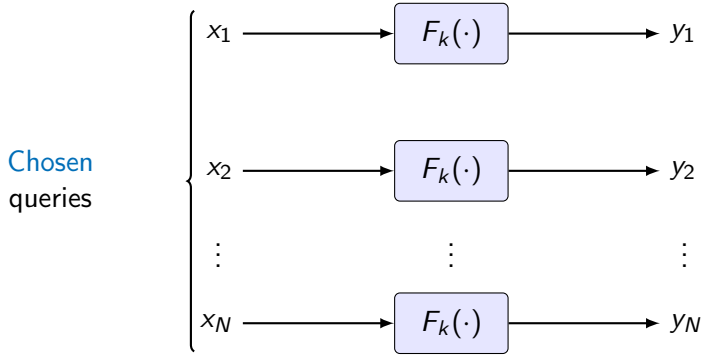
Cryptanalysis of the Alternative Mod-2/Mod-3 Weak PRF

Augustin Bariant[1], Christina Boura[2], Baptiste Germon[2,3], Rachelle Heim Boissier[4], Charles Meyer-Hilfiger[3], Tyge Tiessen[5]

[1] ANSSI
[2] Université Paris Cité, IRIF
[3] Univ Rennes, Inria, CNRS, IRISA
[4] ULB, Belgium, [5] Technical University of Denmark

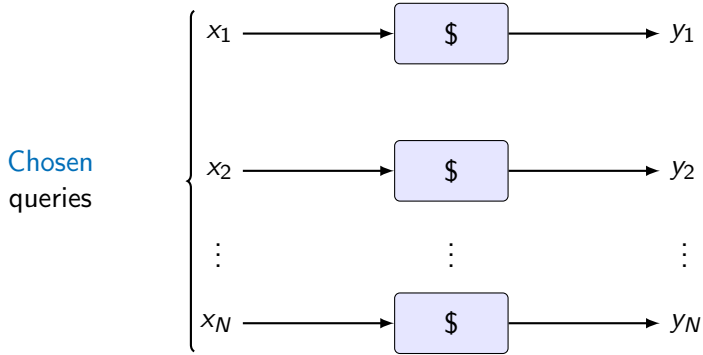
Pseudorandom Function (PRF)

Let F_k be a family of functions parameterized by a secret key.



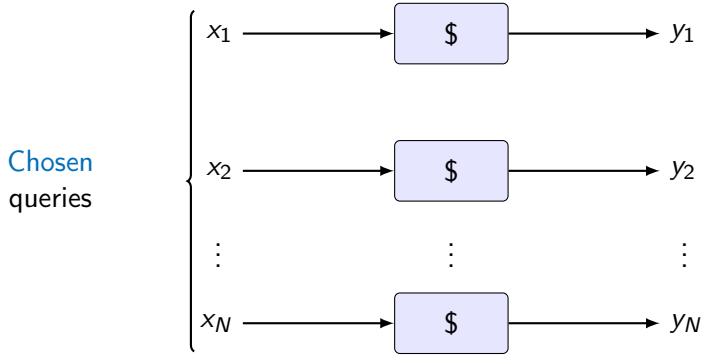
Pseudorandom Function (PRF)

Let F_k be a family of functions parameterized by a secret key.



Pseudorandom Function (PRF)

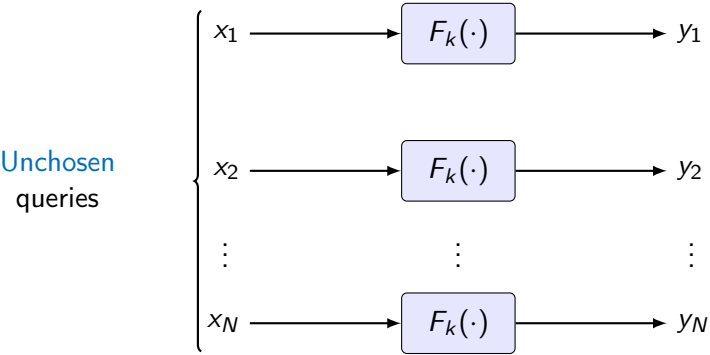
Let F_k be a family of functions parameterized by a secret key.



Given chosen queries, can an adversary distinguish F_k from random in reasonable time?

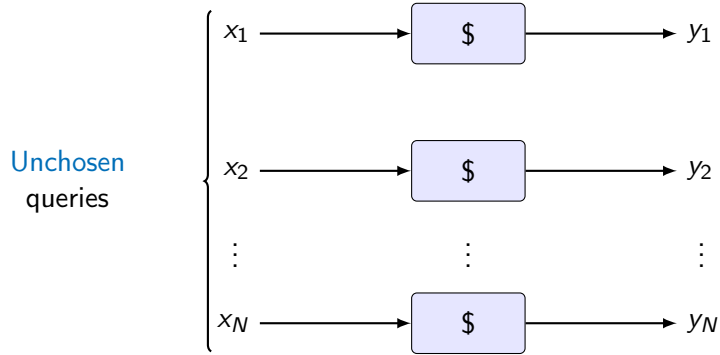
Weak Pseudorandom Function (w-PRF)

Let F_k be a family of functions parameterized by a secret key.



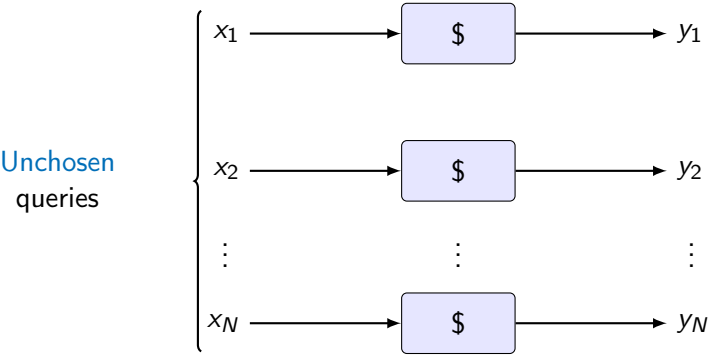
Weak Pseudorandom Function (w-PRF)

Let F_k be a family of functions parameterized by a secret key.



Weak Pseudorandom Function (w-PRF)

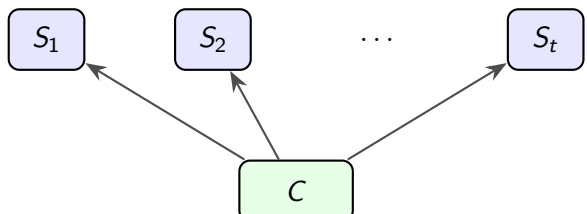
Let F_k be a family of functions parameterized by a secret key.



Given uniformly drawn and independent queries, can an adversary distinguish F_k from random in reasonable time?

What For?

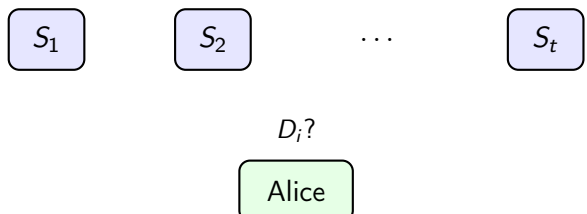
- Can be used to generate OTP randomness. (Less efficient than AES-NI).
- In MPC protocols:



C shares encrypted dataset to servers.

What For?

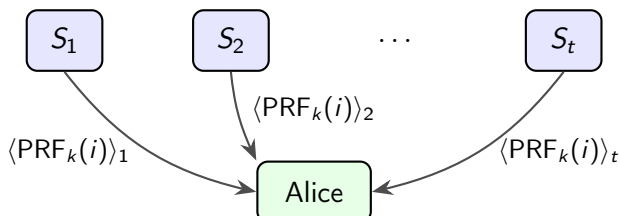
- Can be used to generate OTP randomness. (Less efficient than AES-NI).
- In MPC protocols:



Alice wants to recover D_i.

What For?

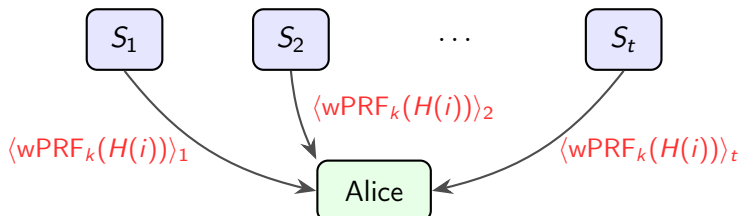
- Can be used to generate OTP randomness. (Less efficient than AES-NI).
- In MPC protocols:



Servers evaluate $\mathrm{PRF}_k(i)$ through the MPC protocol.

What For?

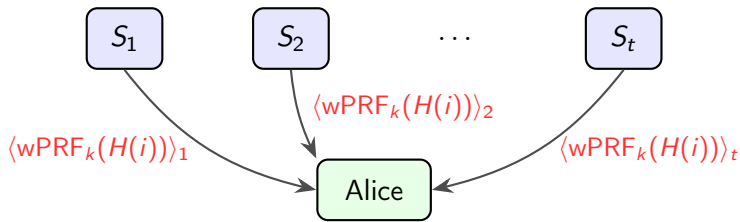
- Can be used to generate OTP randomness. (Less efficient than AES-NI).
- In MPC protocols:



Servers evaluate $\mathrm{wPRF}_k(H(i))$ through the MPC protocol.

What For?

- Can be used to generate OTP randomness. (Less efficient than AES-NI).
- In MPC protocols:

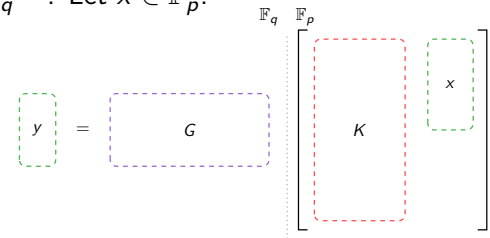


Servers evaluate $\mathrm{wPRF}_k(H(i))$ through the MPC protocol.

- Create efficient Pseudo-Random Correlation Function with wPRF.
- And other stuff...

The Alternating-Moduli Construction

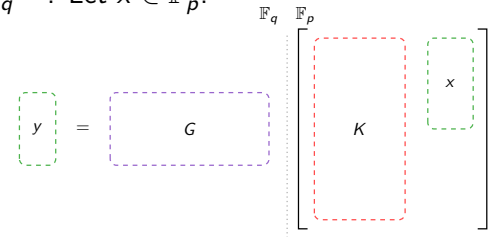
Introduced by Boneh et al. at TCC 2018[1]. Let p, q be two distinct primes. Let $K \xleftarrow{\$} \mathbb{F}_p^{m \times n}$ and $G \xleftarrow{\$} \mathbb{F}_q^{t \times m}$. Let $x \in \mathbb{F}_p^n$.



[1] [BIPSW18], Boneh, Ishai, Passelègue, Sahai, Wu, *Exploring Crypto Dark Matter*, TCC 2018

The Alternating-Moduli Construction

Introduced by Boneh et al. at TCC 2018[1]. Let p, q be two distinct primes. Let $K \xleftarrow{\$} \mathbb{F}_p^{m \times n}$ and $G \xleftarrow{\$} \mathbb{F}_q^{t \times m}$. Let $x \in \mathbb{F}_p^n$.



Several cryptanalysis works: [CCKK21] & [HLR$^+$26]

[2] [CCKK21], Cheon, Cho, Kim, Kim, *Adventures in crypto dark matter*, PKC 2021

[3] [HLR$^+$26], Hu, Leander, Raddum, Sandrib, Udovenko, *Cryptanalysis of Two Alternating Moduli Weak PRFs*, ToSC 2026

Alternative Construction

Boneh et al. proposed another alternative construction. Let $k \xleftarrow{\$} \mathbb{F}_2^n$.

$$\begin{aligned} F_k : \mathbb{F}_2^n &\to \mathbb{F}_2 \\ x &\mapsto (\langle x, k \rangle \bmod 2 + \langle x, k \rangle \bmod 3) \bmod 2 \end{aligned}$$

Alternative Construction

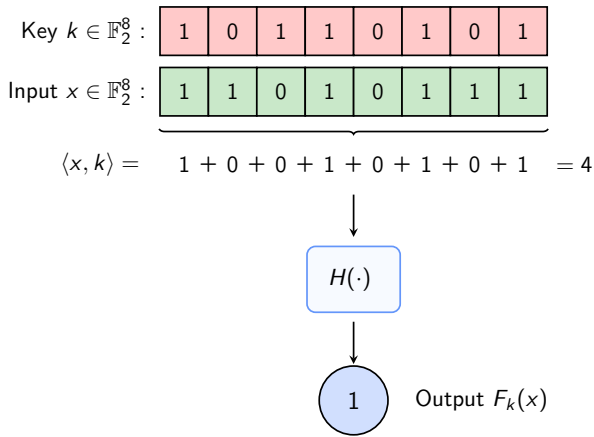
Boneh et al. proposed another alternative construction. Let $k \xleftarrow{\$} \mathbb{F}_2^n$.

$$
\begin{aligned}
F_k : \mathbb{F}_2^n &\to \mathbb{F}_2 \\
x &\mapsto (\langle x, k \rangle \bmod 2 + \langle x, k \rangle \bmod 3) \bmod 2 \\
&= H(\langle x, k \rangle)
\end{aligned}
$$

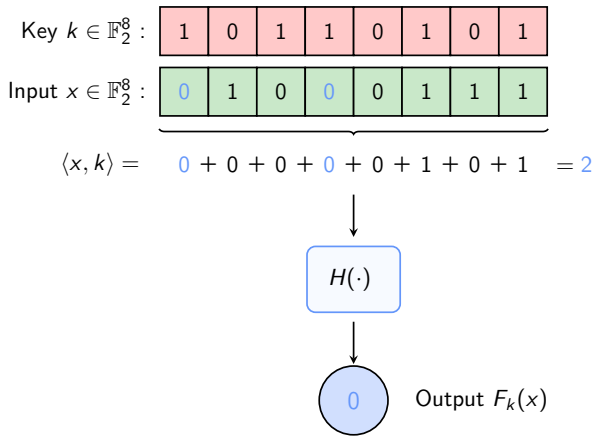
where

$$
H(r) = \begin{cases} 1 \text{ if } r \bmod 6 \in \{3, 4, 5\} \\ 0 \text{ otherwise} \end{cases}
$$

Example


Key $k \in \mathbb{F}_2^8$:
1 0 1 1 0 1 0 1
Input $x \in \mathbb{F}_2^8$:
1 1 0 1 0 1 1 1
$\langle x, k \rangle =$ $1 + 0 + 0 + 1 + 0 + 1 + 0 + 1$ $= 4$
$H(\cdot)$
1
Output $F_k(x)$

Example


Key $k \in \mathbb{F}_2^8$:
1 0 1 1 0 1 0 1
Input $x \in \mathbb{F}_2^8$:
0 1 0 0 0 1 1 1
$\langle x, k \rangle =$ 0 + 0 + 0 + 0 + 0 + 1 + 0 + 1 = 2
$H(\cdot)$
0
Output $F_k(x)$

Security Claims

- Can be viewed as an LPN problem:

Decisional LPN(n, ϵ)

Let $k \in \mathbb{F}_2^n$. ϵ is called the noise rate. We are given samples of the form $(x, \langle x, k \rangle + e)$ where $x \xleftarrow{\$} \mathbb{F}_2^n$ and $e \sim \mathcal{B}(\epsilon)$. The goal is to distinguish this distribution from samples of the form (x, r) with $r \xleftarrow{\$} \{0, 1\}$.

[1] [BKW00], Blum, Kalai, Wasserman, *Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model*, ACM STOC 2000

Security Claims

- Can be viewed as an LPN problem:

Decisional LPN(n, ϵ)

Let $k \in \mathbb{F}_2^n$. ϵ is called the noise rate. We are given samples of the form $(x, \langle x, k \rangle + e)$ where $x \xleftarrow{\$} \mathbb{F}_2^n$ and $e \sim \mathcal{B}(\epsilon)$. The goal is to distinguish this distribution from samples of the form (x, r) with $r \xleftarrow{\$} \{0, 1\}$.

$$F_k(x) = \underbrace{\langle x, k \rangle \bmod 2}_{\textit{parity}} + \underbrace{(\langle x, k \rangle \bmod 3) \bmod 2}_{\textit{noise},\ \epsilon = 1/3}$$

- Generic solvers: BKW[1] with asymptotic time and data complexity $2^{\mathcal{O}(\frac{n}{\log_2 n})}$.
- The designers claimed 128 bits of security for $n = 384$.

[1] [BKW00], Blum, Kalai, Wasserman, *Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model*, ACM STOC 2000

Overview & Contributions

Attack	Asymptotic		$n = 384$		$n = 768$	
	Time	Data	Time	Data	Time	Data
[CCKK21]	$\mathcal{O}(2^{0.21n})$	$\mathcal{O}(2^{0.21n})$	$2^{80.6}$	$2^{80.6}$	$2^{161.3}$	$2^{161.3}$
[JMN23][1]	$\mathcal{O}(2^{0.166n})$	$\mathcal{O}(2^{0.1n})$	$2^{74.27}$	$2^{44.5}$	$2^{138.8}$	$2^{83.62}$
[HLR+26]	—	—	$2^{149.2}$	$2^{108.31}$	$2^{207.9}$	$2^{206.31}$

[1] [JMN23], Johansson, Meier, Nguyen, *Differential cryptanalysis of Mod-2/Mod-3 constructions of binary weak PRFs*, IEEE ISIT 2023

Overview & Contributions

Attack	Asymptotic		$n = 384$		$n = 768$	
	Time	Data	Time	Data	Time	Data
[CCKK21]	$\mathcal{O}(2^{0.21n})$	$\mathcal{O}(2^{0.21n})$	$2^{80.6}$	$2^{80.6}$	$2^{161.3}$	$2^{161.3}$
[JMN23][1]	$\mathcal{O}(2^{0.166n})$	$\mathcal{O}(2^{0.1n})$	$2^{74.27}$	$2^{44.5}$	$2^{138.8}$	$2^{83.62}$
[HLR+26]	—	—	$2^{149.2}$	$2^{108.31}$	$2^{207.9}$	$2^{206.31}$
Ours	$\mathcal{O}(2^{0.099n})$	$\mathcal{O}(2^{0.099n})$	$2^{43.16}$	$2^{43.16}$	$2^{81.58}$	$2^{81.58}$
+ Splitting strategy	$\tilde{\mathcal{O}}(2^{0.09n})$	$\tilde{\mathcal{O}}(2^{0.09n})$	$2^{40.72}$	$2^{40.46}$	$2^{75.76}$	$2^{75.67}$

[1] [JMN23], Johansson, Meier, Nguyen, *Differential cryptanalysis of Mod-2/Mod-3 constructions of binary weak PRFs*, IEEE ISIT 2023

Overview & Contributions

Attack	Asymptotic		$n = 384$		$n = 768$	
	Time	Data	Time	Data	Time	Data
[CCKK21]	$\mathcal{O}(2^{0.21n})$	$\mathcal{O}(2^{0.21n})$	$2^{80.6}$	$2^{80.6}$	$2^{161.3}$	$2^{161.3}$
[JMN23][1]	$\mathcal{O}(2^{0.166n})$	$\mathcal{O}(2^{0.1n})$	$2^{74.27}$	$2^{44.5}$	$2^{138.8}$	$2^{83.62}$
[HLR+26]	—	—	$2^{149.2}$	$2^{108.31}$	$2^{207.9}$	$2^{206.31}$
Ours	$\mathcal{O}(2^{0.099n})$	$\mathcal{O}(2^{0.099n})$	$2^{43.16}$	$2^{43.16}$	$2^{81.58}$	$2^{81.58}$
+ Splitting strategy	$\tilde{\mathcal{O}}(2^{0.09n})$	$\tilde{\mathcal{O}}(2^{0.09n})$	$2^{40.72}$	$2^{40.46}$	$2^{75.76}$	$2^{75.67}$
[JMN23] revisited	$\mathcal{O}(2^{0.162n})$	$\mathcal{O}(2^{0.097n})$	—	—	—	—
+ Weak keys	$\mathcal{O}(2^{0.147n})$	$\mathcal{O}(2^{0.08n})$	—	—	—	—

[1] [JMN23], Johansson, Meier, Nguyen, *Differential cryptanalysis of Mod-2/Mod-3 constructions of binary weak PRFs*, IEEE ISIT 2023

Outline

1. Attack based on singletons
2. Splitting strategy
3. Johansson et al.'s attack revisited
4. Conclusion

Cheon et al.'s Attack

In their attack, they look at:

$$\Pr\left[x_j = 0 \mid k_j = 1 \wedge F_k(x) = 0\right] \approx \frac{1}{2} \pm \underbrace{\frac{1}{2^{0.21\beta n}}}_{bias(n,\beta)}$$

where $hw(k) = \beta n$.

Therefore, they have an attack in $\mathcal{O}\left(\frac{1}{bias(n, 0.5)^2}\right)$ time and data complexity.

Cheon et al.'s Attack

In their attack, they look at:

$$\Pr\left[x_j = 0 \mid k_j = 1 \wedge F_k(x) = 0\right] \approx \frac{1}{2} \pm \underbrace{\frac{1}{2^{0.21\beta n}}}_{bias(n,\beta)}$$

where $hw(k) = \beta n$.
Therefore, they have an attack in $\mathcal{O}(2^{0.21n})$ time and data complexity.

Cheon et al.'s Attack

In their attack, they look at:

$$\Pr\left[x_j = 0 \mid k_j = 1 \wedge F_k(x) = 0\right] \approx \frac{1}{2} \pm \underbrace{\frac{1}{2^{0.21\beta n}}}_{bias(n,\beta)}$$

where $hw(k) = \beta n$.
Therefore, they have an attack in $\mathcal{O}(2^{0.21n})$ time and data complexity.

Main Takeaway

Under **fixed-weight condition** on the key, they have a **strong** bias!

Bias

We want to fix the weight of both the **key AND the input**!

Bias

We want to fix the weight of both the **key AND the input**!

Bias

Let k be a key of weight βn and $0 \leq \alpha \leq 1$ such that $\alpha n \in \mathbb{N}$. We consider:

$$\Pr\left[F_k(x) = 1 \mid hw(x) = \alpha n\right] = \frac{1}{2} + \epsilon(n, \alpha, \beta)$$

Exact Formula

Let's count $C_\alpha = |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge F_k(x) = 1\}|$.

$$F_k(x) = 1 \Leftrightarrow H(\langle x, k \rangle) = 1$$

Exact Formula

Let's count $C_\alpha = |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge F_k(x) = 1\}|$.

$$F_k(x) = 1 \Leftrightarrow H(\langle x, k \rangle) = 1$$

$$C_\alpha = \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge \langle x, k \rangle = i\}|$$

Exact Formula

Let's count $C_\alpha = |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge F_k(x) = 1\}|$.

$$F_k(x) = 1 \Leftrightarrow H(\langle x, k \rangle) = 1$$

$$\begin{aligned} C_\alpha &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge \langle x, k \rangle = i\}| \\ &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} \binom{\beta n}{i} \binom{n - \beta n}{\alpha n - i} \end{aligned}$$

Exact Formula

Let's count $C_\alpha = |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge F_k(x) = 1\}|$.

$$F_k(x) = 1 \Leftrightarrow H(\langle x, k \rangle) = 1$$

$$\begin{aligned} C_\alpha &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge \langle x, k \rangle = i\}| \\ &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} \binom{\beta n}{i} \binom{n-\beta n}{\alpha n - i} \end{aligned}$$

$$\Pr[F_k(x) = 1 \mid hw(x) = \alpha n] = \binom{n}{\alpha n}^{-1} \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} \binom{\beta n}{i} \binom{n-\beta n}{\alpha n - i}$$

Exact Formula

Let's count $C_\alpha = |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge F_k(x) = 1\}|$.

$$F_k(x) = 1 \Leftrightarrow H(\langle x, k \rangle) = 1$$

$$\begin{aligned} C_\alpha &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} |\{x \in \mathbb{F}_2^n \mid hw(x) = \alpha n \wedge \langle x, k \rangle = i\}| \\ &= \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} \binom{\beta n}{i} \binom{n - \beta n}{\alpha n - i} \end{aligned}$$

$$\epsilon(n, \alpha, \beta) = \binom{n}{\alpha n}^{-1} \sum_{\substack{i=0 \\ H(i)=1}}^{\beta n} \binom{\beta n}{i} \binom{n - \beta n}{\alpha n - i} - \frac{1}{2}.$$

Asymptotic Formula

We have the following formula:

$$\epsilon(n,\alpha,\beta) \underset{n\to+\infty}{\sim} \frac{1}{6}\exp\left(-\frac{\pi^2\sigma^2}{18}n\right)\cos\left(\frac{\pi}{3}\mu n+\frac{2\pi}{3}\right)+\mathcal{O}\left(\exp\left(\frac{-\pi^2\sigma^2}{2}n\right)\right).$$

with $\mu=\alpha\beta n$ and $\sigma^2=\alpha(1-\alpha)\beta(1-\beta)$.

Asymptotic Formula

We have the following formula:

$$\epsilon(n,\alpha,\beta) \underset{n\to+\infty}{\sim} \frac{1}{6}\exp\left(-\frac{\pi^2\sigma^2}{18}n\right)\cos\left(\frac{\pi}{3}\mu n+\frac{2\pi}{3}\right)+\mathcal{O}\left(\exp\left(\frac{-\pi^2\sigma^2}{2}n\right)\right).$$

with $\mu=\alpha\beta n$ and $\sigma^2=\alpha(1-\alpha)\beta(1-\beta)$.

The decay rate is symmetric in α and β around 0.5!

Asymptotic Formula

We have the following formula:

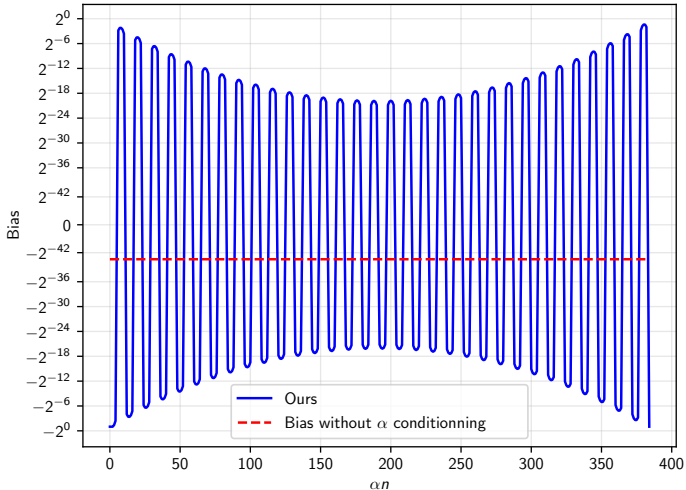
$$\epsilon(n,\alpha,\beta) \underset{n\to+\infty}{\sim} \frac{1}{6}\exp\left(-\frac{\pi^2\sigma^2}{18}n\right)\cos\left(\frac{\pi}{3}\mu n+\frac{2\pi}{3}\right)+\mathcal{O}\left(\exp\left(\frac{-\pi^2\sigma^2}{2}n\right)\right).$$

with $\mu=\alpha\beta n$ and $\sigma^2=\alpha(1-\alpha)\beta(1-\beta)$.

The decay rate is symmetric in α and β around 0.5!

The worst case is at $\alpha=\beta=0.5$ and $\epsilon(n,0.5,0.5)\approx\mathcal{O}(2^{-0.0495n})$.

Why Such Quadratic Improvement?



Université de Rennes

Outline

Outline

Statistical Reminders

We want to distinguish whether $(y_1, \ldots, y_N)$ are drawn i.i.d with:

$$\overbrace{\mathcal{B}(0.5)}^{\$} \text{ or with } \overbrace{\mathcal{B}(0.5 + \epsilon(n, \alpha, \beta))}^{F_k(\cdot)}.$$

Score Function

$$S(y_1, \ldots, y_N) = \sum_{i=0}^{N} (-1)^{y_i}$$

Random Case:
$S(y_1, \ldots, y_N) \approx 0$

w-PRF Case:
$|S(y_1, \ldots, y_N)|$ is high

Distinguishing Contraint

We need $N > \frac{1}{\epsilon(n,\alpha,\beta)^2}$ to correctly distinguish.

Basic Attack

1. Query $N' = N \cdot \frac{1}{p_\alpha}$ samples (x_i, y_i) where $p_\alpha = \Pr(hw(x) = \alpha n)$.
2. Filter all (x_i, y_i) such that $hw(x_i) = \alpha n$ and store them in L_α.
3. Compute $S(L_\alpha) = \sum_{(x_i, y_i) \in L_\alpha} (-1)^{y_i}$.
4. If $S(L_\alpha)$ is "big enough" return "w-PRF".
5. Otherwise return "Random".

Basic Attack

1. Query $N' = N \cdot \frac{1}{p_\alpha}$ samples (x_i, y_i) where $p_\alpha = \Pr(hw(x) = \alpha n)$.
2. Filter all (x_i, y_i) such that $hw(x_i) = \alpha n$ and store them in L_α.
3. Compute $S(L_\alpha) = \sum_{(x_i, y_i) \in L_\alpha} (-1)^{y_i}$.
4. If $S(L_\alpha)$ is "big enough" return "w-PRF".
5. Otherwise return "Random".

The worst case is $\beta = 0.5$, in this case $\alpha = 0.5$ gives the best complexity:

$$\mathcal{T} = \mathcal{D} = \mathcal{O}\Big(\underbrace{2^{0.099n}}_{\epsilon(n, \frac{1}{2}, \frac{1}{2})^{-2}} \quad \underbrace{\sqrt{n}}_{\Pr(hw(x) = \frac{n}{2})^{-1}} \Big)$$

Aggregated Statistics

We want to further optimize the data complexity $\Rightarrow$ keep all the samples.

Aggregated Statistics

We now want to distinguish whether $(y_1, \ldots, y_N)$ are independently drawn with:

$$\overbrace{\mathcal{B}(0.5)}^{\$} \text{ or with } \overbrace{\mathcal{B}(0.5 + \epsilon(n, \alpha_i, \beta))}^{F_k(\cdot)} \text{ where } hw(y_i) = \alpha_i n.$$

Score Function

$$S(y_1, \ldots, y_N) = \sum_{i=0}^{N} (-1)^{y_i} \epsilon(n, \alpha_i, \beta)$$

Case Random:
$S(y_1, \ldots, y_N) \approx 0$

Case w-PRF:
$|S(y_1, \ldots, y_N)|$ is high

New Distinguishing Constraint

$$N > \frac{1}{\sum_\alpha p_\alpha \epsilon(n, \alpha, \beta)^2}$$

Aggregated Attack

1. Query N samples (x_i, y_i).
2. Split all samples (x_i, y_i) in sublists L_α depending on $hw(x_i)$.
3. Compute the score function $S = \sum_{\alpha} \epsilon(n, \alpha, \beta) \sum_{(x_i, y_i) \in L_\alpha} (-1)^{y_i}$.
4. If S is "big enough" return "w-PRF".
5. Otherwise return "Random".

Aggregated Attack

1. Query N samples (x_i, y_i).
2. Split all samples (x_i, y_i) in sublists L_α depending on $hw(x_i)$.
3. Compute the score function $S = \sum_{\alpha} \epsilon(n, \alpha, \beta) \sum_{(x_i, y_i) \in L_\alpha} (-1)^{y_i}$.
4. If S is "big enough" return "w-PRF".
5. Otherwise return "Random".

We gain a factor $\sqrt{n}$ on the data complexity.
On $n = 384$, $\mathcal{D} = 2^{43.16}, \mathcal{T} = 2^{43.16}$ which runs on a 32-core server in a few hours!

Outline

Outline

Splitting Strategy

Main Takeaway from previous attacks

$H(\langle x, k \rangle)$ is highly biased with **exponential decay in** n, α, β.

Splitting Strategy

Main Takeaway from previous attacks

$H(\langle x, k \rangle)$ is highly biased with **exponential decay in** n, α, β.

Can we "lower" those parameters?

Splitting Strategy

Main Takeaway from previous attacks

$H(\langle x, k \rangle)$ is highly biased with **exponential decay in** n, α, β.

Can we "lower" those parameters?

Splitting

Let $\mathcal{P} = \{1, \ldots, s\}$ and $\mathcal{N} = \{s+1, \ldots, n\}$ for some s. Then,

$$H(\langle x, k \rangle) = H(\langle x, k \rangle_{\mathcal{P}} + \langle x, k \rangle_{\mathcal{N}})$$

Splitting Strategy

Main Takeaway from previous attacks

$H(\langle x, k \rangle)$ is highly biased with **exponential decay in** n, α, β.

Can we "lower" those parameters?

Splitting

Let $\mathcal{P} = \{1, \ldots, s\}$ and $\mathcal{N} = \{s+1, \ldots, n\}$ for some s. Then,

$$H(\langle x, k \rangle) = H(\langle x, k \rangle_{\mathcal{P}} + \langle x, k \rangle_{\mathcal{N}})$$

If we know $\langle x, k \rangle_{\mathcal{P}}$, the bias is $\approx |\epsilon(n - s, \alpha - |x_{\mathcal{P}}|, \beta - |k_{\mathcal{P}}|)|$ which is exponentially better.

Splitting Strategy

Main Takeaway from previous attacks

$H(\langle x, k \rangle)$ is highly biased with **exponential decay in** n, α, β.

Can we "lower" those parameters?

Splitting

Let $\mathcal{P} = \{1, \ldots, s\}$ and $\mathcal{N} = \{s+1, \ldots, n\}$ for some s. Then,

$$H(\langle x, k \rangle) = H(\langle x, k \rangle_{\mathcal{P}} + \langle x, k \rangle_{\mathcal{N}})$$

We can guess $k_{\mathcal{P}}$!

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{old}(n)$ and $\mathcal{D}_{old}(n)$ are the complexities for the **basic** attack then,

$$\text{and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{old}(n-s)$$

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{old}(n)$ and $\mathcal{D}_{old}(n)$ are the complexities for the **basic** attack then,

$$\mathcal{T}_{\text{splitting}}(n) = 2^s s\mathcal{T}_{old}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{old}(n-s)$$

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{old}(n)$ and $\mathcal{D}_{old}(n)$ are the complexities for the **basic** attack then,

$$\mathcal{T}_{\text{splitting}}(n) = 2^s s\mathcal{T}_{old}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{old}(n-s)$$

But we can do much better!

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{old}(n)$ and $\mathcal{D}_{old}(n)$ are the complexities for the **basic** attack then,

$$\mathcal{T}_{\text{splitting}}(n) = 2^s s\mathcal{T}_{old}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{old}(n-s)$$

But we can do much better! $\rightarrow$ Compute score functions in batch with FFT-like techniques.

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{old}(n)$ and $\mathcal{D}_{old}(n)$ are the complexities for the **basic** attack then,

$$\mathcal{T}_{\text{splitting}}(n) = s2^s + s\mathcal{T}_{old}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{old}(n-s)$$

But we can do much better! $\rightarrow$ Compute score functions in batch with FFT-like techniques.

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{agr}(n)$ and $\mathcal{D}_{agr}(n)$ are the complexities for the aggregated attack then,

$$\mathcal{T}_{\text{splitting}}(n) = s2^s + s\mathcal{T}_{agr}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{agr}(n-s)$$

We can further optimize by combining **splitting strategy** and the **aggregated** attack.

Attack Strategy

- Guess that $k_{\mathcal{P}} = z$.
- Compute the score function $S(z)$.
- Loop over all possible values of z.
- If $\max_{z \in \mathbb{F}_2^s} |S(z)|$ is "high enough" return "w-PRF".
- Otherwise, return "Random".

Complexity

If $\mathcal{T}_{agr}(n)$ and $\mathcal{D}_{agr}(n)$ are the complexities for the aggregated attack then,

$$\mathcal{T}_{\text{splitting}}(n) = s2^s + s\mathcal{T}_{agr}(n-s) \text{ and } \mathcal{D}_{\text{splitting}}(n) = s\mathcal{D}_{agr}(n-s)$$

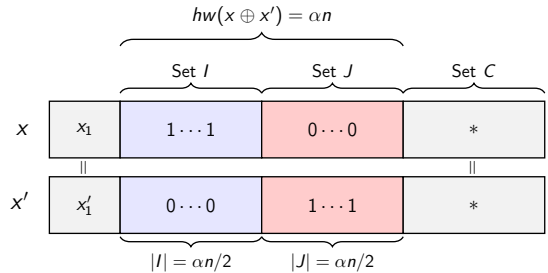
For $n = 384$: $\mathcal{T} = 2^{40.72}, \mathcal{D} = 2^{40.46}$.

Outline

Outline

Johansson et al's Attack

Investigate the bias between pairs of inputs (x, x') satisfying:



For those pairs we have:

$$\langle x, k \rangle = x_1 k_1 + \langle x, k \rangle_C + \langle x, k \rangle_I \qquad \langle x', k \rangle = x'_1 k_1 + \langle x', k \rangle_C + \langle x', k \rangle_J$$
$$= s + \langle x, k \rangle_I \qquad = s + \langle x', k \rangle_J$$

For small α, $\langle x, k \rangle_I \approx \langle x', k \rangle_J$

Bias

For pairs $(x, x') \in \mathcal{X}_\alpha$, $F_k(x)$ is highly dependent on $F_k(x')$.

Bias from [JMN23]

$$\Pr(F_k(x) = 0 \mid F_k(x') = 0, (x, x') \in \mathcal{X}_\alpha) = \frac{1}{2} \pm \frac{1}{2^{0.205\alpha n + 1.17}}$$

Bias

For pairs $(x, x') \in \mathcal{X}_\alpha$, $F_k(x)$ is highly dependent on $F_k(x')$.

Bias from [JMN23]

$$\Pr(F_k(x) = 0 \mid F_k(x') = 0, (x, x') \in \mathcal{X}_\alpha) = \frac{1}{2} \pm \frac{1}{2^{0.205\alpha n + 1.17}}$$

In the attack scenario the weight of the key is fixed!

Bias

For pairs $(x, x') \in \mathcal{X}_\alpha$, $F_k(x)$ is highly dependent on $F_k(x')$.

Bias from [JMN23]

$$\Pr(F_k(x) = 0 \mid F_k(x') = 0, (x, x') \in \mathcal{X}_\alpha) = \frac{1}{2} \pm \frac{1}{2^{0.205\alpha n + 1.17}}$$

In the attack scenario the weight of the key is fixed!

Our Bias

Let $k \in \mathbb{F}_2^n$ such that $hw(k) = \beta n$.

$$\Pr(F_k(x) = 0 \mid F_k(x') = 0, (x, x') \in \mathcal{X}_\alpha) = \frac{1}{2} + \gamma(n, \alpha, \beta, k_1)$$

Comparison with Johansson et al.

We found both an exact formula and a heuristic asymptotic approximation for $\gamma(n, \alpha, \beta, k_1)$.

Comparison with Johansson et al.

We found both an exact formula and a heuristic asymptotic approximation for $\gamma(n, \alpha, \beta, k_1)$.

$\ell = \alpha n/2$	6	8	12	15	18
Bias observed for $hw(k) = 310$	0.161	0.114	0.056	0.0363	0.0211
Bias for a random key	0.079	0.044	0.014	0.0059	0.0025

Table: Table I of [JMN23] with $n = 384$

Our formula correctly predicts the unexpected behavior for $\beta = \frac{310}{384}$

Comparison with Johansson et al.

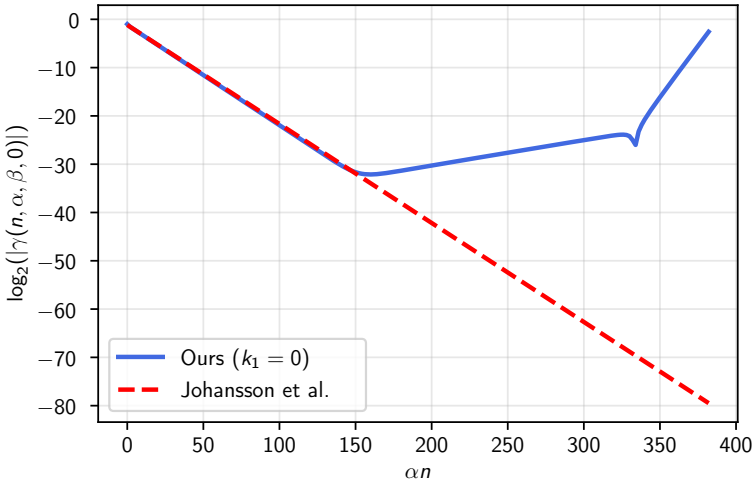


Figure: Evolution in α of $\log_2(|\gamma(384, \alpha, 0.5, 0)|)$

Bias Oscillations

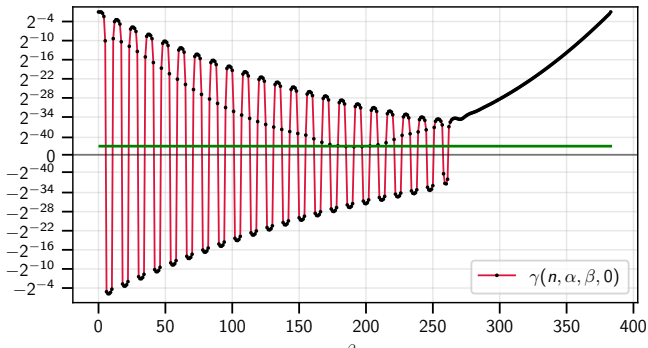


Figure: Evolution in β of $\gamma(384, 0.5, \beta, 0)$. The green line is the average bias over β

Bias Oscillations

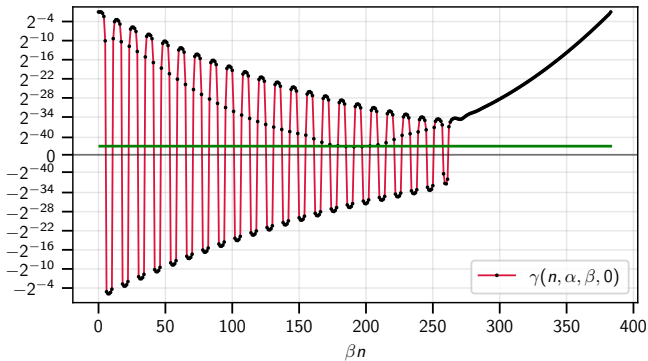


Figure: Evolution in β of $\gamma(384, 0.5, \beta, 0)$. The green line is the average bias over β

Observation for later

Values of β such that $\beta n \equiv 5 \bmod 6$ have way lower biases than others!

Analysis of the attack

- For a fixed α, they compute the bias $\gamma(n, \alpha)$.
- One needs $\mathcal{O}(\frac{1}{\gamma(n,\alpha)^2})$ pairs to distinguish.
- Those pairs can be found with Nearest-Neighbor-Search(NNS) algorithm.
- Search for α^* which gives the best trade-off between NNS complexity and distinguishing complexity.

They obtained $\alpha^* \approx 0.37$.

New Possible Parameters

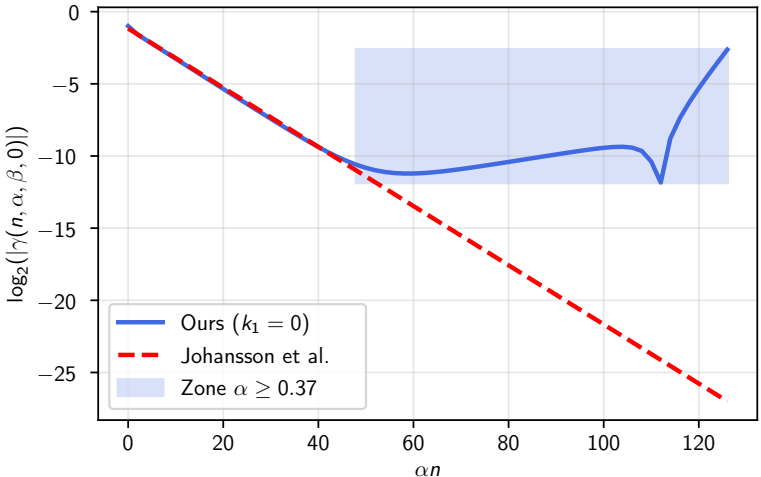
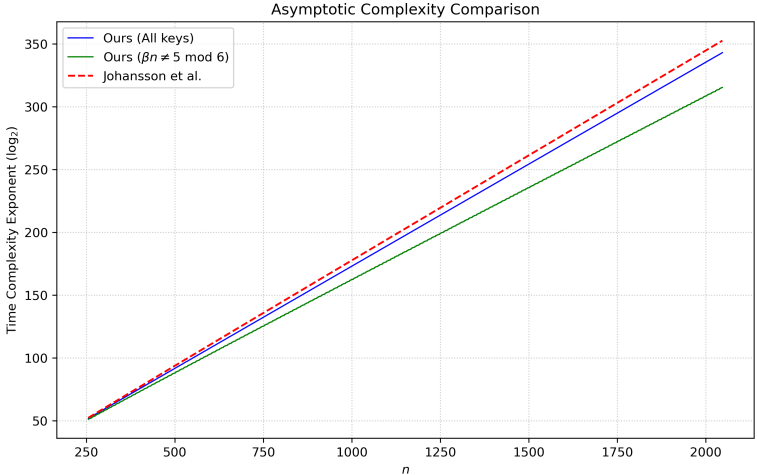


Figure: Evolution in α of $\log_2(|\gamma(384, \alpha, 0.5, 0)|)$

Analysis With Our Bias



$\mathcal{O}(2^{0.166n})$ $\mathcal{O}(2^{0.162n})$ $\mathcal{O}(2^{0.147n})$

Outline

Outline

Conclusion

- We found a much better attack by fixing both key and input weights.
- Combined with the splitting strategy we obtain the best known attack!
- We break the $n = 768$ instance for the first time and provide the first practical attack on $n = 384$.
- We refine the analysis of Johansson et al. and improve their original attack.

Conclusion

- We found a much better attack by fixing both key and input weights.
- Combined with the splitting strategy we obtain the best known attack!
- We break the $n = 768$ instance for the first time and provide the first practical attack on $n = 384$.
- We refine the analysis of Johansson et al. and improve their original attack.

Open directions

- Reanalyze Cheon et al.'s attack with fixed-weight inputs.
- Reduce even more the data complexity by looking at ℓ-tuple?

Conclusion

- We found a much better attack by fixing both key and input weights.
- Combined with the splitting strategy we obtain the best known attack!
- We break the $n = 768$ instance for the first time and provide the first practical attack on $n = 384$.
- We refine the analysis of Johansson et al. and improve their original attack.

Open directions

- Reanalyze Cheon et al.'s attack with fixed-weight inputs.
- Reduce even more the data complexity by looking at ℓ-tuple?

  We tried but failed :(

Conclusion

- We found a much better attack by fixing both key and input weights.
- Combined with the splitting strategy we obtain the best known attack!
- We break the $n = 768$ instance for the first time and provide the first practical attack on $n = 384$.
- We refine the analysis of Johansson et al. and improve their original attack.

Open directions

- Reanalyze Cheon et al.'s attack with fixed-weight inputs.
- Reduce even more the data complexity by looking at ℓ-tuple?
  We tried but failed :(

Thanks for your attention!


Jung Hee Cheon, Wonhee Cho, Jeong Han Kim, and Jiseung Kim.
Adventures in crypto dark matter: Attacks and fixes for weak pseudorandom functions.
In Juan Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 739–760. Springer, Cham, May 2021.


Kai Hu, Gregor Leander, Håvard Raddum, Arne Sandrib, and Aleksei Udovenko.
Cryptanalysis of two alternating moduli weak prfs.
IACR Trans. Symmetric Cryptol., 2026(1):95–118, 2026.


Thomas Johansson, Willi Meier, and Vu Nguyen.
Differential cryptanalysis of mod-2/mod-3 constructions of binary weak prfs.
In *ISIT 2023*, pages 477–482. IEEE, 2023.