Zeroed Out: Cryptanalysis of Weak PRFs in Alternating Moduli

Irati Manterola Ayala[1] and Håvard Raddum[2]

[1] Simula UiB, Bergen, Norway, irati@simula.no
[2] Simula UiB, Bergen, Norway, haavardr@simula.no

Abstract. The growing adoption of secure multi-party computation (MPC) has driven the development of efficient symmetric key primitives tailored for MPC. Recent advances, such as the alternating moduli paradigm, have shown promise but leave room for cryptographic and practical improvements. In this paper, we analyze a family of weak pseudorandom functions (wPRF) proposed at Crypto 2024, focusing on their One-to-One parameter sets. We demonstrate that these configurations fail to achieve their intended one-to-one mappings and exploit this observation to develop an efficient key recovery attack.

Our analysis reveals critical vulnerabilities, reducing the complexity of key recovery to $\mathcal{O}(2^{\lambda/2}\log_2\lambda)$ for the Standard One-to-One wPRF and $\mathcal{O}(2^{0.84\lambda})$ for the Reversed Moduli variant – both substantially below their claimed λ-bit security. We validate our findings through experimental evaluation, confirming alignment between predicted and observed attack complexities.

Keywords: Multi-Party Computation · Weak pseudorandom functions · Alternating moduli paradigm · Symmetric cryptanalysis · Key recovery attack

1 Introduction

The rise of interest in secure multi-party computation (MPC) and the growing threat of quantum computers have created an urgent need for efficient and quantum-resistant symmetric key primitives specifically designed for use in MPC settings. While classic symmetric key primitives hold promise due to their simplicity and performance potential, existing constructions were developed for different (and usually incompatible) settings. This creates a pressing need for new designs that avoid such vulnerabilities while remaining suitable for MPC applications.

Important cryptographic tasks, such as ring signatures, oblivious pseudorandom functions (OPRFs), verifiable random functions (VRFs), and blind signatures, require efficient solutions tailored to these evolving challenges [RST01, FIPR05, NR97, MRV99, Cha82]. Ideally, these primitives should be evaluable in a single round of communication using linear secret-sharing techniques. While there has been progress in adapting existing symmetric key primitives for MPC [AGP+19, ARS+16, DEG+18, DGH+21, GØSW22, GRR+16], many constructions still require too many communication rounds or involve high overheads [BIP+18]. This inefficiency stems in part from the difficulty of balancing low-depth functions, which are essential for efficiency in MPC settings, with security requirements.

To address these issues, Boneh et al. [BIP+18] introduced the alternating moduli paradigm, separating the requirements for MPC efficiency from those for cryptographic security. By alternating linear operations over different moduli, they built a depth-2 weak pseudorandom function (wPRF) that can be securely evaluated in a single communication

round after preprocessing. Dinur et al. [DGH+21] extended this work by introducing new one-way functions (OWFs), pseudorandom generators (PRGs), and wPRFs within the same framework. They showed that their OWF could be used to build a post-quantum signature scheme with good efficiency. Despite these advances, the protocols built around these constructions often fell short of state-of-the-art performance. Moreover, the two-party computation (2PC) protocols for these constructions require significant preprocessing time to generate correlated randomness, with communication overheads remaining higher than optimal.

Building on this line of work, Alamati et al. [APRR24] revisited the alternating moduli paradigm to propose a new wPRF that improves on previous constructions in terms of efficiency and practicality. According to the authors, their design significantly reduces communication and computational costs, particularly in the main evaluation phase, and minimizes the need for oblivious transfers. In terms of cryptanalysis, they argue that the security of their wPRF depends on the hardness of solving sparse multivariate polynomial systems over $\mathbb{F}_3$ or, in the dual form, on sparse multilinear interpolation. This argument is used by the authors to justify their focus on subset-sum attacks as the primary cryptanalytic threat. However, our analysis shows that this focus may be too narrow, as other potential attack vectors remain relevant and deserve further attention.

Our Contributions. In this paper, we present cryptanalysis of the One-to-One parameter sets proposed by Alamati et al. for their alternating moduli wPRF. We show that these do not provide the approximately one-to-one mappings they were designed to achieve. Leveraging this observation, we present a novel key recovery attack against the Standard One-to-One parameter set of the wPRF. Our attack achieves a complexity of $\mathcal{O}(2^{\lambda/2}\log_2\lambda)$, significantly lower than the claimed λ-bit security level.

Next, we adapt the key recovery attack to the Reversed Moduli One-to-One parameter set. While this variant introduces additional challenges, our modified attack successfully recovers the key with a complexity of $\mathcal{O}(2^{0.84\lambda})$, once again breaking the claimed 2^λ security level. We have also considered the Many-to-One parameter sets but did not find any successful attacks against these variants.

We provide both theoretical complexity analyses and experimental verification of our attacks. Our experiments confirm that the observed attack complexities closely align with the theoretical predictions. Beyond identifying these vulnerabilities, we propose potential countermeasures to mitigate these attacks, aiming to enhance the security of future designs.

Outline of this Paper. This paper is structured as follows. In Section 2, we provide the necessary preliminaries, namely the definition and security notions of weak pseudorandom functions, as well as the classical and generalized birthday paradox. Section 3 presents the wPRF construction by Alamati et al., detailing its specification, variants, and recommended parameter sets. In Section 4, we describe our primary contributions, offering a comprehensive cryptanalysis of two of the proposed wPRF parameter sets and analyzing the theoretical complexity of our key recovery attack. Section 5 validates our theoretical analysis with experimental results, showcasing the feasibility and accuracy of our approach. Finally, in Section 6, we conclude by summarizing our findings, discussing potential countermeasures, and outlining open problems for future research.

2 Preliminaries

In this section, we present the foundational concepts necessary for understanding the results and analysis in this paper. These include the definition and security notions of weak pseudorandom functions, along with the classical and generalized forms of the birthday paradox.

2.1 Weak Pseudorandom Functions

The definition of a weak pseudorandom function below follows Definition 2.1 from [BIP+18].

Definition 1. A **weak pseudorandom function (wPRF)** is a keyed function $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ that, when queried on random inputs $x \in \mathcal{X}$, is computationally indistinguishable from a truly random function. More formally, for a randomly selected key $k \in \mathcal{K}$, the output $f(k, x)$ for x sampled uniformly at random from $\mathcal{X}$ is indistinguishable from the output $g(x)$ of a random function $g : \mathcal{X} \to \mathcal{Y}$ to any adversary running in time $t(\lambda)$ with access to an oracle for f.

The distinction between a *weak PRF* and a *strong PRF* lies in the adversarial query model: wPRFs restrict adversaries to query only random inputs, whereas strong PRFs permit the adversary to query adaptive, chosen inputs.

Security Notion of a wPRF. The security of a wPRF f is quantified by the advantage an adversary $\mathcal{A}$ running in time $t(\lambda)$ has in distinguishing $f(k, x)$ from a random function. We say that f is *secure* if

$$\mathrm{Adv}^{\mathrm{wPRF}}_{f,\mathcal{A}} = \left|\Pr[\mathcal{A}^{f(k,\cdot)} = 1] - \Pr[\mathcal{A}^{g(\cdot)} = 1]\right| \leq \epsilon(\lambda),$$

where $\epsilon(\lambda)$ is negligible in λ. If a wPRF claims to provide λ-*bit security*, it means that the above security notion holds when $t(\lambda) = 2^\lambda$. That is, the above advantage remains negligible even when $\mathcal{A}$ is allowed up to 2^λ queries and runs in time 2^λ.

2.2 The Birthday Paradox

The *birthday paradox* is a probabilistic phenomenon that explains the counterintuitive likelihood of repeated outcomes when drawing samples from a finite set. It is particularly relevant in cryptographic contexts, where it is used to estimate the probability of repeated outputs in hash functions and similar structures.

Given a function that maps inputs to $|\mathcal{Y}|$ equally likely outputs, the birthday paradox quantifies the number of samples required to observe the same output at least twice.

Lemma 1. *[CLR90, Sec. 5.4.1]* ***Classical Birthday Paradox****. For a uniform random distribution over $|\mathcal{Y}|$ possible outputs, the expected number of samples S required to observe the first repeated outcome is:*

$$S \approx \sqrt{2|\mathcal{Y}|}.$$

The analysis given in [CLR90] naturally extends to estimating the number of random samples needed to find multiple pairs of repeated outcomes. In this paper, we refer to this as the *generalized birthday paradox*, noting that it differs from other common generalizations [Wag02, Das05].

Lemma 2. ***Generalized Birthday Paradox****. For a uniform random distribution over $|\mathcal{Y}|$ possible outputs, the expected number of samples S required to observe c pairs of repeated outcomes is:*

$$S \approx \sqrt{2|\mathcal{Y}|c}.$$

The generalized form reveals that the sample complexity scales proportionally to $\sqrt{c}$, meaning that detecting more collisions requires only a sublinear increase in samples.

3 A New Weak PRF

At Crypto 2024, Alamati et al. [APRR24] introduced a novel wPRF tailored for efficient multiparty computation (MPC) applications. This construction builds upon and generalizes the alternating-moduli paradigm initially proposed by Boneh et al. [BIP+18]. By alternating computations over two distinct moduli, typically $\mathbb{F}_2$ followed by $\mathbb{F}_3$, this approach has demonstrated significant potential in achieving both simplicity and efficiency in advanced cryptographic protocols.

We explore the details of Alamati et al.'s new wPRF constructions, and discuss their recommended parameter sets for achieving λ-bit security under various constraints.

3.1 Specification.

In their work [BIP+18], Boneh et al. considered the function

$$f(\mathbf{K}, x) := g(\mathbf{K} \cdot_2 x), \quad \text{where } g(w) = \sum_i w_i \mod 3.$$

Here, the operation $\cdot_p$ denotes multiplication modulo p, the matrix $\mathbf{K} \in \mathbb{F}_2^{m \times n}$ is the secret key and the term $\mathbf{K} \cdot_2 x \in \mathbb{F}_2^m$ is embedded into $\mathbb{F}_3^m$ component-wise in the natural way. Extensions to this idea defined the wPRF

$$f(\mathbf{K}, x) := \mathbf{B} \cdot_3 (\mathbf{K} \cdot_2 x),$$

where $\mathbf{K}$ is a square matrix and $\mathbf{B}$ is a compressing matrix.

To improve upon Boneh et al.'s construction, Alamati et al. propose a new wPRF that optimizes the end-to-end cost of MPC protocols while enhancing performance during the main computation phase, leading to significant gains in both communication complexity and computational efficiency. Their construction relies on three core components:

1. Non-linear combination of the input and key modulo two.
2. Matrix multiplication modulo two.
3. Natural modulus conversion followed by a public compressing linear map $\mathbf{B}$.

3.2 Definition of the Standard wPRF.

The proposed standard $(\mathbb{F}_2, \mathbb{F}_3)$-wPRF is formulated as follows:

$$F(k, x) := \mathbf{B} \cdot_3 (\mathbf{A} \cdot_2 [k \odot_2 x]),$$

where:

- $x, k \in \mathbb{F}_2^n$ are random vectors representing the input and key,
- $\mathbf{A} \in \mathbb{F}_2^{m \times n}$ is a random matrix,
- $\mathbf{B} \in \mathbb{F}_3^{t \times m}$ is a random compressing matrix (i.e., $t < m$).

Here, the operation $\odot_p$ denotes component-wise multiplication modulo p.
A visual representation of the standard wPRF construction is given in Fig. 1.

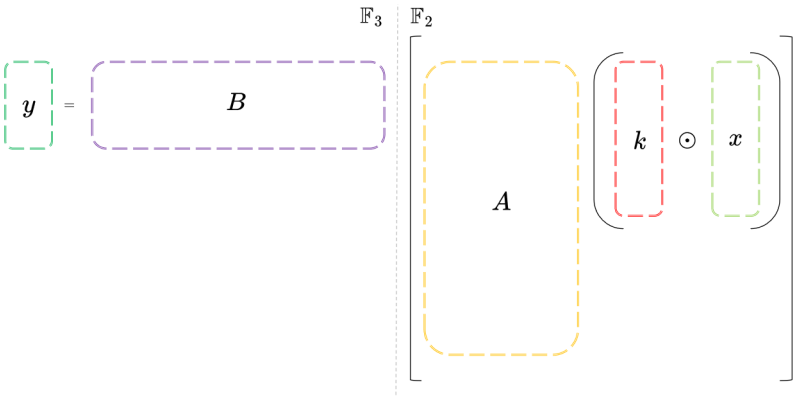

Figure 1: Construction of the standard $(\mathbb{F}_2, \mathbb{F}_3)$-wPRF.

Variants of the Standard wPRF. The generalized $(\mathbb{F}_p, \mathbb{F}_q)$-wPRF extends this concept to arbitrary primes p and q in a straightforward manner as

$$F(k, x) := \mathbf{B} \cdot_q (\mathbf{A} \cdot_p [k \odot_p x]),$$

where $x, k \in \mathbb{F}_p^n$, $\mathbf{A} \in \mathbb{F}_p^{m \times n}$, and $\mathbf{B} \in \mathbb{F}_q^{t \times m}$.
For scenarios requiring binary secret-sharing outputs, Alamati et al. propose the Reversed Moduli $(\mathbb{F}_3, \mathbb{F}_2)$-wPRF:

$$F(k, x) := \mathbf{B} \cdot_2 (\mathbf{A} \cdot_3 [k \odot_3 x]),$$

where the roles of the moduli are reversed.

3.3 Parameters.

Table 2 summarizes the recommended parameter sets from [APRR24] across the different wPRF constructions. Alamati et al. divide the parameter sets into two groups, namely One-to-One and Many-to-One parameters. The authors assert that each parameter set achieves λ-bit security.

One-to-One Parameters. The One-to-One parameter set is designed to provide a (roughly) one-to-one mapping between inputs and outputs. Specifically, the input and output spaces are of the same size, and for any given input x, the authors claim we can expect a unique corresponding output y. This setup represents their most conservative alternative.

Many-to-One Parameters. As the name suggests, the Many-to-One parameter set has a larger input space than output space. This means that for any given output y, there should be multiple input values x mapping to y, leading to a many-to-one mapping between input and output values.

4 Our Attack

In this section, we present a key recovery attack against the two One-to-One parameter sets proposed by the authors. Our method exploits weaknesses in these parameter sets, and efficiently identifies key bits using collisions. The attack is able to recover the key in $\mathcal{O}(2^{\lambda/2} \log_2 \lambda)$ calls to the wPRF in the standard version, and in $\mathcal{O}(2^{0.84\lambda})$ calls in the reversed moduli variant, demonstrating a significant reduction in complexity compared to

Table 1: Recommended parameter sets for the wPRF for λ-bit security.

Variant	*One-to-One*			*Many-to-One*		
	n	m	t	n	m	t
$(\mathbb{F}_2, \mathbb{F}_3)$-wPRF	2λ	7.06λ	$\frac{2\lambda}{\log_2(3)}$	4λ	2λ	$\frac{\lambda}{\log_2(3)}$
$(\mathbb{F}_3, \mathbb{F}_2)$-wPRF	$\frac{2\lambda}{\log_2(3)}$	$\frac{7.06\lambda}{\log_2(3)}$	2λ	$\frac{4\lambda}{\log_2(3)}$	2λ	λ

the claimed 2^λ calls. We begin with an analysis of the Standard One-to-One wPRF to establish the basic methodology. Following this, we demonstrate how the attack can be modified for the Reversed Moduli variant, overcoming its additional complexities.

In the following, let X denote the input space of the wPRF, let M denote the output space of the multiplication with the matrix $\mathbf{A}$, and let Y denote the output space of the wPRF.

One-to-One? We target the proposed parameter sets where the input space size is $|X| = 2^{2\lambda}$ $(= 3^{(2\lambda/\log_2 3)})$, the intermediate space has size $|M| = 2^{7.06\lambda}$ $(= 3^{(7.06\lambda/\log_2 3)})$, and the output space size is $|Y| = 3^{2\lambda/\log_2 3} = 2^{2\lambda}$. The authors argue that these configurations result in a (roughly) one-to-one mapping between inputs and outputs of the wPRF. However, this assumption does not hold once the wPRF is instantiated with a fixed key k. We exploit this observation to construct a key recovery attack.

- *Standard One-to-One.* Define h_1 as the Hamming weight of k, and let $h_0 = 2\lambda - h_1$ denote the number of zeros in k. For a key k chosen uniformly at random, we expect $h_1 \approx h_0 \approx \lambda$, following a binomial distribution. In positions where $k_i = 0$, the value of x_i is irrelevant, as $k_i \odot x_i$ will always equal zero. This leads to 2^{h_0} distinct values of x producing the same input to the multiplication with $\mathbf{A}$, creating a 2^{h_0}-to-1 sub-mapping in the wPRF. Consequently, once the key is fixed, the wPRF becomes a 2^{h_0}-to-1 mapping. The image of the wPRF F, denoted as $im(F)$, thus has size $2^{h_1} \approx 2^\lambda$ instead of $2^{2\lambda}$.

- *Reversed Moduli One-to-One.* Extending notation, let h_i^* be the number of elements in k that take the value i (for $i = 0, 1, 2$). For a uniformly random key k, we expect $h_0^* \approx h_1^* \approx h_2^* \approx \frac{2\lambda}{3\log_2(3)}$. Similarly to the standard case, the operation $k \odot x$ induces a $3^{h_0^*}$-to-1 sub-mapping, which can be expressed as $2^{2\lambda - \log_2(3)(h_1^* + h_2^*)}$-to-1. As a result, $im(F)$ has size $2^{\log_2(3)(h_1^* + h_2^*)}$, with an expected value of $2^{4\lambda/3}$, instead of the intended $2^{2\lambda}$.

4.1 Key Recovery Attack on Standard One-to-One wPRF

Our attack aims to recover the key k by finding pairs x, x' such that $F(k, x) = F(k, x')$. Whenever this occurs we say we have a *collision*. The attack is described in Algorithm 1 and explained in the following.

We initialize a key K as $K = [1, 1, 1, \ldots, 1]$ and iteratively refine it towards the correct key k by identifying positions in k that must be 0. The idea is to query the wPRF on random inputs, building up a table of input and output values (x, y). By the birthday paradox (see Lemma 1), collisions are expected to appear after approximately $\sqrt{2|im(F)|} = 2^{(h_1+1)/2}$ samples.

Let x and x' be two inputs producing the same output y. If $x_i \neq x'_i$, then it must hold that $k_i = 0$. To understand this, note that $|M| = 2^{7.06\lambda}$ is much larger than $|X| = 2^{2\lambda}$, and only 2^{h_1} different values go into multiplication with $\mathbf{B}$, which is much smaller than $|Y| = 2^{2\lambda}$. Thus, the probability of creating collisions *after* multiplying $k \odot x$ with $\mathbf{A}$ becomes negligible. Therefore, with overwhelming probability, the only source of collisions is the 2^{h_0}-to-1 mapping of $k \odot x$. So, if $x_i \neq x'_i$, then k_i must be zero, as the differing input bits would otherwise result in different values in M (and therefore a different output y).

To further support this and address potential false positives, we observe that both the input and output spaces have cardinalities $|X| = |Y| = 2^{2\lambda}$, so the probability that a difference in input results in the same output is analogous to finding a collision in a random mapping from a space of size $2^{2\lambda}$ to itself. By the birthday bound, the probability of any collision after I random inputs is approximately $I^2/2^{2\lambda+1}$. In our setting, we will later show that the attack requires approximately $I \approx 2^{\lambda/2}$ such queries, yielding a collision probability of $1/2^{\lambda+1}$, which is exponentially small in λ. This makes false positives—cases where $x_i \neq x'_i$ but $k_i = 1$—highly improbable, so each collision reveals with high confidence that the differing positions in x and x' correspond to zero entries in k.

To further analyze the key recovery, let

$$J_0 = \{i | k_i = 0\} \text{ and } J_1 = \{i | k_i = 1\}.$$

For two colliding inputs x, x', let $X_= = X_=(x, x') = \{i | x_i = x'_i\}$ and $X_\neq = X_\neq(x, x') = \{i | x_i \neq x'_i\}$.

As collisions accumulate, we progressively update K by changing 1-bits in K to 0 for all indices in $X_\neq$. For each collision, we know that $J_1 \subseteq X_=$ and $X_\neq \subseteq J_0$. For positions $i \in J_0$, we have either $x_i = x'_i$ or $x_i \neq x'_i$ with equal probability since both x and x' are drawn uniformly at random. Consequently, we expect that only half of the set J_0 will be revealed from any one collision. Thus, each new independent collision is expected to reveal half of the previously unrevealed positions where $k_i = 0$. This suggests that the Hamming distance between K and k is halved with each new collision. Specifically, the expected Hamming distance after c collisions can be expressed as

$$d_c = h_0/2^c. \tag{1}$$

4.1.1 Collision Saturation Point.

As the attack progresses and additional collisions are found, the rate of discovering new J_0 positions decreases, as many zeros in k have already been determined. At a certain point, it becomes more efficient to perform an exhaustive search among keys within a small Hamming distance of the current guess K. The transition occurs when the expected cost of generating the $(c+1)$-th collision surpasses the cost of exhaustive search among vectors with hamming distance at most d_c from K.

Cost of New Collision. By the generalized birthday paradox (see Lemma 2), the expected number of samples required to find c collisions is $\sqrt{2^{h_1+1}c}$. To find the $(c+1)$-th collision after already obtaining c collisions, the number of new queries needed is

$$\sqrt{2^{h_1+1}(c+1)} - \sqrt{2^{h_1+1}c} = 2^{(h_1+1)/2}(\sqrt{c+1} - \sqrt{c}). \tag{2}$$

The total cost of generating the $(c+1)$-th collision is dominated by this term, since verifying collisions can be done in constant time by storing (x, y)-pairs in a hash table.

Cost of Exhaustive Search. For exhaustive search, we consider all keys within a Hamming distance of at most d_c from K. Notably, we only flip 1-bits in K to 0, never changing

0-bits to 1. This restricted search space is referred to as the *one-sided Hamming distance* from K. Thus, the number of candidate keys to search is $\sum_{j=1}^{\lceil d_c \rceil} \binom{H_1}{j}$, where H_1 is the current Hamming weight of K. To verify each key candidate, we compute an output using the current key guess and a previously queried input. Since the number of candidate keys remains below $2^{(\lambda+1)/2} \cdot (\sqrt{c+1} - \sqrt{c}) < 2^{(\lambda+1)/2}$ (see Inequality 3 below), the probability of an incorrect key producing the expected output is negligibly small. This should guarantee that only the correct key passes with very high probability. In the worst case, this results in a total query cost of

$$\sum_{j=1}^{\lceil d_c \rceil} \binom{H_1}{j}.$$

To determine the optimal transition point to exhaustive search, we substitute the expected values $h_0 = h_1 = \lambda$ into Equations (2) and (1). Minimizing the total attack cost requires switching strategies once c collisions have been found and the following inequality holds:

$$\sum_{j=1}^{\lceil \lambda/2^c \rceil} \binom{H_1}{j} < 2^{(\lambda+1)/2} \cdot (\sqrt{c+1} - \sqrt{c}). \tag{3}$$

This condition does not guarantee that the correct key lies within d_c Hamming distance of K. If exhaustive search fails to find the correct key at this stage, we simply identify one more collision and retry.

4.1.2 Complexity Analysis.

We measure the complexity of the attack in terms of the needed number of queries to the wPRF. The attack follows a two-phase approach: first, collisions are accumulated until reaching the transition point; then, exhaustive search on the key is applied. Let C denote the number of collisions at the transition point. We know that $C \leq \log_2 \lambda$, since for $C = \log_2 \lambda$ and $H_1 \leq n = 2\lambda$ Inequality (3) always holds for $\lambda \geq 17$.

As discussed above, the total number of samples required to recover the key is approximately

$$\sqrt{2^{\lambda+1} C} + \sum_{j=1}^{\lceil \lambda/2^C \rceil} \binom{H_1}{j},$$

where H_1 represents the Hamming weight of the guessed key after C collisions. By construction of K, we estimate H_1 as $h_1 + \frac{h_0}{2^C} \approx \lambda + \frac{\lambda}{2^C}$.

For $C = \log_2 \lambda$, the sum in the expression above stops at $j = 1$, leading to an attack complexity of the order

$$\mathcal{O}\left(2^{\lambda/2} \log_2 \lambda\right).$$

The total cost of the attack is thus significantly lower than 2^λ, demonstrating a clear compromise of the claimed security level. We explicitly note that this complexity analysis assumes $h_1 = \lambda$, which corresponds to the expected Hamming weight of a uniformly random key. While this assumption provides a realistic estimate of the computational cost for a typical key, we acknowledge that the actual complexity may vary slightly for specific instances where the key deviates significantly from the expected Hamming weight.

Algorithm 1 Key Recovery Attack

Require: Input-output oracle $\mathcal{O}$, security parameter λ
Ensure: Recovered key k

$K \leftarrow [1, 1, \ldots, 1]$
$\mathcal{P} \leftarrow \emptyset$
$c \leftarrow 0$
$H_1 \leftarrow n$
while Correct key not found **do**
  repeat
    Collect a new input-output pair (x, y) using $\mathcal{O}$
    if $(x', y) \in \mathcal{P}$ for some $x' \neq x$ **then**
      for $i \in X_{\neq}$ **do**
        if $K_i = 1$ **then**
          $K_i \leftarrow 0$
          $H_1 \leftarrow H_1 - 1$
        end if
      end for
      $c \leftarrow c + 1$
    end if
    Add (x, y) to $\mathcal{P}$
  until Collision is found
  if $\sum_{j=1}^{\lceil \lambda / 2^c \rceil} \binom{H_1}{j} \leq 2^{(\lambda+1)/2} \cdot (\sqrt{c+1} - \sqrt{c})$ **then**
    for each k' with one-sided Hamming distance at most d_c from K **do**
      if k' matches an input-output pair from $\mathcal{P}$ **then**
        return k'
      end if
    end for
  end if
end while

4.2 Attack on Reversed Moduli One-to-One wPRF

We adapt the collision-based key recovery attack methodology used in the standard parameter set to the reversed moduli One-to-One wPRF. The key difference in this variant is that non-zero key positions can take two distinct values, requiring modifications to our approach. The modified attack still remains feasible and reveals vulnerabilities in the construction. Below, we detail the process and analyze its computational complexity.

4.2.1 Collisions: Identifying Zero Key Positions.

The first step of the attack is to identify the positions in the key k where $k_i = 0$. To achieve this, we employ the same collision-finding method used previously in the standard case. By the birthday paradox (see Lemma 1), we expect collisions to appear after collecting approximately

$$\sqrt{2|im(F)|} = 2^{(\log_2(3)(h_1^*+h_2^*)+1)/2} \approx 2^{(4\lambda+3)/6}$$

samples.

In this setting, the size of the domain M is again significantly larger than the size of the input space X, ensuring that collisions arise solely from the $3^{h_0^*}$-to-1 mapping induced by $k \odot x$ with overwhelming probability. Therefore, each collision reveals information about positions in k where $k_i = 0$.

Let $J_0 = \{i \mid k_i = 0\}$ and x and x' two colliding inputs as before. We again have

$$X_{\neq} = X_{\neq}(x, x') = \{i | x_i \neq x'_i\} \subseteq J_0.$$

For positions $i \in J_0$, we have either $x_i = x'_i$ or $x_i \neq x'_i$, but these events do not occur with equal probability in the reversed moduli case. Since x takes values in $\mathbb{F}_3$, we have $x_i \neq x'_i$ with probability $2/3$. Thus, we expect to recover approximately $2/3$ of J_0 from any given collision. This higher recovery rate, compared to the standard wPRF, reduces the number of collisions required to fully determine J_0.

We continue generating collisions until all zero positions in the key are likely identified. To estimate the number of collisions required, we analyze the probability of revealing additional zeroes as we accumulate collisions. As discussed, the first collision is expected to reveal approximately $2/3 \cdot h_0^*$ zeroes. The second collision builds upon this, revealing another $2/3^2 \cdot h_0^*$ zeroes. More generally, after c collisions, the total number of recovered zeroes is

$$\sum_{i=1}^{c} \frac{2}{3^i} \cdot h_0^*.$$

Thus, the number of remaining zero positions in k yet to be identified after c collisions is given by

$$h_0^* - \sum_{i=1}^{c} \frac{2}{3^i} \cdot h_0^* = \left(1 - \sum_{i=1}^{c} \frac{2}{3^i}\right) h_0^*.$$

To ensure all zero positions are likely identified, the number of remaining positions must be less than 1, i.e.,

$$\left(1 - \sum_{i=1}^{c} \frac{2}{3^i}\right) h_0^* \approx \left(1 - \sum_{i=1}^{c} \frac{2}{3^i}\right) \frac{2\lambda}{3\log_2(3)} < 1.$$

Solving this expression for c gives the minimum number of expected collisions required to recover all zero positions in the key. To ensure high-probability recovery, we introduce a small safety margin by multiplying the derived value for c by three. In any case, the complexity of determining all zero positions remains of the order $\mathcal{O}(\log_3(\lambda))$ collisions.

4.2.2 Exhaustive Search over Non-Zero Key Positions.

Once the positions in J_0 are determined, the values of the remaining positions $J_1 \cup J_2 = \{i \mid k_i \in \{1, 2\}\}$ remain unknown. These positions are expected to constitute $2/3$ of the key. However, for these positions, each k_i can only take one of two possible values, 1 or 2, since all zeroes have already been detected. For a key of length $n = \dfrac{2\lambda}{\log_2 3}$, the total number of candidates for the remaining key components is therefore

$$2^{(2/3)\cdot(2\lambda/\log_2 3)} \approx 2^{0.84\lambda}.$$

The correctness of any candidate key can be verified by querying the wPRF on an input-output pair as before. Thus, the exhaustive search over all possible keys in $J_1 \cup J_2$ requires at most $2^{0.84\lambda}$ queries.

4.2.3 Complexity Analysis.

The overall complexity of the attack consists of two main components: identifying zero positions via collisions and performing an exhaustive search over non-zero key positions.

Collision Complexity. By the generalized birthday paradox (see Lemma 2), the expected cost of finding enough collisions to identify all zero positions of the key is

$$\sqrt{2^{(4\lambda+3)/3} C},$$

where $C = \mathcal{O}(\log_3(\lambda))$ denotes the number of collisions required to fully determine J_0. The total complexity of this step thus becomes $\mathcal{O}(2^{2\lambda/3} \log_3(\lambda))$.

Exhaustive Search Complexity. Once the zero positions are known, the exhaustive search requires testing $2^{0.84\lambda}$ key candidates, each verified with a query. This results in a total cost of $2^{0.84\lambda}$.

Total Complexity. The overall complexity of the attack is the sum of the costs of the collision and exhaustive search steps. Notably, the complexity is dominated by the exhaustive search step, and so the attack has a total cost of $\mathcal{O}(2^{0.84\lambda})$. This is well below the claimed security level of 2^λ, demonstrating that the Reversed Moduli One-to-One parameter set also fails to provide the intended security guarantees.

4.3 Applicability beyond the One-to-One Parameter Sets

Many-to-One Parameter Sets. The attack described above does not apply to the Many-to-One variants of the wPRF. In these cases, the input space size $|X| = 2^{4\lambda}$ significantly exceeds the output space size $|Y| = 2^\lambda$, making collisions unavoidable. Since the intermediate output space of the pointwise multiplication followed by multiplication with $\mathbf{A}$ has size $|M| = 2^{2\lambda}$, most collisions occur independently of the term $k \odot x$. More specifically, distinct points in M produce a collision in Y at a rate of once every $2^{\lambda/2}$ queries, while collisions due to $k \odot x$ being a $2^{2\lambda}$-to-1 mapping only appear at a rate of once every 2^λ queries. As a result, generating even a single collision where $k \odot x = k \odot x'$ will take $\mathcal{O}(2^\lambda)$ time, making our approach ineffective for these parameter sets.

Boneh et al.'s wPRF. Recall that Alamati et al.'s construction builds upon the wPRF derived by Boneh et al.'s alternating-moduli function [BIP+18]. This wPRF is defined as $f(\mathbf{K}, x) := \mathbf{B} \cdot_3 (\mathbf{K} \cdot_2 x)$, where $\mathbf{K}$ is a square matrix acting as the secret key, and x is an input vector. Unlike Alamati et al.'s wPRF, which employs component-wise

multiplication $k \odot x$ before a linear transformation, Boneh et al.'s construction instead performs a full matrix-vector multiplication $\mathbf{K} \cdot_2 x$ directly. This difference has a crucial impact on the applicability of our attack. In Alamati et al.'s wPRF, zero entries in k eliminate contributions from corresponding positions in x, effectively reducing the entropy of the input to subsequent computations and enabling the attack. However, in Boneh et al.'s construction, each x_i in the input x is multiplied with n different bits of the key $\mathbf{K}$, ensuring that no input component is entirely zeroed out due to a zero entry in $\mathbf{K}$. Consequently, the structure that enables our attack in Alamati et al.'s scheme is absent here, making our approach ineffective against Boneh et al.'s wPRF.

5 Experimental Verification

To validate our proposed approach, we have conducted a series of low-scale experiments[1] in the Standard One-to-One parameter set, using $\lambda = 28$ and $\lambda = 34$ as test cases. For each scenario, we have performed 1000 independent experiments to ensure statistical significance, recording the average results obtained. Table 2 summarizes our experimental findings, which corroborate the theoretical estimations presented in Section 4 and demonstrate the feasibility of a successful key recovery attack.

We analyze the average complexity of the two principal components outlined in Section 4: finding collisions and exhaustive search.

- **Collision Finding ($C_{\mathbf{col}}$):** We measure the average number of samples required to generate a sufficient number of collisions necessary for key recovery.
- **Exhaustive Search ($C_{\mathbf{exs}}$):** Once the sufficient number of collisions is identified, we perform an exhaustive search over the key candidates with a small Hamming distance from the current key guess. We record the average number of calls to F for this step.

By combining these components, we compute the total complexity $C_{\text{tot}} = C_{\text{col}} + C_{\text{exs}}$ of the attack. Our results demonstrate that we achieve key recovery with complexity closely aligned with the theoretical expectation of $\mathcal{O}(2^{\lambda/2} \log_2(\lambda))$:

- For $\lambda = 28$, the observed average total complexity is $C_{\text{tot}} = 2^{16.6}$, which is consistent with the estimated complexity of $2^{\lambda/2} \log_2(\lambda) = 2^{16.27}$.
- For $\lambda = 34$, the observed average total complexity is $C_{\text{tot}} = 2^{19.82}$, closely matching the estimated complexity of $2^{\lambda/2} \log_2(\lambda) = 2^{19.35}$.

Note that all 1000 experiments recovered the correct key, and that the numbers used to calculate C_{col} and C_{exs} represent the total number of calls to the wPRF oracle, including instances where the attack had to go back and find an additional collision before trying exhaustive search again.

Additionally, we evaluate the accuracy of the transition step discussed in Section 4, which estimates the optimal transition point between collision finding and exhaustive search. Specifically, we measure the success rate of the computed transition point C from Inequality 3 by verifying whether, after finding C collisions, the key is successfully recovered on the first attempt at exhaustive search. The measured success rate is 76.6% for $\lambda = 28$ and 88.8% for $\lambda = 34$, indicating that the theoretical model becomes increasingly accurate for larger values of λ.

Furthermore, we report the average number of collisions required to recover the full key. Our results show that the attack requires approximately 4.39 collisions for $\lambda = 28$

[1] The implementation details and source code are available at `https://github.com/Simula-UiB/wPRF-Collision-Attack`.

and 4.19 collisions for $\lambda = 34$ to achieve full key recovery. We would expect the number of necessary collisions to increase for higher values of λ, as a higher λ corresponds to a higher Hamming weight of the key on average, requiring more bits to be flipped to 0 to reach the final correct guess of the key. However, our experiments indicate that this is not necessarily the case. This discrepancy may be attributed to the significant differences in the accuracy of the computed transition point C. For $\lambda = 28$, fewer collisions should, in theory, have been required before successfully switching to exhaustive search. Nonetheless, due to inaccuracies in approximating the transition point correctly in nearly 25% of the cases, more collisions were needed than expected. As the accuracy of C improves with higher values of λ, we observe fewer such deviations. We therefore hypothesize that this theoretical trend persists for higher values of λ, where the discrepancy in the transition point accuracy is likely to diminish further, thereby reducing unexpected variations in the number of collisions needed to recover the key through exhaustive search.

Table 2: Summary of experimental results for $\lambda = 28$ and $\lambda = 34$. The columns represent the average complexity of collision finding (C_{col}), exhaustive search (C_{exs}), and total complexity (C_{tot}). Additionally, the table reports the average number of collisions required to achieve full key recovery and the success rate of the transition point C estimated using Inequality 3. All values are averaged over 1000 independent experiments.

λ	C_{col}	C_{exs}	C_{tot}	# Collisions	Accuracy of C (%)
28	$2^{16.6}$	$2^{7.64}$	$2^{16.6}$	4.39	76.6
34	$2^{19.82}$	$2^{10.88}$	$2^{19.82}$	4.19	88.8

5.1 Hamming Distance Analysis

In addition to the previously described experiments, we also verify the assumption that the Hamming distance between the actual key k and the guessed key K is approximately halved with each new collision.

We have performed 100 independent experiments for various values of λ and recorded the average results. While the findings are consistent across different values of λ, we present the case of $\lambda = 34$ as a representative example. Figure 2 illustrates the average decrease in Hamming distance between the current guessed key and the actual key after each identified collision. The graph shows that the Hamming distance roughly halves with each collision, as expected.

6 Conclusions

In this paper, we conducted a detailed cryptanalysis of the One-to-One parameter sets in the alternating moduli wPRFs proposed by Alamati et al. Our analysis reveals critical vulnerabilities in these constructions, allowing for efficient key recovery attacks that compromise the claimed λ-bit security levels. Specifically, we presented an attack with complexity $\mathcal{O}(2^{\lambda/2} \log_2 \lambda)$ against the Standard One-to-One wPRF and $\mathcal{O}(2^{0.84\lambda})$ against the Reversed Moduli variant. Both attacks exploit the reduction in output space caused by the 0-values in the random but fixed key, which induces sub-mappings that deviate significantly from the intended one-to-one mappings. The effectiveness of the attacks was further validated through experimental implementations.

To address these vulnerabilities, we propose potential countermeasures. One strategy is to restrict the selection of keys to elements in $\mathbb{F}_p^*$, thereby excluding zero values as

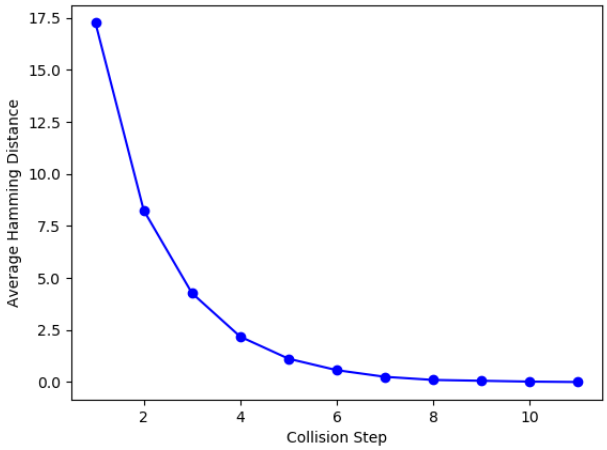

Figure 2: Number of found collisions vs. the average Hamming distance between the guessed key and the actual key for $\lambda = 34$.

coefficients in the key and ensuring that no part of the input is zeroed out in the first operation of the wPRF. The drawback of this mitigation is that p must be greater than 2 for this countermeasure to be applicable, and so one can not have $\mathbb{F}_2^n$ as the space for inputs and keys. Another approach is to replace the pointwise multiplication operation with addition or another operation that does not make any part of the input irrelevant.

We also identify open problems for future research. A deeper analysis of the Many-to-One parameter sets, which were not susceptible to our current attack, could shed light on the resilience of alternating moduli constructions in different configurations. Additionally, studying the trade-offs between mitigation techniques and their impact on performance in secure MPC environments requires further investigation. Finally, exploring alternative low-depth cryptographic designs that balance efficiency and security remains an important direction.

References

[AGP+19] Martin R. Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *ESORICS 2019, Part II*, volume 11736 of *LNCS*, pages 151–171. Springer, Cham, September 2019.

[APRR24] Navid Alamati, Guru-Vamsi Policharla, Srinivasan Raghuraman, and Peter Rindal. Improved alternating-moduli PRFs and post-quantum signatures. Cryptology ePrint Archive, Report 2024/582, 2024.

[ARS+16] Martin Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. Cryptology ePrint Archive, Report 2016/687, 2016.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 699–729. Springer, Cham, November 2018.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203. Plenum Press, New York, USA, 1982.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.

[Das05] Anirban DasGupta. The matching, birthday and the strong birthday problem: a contemporary review. *Journal of Statistical Planning and Inference*, 130(1):377–389, 2005. Herman Chernoff: Eightieth Birthday Felicitation Volume.

[DEG+18] Christoph Dobraunig, Maria Eichlseder, Lorenzo Grassi, Virginie Lallemand, Gregor Leander, Eik List, Florian Mendel, and Christian Rechberger. Rasta: A cipher with low ANDdepth and few ANDs per bit. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 662–692. Springer, Cham, August 2018.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 517–547, Virtual Event, August 2021. Springer, Cham.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 303–324. Springer, Berlin, Heidelberg, February 2005.

[GØSW22] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From farfalle to megafono via ciminion: The PRF hydra for MPC applications. Cryptology ePrint Archive, Report 2022/342, 2022.

[GRR+16] Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart. MPC-friendly symmetric key primitives. Cryptology ePrint Archive, Report 2016/542, 2016.

[MRV99] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th FOCS*, pages 120–130. IEEE Computer Society Press, October 1999.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, Berlin, Heidelberg, December 2001.

[Wag02] David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Berlin, Heidelberg, August 2002.