Zeroed Out: Cryptanalysis of Weak PRFs in Alternating Moduli

Irati Manterola Ayala and Håvard Raddum

Simula UiB, Bergen, Norway
{irati,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 advancements, 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 the 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.
The attacks reveal significant 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 evaluations, 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 and often have algebraic structures that quantum computers could exploit. This necessitates new designs that avoid such vulnerabilities while being suitable for MPC.

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 [16,11,15,14,5]. 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 [2,3,10,8,12,13], many constructions still require too many communication rounds or involve high overheads [4]. 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. [4] 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. [9] 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 2PC protocols for these constructions require significant preprocessing time to generate correlated randomness, with communication overheads higher than desired.

Building on this line of work, Alamati et al. [1] 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 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 the suggested parameter sets named One-to-One do not give the approxiamtely one-to-one mappings they are supposed to do. We use this observation to 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 and apply it to the Reversed Moduli One-to-One parameter set. This variant presents some extra challenges, but the 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 could not find any successful attacks on 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. In addition to highlighting the vulnerabilities in the current wPRF constructions, we propose potential countermeasures to mitigate these attacks, aiming to make future constructions secure.

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 and the classical and generalized birthday paradox. Section 3 introduces the wPRF construction by Alamati et al., explaining its specification, variants, and recommended parameter sets. In Section 4, we detail our primary contributions, including a comprehensive cryptanalysis of two proposed wPRF parameter sets and an analysis of the theoretical complexity of our key recovery attack. Section 5 validates our theoretical analysis through experimental results, showcasing the feasibility and accuracy of our approach. Finally, in Section 6, we conclude by summarizing our findings, discussing potential countermeasures, and highlighting 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, as well as 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 [4].

Definition 1. *A **weak pseudorandom function (wPRF)** is a keyed function $f : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ 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 $g : \mathcal{X} \rightarrow \mathcal{Y}$ 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 give λ-*bit security*, it means that the above security notion holds when $t(\lambda) = 2^{\lambda}$. That is, $\mathcal{A}$ is allowed to make up to 2^{λ} queries to the oracle and can run in time up to 2^{λ}.

2.2 The Birthday Paradox and Its Generalization

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 probabilities for repeated outcomes of hash functions and related structures.

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

Lemma 1. *[6, 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 [6] can be easily generalized to determine how many random samples are needed to find multiple pairs of repeated outcomes. In this paper we call this the generalized birthday paradox, noting that it differs from other generalizations [17,7].

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 given by:*

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

The generalized form reveals that the sample complexity scales proportionally to $\sqrt{c}$.

3 A new weak PRF

At Crypto 2024, Alamati et al. [1] 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. [4]. This paradigm, which alternates computations over two distinct moduli, typically $\mathbb{F}_2$ followed by $\mathbb{F}_3$, has demonstrated significant potential for 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 to achieve λ-bit security under various constraints.

Specification. In their work [4], 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 $\cdot_p$ denotes multiplication modulo p, the matrix $\mathbf{K} \in \mathbb{F}_2^{m \times n}$ is the secret key and $\mathbf{K} \cdot_2 x \in \mathbb{F}_2^m$ is embedded into $\mathbb{F}_3^m$ component-wise in the natural way. Extensions [9] 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 optimises the end-to-end cost of MPC protocols while enhancing performance during the main computation phase, thus significantly improving communication complexity and computational efficiency. The 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 $\mathbf{B}$.

Definition of the Standard wPRF. The proposed standard $(\mathbb{F}_2, \mathbb{F}_3)$-wPRF is defined as:

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

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

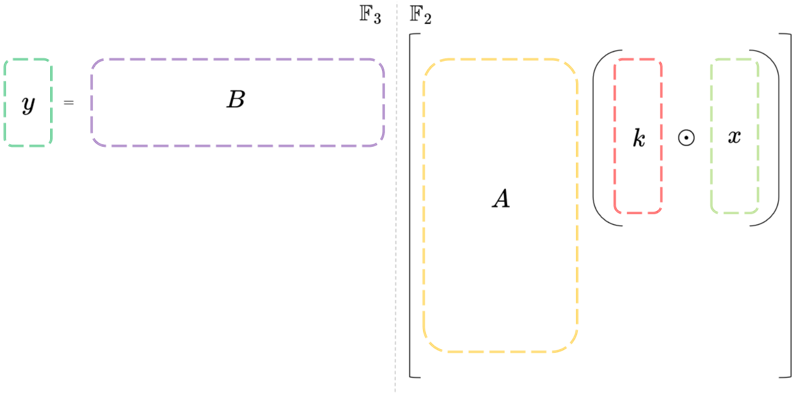

Fig. 1: Construction of the standard version of the new wPRF.

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

Parameters. Table 2 summarizes the recommended parameter sets from [1] across the different wPRF constructions. Alamati et al. divide the parameter sets into two groups, namely One-to-One parameters and Many-to-One parameters. The authors claim λ-bit security for each parameter set.

One-to-One Parameters. The One-to-One parameter set is designed with the aim of giving a (roughly) one-to-one mapping between inputs and outputs. Specifically, the input space and output space are of the same size, and for any given input x the authors claim we can expect a unique corresponding output y. This setup is 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. For any given output y there should be many multiple values for x mapping to y, leading to a many-to-one mapping between inputs and outputs.

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

4 Our Attack

In this section, we present a key recovery attack against the two One-to-One parameter sets proposed by the authors. This attack 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 to the wPRF in the reversed moduli variant, demonstrating a significant reduction in complexity compared to the claimed 2^{λ} calls. The attack begins 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 to the Reversed Moduli variant, overcoming its additional complexities.

In the following, let X denote the input space, let M denote the output space of the multiplication using 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 has size $|X| = 2^{2\lambda}$ $(= 3^{(2\lambda/\log_2 3)})$, 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, but this 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 $h_0 = 2\lambda - h_1$ as 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 implies that 2^{h_0} distinct values of x will give the same input to the multiplication with $\mathbf{A}$, and the weak PRF can never recover from this 2^{h_0}-to-1 sub-mapping, meaning that the whole F becomes a 2^{h_0}-to-1 mapping once the key is fixed. The image of F, denoted as $im(F)$, will therefore be of size $2^{h_1} \approx 2^{\lambda}$ instead of $2^{2\lambda}$.
- *Reversed Moduli One-to-One.* Extending the notation, let h_i be the number of elements in k that have the value i (for $i = 0, 1, 2$). For a uniformly random key k, we expect $h_0 \approx h_1 \approx h_2 \approx \dfrac{2\lambda}{3\log_2(3)}$. Similarly to the standard case, the operation $k \odot x$ will induce 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. Thus, $im(F)$ will be of 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 One-to-One Standard 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 weak PRF on random inputs, building up a table of input and output values (x, y). By the birthday paradox, collisions are expected to appear after collecting $\sqrt{2|im(F)|} = 2^{(h_1+1)/2}$ samples.

For two inputs x and x' producing the same output y, if $x_i \neq x'_i$, then k_i must be zero because the differing input bits would otherwise result in different values in M. Note that the size of M is $2^{7.06\lambda}$ which is much larger than the size of X. So the probability of creating collisions *after* multiplying $k \odot x$ with $\mathbf{A}$ becomes negligible since there are only 2^{h_1} different values going into multiplication with $\mathbf{B}$, which is much smaller than $|Y| = 2^{2\lambda}$. Therefore, with overwhelming probability, the only source of collisions comes from the 2^{h_0}-to-1 mapping of $x \odot k$, which means that each collision reveals some positions in k that must be zero.

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 we find more collisions, 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 that $X_{\neq} \subseteq J_0$. 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. We therefore expect that only half of the set J_0 will be revealed from any one collision. With further collisions we learn more positions of J_0, but as the inputs are drawn uniformly at random, collisions are independent of each other and new collisions are only expected to reveal half of the so far unrevealed positions where $k_i = 0$. This suggests that the Hamming distance between K and k is halved with each collision. Specifically, the expected Hamming distance after c collisions have been found can be expressed as

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

Collision Saturation Point. As the attack progresses, the number of new revealed positions in J_0 diminishes with each new collision, as many zeros in k have already been determined. At some stage it becomes more efficient to perform an exhaustive search among keys with small Hamming distances from the current guess K. The switch should occur when the expected cost of generating the $(c+1)$-th collision exceeds the expected cost of exhaustive search among the vectors with hamming distance d_c from K.

Cost of New Collision. Using the generalized birthday paradox, 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 having generated c collisions, the number of new queries we need 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 number of queries since verifying whether we have a new collision 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 d_c from K. Note that we only need to search for possible keys by changing 1-bits in K to 0, and never changing 0-bits to 1. We therefore call this search as searching within the *one-sided Hamming distance* from K. The number of keys to search is therefore $\sum_{j=1}^{\lceil d_c \rceil} \binom{H_1}{j}$, where H_1 is the current Hamming weight of K. To verify each key candidate, we compute three outputs using the current key guess and some x that has already been queried. The number three is chosen somewhat arbitrarily, but should guarantee that only the correct key will pass the verification. In the worst case, this results in a total cost of

$$3 \cdot \sum_{j=1}^{\lceil d_c \rceil} \binom{H_1}{j}$$

queries.

Finally, we insert $h_0 = h_1 = \lambda$, their expected values, in equations (2) and (1) to compute the point to switch to exhaustive search. To minimize the attack cost, the transition to exhaustive search should occur when c collisions have been found and

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

Of course, we are not guaranteed that the correct key has Hamming distance less than d_c whenever (3) is satisfied. If the exhaustive search fails to find the correct key on the first try, we simply find one more collision and then try exhaustive search again.

Complexity Analysis. We measure the complexity of the attack in terms of the needed number of queries to the weak PRF. The attack proceeds by identifying collisions until a transition point is reached, at which point exhaustive search on the key is performed. Let C represent the number of collisions found 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

$$\sqrt{2^{\lambda+1} C} + 3 \cdot \sum_{j=1}^{\lceil \lambda/2^C \rceil} \binom{H_1}{j},$$

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

With $C = \log_2 \lambda$ the sum in the expression above will stop at $j = 1$, and consequently, the complexity of the attack is of the order

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

Algorithm 1 Key Recovery Attack

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

  $K \leftarrow [1, 1, \dots, 1]$
  $\mathcal{P} \leftarrow \emptyset$
  $c \leftarrow 0$
  $H_1 \leftarrow n$
  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 $3 \cdot \sum_{j=1}^{\lceil \lambda/2^c \rceil} \binom{H_1}{j} \leq 2^{(\lambda+1)/2} \cdot (\sqrt{c+1} - \sqrt{c})$ then
          for  each $k'$ with one-sided Hamming distance at most $d_c$ from $K$ do
              if $k'$ matches 3 input-output pairs from $\mathcal{P}$ then
                  return $k'$
              end if
          end for
      end if
  end while
```

The total cost of the attack is thus significantly lower than 2^λ, highlighting a clear compromise of the claimed security level.

4.2 Attack on One-to-One Reversed Moduli wPRF

We adapt the collision-based key recovery attack methodology used in the standard parameter set to the One-to-One reversed moduli 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.

Collisions: Finding 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 method of finding collisions used previously in the standard case. By the birthday paradox, 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 from the 3^{h_0}-to-1 mapping induced by $x \odot k$ 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 we have likely identified all zero positions in the key. 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 and reveals another $2/3^2 \cdot h_0$ zeroes. Following this reasoning, after c collisions, the total number of zeroes revealed can be expressed as

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

After having found c collisions, the number of remaining zero positions in k yet to be identified is therefore 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.,

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

This expression can be solved for c in order to find the minimum expected number of collisions after which we should identify all zero positions of the key. To ensure that all zero positions are found with high probability, we add a small safety margin. Specifically, we can multiply the derived value for c by three, and in any case the complexity of determining all zero positions is of the order $\mathcal{O}(\log_3(\lambda))$ collisions.

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 two possible values, 1 or 2 since all 0's have been detected. For a key of length $n = \frac{2\lambda}{\log_2 3}$, 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 three input-output pairs as before. Thus, the exhaustive search over all possible keys in $J_1 \cup J_2$ requires $3 \cdot 2^{0.84\lambda}$ queries.

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, 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 3 queries. This results in a total cost of

$$3 \cdot 2^{0.84\lambda}.$$

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 below the claimed security level of 2^{λ}, demonstrating that the One-to-One Reversed Moduli parameter set also is broken.

4.3 Trying the Attack on 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 $|Y| = 2^{\lambda}$, so collisions are inevitable. Since the intermediate output space of the pointwise multiplication followed by multiplication with $\mathbf{A}$ has size $|M| = 2^{2\lambda}$, these collisions will mostly arise without being linked to $k \odot x$. More specifically, distinct points in M produce a collision in Y at a rate of once per $2^{\lambda/2}$ queries, while collisions due to $k \odot x$ being a $2^{2\lambda}$-to-1 mapping only appear at a rate of once for every 2^{λ} queries. As a result, generating even a single collision where $x \odot k = x' \odot k$ will take $\mathcal{O}(2^{\lambda})$ time, making our approach ineffective for these parameter sets.

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 analyse the average complexity of the two principal components as 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 a sufficient number of collisions have been identified, we perform an exhaustive search over the key candidates with small Hamming distances from the current key guess. We record the average number of calls to F for this step.

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

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} are the total number of calls to the PRF oracle, including the cases where the attack had to go back and find one more 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 having found C collisions, the key is successfully recovered on the first attempt at exhaustive search. The exhaustive search will be over all key candidates with one-sided Hamming distance d_C from the current key guess K. The measured success rates for recovering the correct key on the first attempt at exhaustive search 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, on the average, the attack requires approximately 4.39 collisions for $\lambda = 28$ 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 implies, on the average, a higher Hamming weight of the key, 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 our failure to approximate 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, as seen in our results, we observe fewer such deviations. We therefore hypothesize that this theoretical trend would hold 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.

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.

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

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 collision found.

The graph shows that the Hamming distance roughly halves with each collision, as expected.

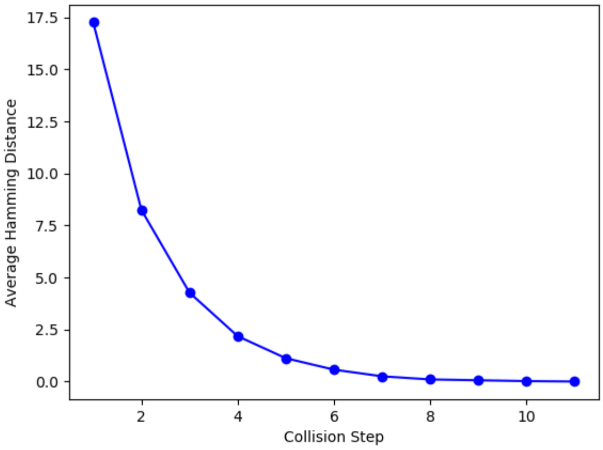

Fig. 2: Number of found collisions vs. the average Hamming distance between the guessed key and the actual key for $\lambda = 34$.

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 demonstrated 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 are far 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 0's as coefficients in the key and making sure that no part of the input is zeroed out as the first operation of the wPRF. The drawback of this mitigation is that p must be greater than 2 for this countermeasure to make sense, 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

1. Alamati, N., Policharla, G.V., Raghuraman, S., Rindal, P.: Improved Alternating-Moduli PRFs and Post-quantum Signatures. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024. pp. 274–308. Springer Nature Switzerland, Cham (2024)
2. Albrecht, M.R., Grassi, L., Perrin, L., Ramacher, S., Rechberger, C., Rotaru, D., Roy, A., Schofnegger, M.: Feistel Structures for MPC, and More. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) Computer Security – ESORICS 2019. pp. 151–171. Springer International Publishing, Cham (2019)
3. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015. pp. 430–454. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)
4. Boneh, D., Ishai, Y., Passelègue, A., Sahai, A., Wu, D.J.: Exploring Crypto Dark Matter. In: Beimel, A., Dziembowski, S. (eds.) Theory of Cryptography. pp. 699–729. Springer International Publishing, Cham (2018)
5. Chaum, D.: Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology. pp. 199–203. Springer US, Boston, MA (1983)

6. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Second Edition. MIT Press (2001)
7. DasGupta, A.: The matching, birthday and the strong birthday problem: a contemporary review. Journal of Statistical Planning and Inference **130**(1), 377–389 (2005). https://doi.org/https://doi.org/10.1016/j.jspi.2003.11.015, https://www.sciencedirect.com/science/article/pii/S0378375804002721, herman Chernoff: Eightieth Birthday Felicitation Volume
8. Dinur, I., Goldfeder, S., Halevi, T., Ishai, Y., Kelkar, M., Sharma, V., Zaverucha, G.: MPC-Friendly Symmetric Cryptography from Alternating Moduli: Candidates, Protocols, and Applications. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 517–547. Springer International Publishing, Cham (2021)
9. Dinur, I., Goldfeder, S., Halevi, T., Ishai, Y., Kelkar, M., Sharma, V., Zaverucha, G.: MPC-Friendly Symmetric Cryptography from Alternating Moduli: Candidates, Protocols, and Applications. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 517–547. Springer International Publishing, Cham (2021)
10. Dobraunig, C., Eichlseder, M., Grassi, L., Lallemand, V., Leander, G., List, E., Mendel, F., Rechberger, C.: Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018. pp. 662–692. Springer International Publishing, Cham (2018)
11. Freedman, M.J., Ishai, Y., Pinkas, B., Reingold, O.: Keyword Search and Oblivious Pseudorandom Functions. In: Kilian, J. (ed.) Theory of Cryptography. pp. 303–324. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
12. Grassi, L., Øygarden, M., Schofnegger, M., Walch, R.: From Farfalle to Megafono via Ciminion: The PRF Hydra for MPC Applications. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 255–286. Springer Nature Switzerland, Cham (2023)
13. Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., Smart, N.P.: MPC-Friendly Symmetric Key Primitives. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 430–443. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978332, https://doi.org/10.1145/2976749.2978332
14. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: 40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039). pp. 120–130 (1999). https://doi.org/10.1109/SFFCS.1999.814584
15. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: Proceedings 38th Annual Symposium on Foundations of Computer Science. pp. 458–467 (1997). https://doi.org/10.1109/SFCS.1997.646134
16. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) Advances in Cryptology — ASIACRYPT 2001. pp. 552–565. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
17. Wagner, D.: A Generalized Birthday Problem. In: Yung, M. (ed.) Advances in Cryptology — CRYPTO 2002. pp. 288–304. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)