^\mu}\mu\sqrt{p}\log p \;=\; 2^{\frac{\mu}{2}+1}\mu\sqrt{p}\log p,$$

since the number of subsets $\mathcal{I}\subseteq\mathcal{L}$ is equal to 2^μ. □

4 Our Construction of wPRF

In this section, we first present our wPRF construction followed by its security proof, based on the hardness of the quadratic residuosity assumption. For this, first recall that Cornacchia's algorithm [17, Algorithm 1.5.2] on input an integer

prime $p \equiv 1 \bmod 4$, outputs $\pi \in \mathbb{Z}[i]$ such that $p = \pi\bar{\pi}$. However, the security proof of our wPRF scheme, in particular Theorem 9, requires one to efficiently sample from $\mathbb{J}_N \backslash \mathbb{QR}_N$ without knowledge of the factorization $N = pq$ (mentioned in Section 2.5) which restricts us to work with primes $p \equiv q \equiv 5 \bmod 8$. Under this assumption, a simplified variant of Cornacchia's algorithm can be used to compute a Gaussian integer π for a given prime $p \equiv 5 \bmod 8$ such that $p = \pi\bar{\pi}$, where instead of invoking Shanks's algorithm [17, Algorithm 1.5.1] to find a quadratic non-residue, one can directly work with 2, since the Legendre symbol $\left(\frac{2}{p}\right) = -1$ for any prime $p \equiv 5 \bmod 8$. The full algorithmic description and its correctness proof of this simplified algorithm are given as Algorithm 6 in Appendix A.4.

Next, define an encoding function $F : \{\pm 1, \pm i\} \to \{0,1\}^2$ by $1 \mapsto 00, i \mapsto 01, -1 \mapsto 10$ and $-i \mapsto 11$. Note that any of the possible 24 bijective functions can be taken. Using this encoding function F, we now define *a family of Hidden Prime Quartic PRFs* ($\mathcal{Q}^{\mathsf{wPRF}}$) followed by our *Generalized Hidden Prime Quartic PRG*.

Definition 5 (Hidden Prime Quartic PRF ($\mathcal{Q}^{\mathsf{wPRF}}$)). *Let λ be a security parameter. Define a family of **Hidden Prime Quartic PRFs** $Q_\lambda : (\mathsf{QPrimes}_\lambda \times [2^\lambda]) \times [2^{3\lambda}] \to \{0,1\}^2$ that generate 2-bit output as follows.*

- *Use variants of primality testing algorithms such as Miller-Rabin [45,53] to find a λ-bit quartic prime p such that $p \equiv 5 \bmod 8$.*
- *Compute the corresponding Gaussian prime π using the simplified Cornacchia's algorithm (see Algorithm 6 of Appendix A.4).*
- *Finally, for a shift $x \in [2^\lambda]$ generate the 2-bit output as*

$$Q_\lambda((p,x),a) := F\left(\chi_\pi(x+a)\right),$$

where $p = \pi\bar{\pi}$ and F denote the encoding function defined above.

(Refer to Algorithm 7 in Appendix A.5 for an algorithmic description of Q_λ.) We then define the family of functions $\mathcal{Q}^{\mathsf{wPRF}} = \{Q_\lambda\}_{\lambda \in \mathbb{N}}$ as the secret-prime Quartic weak pseudorandom function.

Definition 6. *(Generalized Hidden Prime Quartic PRG) Let $\lambda \in \mathbb{N}$ be a security parameter. For a sequence length $\ell = \mathsf{poly}(\lambda)$ and a vector $\boldsymbol{a} = (a_1, \ldots, a_\ell) \in \mathbb{Z}^\ell$, we define our family of* generalized hidden prime quartic sequence function *$Q_{\boldsymbol{a},\ell} : \mathsf{QPrimes}_\lambda \times [2^\lambda] \to \{\{0,1\}^2\}^\ell$ as follows.*

- *Use variants of primality testing algorithms such as Miller-Rabin [45,53] to find a λ-bit quartic prime p such that $p \equiv 5 \bmod 8$.*
- *Compute the corresponding Gaussian prime π using the simlified Cornacchia's algorithm (see Algorithm 6 of Appendix A.4).*
- *Finally, for a shift $x \in [2^\lambda]$ output the sequence $Q_{\boldsymbol{a},\ell}(p,x) := \Big(F\left(\chi_\pi(x+a_1)\right), F\left(\chi_\pi(x+a_2)\right), \ldots, F\left(\chi_\pi(x+a_\ell)\right)\Big)$, $p = \pi\bar{\pi}$ and F denotes an encoding function defined above.*

For a security parameter $\lambda \in \mathbb{N}$, define the set of all such ℓ-length functions, i.e., generalized hidden prime quartic sequence function $Q_{\boldsymbol{a},\ell}$ with 2ℓ-bit output, as

$$\mathcal{Q}_{\lambda,\ell} \;:=\; \left\{ Q_{\boldsymbol{a},\ell} \mid \boldsymbol{a} \in [2^{3\lambda}]^{\ell} \right\}.$$

Then, the generalized hidden prime quartic PRG family with output length ℓ is defined as

$$\mathcal{Q}_{\ell}^{\mathsf{PRG}} \;:=\; \left\{ \mathcal{Q}_{\lambda,\ell} \mid \lambda \in \mathbb{N} \right\}.$$

Complexity of Quartic vs. Legendre PRF: First note that the first two steps in both Definition 5 and 6 are both common, and as such, the first step is identical to the first step required for Legendre wPRFs [18]. However, the second step is not required in case of Legendre wPRFs [18]. Although the second step is extra in case of quartic wPRFs, we mention that this cost is one-time and as such is executed at the beginning to generate a prime integer $p \equiv 5 \bmod 8$ and a Gaussian prime π. Using the observation that $n^{(p-1)/4} \bmod \pi \equiv \left(n^{(p-1)/4} \bmod p\right) \bmod \pi$ and noting that the exponent in case of quartic is $(p-1)/4$ is 1-bit less than that of $(p-1)/2$, which is the exponent in case of the quadratic character, it is straightforward to show that the computational cost of one PRF evaluation in case of quartic character is almost similar to that of the Legendre PRF. However, the fact that the quartic character outputs 2 bits instead of one makes it approximately two times as efficient compared to the Legendre PRF. A more detailed discussion is given in Appendix B.

The remainder of this section is devoted to proving that our generalized hidden-prime quartic sequence function constitutes a wPRF, under the assumption that the Quadratic Residuosity problem with respect to RSA moduli is hard. The proof proceeds via a sequence of reductions. In Section 4.1, we first show that the Quadratic Residuosity problem over RSA moduli reduces to the Decisional Quadratic Residuosity problem with respect to quartic integers, i.e., $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{Q}}$. In Section 4.2, we then establish that the QR_{Q} problem further reduces to the Decisional Quartic Residuosity problem. Finally, in Section 4.3, we demonstrate that the Decisional Quartic Residuosity problem reduces to the task of distinguishing our wPRF construction (Definition 5) from a uniformly random 2ℓ-bit string.

4.1 Quadratic Residuosity for RSA Modulus Reduces to Quadratic Residuosity for Quartic modulus

In this section, we prove that the quadratic residuosity problem with respect to the RSA modulus reduces in polynomial-time to the quartic residuosity problem, i.e., $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{Q}}$. The idea behind the proof is similar to the Corrigan-Gibbs and Wu [18]. We begin by recalling the definition of *Quadratic residuosity problem*, given by Goldwasswer and Micali [29], relative to a standard RSA modulus $N = pq$, for arbitrary large (distinct) primes p and q.

Definition 7 (Quadratic Residuosity Problem for RSA Modulus). *Let $\mathsf{Primes}_{\lambda}$ be the set of λ-bit primes. For an PPT algorithm $\mathcal{A}$, a security parameter*

λ, *and a bit* $b \in \{0,1\}$, *define* $\rho_{\mathcal{A},b}(\lambda)$ *as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N,x_b)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{Primes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\; x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right].$$

Then the ***quadratic-residuosity problem with respect to an RSA modulus*** N *(*$\mathsf{QR}_{\mathsf{RSA}}$ *in short) is for the algorithm* $\mathcal{A}$ *to distinguish if the input came from* $\mathbb{QR}_N$ *or from* $\mathbb{J}_N \setminus \mathbb{QR}_N$*. The corresponding* ***quadratic-residuosity advantage*** *for algorithm* $\mathcal{A}$ ***with respect to an RSA modulus*** N *is defined as follows.*

$$\mathsf{QRAdv}_{\mathsf{RSA}}[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

Let $\mathsf{BPrimes}_\lambda$ denote the set of λ-bit Blum primes, i.e., the set of all λ-bit primes p such that $p \equiv 3 \bmod 4$. Then the set of ***Non-Blum primes*** is defined as $\mathsf{NBPrimes}_\lambda := \mathsf{Primes}_\lambda \setminus \mathsf{BPrimes}_\lambda$. Let, $\delta(\lambda) := \frac{|\mathsf{NBPrimes}_\lambda|}{|\mathsf{Primes}_\lambda|}$ denote the density of λ-bit Non-Blum primes. We call N a ***special integer*** if $N = pq$ and $p \equiv q \equiv 1 \bmod 4$.

Definition 8 (Quadratic Residuosity Problem for Special **Modulus).** *Let* $\mathsf{NBPrimes}_\lambda$ *be the set of* λ*-bit non-Blum primes. For an* PPT *algorithm* $\mathcal{A}$*, a security parameter* λ*, and a bit* $b \in \{0,1\}$*, define* $\rho_{\mathcal{A},b}(\lambda)$ *as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N,x_b)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{NBPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\; x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right].$$

Then the ***quadratic-residuosity problem with respect to a special modulus*** N *(*$\mathsf{QR}_{\mathsf{SP}}$ *in short) is for the algorithm* $\mathcal{A}$ *to distinguish if the input came from* $\mathbb{QR}_N$ *or from* $\mathbb{J}_N \setminus \mathbb{QR}_N$*. The corresponding* ***quadratic-residuosity advantage*** *for algorithm* $\mathcal{A}$ ***with respect to a special modulus*** N *is defined as follows.*

$$\mathsf{QRAdv}_{\mathsf{SP}}[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

The next theorem, together with the subsequent result, can be equivalently reformulated as $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{SP}}$.

Theorem 5. *Let* $\delta(\lambda) := \frac{|\mathsf{NBPrimes}_\lambda|}{|\mathsf{Primes}_\lambda|}$ *be the density of Non-Blum primes. For every efficient (*PPT*) adversary* $\mathcal{A}$ *for the* $\mathsf{QR}_{\mathsf{SP}}$ *problem there exists an efficient adversary* $\mathcal{B}$ *for* $\mathsf{QR}_{\mathsf{RSA}}$ *such that the following holds.*

$$\mathsf{QRAdv}_{\mathsf{RSA}}[\mathcal{B}](\lambda) \;\geq\; \delta(\lambda)^2 \cdot \mathsf{QRAdv}_{\mathsf{SP}}[\mathcal{A}](\lambda)$$

Proof (Theorem 5). See Appendix C.5.

Finally, take $x = 2^\lambda - 1$ (the maximum value of a λ-bit number), $a = 1$ and $d = 4$. Then applying Theorem 1 we get $\pi(x;4,1) \approx \frac{x}{\varphi(4)\log x} = \frac{x}{2\log x}$

as $x \to \infty$. Also from Prime number theorem we know that $\pi(x) \approx \frac{x}{\log x}$ as $x \to \infty$. Clearly, $\frac{x}{\log x} \neq 0$ as $x \to \infty$. Therefore, $\delta(\lambda) := \frac{|\mathsf{NBPrimes}_\lambda|}{|\mathsf{Primes}_\lambda|} \to \frac{x/2\log x}{x/\log x} = \frac{1}{2}$ as $x \to \infty$. In other words, as $p \to \infty$ the density of Non-Blum primes tends to $\frac{1}{2}$. Combined with Theorem 5, it shows that the hardness of the quadratic-residuosity assumption with respect to general RSA modulus implies the hardness of quadratic residuosity with respect to quartic integers, i.e. $\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathsf{QR}_{\mathsf{SP}}$.

We call all such non-Blum primes p with $p \equiv 5 \bmod 8$ as ***quartic primes*** and call $N = pq$ such that $p \equiv q \equiv 5 \bmod 8$ as ***quartic integers***. Then the density of quartic primes with respect to non-Blum primes is given by $\delta(\lambda) := \frac{|\mathsf{QPrimes}_\lambda|}{|\mathsf{NBPrimes}_\lambda|}$, where $\mathsf{QPrimes}_\lambda$ denotes the set of all λ-bit primes p such that $p \equiv 5 \bmod 8$.

Definition 9 (Quadratic Residuosity Problem for Quartic Modulus). *Let $\mathsf{QPrimes}_\lambda$ be the set of λ-bit quartic primes. For an PPT algorithm $\mathcal{A}$, a security parameter λ, and a bit $b \in \{0,1\}$, define $\rho_{\mathcal{A},b}(\lambda)$ as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N,x_b)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{QPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\; x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right].$$

*Then the **quadratic-residuosity problem with respect to a Quartic modulus** N (QR_Q in short) is for the algorithm $\mathcal{A}$ to distinguish if the input came from $\mathbb{QR}_N$ or from $\mathbb{J}_N \setminus \mathbb{QR}_N$. The corresponding **quadratic-residuosity advantage** for algorithm $\mathcal{A}$ **with respect to a Quartic modulus** N is defined as follows.*

$$\mathsf{QRAdv}_\mathsf{Q}[\mathcal{A}](\lambda) \;:=\; |\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)|\,.$$

The next theorem, together with the subsequent result, can be equivalently reformulated as $\mathsf{QR}_{\mathsf{SP}} \leq_p \mathsf{QR}_\mathsf{Q}$.

Theorem 6. *Let $\delta(\lambda)$ denote the density of Quartic primes with respect to non-Blum primes. Then, for every efficient quadratic-residuosity adversary $\mathcal{A}$ with respect to Quartic integers, there exists an efficient quadratic-residuosity adversary $\mathcal{B}$ with respect to a special integer modulus such that*

$$\mathsf{QRAdv}_{\mathsf{SP}}[\mathcal{B}](\lambda) \;\geq\; \delta(\lambda)^2 \cdot \mathsf{QRAdv}_\mathsf{Q}[\mathcal{A}](\lambda).$$

Proof (Theorem 6). See Appendix C.6.

Finally, as $p \to \infty$, the density of Quartic primes with respect to non-Blum primes tends to $\frac{1}{2}$. This again follows from Theorem 1, where $x = 2^\lambda - 1$, $a = 5$ and $d = 8$. Hence $\pi(x;8,5) \approx \frac{x}{\varphi(8)\log x} = \frac{x}{4\log x}$ as $x \to \infty$. Also we have seen that for Non-Blum primes $\pi(x;4,1) \approx \frac{x}{2\log x}$ as $x \to \infty$. Therefore, $\delta(\lambda) = \frac{|\mathsf{QPrimes}_\lambda|}{|\mathsf{NBPrimes}_\lambda|} \to \frac{x/4\log x}{x/2\log x} = \frac{1}{2}$ as $x \to \infty$. Combined with Theorem 6, this shows that the hardness of the quadratic-residuosity assumption with respect to the special integer modulus implies the hardness of quadratic residuosity with respect to quartic integers, i.e. $\mathsf{QR}_{\mathsf{SP}} \leq_p \mathsf{QR}_\mathsf{Q}$.

4.2 Quadartic Residuosity for Quartic modulus Reduces to Decisional Quartic Residuosity

In this section we show that $\mathsf{QR}_{\mathsf{Q}} \leq_p \mathsf{QuarticRD}_{\mathsf{Q}}$, where $\mathsf{QuarticRD}_{\mathsf{Q}}$ denotes the quartic residuosity problem (see Definition 11). We start by showing that QR_{Q} is polynomial-time reducible to the $\mathsf{QRD}_{\mathsf{Q}}$ problem, i.e., $\mathsf{QR}_{\mathsf{Q}} \leq_p \mathsf{QRD}_{\mathsf{Q}}$, where $\mathsf{QRD}_{\mathsf{Q}}$ denotes the *Decisional Quadratic Residuosity for quartic integers* defined below.

Definition 10 (Decisional Quadratic Residuosity Problem for Quartic Integers ($\mathsf{QRD}_{\mathsf{Q}}$)). *Let $\lambda \in \mathbb{N}$ denote the security parameter and $b \in \{0,1\}$ a bit. For a* PPT *adversary $\mathcal{A}$ define $\rho_{\mathcal{A},b}(\lambda)$ as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N,x,y_b)=1 : \begin{array}{r} p,q \xleftarrow{\$} \mathsf{QPrimes}_\lambda,\; N \leftarrow pq \\ x \xleftarrow{\$} \mathbb{J}_N \\ y_0 \leftarrow \left(\frac{x}{p}\right),\; y_1 \xleftarrow{\$} \{-1,1\} \end{array}\right]$$

Then decisional quadratic residuosity advantage for quartic integers *of $\mathcal{A}$ is defined as follows.*

$$\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

Theorem 7 ($\mathsf{QR}_{\mathsf{Q}} \leq_p \mathsf{QRD}_{\mathsf{Q}}$). *For every* PPT decisional-quadratic-residuosity *adversary $\mathcal{A}$* for quartic integers, *there exists* PPT quadratic-residuosity *adversary $\mathcal{B}$ such that*

$$\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda) \;=\; \frac{1}{2}\mathsf{QRAdv}_{\mathsf{Q}}[\mathcal{B}](\lambda).$$

Proof (Theorem 7). See Appendix C.7.

Next, we define the following problem which is analogous to the decisional quadratic residuosity problem. We call this problem as the *Decisional Quartic Residuosity Problem* or $\mathsf{QuarticRD}_{\mathsf{Q}}$ in short.

Definition 11 (Decisional Quartic Residuosity Problem ($\mathsf{QuarticRD}_{\mathsf{Q}}$)). *Let $\mathcal{A}$ be a* PPT *algorithm, $\lambda \in \mathbb{N}$ denote the security parameter, and $b \in \{0,1\}$ a bit. Define the quantity $\rho_{\mathcal{A},b}(\lambda)$ as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N,x,y_b)=1 : \begin{array}{r} p,q \xleftarrow{\$} \mathsf{QPrimes}_\lambda, N \leftarrow pq, p=\pi\bar{\pi}, \\ x \xleftarrow{\$} \mathbb{J}_N \\ y_0 \leftarrow F\left(\chi_\pi(x)\right),\; y_1 \xleftarrow{\$} \{0,1\}^2 \end{array}\right].$$

Then the decisional quartic residuosity advantage *of $\mathcal{A}$ is defined as follows.*

$$\mathsf{QuarticRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

Theorem 8 ($\mathsf{QRD}_{\mathsf{Q}} \leq_p \mathsf{QuarticRD}_{\mathsf{Q}}$)**.** *For every* PPT decisional quartic residuosity *adversary* $\mathcal{A}$*, there exists a* PPT decisional quadratic residuosity *adversary for quartic integers* $\mathcal{B}$ *such that*

$$\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{B}](\lambda) \;=\; \frac{1}{2}\mathsf{QuarticRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda)$$

Proof (Theorem 8). We construct an Algorithm $\mathcal{B}$ that solves the decisional quadratic residuosity problem with respect to quartic integers given an adversary $\mathcal{A}$ that can solve the decisional quartic residuosity problem. On input a decisional quadratic residuosity challenge (N, x, y) Algorithm $\mathcal{B}$ does the following. It first samples a bit $r \xleftarrow{\$} \{0,1\}$. Then sets $z \leftarrow F((-1)^r)$ if $y = 1$, otherwise it sets $z \leftarrow F(i \cdot (-1)^r)$. The algorithm then runs the subroutine $\mathcal{A}$ on input (N, x, z). Finally it outputs whatever $\mathcal{A}$ outputs. Now, conditioning on the independent events "$r = b_r$" and "$y_1 = b_{y_1}$", where $b_r \in \{0,1\}$ and $b_{y_1} \in \{-1,1\}$, and following some routine calculations, we get $\rho_{\mathcal{B},1} = \rho_{\mathcal{A},1}$. Similarly, using the facts that $\Pr[y_0 = 1] = \Pr[y_0 = -1] = \frac{1}{2}$, for any $b_1, b_2 \in \{0,1\}$,

$$\Pr[F(\chi_\pi(x)) = b_1 b_2 \mid y_0 = b_{y_0}] \;=\; \begin{cases} \frac{1}{2}; \text{ if } b_2 = (1 - b_{y_0})/2 \\ 0; \text{ otherwise} \end{cases},$$

$\Pr[F(\chi_\pi(x)) = b_1 b_2] = \frac{1}{4}$ and by some routine calculations, we get $\rho_{\mathcal{B},0} = \frac{1}{2} \cdot (\rho_{\mathcal{A},0} + \rho_{\mathcal{A},1})$. Therefore, $\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{B}](\lambda) = \frac{1}{2} \cdot \mathsf{QuarticRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda)$. □

4.3 Decisional Quartic Residuosity Reduces to wPRFs

In this section we show that $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$, where $\mathcal{Q}^{\mathsf{wPRF}}$ denotes our hidden prime quartic PRF family (see Definition 5). We begin by defining another intermediate problem which we call the ℓ*-time Full Domain Quartic Problem (*$\mathsf{Quartic}_\ell$*)* and show that $\mathsf{QuarticRD}_{\mathsf{Q}} \leq_p \mathsf{Quartic}_\ell$.

Definition 12 (ℓ-time Full Domain Quartic Problem ($\mathsf{Quartic}_\ell$)). *Let* $\mathcal{A}$ *be a* PPT *algorithm,* $\lambda \in \mathbb{N}$ *denote the security parameter, and* $b \in \{0,1\}$ *a bit. Define the quantity* $\rho_{\mathcal{A},b}(\lambda)$ *as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}\left(N, \begin{matrix} x_1, \cdots, x_\ell \\ y_{b,1}, \cdots, y_{b,\ell} \end{matrix}\right) = 1 : \begin{matrix} p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda, N \leftarrow pq, p = \pi\bar{\pi} \\ x_1, \cdots, x_\ell \xleftarrow{\$} \mathbb{Z}_N^\star \\ \forall\; i \in [\ell] : y_{0,i} \leftarrow F\left(\chi_\pi(x_i)\right) \\ \forall\; i \in [\ell] : y_{1,i} \xleftarrow{\$} \{0,1\}^2 \end{matrix}\right].$$

Then the ℓ***-time full domain Quartic advantage*** *of* $\mathcal{A}$ *is defined as follows.*

$$\mathsf{QuarticAdv}_\ell[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

It is important to note that the sampling procedure of [18, Lemma 3.5] will not work here as for any $r \xleftarrow{\$} \mathbb{QR}_N$, $\chi_\pi(r) \in \{-1,1\}$. Hence, one cannot predict

the quartic character of the resampled value x. In the following (see Algorithm 1), we provide a slight modification of the way r is sampled from $\mathbb{QR}_N$ so that an equivalent resampling procedure can be designed. It is easy to check that the output r of $\mathsf{ResamplingQR}(N)$ is distributed uniformly at random over $\mathbb{QR}_N$. Let, $z_0 \in \mathbb{Z}_N^\star$ be a *quadratic non-residue* modulo both p and q, $z_1 \in \mathbb{Z}_N^\star$ be a

Algorithm 1 $\mathsf{ResamplingQR}(N)$

$b \xleftarrow{\$} \{0,1\}$
if $b = 0$ **then**
  $x \xleftarrow{\$} \mathbb{Z}_N^*$
  Set $r \leftarrow x^4 \bmod N$ and $\beta \leftarrow F(1)$
else
  $b' \xleftarrow{\$} \{0,1\}$
  if $b' = 0$ **then**
    $x \xleftarrow{\$} \mathbb{Z}_N^\star \setminus \mathbb{J}_N$ (see Section 2.5)
  else
    $x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$ (see Section 2.5)
  end if
  Set $r \leftarrow x^2 \bmod N$ and $\beta \leftarrow F(-1)$
end if
Return (r, β)

quadratic non-residue modulo p and *quadratic residue* modulo q, and $u \in \mathbb{J}_N$. The next algorithm (see Algorithm 2) which on input an element $u \in \mathbb{J}_N$ and its corresponding quartic character $\chi_\pi(u)$ uses Algorithm 1 to output (x, β) such that $x \xleftarrow{\$} \mathbb{Z}_N^*$ and $\beta = F(\chi_\pi(x))$. This leads us to the following lemma.

Algorithm 2 $\mathsf{ResamplingQuartic}(u \in \mathbb{J}_N, \alpha = F(\chi_\pi(u)))$

$(r, \gamma) \leftarrow \mathsf{ResamplingQR}(N)$
$\rho_0, \rho_1 \xleftarrow{\$} \{0,1\}$
Set $x \leftarrow u \cdot r \cdot z_0^{\rho_0} \cdot z_1^{\rho_1} \bmod N$
Set $\beta \leftarrow \alpha \cdot \gamma \cdot F(\chi_\pi(z_0)^{\rho_0}) \cdot F(\chi_\pi(z_1)^{\rho_1})$
Return (x, β)

Lemma 3. *For all composite integers $N = pq$, where p and q are distinct odd primes and for every*

- $z_0 \in \mathbb{Z}_N^\star$ *that is a* quadratic non-residue *modulo both p and q*
- $z_1 \in \mathbb{Z}_N^\star$ *that is a* quadratic non-residue *modulo p and* quadratic residue *modulo q*
- $u \in \mathbb{J}_N$

The following distributions are identical.

$$\left\{(x,\beta): \begin{array}{l}\rho_0,\rho_1 \xleftarrow{\$} \{0,1\}\\ (x,\beta) \leftarrow \mathsf{ResamplingQuartic}(u, F(\chi_\pi(u)))\end{array}\right\} \equiv \left\{(x,\beta): \begin{array}{l} x \xleftarrow{\$} \mathbb{Z}_N^\star \\ \beta \leftarrow F(\chi_\pi(x))\end{array}\right\}$$

Proof (Lemma 3). See Appendix C.8.

Note that Lemma 3 holds for any z_0, z_1 such that z_0 belongs to both $\mathbb{QNR}_p$ and $\mathbb{QNR}_q$; and z_1 belongs to both $\mathbb{QNR}_p$ and $\mathbb{QR}_q$. In particular, the lemma holds for z_0, z_1 satisfying $\chi_\pi(z_0) = \chi_\pi(z_1) = i$, which implies that the output of Algorithm 2 is set to (x, β), where $\beta \leftarrow \alpha \cdot \gamma \cdot i^{\rho_0+\rho_1}$[5].

Using Lemma 3 we next show in Theorem 9 that the decisional quartic residuosity problem is polynomial-time reducible to the $\mathsf{Quartic}_\ell$ problem. The proof of Theorem 9 is again similar to that of [18, Lemma 3.5]. We defer the detailed proof to Appendix C.9.

Theorem 9 ($\mathsf{QuarticRD}_\mathsf{Q} \leq_p \mathsf{Quartic}_\ell$)**.** *For every polynomial $\ell = \mathsf{poly}(\lambda)$ and for every* PPT ℓ-time full-domain Quartic *adversary $\mathcal{A}$, there exists* PPT decisional quartic residuosity *adversary $\mathcal{B}$ such that*

$$\mathsf{QuarticAdv}_\ell[\mathcal{A}](\lambda) \;=\; \ell \cdot \mathsf{QuarticRDAdv}[\mathcal{B}](\lambda).$$

Corollary 1. *For every polynomial $\ell(\lambda) = \mathsf{poly}(\lambda)$ and every* PPT ℓ-time full-domain Quartic *adversary $\mathcal{A}$, there exists a* PPT decisional quadratic residuosity *adversary for quartic integers $\mathcal{B}$ such that*

$$\mathsf{QuarticAdv}_\ell[\mathcal{A}](\lambda) \;=\; \ell(\lambda) \cdot \mathsf{QRAdv}[\mathcal{B}](\lambda).$$

Proof (Corollary 1). Combining Theorems 7, 8 and 9 the result follows. □

The next result, which is one of the main results of this paper shows that distinguishing our family of hidden prime quartic wPRF ($\mathcal{Q}^{\mathsf{wPRF}}$) from a uniformly random 2ℓ-bit string is at least as hard as solving the quadratic residuosity problem with respect to a RSA modulus (QR_RSA), i.e., $\mathsf{QR}_\mathsf{RSA} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$. The idea behind the proof of Theorem 10 is inspired by the proof of [18, Theorem 4.2].

Theorem 10 ($\mathsf{QR}_\mathsf{RSA} \leq_p \mathcal{Q}^{\mathsf{wPRF}}$)**.** *For every efficient ℓ-query algorithm $\mathcal{A}$ that breaks weak pseudorandomness of $\mathcal{Q}^{\mathsf{wPRF}}$ with advantage $\mathsf{wPRFAdv}[\mathcal{A}](\lambda)$, there exists an efficient algorithm $\mathcal{B}$ for the quadratic residuosity problem for RSA modulus with advantage $\mathsf{QRAdv}[\mathcal{B}](\lambda)$ such that*

$$\mathsf{wPRFAdv}[\mathcal{A}, \mathcal{Q}^{\mathsf{wPRF}}](\lambda) \;=\; \ell \cdot \mathsf{QRAdv}[\mathcal{B}](\lambda) + \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\cdot)$ denotes a negligible function and $\ell = \mathsf{poly}(\lambda)$.

[5] For a fixed $z_0 \in \mathbb{QNR}_p, \mathbb{QNR}_q$ such that $\chi_\pi(z_0) = i$, we will always find such $z_1 \in \mathbb{QNR}_p, \mathbb{QR}_q$ with $\chi_\pi(z_1) = i$. If not, let us assume that $\chi_\pi(z_1) = -i$ (since $\left(\frac{z_1}{p}\right) = -1$). Then, by Fact 4 there exists z_1^{-1} such that $\chi_\pi(z_1^{-1}) = -i$. Implying that $z_1^{-1} \in \mathbb{QNR}_p$ and the multiplicative property of $\mathbb{QR}_q$ ensures that $z_1^{-1} \in \mathbb{QR}_q$.

Proof (Theorem 10). Consider the distributions of the following two types of outputs.

- **Output of Quartic wPRF:** An adversary making exactly ℓ queries sees a random vector $\boldsymbol{a} = (a_1, \ldots, a_\ell) \xleftarrow{\$} [2^{3\lambda}]^\ell$ and a sequence of ℓ Quartic symbols $(F(\chi_\pi(x + a_1)), \ldots, F(\chi_\pi(x + a_\ell)))$, where $p = \pi\bar{\pi}$, $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ and $x \xleftarrow{\$} [2^\lambda]$.
- **Output full domain Quartic instance ℓ times:** The adversary sees a random vector $\boldsymbol{x} = (x_1, \ldots, x_\ell) \xleftarrow{\$} \mathbb{Z}_N^\star$ and Quartic symbols $F(\chi_\pi(x_1)), \ldots, F(\chi_\pi(x_\ell))$, where $p = \pi\bar{\pi}$ and $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda, N = pq$.

The key argument in the proof is that when $a_i \xleftarrow{\$} [2^{3\lambda}]$, the distribution of $(a_i + x) \bmod p$ is statistically indistinguishable from $x_i \bmod p$, where $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$. The theorem then follows from the following claim.

Claim 1 *For every polynomial $\ell = \mathsf{poly}(\lambda)$, every efficient ℓ query algorithm $\mathcal{A}$ that breaks the pseudorandomness of $\mathcal{Q}^{\mathsf{wPRF}}$ with advantage $\mathsf{wPRFAdv}[\mathcal{A}](\lambda)$, there exists an efficient algorithm $\mathcal{B}$ for the ℓ time full domain Quartic problem with advantage $\mathsf{QuarticAdv}_l[\mathcal{B}](\lambda)$ such that*

$$\mathsf{wPRFAdv}[\mathcal{A}, \mathcal{Q}^{\mathsf{wPRF}}] = \mathsf{QuarticAdv}_\ell[\mathcal{B}](\lambda) + \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\cdot)$ denotes some negligible function.

Proof (Claim 1). Consider any polynomial $\ell = \mathsf{poly}(\lambda)$, and let $\mathcal{A}$ be an efficient adversary in the wPRF distinguishing game that issues ℓ queries. Without loss of generality, we may assume that $\mathcal{A}$ makes exactly ℓ queries, as any adversary making fewer queries can also be made to issue ℓ queries without affecting its distinguishing advantage. We begin by defining the following sequence of hybrid experiments.

Hyb_0: Challenger starts by sampling the following components.
- Samples $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$. It also samples $x \xleftarrow{\$} [2^\lambda]$.
- Samples $a_i \xleftarrow{\$} [2^{3\lambda}]$ $\forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} \leftarrow [F(\chi_\pi(x + a_1)), \ldots, F(\chi_\pi(x + a_\ell))]$

The challenger gives $(\boldsymbol{a}, \boldsymbol{y})$ to algorithm $\mathcal{A}$ as the answer to its ℓ queries. Algorithm $\mathcal{A}$ outputs a bit $b \in \{0, 1\}$, which is the output of the experiment.

Hyb_1: In this game only the sampling of a_i is changed.
- Samples $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$. It also samples $x \xleftarrow{\$} [2^\lambda]$
- Samples $a_i' \xleftarrow{\$} [2^{3\lambda}]$ and sets $a_i \leftarrow a_i' - x, \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} \leftarrow (F(\chi_\pi(a_1{}')), \ldots, F(\chi_\pi(a_\ell{}')))$

Hyb_2: In this game only sampling of a_i' is changed with respect to Hyb_1 along with the creation of N.

- Samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv q \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$, computes $N \leftarrow pq$ and samples $x \xleftarrow{\$} [2^\lambda]$.
- Sets $B \leftarrow \lfloor 2^{3\lambda}/N \rfloor$
- Samples $x_i \xleftarrow{\$} \mathbb{Z}_N, w_i \xleftarrow{\$} [0, B-1], a'_i \leftarrow w_i N + x_i$, $a_i \leftarrow a'_i - x, \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} \leftarrow (F(\chi_\pi(x_1)), \ldots, F(\chi_\pi(x_\ell)))$

Hyb_3**:** In this game only sampling of x_i is changed with respect to Hyb_2.

- Samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv q \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$, computes $N \leftarrow pq$ and samples $x \xleftarrow{\$} [2^\lambda]$.
- Sets $B \leftarrow \lfloor 2^{3\lambda}/N \rfloor$
- Samples $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$, $w_i \xleftarrow{\$} [0, B-1], a'_i \leftarrow w_i N + x_i$, $a_i \leftarrow a'_i - x\ \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} \leftarrow (F(\chi_\pi(x_1)), \ldots, F(\chi_\pi(x_\ell)))$

Hyb_4**:** In this game only $\boldsymbol{y}$ is changed with respect to Hyb_3.

- Samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv q \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$, computes $N \leftarrow pq$ and samples $x \xleftarrow{\$} [2^\lambda]$.
- Sets $B \leftarrow \lfloor 2^{3\lambda}/N \rfloor$
- Samples $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$, $w_i \xleftarrow{\$} [0, B-1], a'_i \leftarrow w_i N + x_i$, $a_i \leftarrow a'_i - x\ \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and samples $\boldsymbol{y} \xleftarrow{\$} \{\{0,1\}^2\}^\ell$

Hyb_5**:** In this game, the sampling of x_i and the creation of N are changed with respect to Hyb_4.

- Samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ such that $p \equiv q \equiv 5 \bmod 8$. Finds π such that $p = \pi\bar{\pi}$, computes $N \leftarrow pq$ and samples $x \xleftarrow{\$} [2^\lambda]$.
- Sets $B \leftarrow \lfloor 2^{3\lambda}/N \rfloor$
- Samples $x_i \xleftarrow{\$} \mathbb{Z}_N$, $w_i \xleftarrow{\$} [0, B-1]$. Computes $a'_i \leftarrow w_i N + x_i$ and $a_i \leftarrow a'_i - x, \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and samples $\boldsymbol{y} \xleftarrow{\$} \{\{0,1\}^2\}^\ell$

Hyb_6**:** In this game, only the sampling of a'_i changes with respect to Hyb_5. Furthermore, instead of sampling p and q only p is sampled.

- Sample $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ and find π such that $p = \pi\bar{\pi}$. Also sample $x \xleftarrow{\$} [2^\lambda]$
- Sample $a'_i \xleftarrow{\$} [2^{3\lambda}]$, computes $a_i \leftarrow a'_i - x, \forall\ i \in [\ell]$
- Sets $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and samples $\boldsymbol{y} \xleftarrow{\$} \{\{0,1\}^2\}^\ell$

Hyb_7**:** In this game, only sampling of a_i changes with respect to Hyb_6.

- Sample $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ and find π such that $p = \pi\bar{\pi}$. Also sample $x \xleftarrow{\$} [2^\lambda]$.
- Sample $a_i \xleftarrow{\$} [2^{3\lambda}], \forall\ i \in [\ell]$
- Set $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and sample $\boldsymbol{y} \xleftarrow{\$} \{\{0,1\}^2\}^\ell$

Claim 2 $\mathsf{Hyb}_0(\mathcal{A})$ *and* $\mathsf{Hyb}_1(\mathcal{A})$ *are statistically indistinguishable.*

Proof (Claim 2). Note that the sole distinction between the two games lies in the sampling procedure for the a_i. In Hyb_0, $a_i \xleftarrow{\$} [2^{3\lambda}]$ whereas in Hyb_1, $a_i \leftarrow a'_i - x$, where $a'_i \xleftarrow{\$} [2^{3\lambda}]$ and $x \in [2^\lambda]$. Notice that in both cases, the values $x+a_i$ are distributed almost uniformly over $[2^{3\lambda}]$, with only a negligible deviation arising from the possibility of wraparound when $x+a_i \geq 2^{3\lambda}$. Since $x \in [2^\lambda]$ and $a_i \in [2^{3\lambda}]$, wraparound can occur only if a_i lies in the interval $[2^{3\lambda} - 2^\lambda, 2^{3\lambda})$. This interval has size 2^λ, and hence the probability of wraparound is at most $\frac{2^\lambda}{2^{3\lambda}} = 2^{-2\lambda}$. Therefore, by the union bound the statistical distance between the two hybrids is bounded above by $\ell \cdot 2^{-2\lambda}$, which is negligible in λ. Now, in Hyb_1 we have $x + a_i = x + a'_i - x = a'_i$. Hence we can conclude that, the vector $\boldsymbol{y}$ in Hyb_1 is constructed exactly as in Hyb_0. □

Claim 3 *$\mathsf{Hyb}_1(\mathcal{A})$ and $\mathsf{Hyb}_2(\mathcal{A})$ are statistically indistinguishable.*

Proof (Claim 3). Notice that the only difference between the two games lies in the sampling procedure for the a'_i. In Hyb_1, $a'_i \xleftarrow{\$} [2^{3\lambda}]$ whereas in Hyb_2, $a'_i = w_i N + x_i$, where $w_i \xleftarrow{\$} [0, B-1]$ and $x_i \xleftarrow{\$} \mathbb{Z}_N$. By construction, the distribution of a'_i in Hyb_2 is $\mathcal{U}([0, N \cdot B - 1])$. In Hyb_2, we have $B = \lfloor 2^{3\lambda}/N \rfloor$, i.e. $2^{3\lambda} = NB + r$ where $r \in [0, N-1]$. Since $N < 2^{2\lambda}$, the statistical distance between $\mathcal{U}([2^{3\lambda}])$ and $\mathcal{U}([0, NB-1])$ is bounded by $r/2^{3\lambda} < N/2^{3\lambda} < 2^{-\lambda}$, which is a negligible function on λ. Therefore, by the union bound the statistical distance between the two hybrids is bounded above by $\ell \cdot 2^{-\lambda}$ which is negligible in λ. Now, in Hyb_2 we have $\forall\ i \in [\ell]$, $F(\chi_\pi(a'_i)) = F(\chi_\pi(w_i N + x_i)) = F(\chi_\pi(w_i N + x_i \bmod p)) = F(\chi_\pi(x_i))$, since $p \mid N$. Hence we can conclude that, the vector $\boldsymbol{y}$ in Hyb_2 is constructed exactly as in Hyb_1. □

Claim 4 *$\mathsf{Hyb}_2(\mathcal{A})$ and $\mathsf{Hyb}_3(\mathcal{A})$ are statistically indistinguishable.*

Proof (Claim 4). The sole distinction between the two games lies in the sampling procedure for the x_i. For all $i \in [\ell]$, in Hyb_2, $x_i \xleftarrow{\$} \mathbb{Z}_N$ whereas in Hyb_3, $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$. Since $N = pq$ is a product of two λ-bit primes, the statistical distance between these two distributions is equal to $1 - \phi(N)/N = (p+q-1)/N < 2^{-(\lambda-1)}$, which is negligible in λ. Therefore, by the union bound the statistical distance between the two hybrids is bounded above by $\ell \cdot 2^{1-\lambda}$, which is also negligible. □

Claim 5 *There exists an efficient adversary $\mathcal{B}$ for the ℓ-time full domain Quartic Problem where the following holds.*

$$\mathsf{QuarticAdv}_\ell[\mathcal{B}](\lambda) = |\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]|$$

Proof (Claim 5). Algorithm $\mathcal{B}$ works as follows. On input N and a challenge $((x_1, y_1), \ldots, (x_\ell, y_\ell))$, algorithm $\mathcal{B}$ samples $x \xleftarrow{\$} [2^\lambda]$ and for each $i \in [\ell]$ it also samples $w_i \xleftarrow{\$} [0, B-1]$, where $B = \lfloor 2^{3\lambda}/N \rfloor$. Sets $a'_i \leftarrow w_i N + x_i$ and $a_i \leftarrow a'_i - x\ \forall\ i \in [\ell]$. Algorithm $\mathcal{B}$ then runs algorithm $\mathcal{A}$ on inputs $\boldsymbol{a} \leftarrow (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} \leftarrow (y_1, \ldots, y_\ell)$ and finally outputs whatever $\mathcal{A}$ outputs.

Recall from the definition of ℓ-time full domain Quartic game, that the challenger samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$, sets $N = pq$ and samples $x_1, \ldots, x_\ell \xleftarrow{\$} \mathbb{Z}_N^\star$. The challenger then sends the ℓ-tuple $((x_1, y_1), \ldots, (x_\ell, y_\ell))$ along with N to the adversary, where y_i is either equal to $F(\chi_\pi(x_i))$ or $y_i \xleftarrow{\$} \{0,1\}^2$ for all $i \in [\ell]$. Now, if $y_i = F(\chi_\pi(x_i)), \forall\ i \in [\ell]$, then notice algorithm $\mathcal{B}$ perfectly simulates an execution of Hyb_3 and outputs 1 with probability $\Pr[\mathsf{Hyb}_3(\mathcal{A}) = 1]$. Else if $y_i \xleftarrow{\$} \{0,1\}^2\ \forall\ i \in [\ell]$, then algorithm $\mathcal{B}$ perfectly simulates an execution of Hyb_4 and outputs 1 with probability $\Pr[\mathsf{Hyb}_4(\mathcal{A}) = 1]$. Hence, the claim follows. □

Claim 6 *$\mathsf{Hyb}_4(\mathcal{A})$ and $\mathsf{Hyb}_5(\mathcal{A})$ are statistically indistinguishable.*

Proof (Claim 6). Same as Claim 4. □

Claim 7 *$\mathsf{Hyb}_5(\mathcal{A})$ and $\mathsf{Hyb}_6(\mathcal{A})$ are statistically indistinguishable.*

Proof (Claim 7). Same as Claim 3. □

Claim 8 *$\mathsf{Hyb}_6(\mathcal{A})$ and $\mathsf{Hyb}_7(\mathcal{A})$ are statistically indistinguishable.*

Proof (Claim 8). Same as Claim 2. □

From the above Claims 2, 3, 4, 5, 6, 7 and, 8 one can conclude that there exists a negligible function $\mathsf{negl}(\cdot)$ and an efficient adversary $\mathcal{B}$ such that for all $\lambda \in \mathbb{N}$, $\mathsf{wPRFAdv}[\mathcal{A}, \mathcal{Q}^{\mathsf{wPRF}}] = |\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_1(\mathcal{A}) = 1]| = \mathsf{QuarticAdv}_\ell[\mathcal{B}](\lambda) + \mathsf{negl}(\lambda)$. □

From Claim 1 and Corollary 1 the result of the theorem follows which states that breaking the hidden prime Quartic wPRF is as hard as the standard quadratic residuosity problem. □

Recall the definition of the family of generalized hidden prime quartic PRG $\mathcal{Q}_\ell^{\mathsf{PRG}}$ (see Definition 6). The next result shows that if there exists an efficient adversary that can break the security of the PRG $\mathcal{Q}_\ell^{\mathsf{PRG}}$ then there also exists an efficient adversary that can break the quadratic residuosity problem with respect to an RSA modulus.

Theorem 11. *($\mathsf{QR}_{\mathsf{RSA}} \leq_p \mathcal{Q}_\ell^{\mathsf{PRG}}$) For all polynomials $\ell = \mathsf{poly}(\lambda)$ and for every efficient algorithm $\mathcal{A}$ that breaks pseudorandomness of $\mathcal{Q}_\ell^{\mathsf{PRG}}$ with advantage $\mathsf{PRGAdv}[\mathcal{A}](\lambda)$, there exists an efficient algorithm $\mathcal{B}$ for the quadratic residuosity problem with advantage $\mathsf{QRAdv}[\mathcal{B}](\lambda)$ and a negligible function $\mathsf{negl}(\cdot)$ where the following holds.*

$$\mathsf{PRGAdv}[\mathcal{A}, \mathcal{Q}_\ell^{\mathsf{PRG}}](\lambda) \leq \ell \cdot \mathsf{QRAdv}[\mathcal{B}](\lambda) + \mathsf{negl}(\lambda)$$

Proof (Theorem 11). The idea of the proof is similar to that of [18, Theorem 4.5]. The result follows by combining the result of Theorem 10 and the result of the following Lemma 4. □

Lemma 4. *($\mathcal{Q}^{\mathsf{wPRF}} \leq_p \mathcal{Q}_\ell^{\mathsf{PRG}}$) For all polynomials $\ell = \mathsf{poly}(\lambda)$ and for every efficient algorithm $\mathcal{A}$ that breaks pseudorandomness of $\mathcal{Q}_\ell^{\mathsf{PRG}}$ with advantage $\mathsf{PRGAdv}[\mathcal{A}](\lambda)$, there exists an efficient ℓ query algorithm $\mathcal{B}$ that breaks weak pseudorandomness of $\mathcal{Q}^{\mathsf{wPRF}}$ with advantage $\mathsf{wPRFAdv}[\mathcal{B}](\lambda)$, where*

$$\mathsf{wPRFAdv}[\mathcal{B}, \mathcal{Q}^{\mathsf{wPRF}}](\lambda) \;=\; \mathsf{PRGAdv}[\mathcal{A}, \mathcal{Q}_\ell^{\mathsf{PRG}}](\lambda).$$

Proof (Lemma 4). We will use $\mathcal{A}$ to construct an algorithm $\mathcal{B}$ as follows. On input the function $Q_{\lambda,\ell}$, algorithm $\mathcal{B}$ makes ℓ queries to the wPRF oracle and obtains values $(a_1, y_1), \ldots, (a_\ell, y_\ell)$. Algorithm $\mathcal{B}$ then sets $\boldsymbol{a} = (a_1, \ldots, a_\ell)$ and $\boldsymbol{y} = (y_1, \ldots, y_\ell)$. It gives the function $Q_{\boldsymbol{a},\ell}$ and the vector $\boldsymbol{y}$ to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs.

From the definition of wPRF, recall that the challenger samples $a_i \xleftarrow{\$} [2^{3\lambda}], \forall\ i \in [\ell]$. This means that the function $Q_{\boldsymbol{a},\ell}$ is distributed uniformly over the set $\mathcal{Q}_{\lambda,\ell}$. Now consider the distribution of $y_1, \ldots, y_\ell$.

- If the challenger sets $y_i = F\left(\chi_\pi(x + a_i)\right)$, where $x \xleftarrow{\$} [2^\lambda]$ and $p \xleftarrow{\$} \mathsf{QPrimes}_\lambda$ are the components of the wPRF key, then $\boldsymbol{y} = Q_{\boldsymbol{a},\ell}(p, x)$. So Algorithm $\mathcal{B}$ perfectly simulates the pseudorandom distribution for $\mathcal{A}$ and outputs 1 with probability $\rho_{\mathcal{A},0}(\lambda)$, where $\rho_{\mathcal{A},0}(\lambda)$ is the function from definition of PRG (see Definition 3).
- On the other hand, if the challenger samples $y_i \xleftarrow{\$} \{0,1\}^2$, $\forall\ i \in [\ell]$, then algorithm $\mathcal{B}$ perfectly simulates the truly random distribution for $\mathcal{A}$ and ouputs 1 with probability $\rho_{\mathcal{A},1}(\lambda)$, where $\rho_{\mathcal{A},1}(\lambda)$ is defined in the definition of PRG (see Definition 3).

Hence one can conclude that, $\mathsf{wPRFAdv}[\mathcal{B}, \mathcal{Q}^{\mathsf{wPRF}}](\lambda) = |\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)| = \mathsf{PRGAdv}[\mathcal{A}, \mathcal{Q}_\ell^{\mathsf{PRG}}](\lambda)$. □

The following result does not require sampling over $\mathbb{J}_N \backslash \mathbb{QR}_N$ and as such one can work with $p \equiv 1 \bmod 4$ instead of $p \equiv 5 \bmod 8$. Given a prime $p \equiv 1 \bmod 4$, define a function $f : \mathbb{Z}_p^\star \to \{-1, 1\}^2$ as $f(n) := (\psi_\pi(n), \psi_{\bar{\pi}}(n))$, where $p = \pi\bar{\pi}$ and $\psi_\pi : \mathbb{Z}_p^\star \to \{-1, 1\}$, as $\psi_\pi(n) = \mathfrak{Re}(\chi_\pi(n)) + \mathfrak{Im}(\chi_\pi(n))$. Next, we define a decisional problem based on this function f, and show that this decisional problem is equivalent to the quartic residuosity problem defined earlier.

Definition 13 (Decisional f Problem (fD)). *Let $\mathcal{A}$ be a* PPT *algorithm, $\lambda \in \mathbb{N}$ denote the security parameter, and $b \in \{0,1\}$ a bit. Define the quantity $\rho_{\mathcal{A},b}(\lambda)$ as follows.*

$$\rho_{\mathcal{A},b}(\lambda) \;:=\; \Pr\left[\mathcal{A}(N, x, y_b) = 1 : \begin{array}{l} p, q \xleftarrow{\$} \mathsf{NBPrimes}_\lambda,\ N \leftarrow pq,\ p = \pi\bar{\pi} \\ x \xleftarrow{\$} \mathbb{J}_N \\ y_0 \leftarrow f(x),\ y_1 \xleftarrow{\$} \{-1,1\}^2 \end{array}\right].$$

Then the decisional f advantage *of $\mathcal{A}$ is defined as follows.*

$$\mathsf{fDAdv}[\mathcal{A}](\lambda) \;:=\; \left|\rho_{\mathcal{A},0}(\lambda) - \rho_{\mathcal{A},1}(\lambda)\right|.$$

Theorem 12 ($\mathsf{QuarticRD_Q} \equiv_p \mathsf{fD}$). *For every* PPT decisional f *adversary* $\mathcal{A}$, *there exists a* PPT decisional quartic residuosity *adversary for quartic integers* $\mathcal{B}$ *such that*

$$\mathsf{fDAdv}[\mathcal{A}](\lambda) \;=\; \mathsf{QuarticRDAdv}[\mathcal{B}](\lambda)$$

Proof (Theorem 12). We construct an Algorithm $\mathcal{B}$ that solves the decisional quartic residuosity problem given an adversary $\mathcal{A}$ that can solve the decisional f problem. On input a decisional quartic residuosity challenge (N, x, y) Algorithm $\mathcal{B}$ does the following. It checks the value of y. If $y = 00$, then $\mathcal{B}$ sets $z = (1, 1)$, else if $y = 01$, it sets $z = (1, -1)$. Otherwise, if $y = 10$, the algorithm sets the value of z to $(-1, -1)$ and finally, if $y = 11$, it sets $z = (-1, 1)$. Algorithm $\mathcal{B}$ then sends (N, x, z) to $\mathcal{A}$. It outputs whatever $\mathcal{A}$ outputs. Then,

$$\begin{aligned}
\rho_{\mathcal{B},1} &= \Pr[\mathcal{B}(N,x,y) = 1 \mid y \xleftarrow{\$} \{0,1\}^2] \;=\; \Pr[\mathcal{A}(N,x,z) = 1 \mid z \xleftarrow{\$} \{-1,1\}^2] \\
&= \rho_{\mathcal{A},1} \quad \text{and} \\
\rho_{\mathcal{B},0} &= \Pr[\mathcal{B}(N,x,y) = 1 \mid y \leftarrow F(\chi_\pi(x))] \\
&= \sum_{b_1,b_2 \in \{0,1\}} \Pr[\mathcal{A}(N,x,z) = 1 \wedge z = (1-2b_1, 1-2b_2) \mid y = b_2 b_1] \\
&= \Pr[\mathcal{A}(N,x,z) = 1 | z \leftarrow f(x)] \;=\; \rho_{\mathcal{A},0}
\end{aligned}$$

Conversely, we can construct an Algorithm $\mathcal{A}$ that solves the decisional f problem given an adversary $\mathcal{B}$ that can solve the decisional quartic residuosity problem. On input a decisional f challenge (N, x, z) Algorithm $\mathcal{A}$ does the following. If $z = (1, 1)$, then $\mathcal{A}$ sets $y = 00$, for $z = (1, -1)$ sets $y = 01$, if $z = (-1, -1)$ sets $y = 10$, for $z = (-1, 1)$ it sets $y = 11$ and runs $\mathcal{B}$ with input (N, x, y). Returns whatever $\mathcal{B}$ outputs. Then, by similar computations as above, we can show that $\rho_{\mathcal{A},j} = \rho_{\mathcal{B},j}$ for $j \in \{0, 1\}$. Therefore, combining these two results we get $\mathsf{fDAdv}[\mathcal{A}](\lambda) = \mathsf{QuarticRDAdv}[\mathcal{B}](\lambda)$. □

Remark 2. Theorem 12 shows that the problem of distinguishing f from random is polynomial time equivalent to the decisional quartic residuosity problem. Earlier we have established that the quadratic residuosity problem reduces in polynomial time to the decisional quartic residuosity problem. Together this means that the quadratic residuosity problem reduces to the decisional f problem. Notice that decisional f basically means that the adversary is given both the values of $\psi_\pi(\cdot)$ and $\psi_{\bar{\pi}}(\cdot)$. Instead, our pseudorandom sequence only outputs $\psi_\pi(\cdot)$. Therefore, it is clear that distinguishing our pseudorandom sequence $\psi_\pi(\cdot)$ from random is a harder problem than the decisional f problem.

5 Conclusion

We presented a dual analytic and cryptographic study of pseudorandomness derived from quartic Dirichlet characters. Analytically, without assuming GRH, we proved strong bounds on the well-distribution and correlation measures of sequences defined via ψ, meeting the Mauduit–Sárközy criteria. On the other

hand, we resolved an open problem of Damgård by showing quartic characters yield secure PRG and wPRF under polynomial-time reductions from Quadratic Residuosity.

6 Open Questions and Future Directions

While our results establish a solid foundation, several important directions remain open:

- **Beyond wPRFs.** We proved that quartic characters give rise to weak pseudorandom functions. One of the central open problems is whether these constructions can be strengthened further to *standard* (strong) pseudorandom functions, possibly under additional hardness assumptions.
- **Higher-order characters.** Our work extends the quadratic (Legendre) case to quartic characters. A natural next step is to investigate pseudorandomness from higher-order Dirichlet characters (e.g., cubic, quintic, octic), both in the analytic and cryptographic settings.
- **Quantum security.** While our reductions rely on the classical hardness of the Quadratic Residuosity problem, the quantum complexity corresponding to quartic variants remains largely unexplored. In particular, following the line of inquiry of [25], it is natural to ask whether allowing superposition queries enables a quantum attack against the quartic PRF.

7 Acknowledgements

The authors would like to acknowledge the valuable comments provided by the anonymous reviewers, which helped to improve the quality of the paper. The second author gratefully acknowledges the financial support of the Max Planck Institute for Mathematics, Bonn, during the preparation of this manuscript. The second author also thanks Prof. Ritabrata Munshi, Theoretical Statistics and Mathematics Division (TSMD), Indian Statistical Institute (ISI) Kolkata, for his valuable suggestions and for the financial support provided through the project grant during this work at ISI Kolkata. The third author would like to thank Prof. Debashish Goswami of TSMD, ISI Kolkata, for his financial support during the initial days of this project, as well as the Department of Mathematics and Statistics of IIT Kanpur for providing an excellent research environment. The third author is suppoFellowshipational Post Doctoral Fellowship PDF/2025/002316 from ANRF DST.

References

1. Aladov, N.: Sur la distribution des résidus quadratiques et non-quadratiques d'un nombre premier p dans la suite 1, 2, $\ldots, p-1$. Matematicheskii Sbornik **18**(1), 61–75 (1896)

2. Bach, E.: Realistic analysis of some randomized algorithms. In: Aho, A.V. (ed.) Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA. pp. 453–461. ACM (1987). https://doi.org/10.1145/28395.28444
3. Bach, E.: Realistic analysis of some randomized algorithms. J. Comput. Syst. Sci. **42**(1), 30–53 (1991). https://doi.org/10.1016/0022-0000(91)90038-7
4. Ball, M., Holmgren, J., Ishai, Y., Liu, T., Malkin, T.: On the complexity of decomposable randomized encodings, or: How friendly can a garbling-friendly PRF be? In: Vidick, T. (ed.) 11th Innovations in Theoretical Computer Science Conference, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020. pp. 86:1–86:22. LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPICS.ITCS.2020.86
5. Beullens, W., Beyne, T., Udovenko, A., Vitto, G.: Cryptanalysis of the legendre PRF and generalizations. IACR Trans. Symmetric Cryptol. **2020**(1), 313–330 (2020). https://doi.org/10.13154/TOSC.V2020.I1.313-330
6. Beullens, W., de Saint Guilhem, C.D.: Legroast: Efficient post-quantum signatures from the legendre PRF. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings. pp. 130–150. Lecture Notes in Computer Science, Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_8
7. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo random bits. In: 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982. pp. 112–117. IEEE Computer Society (1982). https://doi.org/10.1109/SFCS.1982.72
8. Brier, É., Ferradi, H., Joye, M., Naccache, D.: New number-theoretic cryptographic primitives. J. Math. Cryptol. **14**(1), 224–235 (2020). https://doi.org/10.1515/JMC-2019-0035
9. Burde, K.: Verteilungseigenschaften von Potenzresten. J. Reine Angev. Math. **249**, 133–172 (1971). https://doi.org/10.1515/crll.1971.249.133
10. Burgess, D.A.: On character sums and primitive roots. Proc. London Math. Soc. (3) **12**, 179–192 (1962). https://doi.org/10.1112/plms/s3-12.1.179
11. Burgess, D.A.: The Distribution of Quadratic Residues and Non-residues. Mathematika **4**(2), 106–112 (1957). https://doi.org/10.1112/S0025579300001157
12. Cassaigne, J., Ferenczi, S., Mauduit, C., Rivat, J., Sárközy, A.: On Finite Pseudorandom Binary Sequences III: The Liouville function, I. Acta Arithmetica **87**(4), 367–390 (1999). https://doi.org/10.4064/aa-87-4-367-390
13. Cassaigne, J., Ferenczi, S., Mauduit, C., Rivat, J., Sárközy, A.: On Finite Pseudorandom Binary sequences IV: The Liouville Function, II. Acta Arithmetica **95**(4), 343–359 (2000). https://doi.org/10.4064/aa-95-4-343-359
14. Cassaigne, J., Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences. VII. The Measures of Pseudorandomness. Acta Arith. **103**(2), 97–118 (2002). https://doi.org/10.4064/aa103-2-1
15. Caviness, B.F., Collins, G.E.: Algorithms for gaussian integer arithmetic. In: Jenks, R.D. (ed.) Proceedings of the third ACM Symposium on Symbolic and Algebraic Manipulation, SYMSAC 1976, Yorktown Heights, New York, USA, August 10-12, 1976. pp. 36–45. ACM (1976). https://doi.org/10.1145/800205.806321
16. Chaitin, G.J.: Information-theoretic limitations of formal systems. J. ACM **21**(3), 403–424 (1974). https://doi.org/10.1145/321832.321839

17. Cohen, H.: A course in computational algebraic number theory, Graduate texts in mathematics, vol. 138. Springer (1993). https://doi.org/10.1007/978-3-662-02945-9, https://www.worldcat.org/oclc/27810276
18. Corrigan-Gibbs, H., Wu, D.J.: The pseudorandomness of legendre symbols under the quadratic-residuosity assumption. In: Applebaum, B., Lin, H.R. (eds.) Theory of Cryptography - 23rd International Conference, TCC 2025, Aarhus, Denmark, December 1-5, 2025, Proceedings, Part III. pp. 298–322. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-3-032-12296-4_10
19. van Dam, W., Hallgren, S.: Efficient Quantum Algorithms for Shifted Quadratic Character Problems. CoRR **quant-ph/0011067** (2000), http://arxiv.org/abs/quant-ph/0011067
20. Damgård, I.: On the randomness of legendre and jacobi sequences. In: Goldwasser, S. (ed.) Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings. pp. 163–172. Lecture Notes in Computer Science, Springer (1988). https://doi.org/10.1007/0-387-34799-2_13
21. Davenport, H.: On the Distribution of Quadratic Residues (mod p). Journal of the London Mathematical Society **1**(1), 46–52 (1933). https://doi.org/10.1112/jlms/s1-8.1.46
22. Dummit, D.S., Foote, R.M.: Abstract algebra. Wiley, New York, 3rd ed edn. (2004)
23. Elliott, P.: A Restricted Mean Value Theorem. Journal of the London Mathematical Society **2**(1), 447–460 (1969). https://doi.org/10.1112/jlms/s2-1.1.447
24. Feist, D.: Legendre pseudo-random function (2019), https://legendreprf.dankrad.vercel.app/
25. Frixons, P., Schrottenloher, A.: Quantum Security of the Legendre PRF. Mathematical Cryptology **1**(2), 52–69 (2021)
26. Goldreich, O.: The Foundations of Cryptography - Volume 1: Basic Techniques. Cambridge University Press (2001). https://doi.org/10.1017/CBO9780511546891
27. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986). https://doi.org/10.1145/6490.6503
28. Goldreich, O., Krawczyk, H., Luby, M.: On the existence of pseudorandom generators. SIAM J. Comput. **22**(6), 1163–1175 (1993). https://doi.org/10.1137/0222069
29. Goldwasser, S., Micali, S.: Probabilistic encryption and how to play mental poker keeping secret all partial information. In: Lewis, H.R., Simons, B.B., Burkhard, W.A., Landweber, L.H. (eds.) Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA. pp. 365–377. ACM (1982). https://doi.org/10.1145/800070.802212
30. Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., Smart, N.P.: Mpc-friendly symmetric key primitives. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 430–443. ACM (2016). https://doi.org/10.1145/2976749.2978332
31. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999). https://doi.org/10.1137/S0097539793244708
32. Hudson, R.H.: On the First Occurrence of Certain Patterns of Quadratic Residues and Non-residues. Israel Journal of Mathematics **44**(1), 23–32 (1983). https://doi.org/10.1007/BF02763169

33. Iwaniec, H., Kowalski, E.: Analytic Number Theory. American Mathematical Society colloquium publications, American Mathematical Society (2004), https://books.google.co.in/books?id=CL8jDwAAQBAJ
34. Kaluđerović, N., Kleinjung, T., Kostić, D.: Cryptanalysis of the Generalised Legendre Pseudorandom Function. Open Book Series **4**(1), 267–282 (2020)
35. Khovratovich, D.: Key Recovery Attacks on the Legendre PRFs within the Birthday Bound. IACR Cryptol. ePrint Arch. p. 862 (2019), https://eprint.iacr.org/2019/862
36. Knuth, D.E.: The art of computer programming, , Volume III, 2nd Edition. Addison-Wesley (1998), https://www.worldcat.org/oclc/312994415
37. Kolmogorov, A.N.: On Tables of Random Numbers. Sankhyā: The Indian Journal of Statistics, Series A (1961-2002) **25**(4), 369–376 (1963), ISSN: 0581572X, http://www.jstor.org/stable/25049284
38. Kuperberg, G.: A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem. SIAM J. Comput. **35**(1), 170–188 (2005). https://doi.org/10.1137/S0097539703436345
39. Landrock, P.: A new concept in protocols: Verifiable computational delegation (position paper). In: Christianson, B., Crispo, B., Harbison, W.S., Roe, M. (eds.) Security Protocols, 6th International Workshop, Cambridge, UK, April 15-17, 1998, Proceedings. pp. 137–145. Lecture Notes in Computer Science, Springer (1998). https://doi.org/10.1007/3-540-49135-X_21
40. Levin, L.A.: One-way functions and pseudorandom generators. In: Sedgewick, R. (ed.) Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 363–365. ACM (1985). https://doi.org/10.1145/22145.22185
41. Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences. I. Measure of Pseudorandomness, the Legendre Symbol. Acta Arith. **82**(4), 365–377 (1997). https://doi.org/10.4064/aa-82-4-365-377
42. Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences: II. The Champernowne, Rudin–Shapiro, and Thue–Morse Sequences, A Further Construction. Journal of Number Theory **73**(2), 256–276 (1998). https://doi.org/10.1006/jnth.1998.2286
43. Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences, V. On (n α) and (n 2α) Sequences. Monatshefte für Mathematik **129**(3), 197–216 (2000). https://doi.org/10.1007/s006050050071
44. Mauduit, C., Sárközy, A.: On Finite Pseudorandom Binary Sequences, VI, (On Sequences). Monatshefte für Mathematik **130**(4), 281–298 (2000). https://doi.org/10.1007/s006050070028
45. Miller, G.L.: Riemann's hypothesis and tests for primality. In: Rounds, W.C., Martin, N., Carlyle, J.W., Harrison, M.A. (eds.) Proceedings of the 7th Annual ACM Symposium on Theory of Computing, May 5-7, 1975, Albuquerque, New Mexico, USA. pp. 234–239. ACM (1975). https://doi.org/10.1145/800116.803773
46. Monnerat, J., Oswald, Y.A., Vaudenay, S.: Optimization of the MOVA undeniable signature scheme. In: Dawson, E., Vaudenay, S. (eds.) Progress in Cryptology - Mycrypt 2005, First International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, September 28-30, 2005, Proceedings. pp. 196–209. Lecture Notes in Computer Science, Springer (2005). https://doi.org/10.1007/11554868_14

47. Monnerat, J., Vaudenay, S.: Undeniable signatures based on characters: How to sign with one bit. In: Bao, F., Deng, R.H., Zhou, J. (eds.) Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004. pp. 69–85. Lecture Notes in Computer Science, Springer (2004). https://doi.org/10.1007/978-3-540-24632-9_6
48. Montgomery, H.L., Vaughan, R.C.: Exponential sums with multiplicative coefficients. Invent. Math. **43**(1), 69–82 (1977). https://doi.org/10.1007/BF01390204
49. Montgomery, P.L.: Speeding the Pollard and Elliptic Curve Methods of Factorization. Mathematics of computation **48**(177), 243–264 (1987). https://doi.org/10.2307/2007888
50. Oon, S.M.: On pseudorandom properties of some Dirichlet characters. Ramanujan J. **15**(1), 19–30 (2008). https://doi.org/10.1007/s11139-007-9064-2
51. Peralta, R.: On the Distribution of Quadratic Residues and Non-residues Modulo a Prime Number. Mathematics of Computation **58**(197), 433–440 (1992)
52. Perron, O.: Bemerkungen über die verteilung der quadratischen reste. Mathematische Zeitschrift **56**(2), 122–130 (1952). https://doi.org/10.1007/BF01175029
53. Rabin, M.O.: Probabilistic algorithm for testing primality. Journal of Number Theory **12**(1), 128–138 (1980). https://doi.org/https://doi.org/10.1016/0022-314X(80)90084-0, https://www.sciencedirect.com/science/article/pii/0022314X80900840
54. Ross, S.M.: Simulation (2. ed.). Statistical modeling and decision science, Academic Press (1997), https://eprints.uad.ac.id/13/1/Sheldon_M._Ross_-_Simulation.pdf
55. de Saint Guilhem, C.D., Meyer, L.D., Orsini, E., Smart, N.P.: BBQ: using AES in picnic signatures. In: Paterson, K.G., Stebila, D. (eds.) Selected Areas in Cryptography - SAC 2019 - 26th International Conference, Waterloo, ON, Canada, August 12-16, 2019, Revised Selected Papers. pp. 669–692. Lecture Notes in Computer Science, Springer (2019). https://doi.org/10.1007/978-3-030-38471-5_27
56. Sárközy, A.: A finite pseudorandom binary sequence. Studia Sci. Math. Hungar. **38**, 377–384 (2001). https://doi.org/10.1556/SScMath.38.2001.1-4.28
57. Seres, I.A., Horváth, M., Burcsi, P.: The legendre pseudorandom function as a multivariate quadratic cryptosystem: security and applications. Appl. Algebra Eng. Commun. Comput. **36**(2), 223–253 (2025). https://doi.org/10.1007/S00200-023-00599-2
58. Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press (2006). https://doi.org/10.1017/CBO9781139165464
59. Solomonoff, R.J.: A formal theory of inductive inference. part I. Inf. Control. **7**(1), 1–22 (1964). https://doi.org/10.1016/S0019-9958(64)90223-2
60. Solomonoff, R.J.: A formal theory of inductive inference. part II. Inf. Control. **7**(2), 224–254 (1964). https://doi.org/10.1016/S0019-9958(64)90131-7
61. Weil, A.: On Some Exponential Sums. Proceedings of the National Academy of Sciences **34**(5), 204–207 (1948). https://doi.org/10.1073/pnas.34.5.204
62. Yao, A.C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982. pp. 80–91. IEEE Computer Society (1982). https://doi.org/10.1109/SFCS.1982.45

A Algorithms

A.1 Complex Division Algorithm

Algorithm 3 gives a method to divide a complex number α by β to get the complex quotient γ and a complex remainder ρ such that $\alpha = \gamma \cdot \beta + \rho$.

Algorithm 3 Division Algorithm in the Gaussian Integers $\mathbb{Z}[i]$

Input: $\alpha = x_1 + iy_1, \beta = x_2 + iy_2 \in \mathbb{Z}[i]$, with $\beta \neq 0$
Output: γ and ρ such that $\alpha = \beta\gamma + \rho$ with $\gamma, \rho \in \mathbb{Z}[i]$ and $\|\rho\| < \|\beta\|$

1: Compute $q \leftarrow \alpha \cdot \bar{\beta}/\|\beta\|$
2: Write $q = x + iy \in \mathbb{Q}(i)$ where $x = \frac{x_1x_2+y_1y_2}{\|\beta\|}, y = \frac{x_2y_1-x_1y_2}{\|\beta\|}$
3: Round x, y to nearest integers, i.e., $u \leftarrow \lfloor x \rceil$, $v \leftarrow \lfloor y \rceil$ such that $|x - u| \leq \frac{1}{2}$ and $|y - v| \leq \frac{1}{2}$
4: Set $\gamma \leftarrow u + iv \in \mathbb{Z}[i]$
5: Set $\rho \leftarrow \alpha - \beta\gamma$
6: **return** (γ, ρ)

A.2 Euclidean Algorithm for Complex Numbers

Algorithm 4 uses the Euclidean algorithm for complex numbers to find the greatest common divisor (gcd) of two complex numbers α and β.

Algorithm 4 Euclidean Algorithm for gcd in $\mathbb{Z}[i]$

Input: $\alpha, \beta \in \mathbb{Z}[i]$ such that both α and β are not zero simultaneously
Output: $d = \gcd(\alpha, \beta)$, correct up to multiplication by a unit in $\{\pm 1, \pm i\}$

1: **while** $\beta \neq 0$ **do**
2: Run Algorithm 3 on inputs α and β to get corresponding outpus γ and ρ such that $\alpha = \beta\gamma + \rho$ with $\|\rho\| < \|\beta\|$
3: Update $(\alpha, \beta) \leftarrow (\beta, \rho)$
4: **end while**
5: **return** α

A.3 Algorithm to Compute $\psi_\pi(n)$

Algorithm 5 gives the construction of $\psi_\pi(n)$. It takes as input a Gaussian prime π, an integer n and outputs the value of $\psi_\pi(n)$.

A.4 Computing a Gaussian Prime π

Algorithm 6 gives an efficient method for computing a Gaussian prime π given a rational prime $p \equiv 5 \bmod 8$.

Algorithm 5 Computation of $\psi_\pi(n)$

Input: A Gaussian prime π and integer $n \in \mathbb{Z}$
Output: $\psi_\pi(n) \in \{-1, 1\}$
1: Compute $p \leftarrow \pi\bar{\pi}$.
2: Call Algorithm 3 on inputs n and π to get (γ, ρ) such that $n = \gamma \cdot \pi + \rho$
3: **if** $\rho = 0$ **then**
4: **return** 0 // By convention
5: **end if**
6: Compute $\chi_\pi(n) \leftarrow n^{\frac{p-1}{4}} \bmod \pi$ using Square and Multiply Method [49]
// Quartic residue symbol
7: Parse $\chi_\pi(n)$ as a complex number $u + iv$ with $u, v \in \{-1, 0, 1\}$.
8: **return** $u + v$

Algorithm 6 Computing π such that $p = \pi\bar{\pi}$ for $p \equiv 5 \bmod 8$

Input: A prime $p \equiv 5 \bmod 8$
Output: A Gaussian prime $\pi = x + iy$ such that $p = \pi\bar{\pi}$
1: Compute $z \leftarrow 2^{\frac{p-1}{4}} \bmod p$ using Square and Multiply Method [49] //
$z^2 \equiv -1 \bmod p$
2: Compute $\pi \leftarrow \gcd(z + i,\, p)$ in $\mathbb{Z}[i]$ using Algorithm 4
3: **return** π

Correctness of Algorithm 6: Since $2 \in \mathbb{QNR}_p$, so $z^2 \equiv 2^{\frac{(p-1)}{2}} \equiv -1 \bmod p$ which implies that $p \mid (z^2+1)$ which further implies that $p \mid (z+i)(z-i)$. Let $p|(z+i)$, then we have $z + i = p(a + ib)$ for some $a + ib \in \mathbb{Z}[i]$. So we have $pb = 1$ (equating the coefficient of i), which is not possible. Hence $p \nmid (z + i)$. Similarly, one can also show that $p \nmid (z - i)$. Therefore, $\gcd(z + i, p)$ is nontrivial. Suppose $x + iy = \gcd(z + i, p)$. So $x + iy$ is a factor of p. Since $\mathbb{Z}[i]$ is a UFD and p is a product of two Gaussian primes, one can conclude that $x + iy$ is a Gaussian prime.

A.5 Quartic wPRF Q_λ

Algorithm 7 gives a simple algorithmic description of Definition 5 to generate the output of the function Q_λ, given the security parameter λ as input.

B Efficiency: Quartic vs. Quadratic

Let p be an λ-bit prime and $p = \pi\bar{\pi}$ in $\mathbb{Z}[i]$. Using the square-and-multiply method, exponentiation with a λ-bit exponent requires $(\lambda - 1)$ squaring and approximately $(\lambda/2-1)$ multiplications. Then for quadratic exponentiation, one needs to compute $n^{(p-1)/2} \bmod p$. Since $(p-1)/2$ is a $(\lambda-1)$-bit integer, therefore the total number of modular p multiplications required to compute $n^{(p-1)/2} \bmod p$ is

$$(\lambda - 2) + \frac{\lambda - 3}{2} \;=\; \frac{3\lambda - 7}{2}.$$

Algorithm 7 Generation of Q_λ

Input: Security parameter $\lambda \in \mathbb{N}$, inputs $x \in [2^\lambda]$, $a \in [2^{3\lambda}]$
Output: A prime $p \equiv 5 \bmod 8$, π and $Q_\lambda((p,x),a) \in \{0,1\}^2$ such that $p = \pi\bar{\pi}$

1: Generate a random λ-bit odd integer p
2: **while** $p \not\equiv 5 \bmod 8$ **or** p fails the Miller–Rabin [45,53] test **do**
3: Generate a new λ-bit odd integer p.
4: **end while**
5: Use Cornacchia's algorithm to find π such that $p = \pi\bar{\pi}$
6: Using Square & Multiply [49] and Algorithm 3 to compute $\chi_\pi(x+a) \equiv (x+a)^{\frac{p-1}{4}} \bmod \pi$.
7: **return** $((p,\pi); F(\chi_\pi(x+a))))$

Now, consider quartic exponentiation. In this case we use the following fact to reduce the number of multiplication over $\mathbb{Z}[i]$ modulo π, which is more costly than modular p multiplications, from $O(\log(p-1)/4)$ to just 1 in exchange for $O(\log(p-1)/4)$ many modular multiplications over $\mathbb{Z}_p$.

$$n^{(p-1)/4} \bmod \pi \;\equiv\; \left(n^{(p-1)/4} \bmod p\right) \bmod \pi.$$

Here $\log_2\left((p-1)/4\right) = (\lambda - 2)$, which gives

$$(\lambda - 3) + \frac{\lambda - 4}{2} \;=\; \frac{3\lambda - 10}{2}$$

many modular multiplications in $\mathbb{Z}_p$. The subsequent reduction modulo π requires one Gaussian division, which costs at most a constant number (at most 4) of ordinary λ-bit multiplications. Therefore, the total cost involved for one quartic PRF evaluation is given by $\frac{3\lambda-10}{2} + C$, where $C \leq 4$. Then the difference between the cost of one evaluation of the quartic PRF and that of the quadratic PRF is equal to

$$\left(\frac{3\lambda - 10}{2} + C\right) - \frac{3\lambda - 7}{2} \;=\; C - \frac{3}{2}.$$

Hence, both computations have essentially the same linear cost in λ, with only a constant-time overhead for the additional reduction modulo π. However, recall that one quartic PRF evaluation outputs 2-bits compto the ared 1-bit output of the quadratic PRF, thereby making the quartic PRF approximately twice aasficient than that of the quadratic PRF in generating $2l$ many output pseudorandom bits.

C Proofs

C.1 Fact 1

Since $\mathbb{Z}[i]$ is a UFD, every non-unit element can be expressed as a product of Gaussian primes uniquely up to the order of the factors and multiplication by

units. In particular, if there exists a Gaussian prime $\pi \in \mathbb{Z}[i]$ such that $p = \pi\bar{\pi}$, then any other factorization of p in $\mathbb{Z}[i]$ must involve one of the associates of π, namely $\pi, -\pi, i\pi, -i\pi$. We claim that for any β chosen from this set, the value of $\chi_\beta(\cdot)$ remains unchanged. Let us assume that, for some $\alpha \in \mathbb{Z}[i]$, using Algorithm 3 we get $\gamma, \rho \in \mathbb{Z}[i]$ such that $\alpha = \pi\gamma + \rho$, where $\|\rho\| < \|\beta\|$. Now, for the remaining choices of associates of π, the following cases arise (each satisfying the division criterion in $\mathbb{Z}[i]$, namely $|\rho| < |\beta|$).

- If $\beta = -\pi$, then write $\alpha = (-\pi)(-\gamma) + \rho$
- Otherwise, if $\beta = i\pi$, then write $\alpha = (i\pi)(-i\gamma) + \rho$
- Finally, if $\beta = -i\pi$, then write $\alpha = (-i\pi)(i\gamma) + \rho$

In each case, the remainder ρ remains unchanged. This establishes the claim. □

C.2 Fact 2

To prove the result, it is enough to show that the algorithm is correct, i.e., we show that for a given $n \in \mathbb{Z}_n^\star, \chi_\pi(n)^2 = \left(\frac{n}{p}\right)$. Recall that, since $\chi_\pi(\cdot)$ is a quartic character, therefore we have

$$\chi_\pi(n) \;=\; \left(\frac{n}{\pi}\right) \;\equiv\; n^{\frac{p-1}{4}} \bmod \pi \quad\Rightarrow\quad \chi_\pi(n)^2 \;\equiv\; n^{\frac{p-1}{2}} \bmod \pi.$$

Now $p = \pi\bar{\pi}$ and $\pi \mid \left(\chi_\pi(n)^2 - n^{\frac{p-1}{2}}\right)$, where $\chi_\pi(n)^2 - n^{\frac{p-1}{2}} \in \mathbb{Z}$. Therefore, $\pi \mid \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$ which implies that $\bar{\pi} \mid \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$. Since, both π and $\bar{\pi}$ are primes in $\mathbb{Z}[i]$, therefore we must have $p \mid \left(\chi_\pi(n)^2 - n^{\frac{p-1}{2}}\right)$, which implies $\chi_\pi(n)^2 = \left(\frac{n}{p}\right)$.

Note that $p \mid \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$ in $\mathbb{Z}[i]$ implies $p \mid \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$ in $\mathbb{Z}$, since $p \mid \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$ in $\mathbb{Z}[i]$ implies that there exists $\beta \in \mathbb{Z}[i]$ such that

$$\chi_\pi(n)^2 - n^{\frac{p-1}{2}} = p\beta$$

Since both $p, \chi_\pi(n)^2 - n^{\frac{p-1}{2}}$ are integers, taking complex conjugates on both sides yields

$$\chi_\pi(n)^2 - n^{\frac{p-1}{2}} \;=\; p\bar{\beta} \quad\Rightarrow\quad p\beta \;=\; p\bar{\beta} \quad\Rightarrow\quad \beta \;=\; \bar{\beta},$$

where the last equality follows from the fact that $\mathbb{Z}[i]$ is an Integral Domain. □

C.3 Fact 3

From Definition of χ_π, we have $\chi_\pi(n) = n^{\frac{p-1}{4}} \bmod \pi$, for any $n \in \mathbb{Z}$. This implies that there exists an $\alpha \in \mathbb{Z}[i]$ such that $n^{\frac{p-1}{4}} - \chi_\pi(n) = \pi\alpha$. Then taking the complex conjugate on both sides and observing the fact that $n^{\frac{p-1}{4}} \in \mathbb{Z}$, we get

$$\begin{aligned} & n^{\frac{p-1}{4}} - \bar{\chi}_\pi(n) = \bar{\pi}\bar{\alpha} \\ \Rightarrow\; & \bar{\chi}_\pi(n) = n^{\frac{p-1}{4}} \bmod \bar{\pi} \;=\; \chi_{\bar{\pi}}(n), \end{aligned}$$

by definition of $\chi_{\bar{\pi}}(n)$. □

C.4 Fact 4

By definition, $nm \equiv 1 \bmod p$, so $\left(\frac{nm}{\pi}\right) = 1$. Now multiplicative property of the quartic character yields that $\left(\frac{n}{\pi}\right)\left(\frac{m}{\pi}\right) = 1$ which further implies that one of the following conditions holds.

$$\left(\frac{n}{\pi}\right) = \left(\frac{m}{\pi}\right) = 1; \quad \left(\frac{n}{\pi}\right) = \left(\frac{m}{\pi}\right) = -1; \quad \left(\frac{n}{\pi}\right) = i \text{ and } \left(\frac{m}{\pi}\right) = -i;$$

$$\left(\frac{n}{\pi}\right) = -i \text{ and } \left(\frac{m}{\pi}\right) = i.$$

That is, depending on the value of $\left(\frac{n}{\pi}\right)$ (or equivalently, $\left(\frac{m}{\pi}\right)$) we have the followings four cases.

- If $\left(\frac{n}{\pi}\right) \in \{\pm 1\}$, then by Fact 3 we obtain

$$\left(\frac{n}{\bar{\pi}}\right) = \overline{\left(\frac{n}{\pi}\right)} = \left(\frac{n}{\pi}\right).$$

  Hence, $\left(\frac{n}{\bar{\pi}}\right) = \left(\frac{m}{\pi}\right)$.
- If $\left(\frac{n}{\pi}\right) \in \{\pm i\}$, then by Fact 3 we obtain

$$\left(\frac{n}{\bar{\pi}}\right) = \overline{\left(\frac{n}{\pi}\right)} = -\left(\frac{n}{\pi}\right).$$

  In this case as well, $\left(\frac{n}{\bar{\pi}}\right) = \left(\frac{m}{\pi}\right)$.

Combining all of these, one can conclude that $\left(\frac{n}{\bar{\pi}}\right) = \left(\frac{m}{\pi}\right)$ holds in general, where m is the multiplicative inverse of n in $\mathbb{Z}_p$ and $p = \pi\bar{\pi}$. □

C.5 Theorem 5

The Theorem 5 is similar to that of Water's proof that the quadratic residuosity problem with respect to RSA modulus reduces to the quadratic residuosity problem with respect to Blum integers (see [18, Theorem A.2]). However, for the sake of completeness, we present here the proof of Theorem 5.
For a bit $b \in \{0,1\}$, define the following two quantities.

$$\rho^{\mathsf{Blum}}_{\mathcal{A},b}(\lambda) := \Pr\left[\mathcal{A}(N, x_b) = 1 : \begin{array}{l} p, q \xleftarrow{\$} \mathsf{BPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\ x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right]$$

$$\rho^{\mathsf{NBlum}}_{\mathcal{A},b}(\lambda) := \Pr\left[\mathcal{A}(N, x_b) = 1 : \begin{array}{l} p, q \xleftarrow{\$} \mathsf{NBPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\ x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right]$$

Then by definition, the QR_{Q} advantage of an adversary $\mathcal{A}$ is given by

$$\mathsf{QRAdv}[\mathcal{A}](\lambda) \;=\; \left|\rho^{\mathsf{NBlum}}_{\mathcal{A},0}(\lambda) - \rho^{\mathsf{NBlum}}_{\mathcal{A},1}(\lambda)\right|.$$

We build efficient adversaries $\mathcal{B}_\beta$ for $\beta \in \{0,1\}$ using adversary $\mathcal{A}$ as a subroutine. The objective of $\mathcal{B}_\beta$ is to solve the $\mathsf{QR}_{\mathsf{RSA}}$ problem and accordingly receives the input (N, x) from the challenger, where $N = pq$ and with probability $1/2$, x belongs to the set $\mathbb{QR}_N$ or belongs to the set $\mathbb{J}_N \setminus \mathbb{QR}_N$. For $\beta \in \{0,1\}$ and for input (N, x), the algorithm $\mathcal{B}_\beta$ works as follows.

- If $N \equiv 3 \bmod 4$, then $\mathcal{B}_\beta$ outputs 0.
- Otherwise, for $N \equiv 1 \bmod 4$, the algorithm $\mathcal{B}_\beta$ runs algorithm $\mathcal{A}$ as a subroutine on input $(N, (-1)^\beta \cdot x)$ and outputs whatever the subroutine $\mathcal{A}$ returns.

It is obvious that if algorithm $\mathcal{A}$ is efficient, then so is algorithm $\mathcal{B}_\beta$. Next, we analyze the advantage of algorithm $\mathcal{B}_\beta$.

First, consider the case where $x \in \mathbb{QR}_N$. Then, depending on the value of $N \bmod 4$, we have the following two cases.

- **Case 1:** If $N \equiv 3 \bmod 4$, then recall that algorithm $\mathcal{B}_\beta$ always outputs 0.
- **Case 2:** For $N \equiv 1 \bmod 4$ we have the following two sub-cases depending on the values of $p \bmod 4$ and $q \bmod 4$.
  - **Sub-case 2.1:** Consider the case when $N = pq \equiv 1 \bmod 4$ and $p \equiv q \equiv 1 \bmod 4$. In this case, $-1 \in \mathbb{QR}_N$ since $-1 \in \mathbb{QR}_p$ and $-1 \in \mathbb{QR}_q$. So $(-1)^\beta x \in \mathbb{QR}_N$ and the distribution of $(-1)^\beta x$ is uniform over $\mathbb{QR}_N$ since $x \xleftarrow{\$} \mathbb{QR}_N$. Algorithm $\mathcal{B}_\beta$ therefore outputs 1 with probability $\rho^{\mathsf{NBlum}}_{\mathcal{A},0}(\lambda)$.
  - **Sub-case 2.2:** Next, consider the case when $N = pq \equiv 1 \bmod 4$ and $p \equiv q \equiv 3 \bmod 4$. In this case, $-1 \in \mathbb{J}_N \setminus \mathbb{QR}_N$ since $-1 \in \mathbb{QNR}_p$ and $-1 \in \mathbb{QNR}_q$. Now, if $\beta = 0$, then $(-1)^0 \cdot x = x$ is uniformly distributed in $\mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{QR}_N$. Alternatively, if $\beta = 1$, then $(-1)^1 \cdot x = -x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ So, the distribution of $-x$ is uniform on $\mathbb{J}_N \setminus \mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{QR}_N$. Therefore, $\mathcal{B}_\beta$ outputs 1 with probability $\rho^{\mathsf{Blum}}_{\mathcal{A},\beta}(\lambda)$.

Thus, the overall probability of the event that $\mathcal{B}_\beta$ outputs 1 given $x \in \mathbb{QR}_N$ is given by

$$\begin{aligned}
\rho_{\mathcal{B}_\beta,0}(\lambda) \;&=\; \Pr\left[\mathcal{B}_\beta(N,x_0)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{Primes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N \end{array}\right] \\
&= \Pr\left[\mathcal{B}_\beta(N,x_0)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{Primes}_\lambda \\ N \leftarrow pq \\ N \equiv 3 \bmod 4 \\ x_0 \xleftarrow{\$} \mathbb{QR}_N \end{array}\right] \cdot \Pr[N \equiv 3 \bmod 4] + \\
&\quad \Pr\left[\mathcal{B}_\beta(N,x_0)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{NBPrimes}_\lambda \\ N \leftarrow pq \\ N \equiv 1 \bmod 4 \\ x_0 \xleftarrow{\$} \mathbb{QR}_N \end{array}\right] \cdot \Pr[p \equiv q \equiv 1 \bmod 4] + \\
&\quad \Pr\left[\mathcal{B}_\beta(N,x_0)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{BPrimes}_\lambda \\ N \leftarrow pq \\ N \equiv 1 \bmod 4 \\ x_0 \xleftarrow{\$} \mathbb{QR}_N \end{array}\right] \cdot \Pr[p \equiv q \equiv 3 \bmod 4] \\
&= \delta(\lambda)^2 \cdot \rho^{\mathsf{NBlum}}_{\mathcal{A},0}(\lambda) + (1-\delta(\lambda))^2 \cdot \rho^{\mathsf{Blum}}_{\mathcal{A},\beta}(\lambda)
\end{aligned}$$

where the third equality follows from the fact that

$$\Pr\left[\mathcal{B}(N,x_0)=1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{Primes}_\lambda \\ N \leftarrow pq \\ N \equiv 3 \bmod 4 \\ x_0 \xleftarrow{\$} \mathbb{QR}_N \end{array}\right] = 0 \quad \text{[By Case 1].}$$

Next, suppose $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$. Again, depending on the value of N mod 4 we have the following two cases.

- **Case 1:** If $N \equiv 3 \bmod 4$, then recall that algorithm $\mathcal{B}_\beta$ always outputs 0.
- **Case 2:** For $N \equiv 1 \bmod 4$ we have the following two sub-cases depending on the values of p mod 4 and q mod 4.
  - **Sub-case 2.1:** For the case when $N = pq \equiv 1 \bmod 4$ and $p \equiv q \equiv 1 \bmod 4$, $-1 \in \mathbb{QR}_N$ since $-1 \in \mathbb{QR}_p$ and $-1 \in \mathbb{QR}_q$. So $(-1)^\beta \cdot x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ and the distribution of $(-1)^\beta \cdot x$ is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$ since $x \xleftarrow{\$} \mathbb{QR}_N$. Algorithm $\mathcal{B}_\beta$ therefore outputs 1 with probability $\rho^{\mathsf{NBlum}}_{\mathcal{A},1}(\lambda)$.
  - **Sub-case 2.2:** Next, for the case when $N = pq \equiv 1 \bmod 4$ and $p \equiv q \equiv 3 \bmod 4$, $-1 \in \mathbb{J}_N \setminus \mathbb{QR}_N$ since $-1 \in \mathbb{QNR}_p$ and $-1 \in \mathbb{QNR}_q$. Now, if $\beta = 0$, then $(-1)^0 \cdot x = x$ is uniformly distributed in $\mathbb{J}_N \setminus \mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$. Alternatively, if $\beta = 1$, then $(-1)^1 \cdot x = -x \in \mathbb{QR}_N$ So,

the distribution of $-x$ is uniform on $\mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$. Therefore, $\mathcal{B}_\beta$ outputs 1 with probability $\rho^{\mathsf{Blum}}_{\mathcal{A},1-\beta}(\lambda)$.

Thus, the algorithm $\mathcal{B}_\beta$ outputs 1 with probability $\rho^{\mathsf{Blum}}_{\mathcal{A},1-\beta}(\lambda)$ whenever $N \equiv 1 \bmod 4$.

By similar arguments, one can show that the overall probability of the event that $\mathcal{B}_\beta$ outputs 1 given that $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ is given by

$$\rho_{\mathcal{B}_\beta,1}(\lambda) = \delta(\lambda)^2 \cdot \rho^{\mathsf{NBlum}}_{\mathcal{A},1}(\lambda) + (1-\delta(\lambda))^2 \cdot \rho^{\mathsf{Blum}}_{\mathcal{A},1-\beta}(\lambda).$$

Next, define the following functions.

$$\varepsilon_{\mathsf{Blum}}(\lambda) \;:=\; \rho^{\mathsf{Blum}}_{\mathcal{A},0}(\lambda) - \rho^{\mathsf{Blum}}_{\mathcal{A},1}(\lambda) \quad \text{and} \quad \varepsilon_{\mathsf{NBlum}}(\lambda) \;:=\; \rho^{\mathsf{NBlum}}_{\mathcal{A},0}(\lambda) - \rho^{\mathsf{NBlum}}_{\mathcal{A},1}(\lambda).$$

Then the advantages of the two algorithms $\mathcal{B}_0$ and $\mathcal{B}_1$ can then be expressed as follows.

$$\begin{aligned}
\mathsf{QRAdv}_{\mathsf{RSA}}[\mathcal{B}_0](\lambda) &= |\rho_{\mathcal{B}_0,0}(\lambda) - \rho_{\mathcal{B}_0,1}(\lambda)| \\
&= |\delta(\lambda)^2 \cdot \varepsilon_{\mathsf{NBlum}}(\lambda) + (1-\delta(\lambda))^2 \cdot \varepsilon_{\mathsf{Blum}}(\lambda)|, \\
\mathsf{QRAdv}_{\mathsf{RSA}}[\mathcal{B}_1](\lambda) &= |\rho_{\mathcal{B}_1,0}(\lambda) - \rho_{\mathcal{B}_1,1}(\lambda)| \\
&= |\delta(\lambda)^2 \cdot \varepsilon_{\mathsf{NBlum}}(\lambda) - (1-\delta(\lambda))^2 \cdot \varepsilon_{\mathsf{Blum}}(\lambda)|.
\end{aligned}$$

Notice that the sign of $\varepsilon_{\mathsf{NBlum}}(\lambda)$ either matches the sign of $\varepsilon_{\mathsf{Blum}}(\lambda)$ or the sign of $-\varepsilon_{\mathsf{Blum}}(\lambda)$. Therefore we conclude that there exists $\beta \in \{0,1\}$ such that

$$\mathsf{QRAdv}_{\mathsf{RSA}}[\mathcal{B}_\beta](\lambda) \;\geq\; \delta(\lambda)^2 \cdot |\varepsilon_{\mathsf{NBlum}}(\lambda)| \;=\; \delta(\lambda)^2 \cdot \mathsf{QRAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda).$$

□

C.6 Theorem 6

Theorem 6 is again similar to that of Water's proof [18, Theorem A.2]. However, for the sake of completeness, we present the proof of Theorem 6 here. Let $\mathcal{A}$ be an efficient quadratic-residuosity adversary with respect to Quartic integers. Let

$$\mathsf{NQPrimes}_\lambda \;:=\; \mathsf{NBPrimes}_\lambda \setminus \mathsf{QPrimes}_\lambda$$

denote the set of *non-Quartic primes*, i.e., the primes p such that $p \equiv 1 \bmod 8$. For a bit $b \in \{0,1\}$, define the following two quantities.

$$\rho^{\mathsf{Quartic}}_{\mathcal{A},b}(\lambda) := \Pr\left[\mathcal{A}(N,x_b) = 1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{QPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\; x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right]$$

$$\rho^{\mathsf{NQuartic}}_{\mathcal{A},b}(\lambda) := \Pr\left[\mathcal{A}(N,x_b) = 1 : \begin{array}{l} p,q \xleftarrow{\$} \mathsf{NQPrimes}_\lambda \\ N \leftarrow pq \\ x_0 \xleftarrow{\$} \mathbb{QR}_N,\; x_1 \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N \end{array}\right]$$

Then by definition, the Quadratic Residuosity advantage of an adversary $\mathcal{A}$ over Quartic integers is given by

$$\mathsf{QRAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda) \;=\; \left|\rho_{\mathcal{A},0}^{\mathsf{Quartic}}(\lambda) - \rho_{\mathcal{A},1}^{\mathsf{Quartic}}(\lambda)\right|$$

Next, we build efficient adversaries $\mathcal{B}_\beta$ for $\beta \in \{0,1\}$ using adversary $\mathcal{A}$ as a subroutine. The objective of $\mathcal{B}_\beta$ is to solve the QR_{Q} problem and accordingly receives the input (N, x) from the challenger, where $N = pq$ such that $p \equiv q \equiv 1 \bmod 4$ and with probability $1/2$, x belongs to the set $\mathbb{QR}_N$ or belongs to the set $\mathbb{J}_N \setminus \mathbb{QR}_N$. For $\beta \in \{0,1\}$ and for input (N, x), the algorithm $\mathcal{B}_\beta$ works as follows.

- If $N \equiv 5 \bmod 8$, then $\mathcal{B}_\beta$ outputs 0.
- Otherwise, for $N \equiv 1 \bmod 8$, the algorithm $\mathcal{B}_\beta$ runs algorithm $\mathcal{A}$ on input $(N, 2^\beta \cdot x)$.
- Finally $\mathcal{B}_\beta$ outputs whatever the subroutine $\mathcal{A}$ returns.

It is very easy to check that if algorithm $\mathcal{A}$ is efficient, then algorithm $\mathcal{B}_\beta$ is also efficient. In the following, we analyze the advantage of algorithm $\mathcal{B}_\beta$.

First, consider the case when $x \in \mathbb{QR}_N$ where $p \equiv q \equiv 1 \bmod 4$. Depending on the value of N mod 8 we have the following two cases.

- **Case 1:** Recall that for $N \equiv 5 \bmod 8$ algorithm $\mathcal{B}_\beta$ always outputs 0.
- **Case 2:** The case when $N \equiv 1 \bmod 8$ can be further sub-divided into the following two sub-cases.
  - **Sub-case 2.1:** For the case when $p \equiv q \equiv 1 \bmod 8$, notice that $2 \in \mathbb{QR}_N$ since $2 \in \mathbb{QR}_p$ and $2 \in \mathbb{QR}_q$. So the distribution of $2^\beta \cdot x$ is uniform over $\mathbb{QR}_N$. Then algorithm $\mathcal{B}_\beta$ outputs 1 with probability $\rho_{\mathcal{A},0}^{\mathsf{NQuartic}}(\lambda)$.
  - **Sub-case 2.2:** On the other hand, when $p \equiv q \equiv 5 \bmod 8$, $2 \in \mathbb{J}_N \setminus \mathbb{QR}_N$ since $2 \in \mathbb{QNR}_p$ and $2 \in \mathbb{QNR}_q$. For $\beta = 0$, $2^0 \cdot x = x \in \mathbb{QR}_N$ is therefore uniformly distributed in $\mathbb{QR}_N$. Whereas for $\beta = 1$, $2^1 \cdot x = 2x \in \mathbb{J}_N \setminus \mathbb{QR}_N$. So the distribution of $2x$ is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{QR}_N$. Hence, $\mathcal{B}_\beta$ outputs 1 with probability $\rho_{\mathcal{A},\beta}^{\mathsf{Quartic}}(\lambda)$.

Therefore the probability that $\mathcal{B}_\beta$ outputs 1 given $x \in \mathbb{QR}_N$ is given by

$$\rho_{\mathcal{B}_\beta,0}(\lambda) \;=\; (1-\delta(\lambda))^2 \cdot \rho_{\mathcal{A},0}^{\mathsf{NQuartic}}(\lambda) + \delta(\lambda)^2 \cdot \rho_{\mathcal{A},\beta}^{\mathsf{Quartic}}(\lambda)$$

Next, consider the case when $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$. We again consider the following two cases.

- **Case 1:** Again, recall that for $N \equiv 5 \bmod 8$ algorithm $\mathcal{B}_\beta$ always outputs 0.
- **Case 2:** The case when $N \equiv 1 \bmod 8$ can be further sub-divided into the following two sub-cases.
  - **Sub-case 2.1:** For $p \equiv q \equiv 1 \bmod 8$, notice that the distribution of $2^\beta x$ is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$ since $2 \in \mathbb{QR}_p$, $2 \in \mathbb{QR}_q$ and $x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$. So algorithm $\mathcal{B}_\beta$ outputs 1 with probability $\rho_{\mathcal{A},1}^{\mathsf{NQuartic}}(\lambda)$.

- **Case 3:** Otherwise, when $p \equiv q \equiv 5 \bmod 8$, $2 \in \mathbb{QNR}_p$ and $2 \in \mathbb{QNR}_q$. Then for $\beta = 0$ the distribution of $2^0 \cdot x = x$ is uniform over $\mathbb{J}_N \setminus \mathbb{QR}_N$ as $x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N$. Similarly, for $\beta = 1$, then $2^1 \cdot x = 2x \in \mathbb{QR}_N$ is uniformly distributed over $\mathbb{QR}_N$.

Combining we get that the algorithms $\mathcal{B}_\beta$ outputs 1 with probability $\rho^{\mathsf{Quartic}}_{\mathcal{A},1-\beta}(\lambda)$.

Therefore the probability that $\mathcal{B}_\beta$ outputs 1 given $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ is given by

$$\rho_{\mathcal{B},1}(\lambda) \;=\; (1-\delta(\lambda))^2 \cdot \rho^{\mathsf{NQuartic}}_{\mathcal{A},1}(\lambda) + \delta(\lambda)^2 \cdot \rho^{\mathsf{Quartic}}_{\mathcal{A},1-\beta}(\lambda).$$

Next, define the functions

$$\varepsilon_{\mathsf{Quartic}}(\lambda) \;:=\; \rho^{\mathsf{Quartic}}_{\mathcal{A},0}(\lambda) - \rho^{\mathsf{Quartic}}_{\mathcal{A},1}(\lambda), \quad \varepsilon_{\mathsf{NQuartic}}(\lambda) \;:=\; \rho^{\mathsf{NQuartic}}_{\mathcal{A},0}(\lambda) - \rho^{\mathsf{NQuartic}}_{\mathcal{A},1}(\lambda)$$

Then the advantage of the two algorithms $\mathcal{B}_0$ and $\mathcal{B}_1$ are given as follows.

$$\begin{aligned}
\mathsf{QRAdv}_{\mathsf{S}}[\mathcal{B}_0](\lambda) &= |\rho_{\mathcal{B}_0,0}(\lambda) - \rho_{\mathcal{B}_0,1}(\lambda)| \\
&= |(1-\delta(\lambda))^2 \cdot \varepsilon_{\mathsf{NQuartic}}(\lambda) + \delta(\lambda)^2 \cdot \varepsilon_{\mathsf{Quartic}}(\lambda)|, \\
\mathsf{QRAdv}_{\mathsf{S}}[\mathcal{B}_1](\lambda) &= |\rho_{\mathcal{B}_1,0}(\lambda) - \rho_{\mathcal{B}_1,1}(\lambda)| \\
&= |(1-\delta(\lambda))^2 \cdot \varepsilon_{\mathsf{NQuartic}}(\lambda) - \delta(\lambda)^2 \cdot \varepsilon_{\mathsf{Quartic}}(\lambda)|.
\end{aligned}$$

Notice that the sign of $\varepsilon_{\mathsf{Quartic}}(\lambda)$ either matches the sign of $\varepsilon_{\mathsf{NQuartic}}(\lambda)$ or the sign of $-\varepsilon_{\mathsf{NQuartic}}(\lambda)$. Therefore we conclude that there exists $\beta \in \{0,1\}$ such that

$$\mathsf{QRAdv}_{\mathsf{S}}[\mathcal{B}_\beta](\lambda) \geq \delta(\lambda)^2 \cdot |\varepsilon_{\mathsf{Quartic}}(\lambda)| = \delta(\lambda)^2 \cdot \mathsf{QRAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda).$$

□

C.7 Theorem 7

Theorem 7 is similar to the proof of [18, Lemma 3.2]. However, for the sake of completeness we present the proof of Theorem 7 here. On input a quadratic-residuosity challenge (N, x) for quartic integers Algorithm $\mathcal{B}$ behaves as follows.

- It samples a bit $y \xleftarrow{\$} \{-1, 1\}$.
- Suppose $\hat{b} \leftarrow \mathcal{A}(N, x, y)$. Without loss of gnerality, assume that on all inputs (N, x, y), algorithm $\mathcal{A}$ outputs a bit $\hat{b} \in \{0, 1\}$. Note that any non-conforming algorithm can be transformed into one with this property without decreasing its advantage.
- If $\hat{b} = 1$, then Algorithm $\mathcal{B}$ outputs 1 if $y = 1$, otherwise if $y = -1$ it outputs 0.
- Otherwise, if $\hat{b} = 0$, then Algorithm $\mathcal{B}$ outputs 1 if $y = -1$, else if $y = 1$ it outputs 0.

To analyze the advantage of algorithm $\mathcal{B}$, we first define the following quantities, for each $z \in \{-1, 1\}$.

$$\begin{aligned}
\nu_{\mathbb{QR},z}(\lambda) &= \Pr[\mathcal{A}(N, x, z) = 1 : p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda,\ N \leftarrow pq,\ x \xleftarrow{\$} \mathbb{QR}_N] \\
\nu_{\mathbb{QNR},z}(\lambda) &= \Pr[\mathcal{A}(N, x, z) = 1 : p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda,\ N \leftarrow pq,\ x \xleftarrow{\$} \mathbb{J}_N \setminus \mathbb{QR}_N]
\end{aligned}$$

Then,

$$\begin{aligned}\rho_{\mathcal{A},0}(\lambda) =& \Pr\left[\mathcal{A}(N,x,y_0)=1 \wedge y_0 \leftarrow \left(\frac{x}{p}\right) \wedge x \xleftarrow{\$} \mathbb{J}_N\right] \\ =& \frac{1}{2}\left(\nu_{\mathbb{QR},1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)\right),\end{aligned}$$

since $y_0 = \left(\frac{x}{p}\right) = 1$ if $x \in \mathbb{QR}_N$, $y_0 = \left(\frac{x}{p}\right) = -1$ if $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ and $\Pr[x \in \mathbb{J}_N \setminus \mathbb{QR}_N] = \Pr[x \in \mathbb{QR}_N] = \frac{1}{2}$. Similarly,

$$\begin{aligned}&\rho_{\mathcal{A},1}(\lambda) \;=\; \Pr\left[\mathcal{A}(N,x,y_1)=1 \wedge y_1 \xleftarrow{\$} \{-1,1\} \wedge x \xleftarrow{\$} \mathbb{J}_N\right] \\ &= \frac{1}{4}\cdot\Big(\Pr\left[\mathcal{A}(N,x,-1)=1 \mid x \in \mathbb{QR}_N\right] + \Pr\left[\mathcal{A}(N,x,-1)=1 \mid x \in \mathbb{J}_N \setminus \mathbb{QR}_N\right]\Big) + \\ &\quad \frac{1}{4}\Big(\Pr\left[\mathcal{A}(N,x,1)=1 \mid x \in \mathbb{QR}_N\right] + \Pr\left[\mathcal{A}(N,x,1)=1 \mid x \in \mathbb{J}_N \setminus \mathbb{QR}_N\right]\Big) \\ &= \frac{1}{4}\Big(\nu_{\mathbb{QR},-1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)+\nu_{\mathbb{QR},1}(\lambda)+\nu_{\mathbb{QNR},1}(\lambda)\Big).\end{aligned}$$

Putting these values in the definition of $\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda)$ we get

$$\begin{aligned}&\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda) \;=\; |\rho_{\mathcal{A},0}-\rho_{\mathcal{A},1}| \\ &= \left|\frac{1}{2}\Big(\nu_{\mathbb{QR},1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)\Big) - \frac{1}{4}\Big(\nu_{\mathbb{QR},-1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)+\nu_{\mathbb{QR},1}(\lambda)+\nu_{\mathbb{QNR},1}(\lambda)\Big)\right| \\ &= \frac{1}{4}\left|\nu_{\mathbb{QR},1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)-\nu_{\mathbb{QR},-1}(\lambda)-\nu_{\mathbb{QNR},1}(\lambda)\right|\end{aligned}$$

Now, consider a quadratic residuosity challenge (N,x) for quartic integers for the Algorithm $\mathcal{B}$. Recall that Algorithm $\mathcal{B}$ outputs 1 when the following events occur.

- $\mathcal{A}(N,x,y)=1$ and $y=1$ or
- $\mathcal{A}(N,x,y)=0$ and $y=-1$

Then the probability that Algorithm $\mathcal{B}$ outputs 1 given that $x \in \mathbb{QR}_N$ is given by

$$\begin{aligned}&\Pr\left[\mathcal{B}(N,x)=1 \mid x \in \mathbb{QR}_N\right] \\ &= \frac{1}{2}\cdot\Big(\Pr\left[\mathcal{A}(N,x,1)=1 \mid x \in \mathbb{QR}_N\right] + \Pr\left[\mathcal{A}(N,x,-1)\to 0 \mid x \in \mathbb{QR}_N\right]\Big) \\ &= \frac{1}{2}+\frac{1}{2}\cdot\Big(\nu_{\mathbb{QR},1}(\lambda)-\nu_{\mathbb{QR},-1}(\lambda)\Big)\end{aligned}$$

Similarly, the probability that Algorithm $\mathcal{B}$ outputs 1 given that $x \in \mathbb{J}_N \setminus \mathbb{QR}_N$ is given by

$$\Pr\left[\mathcal{B}(N,x)=1 \mid x \in \mathbb{J}_n \setminus \mathbb{QR}_N\right] \;=\; \frac{1}{2}+\frac{1}{2}\cdot\Big(\nu_{\mathbb{QNR},1}(\lambda)-\nu_{\mathbb{QNR},-1}(\lambda)\Big).$$

Then

$$\begin{aligned}\mathsf{QRAdv}_{\mathsf{Q}}[\mathcal{B}](\lambda) &= |\Pr[\mathcal{B}\to 1|x\in\mathbb{QR}_N]-\Pr[\mathcal{B}\to 1|x\in\mathbb{J}_N\setminus\mathbb{QR}_N]|\\ &= \frac{1}{2}\cdot|\nu_{\mathbb{QR},1}(\lambda)-\nu_{\mathbb{QR},-1}(\lambda)-\nu_{\mathbb{QNR},1}(\lambda)+\nu_{\mathbb{QNR},-1}(\lambda)|\\ &= 2\cdot\mathsf{QRDAdv}_{\mathsf{Q}}[\mathcal{A}](\lambda)\end{aligned}$$

□

C.8 Lemma 3

Observe that to show that the distributions are identical it is enough to show that the $x = urz_0^{\rho_0}z_1^{\rho_1}$ of the left hand side distribution is uniformly distributed over $\mathbb{Z}_N^\star$. In order to show this we proceed as follows. For fixed elements $z_0, z_1 \in \mathbb{Z}_N^\star$ and $u \in \mathbb{J}_N$, define the map

$$T_{z_0,z_1,u} : \mathbb{QR}_N \times \{0,1\}\times\{0,1\} \to \mathbb{Z}_N^\star \quad \text{such that} \quad (r,\rho_0,\rho_1) \mapsto u\cdot r\cdot z_0^{\rho_0}z_1^{\rho_1}.$$

We claim that $T_{z_0,z_1,u}$ is a bijection. Indeed, observe that the domain has size $|\mathbb{QR}_N|\cdot 2\cdot 2 = \frac{\phi(N)}{4}\cdot 4 = \phi(N)$, which is equal to the size of the co-domain $|\mathbb{Z}_N^\star| = \phi(N)$. Since the domain and co-domain are of the same cardinality, to prove a bijection, it is enough to show that it is injective. Consider two inputs (r,ρ_0,ρ_1) and (r',ρ_0',ρ_1') in the domain such that

$$T_{z_0,z_1,u}(r,\rho_0,\rho_1) = T_{z_0,z_1,u}(r',\rho_0',\rho_1'),$$

which implies that

$$urz_0^{\rho_0}z_1^{\rho_1} \equiv ur'z_0^{\rho_0'}z_1^{\rho_1'} \bmod N.$$

Since, $u \in \mathbb{J}_N \subset \mathbb{Z}_N^\star$ and $r, z_0, z_1 \in \mathbb{Z}_N^\star$, we get $r(r')^{-1}z_0^{\rho_0-\rho_0'}z_1^{\rho_1-\rho_1'} \equiv 1 \in \mathbb{Z}_N^\star$. which implies that

$$r(r')^{-1}z_0^{\rho_0-\rho_0'}z_1^{\rho_1-\rho_1'} \equiv 1 \in \mathbb{Z}_p^\star \quad \text{and} \quad r(r')^{-1}z_0^{\rho_0-\rho_0'}z_1^{\rho_1-\rho_1'} \equiv 1 \in \mathbb{Z}_q^\star$$

which further implies that

$$\left(\frac{r(r')^{-1}}{p}\right)\left(\frac{z_0^{\rho_0-\rho_0'}}{p}\right)\left(\frac{z_1^{\rho_1-\rho_1'}}{p}\right) = \left(\frac{1}{p}\right) = 1 = \left(\frac{1}{q}\right) = \left(\frac{r(r')^{-1}}{q}\right)\left(\frac{z_0^{\rho_0-\rho_0'}}{q}\right)\left(\frac{z_1^{\rho_1-\rho_1'}}{q}\right) \quad (5)$$

Since $z_1 \in \mathbb{QR}_q$, so we have $\left(\frac{z_1^{\rho_1-\rho_1'}}{q}\right) = 1$. Also $r, r' \in \mathbb{QR}_N \Rightarrow r(r')^{-1} \in \mathbb{QR}_N \Rightarrow r(r')^{-1} \in \mathbb{QR}_q \Rightarrow \left(\frac{r(r')^{-1}}{q}\right) = 1$. Using this in the right hand side of (5) we get $(-1)^{\rho_0-\rho_0'} = 1$ which implies that $\rho_0 - \rho_0'$ must be even which further implies that $\rho_0 = \rho_0'$, since $\rho_0, \rho_0' \in \{0,1\}$. Using this on the left hand side of (5) and using the facts that $r(r')^{-1} \in \mathbb{QR}_N \Rightarrow r(r')^{-1} \in \mathbb{QR}_q \Rightarrow \left(\frac{r(r')^{-1}}{q}\right) = 1$ and $z_1 \in \mathbb{QNR}_p$ we get $(-1)^{\rho_1-\rho_1'} = 1$ which implies that $\rho_1 - \rho_1'$ must be even which further implies that $\rho_1 = \rho_1'$, since $\rho_1, \rho_1' \in \{0,1\}$. Combining these two results in equation $rz_0^{\rho_0}z_1^{\rho_1} \equiv r'z_0^{\rho_0'}z_1^{\rho_1'} \mod N$, we get $r \equiv r' \bmod N$. Hence $T_{z_0,z_1,u}$ is injective. □

C.9 Theorem 9

Let $\ell = \mathsf{poly}(\lambda)$ be any polynomial and let $\mathcal{A}$ be any efficient adversary for the ℓ-time full-domain Quartic problem. For each $i \in \{0, \ldots, \ell\}$, define a sequence of ℓ hybrid experiment Hyb_i parameterized by the security parameter λ as follows.

- **Experiment** Hyb_i**:** The challenger begins by sampling $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$, finds π such that $p = \pi\bar{\pi}$ and sets $N \leftarrow pq$. Then it samples $x_1, \cdots, x_\ell \xleftarrow{\$} \mathbb{Z}_N^\star$. For each $j \in [i]$, it samples $\beta_j \xleftarrow{\$} \{0,1\}^2$ uniformly at random. For each $j \in \{i+1, \ldots, \ell\}$, it sets $\beta_j \leftarrow F(\chi_\pi(x))$. Finally, the challenger sends the tuple $(N; ((x_1, \beta_1), \ldots, (x_\ell, \beta_\ell)))$ to $\mathcal{A}$, which then outputs a bit. The output of the experiment is the same as the output of $\mathcal{A}$.

For all $i \in \{0, \ldots, \ell\}$, we write $\mathsf{Hyb}_i(\mathcal{A})$ to denote the output distribution of an execution of Hyb_i with adversary $\mathcal{A}$. Note that by the definition of the hybrid games it is clear that

$$\mathsf{QuarticAdv}_\ell[\mathcal{A}](\lambda) \;=\; |\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_\ell(\mathcal{A}) = 1]|$$

For convenience, we define the following $\mathsf{Resampling}$ function.

Algorithm 8 $\mathsf{Resampling}_{N,z_0,z_1}(u \in \mathbb{Z}_N^\star, y \in \{0,1\}^2)$

1: Return $(x, \beta) \leftarrow \mathsf{ResamplingQuartic}(u, y)$ // where we use $\chi_\pi(z_0) = \chi_\pi(z_1) = i$

Algorithm $\mathcal{B}$ then proceeds as follows.

- Upon receiving a decisional Quartic Residuosity challenge (N, x, y), algorithm samples an index $i^\star \xleftarrow{\$} [\ell]$
- It samples fixed elements $z_0 \in \mathbb{Z}_N^\star \setminus \mathbb{QR}_N$ and $z_1 \in \mathbb{Z}_N^\star \setminus \mathbb{J}_N$ using the sampling techniques described in Section 2.5.
- For each $j \in [\ell]$, the algorithm constructs (x_j, β_j) as follows.
  - If $j < i^\star$, sample $x_j \xleftarrow{\$} \mathbb{Z}_N^\star$ and $\beta_j \xleftarrow{\$} \{0,1\}^2$
  - If $j = i^\star$, set $(x_j, \beta_j) \xleftarrow{\$} \mathsf{Resampling}_{N,z_0,z_1}(x, y)$
  - If $j > i^\star$, set $(x_j, \beta_j) \xleftarrow{\$} \mathsf{Resampling}_{N,z_0,z_1}(1, F(1))$
- Finally, algorithm $\mathcal{B}$ invokes $\mathcal{A}$ on input $(N; ((x_1, \beta_1), \ldots, (x_\ell, \beta_\ell)))$ and outputs whatever $\mathcal{A}$ outputs.

Notice that since $z_1 \in \mathbb{Z}_N^\star \setminus \mathbb{J}_N$, so either $z_1 \in \mathbb{QNR}_p$ and $z_1 \in \mathbb{QR}_q$ or $z_1 \in \mathbb{QR}_p$ and $z_1 \in \mathbb{QNR}_q$. Without loss of generality, assume that $z_1 \in \mathbb{QNR}_p$ and $z_1 \in \mathbb{QR}_q$.

Next, we compute the advantage of $\mathcal{B}$ conditioned on a particular value of $i^\star = i$. First we analyze the distribution of values (x_j, β_j) for all $j \in [\ell] \setminus \{i\}$.

– If $j < i$, the values (x_j, β_j) are independent random values in $\mathbb{Z}_N^\star \times \{0,1\}^2$ just as in Hyb_{i-1} and Hyb_i
– If $j > i$, by Lemma 3 (x_j, β_j) are distributed exactly as $(x_j, F(\chi_\pi(x_j)))$ where $x_j \xleftarrow{\$} \mathbb{Z}_N^\star$ just as in Hyb_{i-1} and Hyb_i

It is therefore enough to analyze the distribution of the pair (x_i, β_i).

Recall that the decisional quartic residuosity challenger samples $p, q \xleftarrow{\$} \mathsf{QPrimes}_\lambda$, computes π such that $p = \pi\bar{\pi}$ and sets $N = pq$. It also samples $x \xleftarrow{\$} \mathbb{J}_N$ and either sets $y = F(\chi_\pi(x))$ or samples $y \xleftarrow{\$} \{0,1\}^2$. Accordingly, we consider the following two possibilities for (x, y) depending on whether $y = F(\chi_\pi(x))$ or $y \xleftarrow{\$} \{0,1\}^2$.

– If $y = F(\chi_\pi(x))$, then by Lemma 3 the pair (x_i, β_i) is distributed identically to $(x_i, F(\chi_\pi(x_i)))$, where $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$. Therefore, the overall distribution of the tuple $(N; ((x_1, \beta_1), \ldots, (x_\ell, \beta_\ell))$ is exactly the same as in experiment Hyb_{i-1}. Consequently, the output of algorithm $\mathcal{B}$ is distributed identically to the output of $\mathcal{A}$ in Hyb_{i-1}. For $i^\star = i$, algorithm $\mathcal{B}$ therefore outputs 1 with probability $\Pr[\mathsf{Hyb}_{i-1}(\mathcal{A}) = 1]$.
– If $y \xleftarrow{\$} \{0,1\}^2$, then again by Lemma 3, x_i is distributed exactly as before since $x_i \xleftarrow{\$} \mathbb{Z}_N^\star$. The value β_i is distributed as $\beta_i \xleftarrow{\$} \{-1, 1\}$. Therefore, the overall distribution of the tuple $(N; ((x_1, \beta_1), \ldots, (x_\ell, \beta_\ell))$ is exactly the same as in experiment Hyb_i. Consequently, the output of algorithm $\mathcal{B}$ is distributed identically to the output of $\mathcal{A}$ in Hyb_i. For $i^\star = i$, algorithm $\mathcal{B}$ therefore outputs 1 with probability $\Pr[\mathsf{Hyb}_i(\mathcal{A}) = 1]$.

Since $\mathcal{B}$ samples $i^\star \xleftarrow{\$} [\ell]$, our analysis above shows that,

$$\rho_{\mathcal{B},0}(\lambda) \;=\; \frac{1}{\ell}\sum_{i\in[\ell]} \Pr[\mathsf{Hyb}_{i-1}(\mathcal{A}) = 1], \quad \rho_{\mathcal{B},1}(\lambda) \;=\; \frac{1}{\ell}\sum_{i\in[\ell]} \Pr[\mathsf{Hyb}_i(\mathcal{A}) = 1]$$

The advantage of algorithm $\mathcal{B}$ is given by

$$\begin{aligned}
\mathsf{QuarticRDAdv}[\mathcal{B}](\lambda) &= |\rho_{\mathcal{B},0}(\lambda) - \rho_{\mathcal{B},1}(\lambda)| \\
&= \frac{1}{\ell}\left|\sum_{i\in[\ell]} \Pr[\mathsf{Hyb}_{i-1}(\mathcal{A}) = 1] - \sum_{i\in[\ell]} \Pr[\mathsf{Hyb}_i(\mathcal{A}) = 1]\right| \\
&= \frac{1}{\ell}\left|\Pr[\mathsf{Hyb}_0(\mathcal{A}) = 1] - \Pr[\mathsf{Hyb}_\ell(\mathcal{A}) = 1]\right| \\
&= \frac{1}{\ell}\mathsf{QuarticAdv}_\ell[\mathcal{A}](\lambda).
\end{aligned}$$

□