



**HAL**  
open science

# Generalized Feistel Networks with Optimal Diffusion

Léo Perrin

► **To cite this version:**

Léo Perrin. Generalized Feistel Networks with Optimal Diffusion. Dagstuhl Seminar 18021 Symmetric Cryptography, Jan 2018, Dagstuhl, Germany. 10.4230/DAGREP.8.1.1 . hal-01953351

**HAL Id: hal-01953351**

**<https://inria.hal.science/hal-01953351>**

Submitted on 12 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Generalized Feistel Networks with Optimal Diffusion

Léo Perrin

DTU, Lyngby  
Inria, Paris

Dagstuhl 2018 (seminar-18021)



Technical University of Denmark



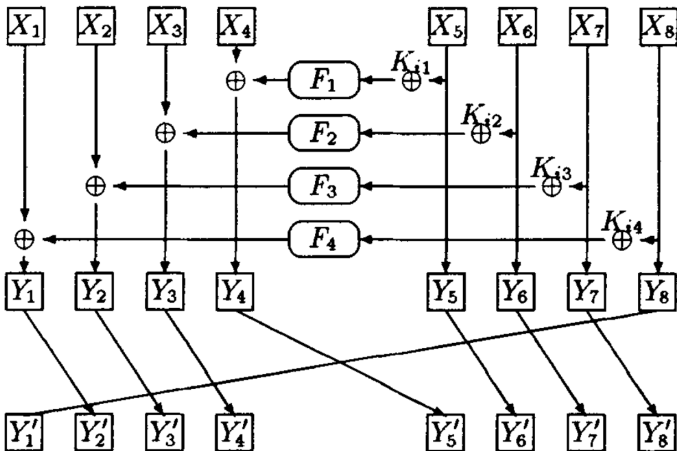
# In this talk

- A new type of generalized Feistel Networks
- Linear layer design
- Wide block cipher/sponge permutation blueprint
- Fibonacci numbers!

# Outline

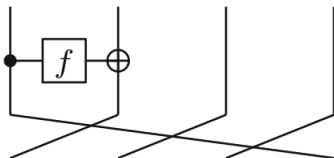
- 1 Introduction
- 2 Observations on GFNs**
- 3 Multi-Rotating Feistel Network (MRFN)
- 4 Possible Applications
- 5 Conclusion

# First GFN

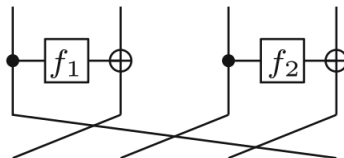


Source: *Generalized Feistel networks*, K. Nyberg (1996)

# Basic GFN



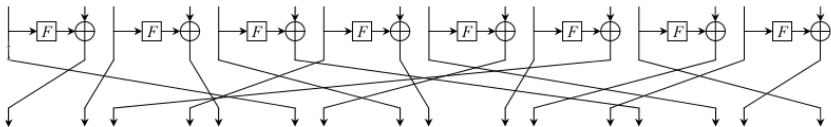
Type-I GFN



Type-II GFN

Source: *Generalized Feistel networks revisited*, A. Bogdanov, K. Shibutani  
(2013)

# Improved GFN



Source: *TWINE: A Lightweight, Versatile Block Cipher*, T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi

# Diffusion in Generalized Feistel networks

How long does it take for each input word to influence each output word?

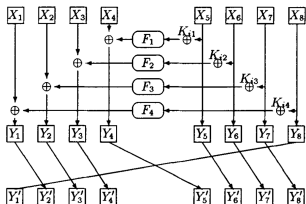
*The state consists of  $2b$  branches.*



# Diffusion in Generalized Feistel networks

How long does it take for each input word to influence each output word?

*The state consists of  $2b$  branches.*



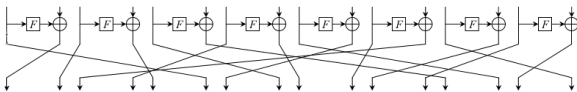
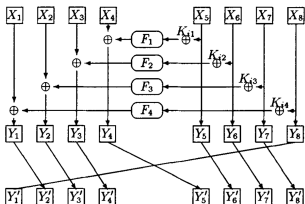
Nyberg/Type-II GFN:

$\approx 2b$  rounds

# Diffusion in Generalized Feistel networks

How long does it take for each input word to influence each output word?

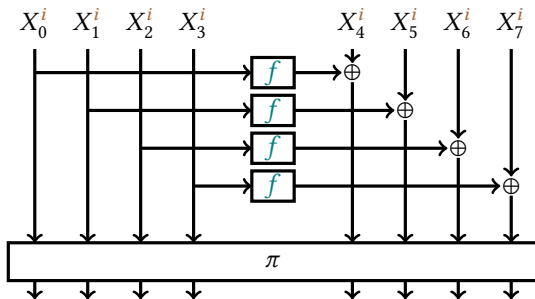
*The state consists of  $2b$  branches.*



Nyberg/Type-II GFN:  
 $\approx 2b$  rounds

TWINE-like GFN:  $\approx 2 \log_2(b)$  rounds

# General Vue



## Optimal Diffusion

The best we can achieve is for  $X_0^0$  to influence  $\phi_{i+2}$  branches at round  $i$ , where

$$\phi_0 = 0, \phi_1 = 1, \phi_{i+2} = \phi_{i+1} + \phi_i.$$

# Diffusion in GFNs

<i>b</i>	8	16	32	64	128	..	2048
Nyberg Type-II/Nyberg	16	32	64	128	256		4096
TWINE-like	6	8	10	12	14		22
Optimal	6	8	9	11	12		18

Number of rounds for full diffusion.

- Can we reach the Fibonacci-based bound?
- Can we have an easy to implement  $\pi$ ?

- Can we reach the Fibonacci-based bound?
- Can we have an easy to implement  $\pi$ ?

**Yes (for both)**

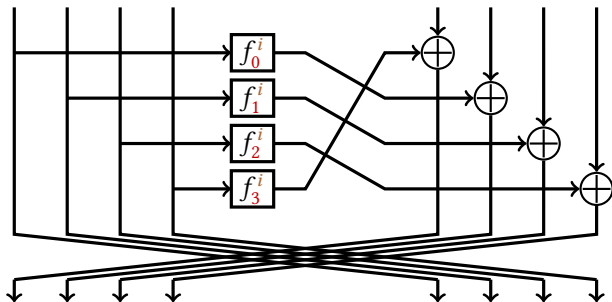
# Outline

- 1 Introduction
- 2 Observations on GFNs
- 3 Multi-Rotating Feistel Network (MRFN)**
- 4 Possible Applications
- 5 Conclusion

# General Structure

- Number of branches:  $2b$
- Number of rounds:  $r$
- $w$ -bit permutations  $f_j^i$  ( $i < r, j < b$ )
- Sequence  $s^i$  of rotations of  $b$  words.

The round  $i$  of a MRFN with  $b = 4$  and  $s^i = 1$  is:





# Some Observations

- Both a Feistel network and a GFN
- $\pi$  is very simple (1 word-wise rotation per round)
- Round function depends on the round index.
- Interesting case:  $s^i = \phi_i$ .

# Some Observations

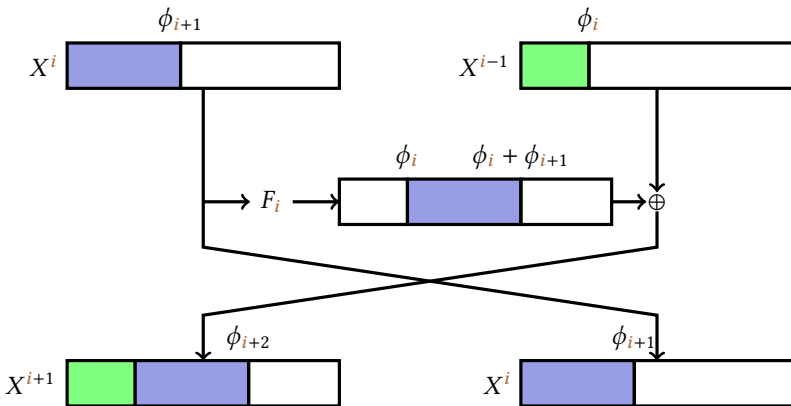
- Both a Feistel network and a GFN
- $\pi$  is very simple (1 word-wise rotation per round)
- Round function depends on the round index.
- Interesting case:  $s^i = \phi_i$ .

## Fibonacci Case

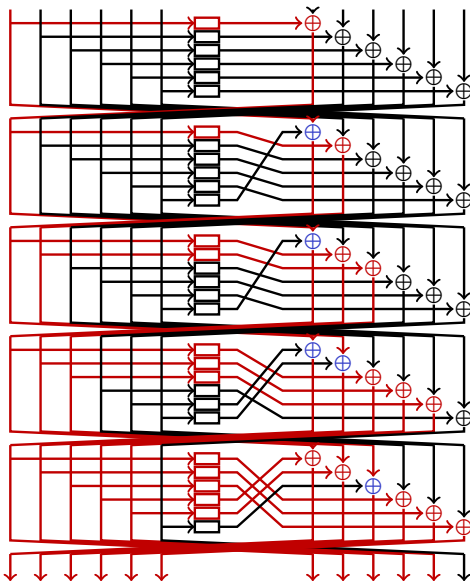
A MRFN with  $s^i = \phi_i$  has optimal diffusion.

# Fibonacci Case

At round 0,  $X_0^0$  has touched the first  $\phi_1 = 1$  branches of one side.



## Example with 12 branches



$$\phi_0 = 0$$

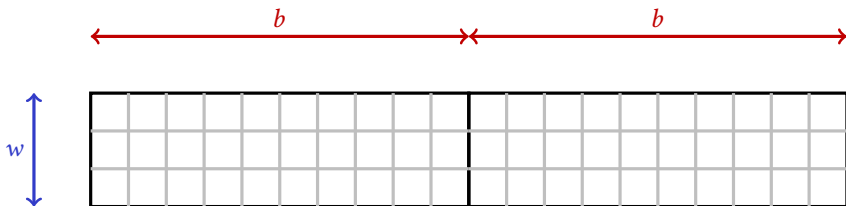
$$\phi_1 = 1$$

$$\phi_2 = 1$$

$$\phi_3 = 2$$

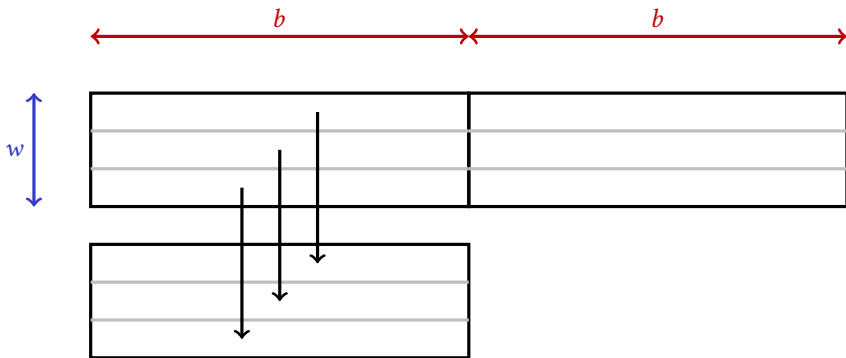
$$\phi_4 = 3$$

# Implementation



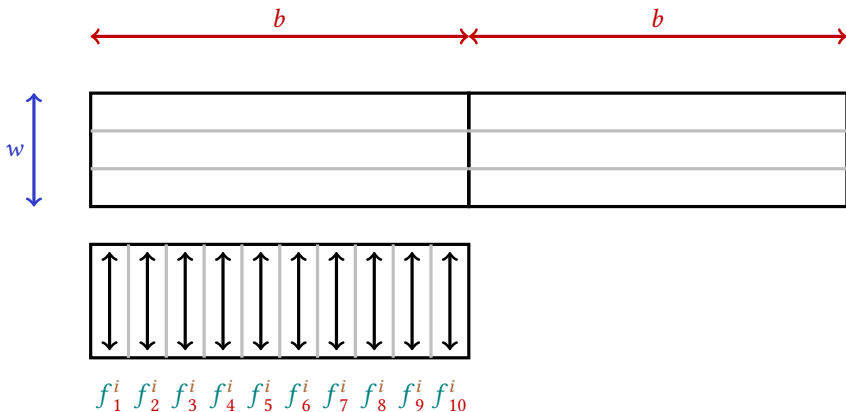
VRound function operating on  $2bw$  bit internal state.

# Implementation



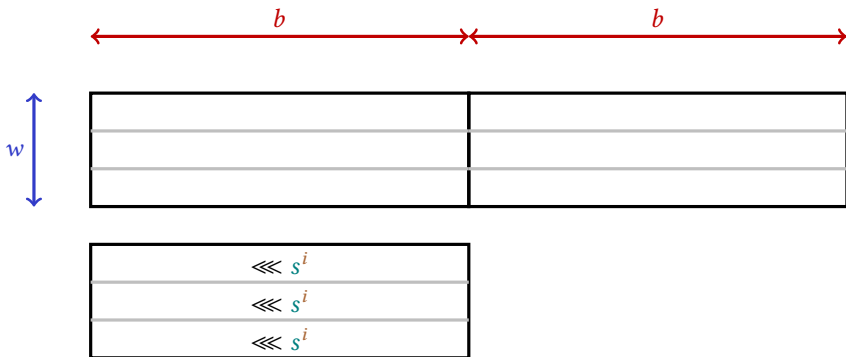
1. copy

# Implementation



2. parallel layer of  $f^i$

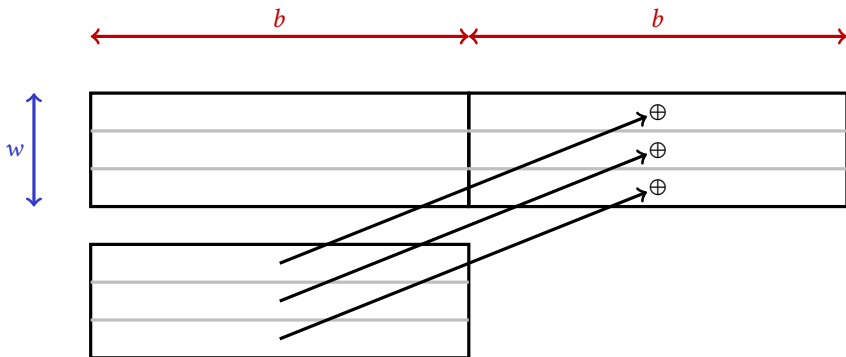
# Implementation



3. rotations

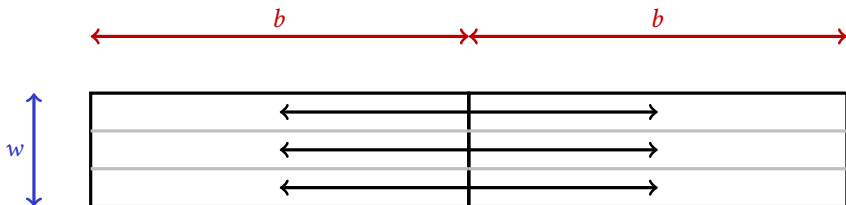


# Implementation



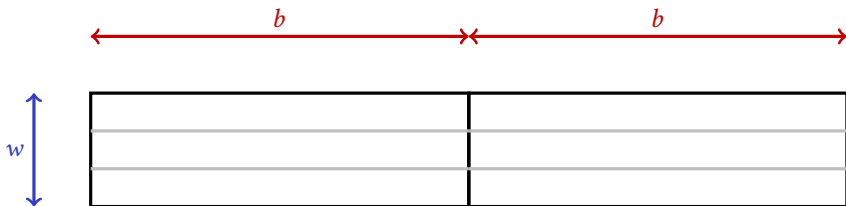
## 4. XOR

# Implementation



5. swap

# Implementation



6. finished!

# Some Observations

- $s^i$  and  $s^i + (-\ell)^i \pmod{b}$  are equivalent
- if  $\gcd(s^i, b) \neq 1$  for all  $i$ , **no full diffusion!**
- Importance of the choice of  $\{s^i\}_{i \geq 0}$

# Security

- If  $s^i = \phi_i$ , then full diffusion in  $\approx \Lambda(n)$  rounds, where  $\Lambda(x) = i$  if  $\phi_{i-1} < x \leq \phi_i$  (optimal).
- If  $s^{2^i} = 0$  and  $i_{2^{i+1}} = 2^i$ , then full diffusion in  $\approx 2 \log_2(n)$  rounds (like TWINE).
- Both are quickly safe from miss-in-the-middle based impossible differential attacks and MitM!

# Security

- If  $s^i = \phi_i$ , then full diffusion in  $\approx \Lambda(n)$  rounds, where  $\Lambda(x) = i$  if  $\phi_{i-1} < x \leq \phi_i$  (optimal).
- If  $s^{2^i} = 0$  and  $i_{2^{i+1}} = 2^i$ , then full diffusion in  $\approx 2 \log_2(n)$  rounds (like TWINE).
- Both are quickly safe from miss-in-the-middle based impossible differential attacks and MitM!
- When  $s^i = \phi_i$ , bad truncated differential with 2 active S-Boxes/round.

# Security

- If  $s^i = \phi_i$ , then full diffusion in  $\approx \Lambda(n)$  rounds, where  $\Lambda(x) = i$  if  $\phi_{i-1} < x \leq \phi_i$  (optimal).
- If  $s^{2^i} = 0$  and  $i_{2^{i+1}} = 2^i$ , then full diffusion in  $\approx 2 \log_2(n)$  rounds (like TWINE).
- Both are quickly safe from miss-in-the-middle based impossible differential attacks and MitM!
- When  $s^i = \phi_i$ , bad truncated differential with 2 active S-Boxes/round.

## Open Problem 1

Differential/Linear bound?

## Open Problem 2

Choice of  $\{s^i\}_{i \geq 0}$ ?

# Outline

- 1 Introduction
- 2 Observations on GFNs
- 3 Multi-Rotating Feistel Network (MRFN)
- 4 Possible Applications**
- 5 Conclusion



# GFN-based Linear Layers

- Use linear  $\{f^i\}_{i \geq 0}; s^i = \phi_i$
- $n$ -bit block divided into  $2b$  branches of  $w$  bits uses:

$$\underbrace{\frac{w^2}{2}}_{f_j^i} \times b \times \underbrace{2 \log_2(b)}_r \text{ XORs .}$$

$f$ layer

# GFN-based Linear Layers

- Use linear  $\{f^i\}_{i \geq 0}; s^i = \phi_i$
- $n$ -bit block divided into  $2b$  branches of  $w$  bits uses:

$$\underbrace{\frac{w^2}{2}}_{f_j^i} \times b \times \underbrace{2 \log_2(b)}_r \text{ XORs .}$$

$f_{\text{layer}}$

- If we fix  $w$  to a small value, then the number of XORs scales with  $n \log_2(n)$  rather than  $n^2$ .

# GFN-based Linear Layers

- Use linear  $\{f^i\}_{i \geq 0}; s^i = \phi_i$
- $n$ -bit block divided into  $2b$  branches of  $w$  bits uses:

$$\underbrace{\frac{w^2}{2}}_{f_j^i} \times b \times \underbrace{2 \log_2(b)}_r \text{ XORs .}$$

$f$ layer

- If we fix  $w$  to a small value, then the number of XORs scales with  $n \log_2(n)$  rather than  $n^2$ .
- Practical gains even for  $n = 256$ :

*Improvements to the Linear Layer of LowMC: A Faster Picnic*, with Angela Promitzer, Sebastian Ramacher and Christian Rechberger (2017/448)

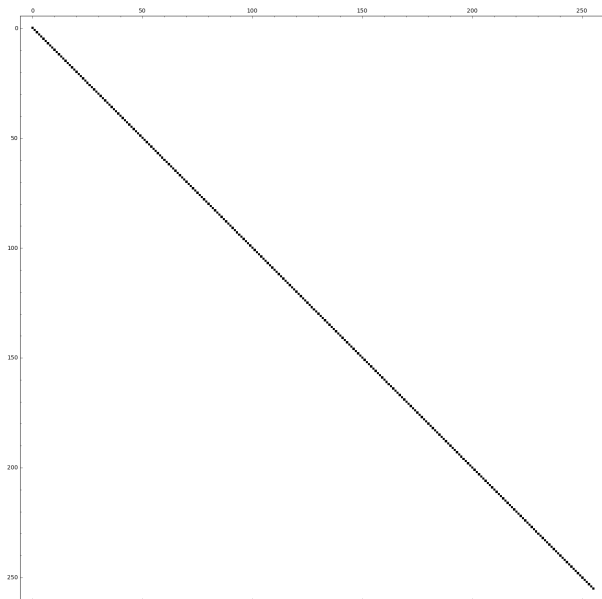
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 0$





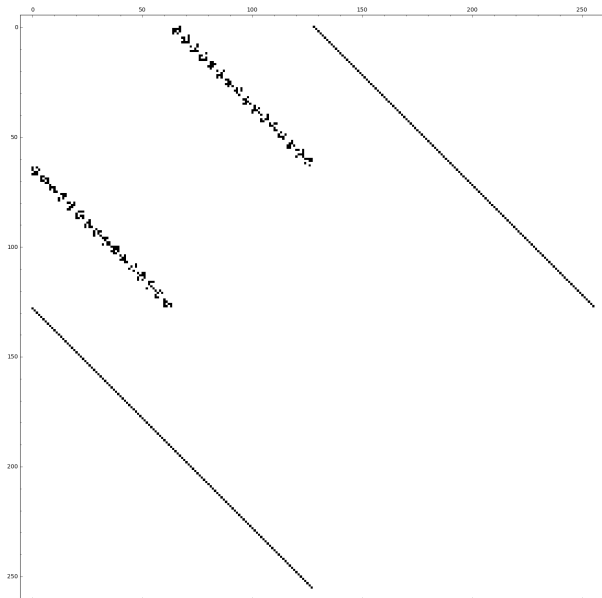
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 1$



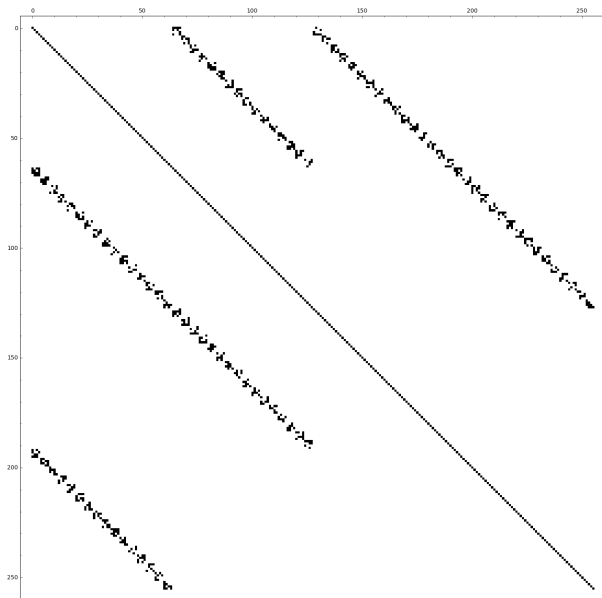
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 2$



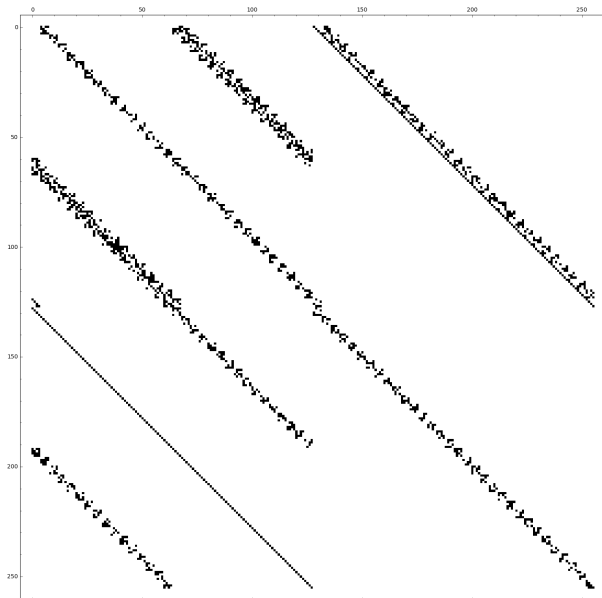
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 3$



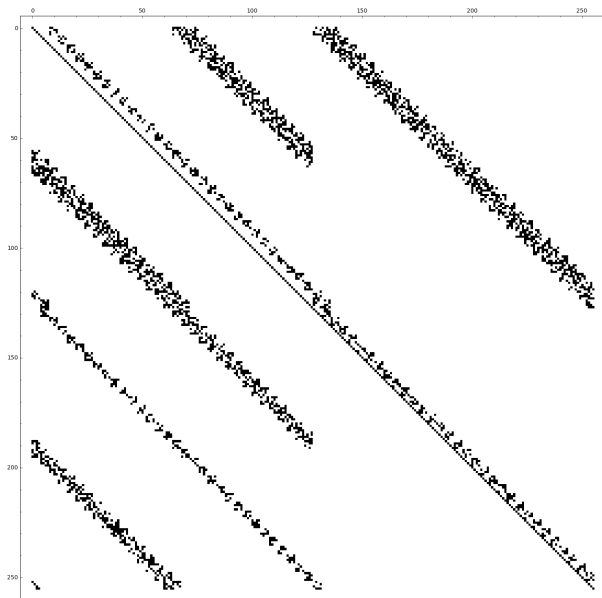
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 4$





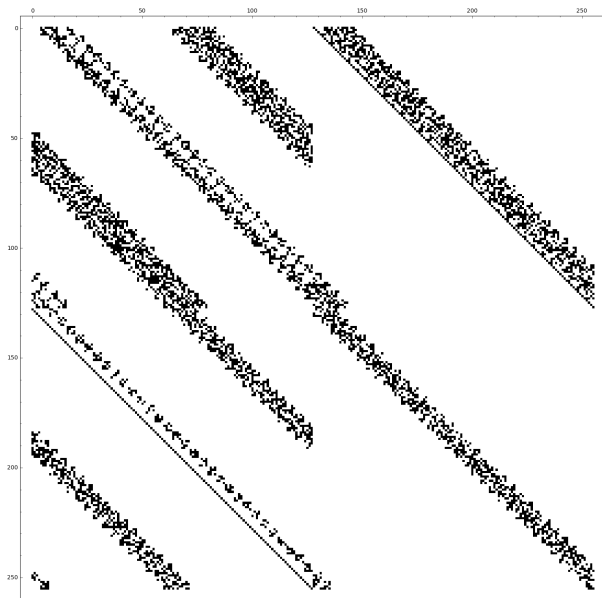
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 5$



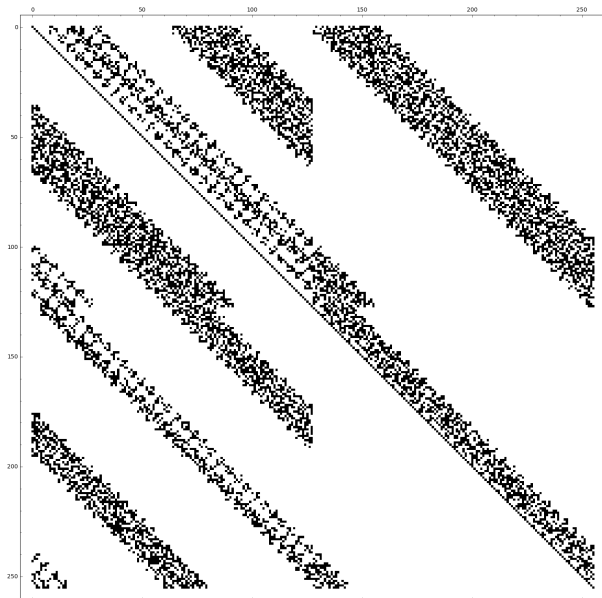
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 6$



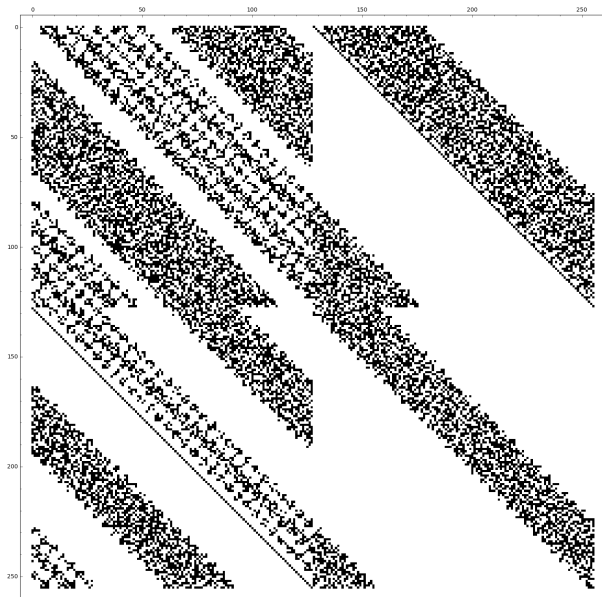
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 7$



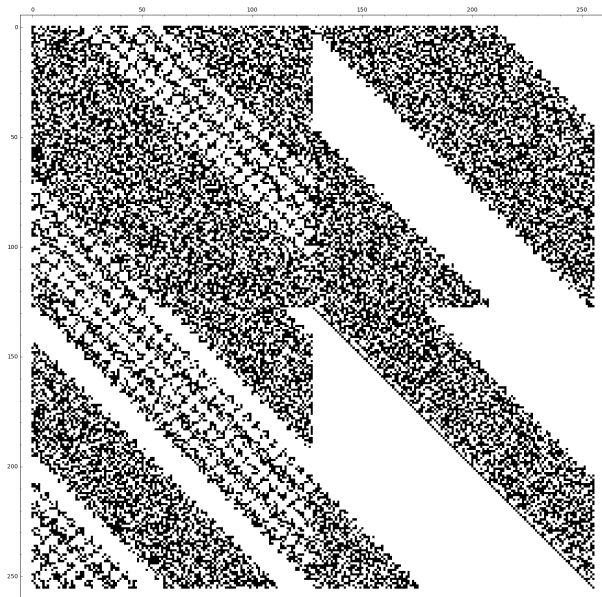
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 8$



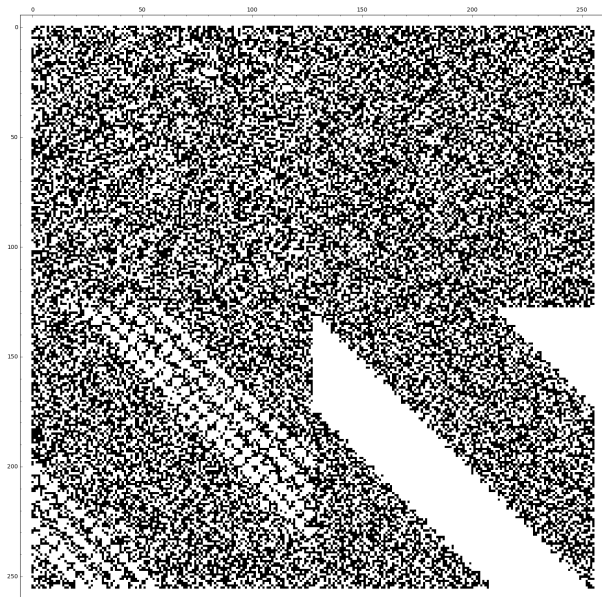
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 9$



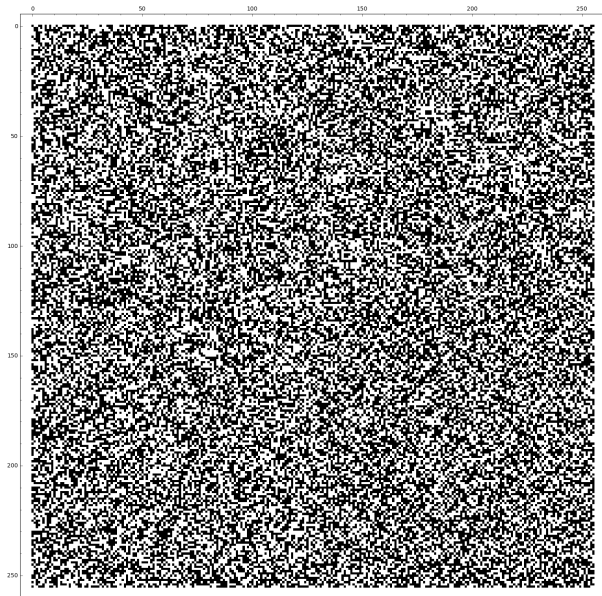
# Example of Linear Layer

■  $n = 256$

■  $w = 4$

■  $b = 32$

$i = 10$



# Sponge function?

■  $n = 384$ , with  $b = 64$  and  $w = 3$

■  $f_j^i(x) = \chi_3(x \oplus c_j^i)$

■  $s^{2i} = 0, s^{2i+1} = 2^i$  for  $0 \leq i < 2 \log_2(b) = 12$ , then repeat (4? times):

$$s = \{0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 32\}$$

# Sponge function?

■  $n = 384$ , with  $b = 64$  and  $w = 3$

■  $f_j^i(x) = \chi_3(x \oplus c_j^i)$

■  $s^{2i} = 0, s^{2i+1} = 2^i$  for  $0 \leq i < 2 \log_2(b) = 12$ , then repeat (4? times):

$$s = \{0, 1, 0, 2, 0, 4, 0, 8, 0, 16, 0, 32\}$$

## Efficiency estimates

On 64-bit processors, for each round:

- 3 word copies
- 3 word-wise AND
- 3+3+3 word-wise XORs
- **Maybe** safe for **48 rounds** if  $\geq 8$  active  $f$  functions/round on average.



# Other?

- MiMC-like construction where  $f_j^i(x) = (x + c_j^i)^3$  (what Arnab just presented).

# Other?

- MiMC-like construction where  $f_j^i(x) = (x + c_j^i)^3$  (what Arnab just presented).
- You tell me!

# Outline

- 1 Introduction
- 2 Observations on GFNs
- 3 Multi-Rotating Feistel Network (MRFN)
- 4 Possible Applications
- 5 Conclusion**

# Conclusion

**Fun stuff happens when we allow the use of different permutations in each round!**

## Open problems

- 1 What are good sequences of rotations?
- 2 How to bound number of active  $f$  functions?
- 3 What can we use it for?
- 4 What happens in other structures (SPN? ARX?) when the linear layers are round-dependent?

# Conclusion

**Fun stuff happens when we allow the use of different permutations in each round!**

## Open problems

- 1 What are good sequences of rotations?
- 2 How to bound number of active  $f$  functions?
- 3 What can we use it for?
- 4 What happens in other structures (SPN? ARX?) when the linear layers are round-dependent?

**Thank you!**