IronMaskArithmetic

Victor Normand

CryptoExperts

February 2025

IronMask

Automatic tool to :

- Check security properties in the probing model (t-NI, t-SNI).
- Check security properties in the random probing model (RP, RPC).
- Do comprehensive verification for the random probing model.

Only available for fields $\mathbb{F}_{2^k}$ of characteristic 2.

Reminder - Random Probing Model

- Security in the (p, ε)-random probing model: given p, the probability to recover information on the secret is bounded by ε.

$$\varepsilon = \sum_{i=1}^{s} c_i p^i (1-p)^{s-i}$$

- c_i is the number of failure tuples of size i.
- Comprehensive verification : Find the exact number of failure tuples of size i for every $i \in \mathbb{N}$.

CryptoExperts

IronMask

Automatic tool to :

- Check security properties in the probing model (t-NI, t-SNI).
- Check security properties in the random probing model (RP, RPC).
- Do comprehensive verification for the random probing model.

Only available for fields $\mathbb{F}_{2^k}$ of characteristic 2.

IronMaskArithmetic

- Same features as IronMask .
- Available on any arithmetic fields $\mathbb{F}_p$ with p prime.

Differences

Some differences between IronMask and IronMaskArithmetic .

- Parsing differences.
- Approach to verification : exhaustive vs constructive.

Parsing gadgets

```
#SHARES 2
#IN a b
#RANDOMS r01 rr01
#OUT e

c0 = a0 + r01
c1 = a1 + r01

d0 = b0 + rr01
d1 = b1 + rr01

e0 = c0 + d0
e1 = c1 + d1
```

(a) Parsing with IronMask

```
#SHARES 2
#IN a b
#RANDOMS r01 rr01
#OUT e
#CAR 3329

c0 = a0 + r01
c1 = a1 + -1 r01

d0 = b0 + rr01
d1 = b1 + -1 rr01

e0 = c0 + d0
e1 = c1 + d1
```

(b) Parsing with IronMaskArithmetic

Exhaustive Approach

- Circuit with ℓ intermediate variables.
- ℓ possible probes.
- To find all the failure tuple of size i :
  - Consider a tuple.
  - Determine if it is a failure tuple.
  - Reitere for all the tuples of size i.
- $\binom{\ell}{i}$ tuples to evaluate.

CRYPTO EXPERTS

Constructive Approach - Incompressible tuple

- q_1, q_2, q_3 probes of some intermediate variables of the circuit.
- (q_1, q_2, q_3) is a failure tuple.
- All sub tuples of (q_1, q_2, q_3) are not failure tuples.

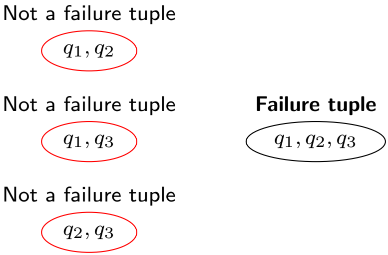
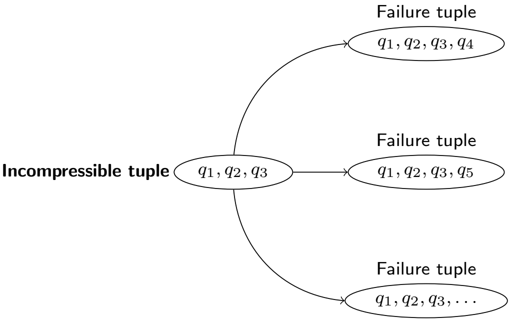


Figure: Illustration of an incompressible tuple

Constructive Approach - Incompressible tuple



Constructive Approach

- $I =$ set of incompressible tuples.
- $s = \min_{\mathcal{T} \in I}(|\mathcal{T}|)$
- $I_j =$ set of incompressible tuples of size j.
- $\mathcal{F}_j =$ set of failures tuples of size j.
- **Expand**(I_j) : Return the tuples of size $j+1$ from the tuples of I_j.

$$\mathcal{F}_s = I_s \xrightarrow{\textsf{Expand}} \mathcal{F} \rightarrow \mathcal{F}_{s+1} = \mathcal{F} \cup I_{s+1} \backslash \mathcal{F}_s$$

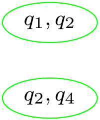
$$\mathcal{F}_{s+1} \xrightarrow{\textsf{Expand}} \mathcal{F} \longrightarrow \mathcal{F}_{s+2} = \mathcal{F} \cup I_{s+2} \backslash \mathcal{F}_{s+1}$$

$\ldots$

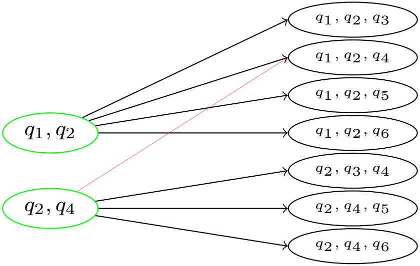
Constructive approach - Example

Example:

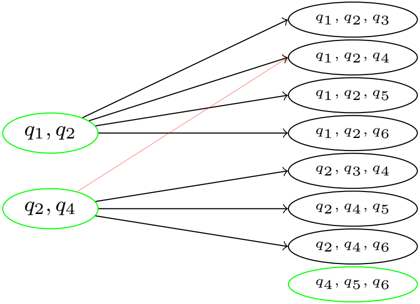
- 6 intermediate variables.
- 2 incompressible failure tuples of size 2: (q_1, q_2) and (q_2, q_4).
- 1 incompressible failure tuple of size 3: (q_4, q_5, q_6).
- We want to compute the number of failure tuples of size 3 : c_3.



Constructive approach - Example



Constructive approach - Example



$c_3 = 8$

Gaussian Elimination Technique

- How to identify if a tuple is a failure tuple ?
- Use the Gaussian Elimination Technique
- Two types of gadgets : Linear gadgets and Non-linear gadgets

Linear Gadget

Gadget with probes q of the form:

$$q = f_q(x_1, \dots, x_n) + \vec{s} \cdot \vec{r}$$

with f_q an arithmetic function over $\mathbb{F}_p$, $\vec{r} \in \mathbb{F}_p^q$ vector of randoms used in the gadget and $\vec{s} \in \mathbb{F}_p^q$ the coefficient values of the randoms.

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \dots, q_\ell)$ the tuple of probes.

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \ldots, q_\ell)$ the tuple of probes.

$$\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix} = M. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S.\vec{r}$$

$M \in \mathbb{M}_{\ell \times n}(\mathbb{F}_p), S \in \mathbb{M}_{\ell \times s}(\mathbb{F}_p)$

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \ldots, q_\ell)$ the tuple of probes.

$$\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix} = M. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S.\vec{r}$$

$S' \xleftarrow{\text{Gauss Pivot}} S$, $S' = \delta.S$

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \ldots, q_\ell)$ the tuple of probes.

$$\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix} = M. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S.\vec{r}$$

$S' \xleftarrow{\text{Gauss Pivot}} S$, $S' = \delta.S$

$M' \xleftarrow{\delta} M$, $M' = \delta.M$

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \ldots, q_\ell)$ the tuple of probes.

$$\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix} = M. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S.\vec{r}$$

$S' \xleftarrow{\text{Gauss Pivot}} S$, $S' = \delta.S$
$M' \xleftarrow{\delta} M$, $M' = \delta.M$

$$\begin{pmatrix} q'_1 \\ \ldots \\ q'_\ell \end{pmatrix} = M'. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S'.\vec{r}$$

CRYPTO EXPERTS

Gaussian step - Linear Gadget

Suppose we have ℓ probes, $\mathcal{Q} = (q_1, \ldots, q_\ell)$ the tuple of probes.

$$\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix} = M. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S.\vec{r}$$

$S' \xleftarrow{\text{Gauss Pivot}} S$, $S' = \delta.S$
$M' \xleftarrow{\delta} M$, $M' = \delta.M$

$$\begin{pmatrix} q'_1 \\ \ldots \\ q'_\ell \end{pmatrix} = M'. \begin{pmatrix} x_1 \\ \ldots \\ x_n \end{pmatrix} + S'.\vec{r}$$

$\begin{pmatrix} q'_1 \\ \ldots \\ q'_\ell \end{pmatrix}$ is a failure tuple iff $\begin{pmatrix} q_1 \\ \ldots \\ q_\ell \end{pmatrix}$ is a failure tuple.

Gaussian step - Linear Gadget

$$\begin{pmatrix} q'_1 \\ \dots \\ q'_\ell \end{pmatrix} = M'.\begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} + S'.\vec{r}$$

Either the i^{th}-line of S' contains a pivot $\implies$ q'_i masked by a **pivot** random, no input shares are necessary to simulate q'_i.

Or the i^{th}-line of S' contains only zeros coefficients $\implies$ q'_i is unmasked, all input shares in q'_i are necessary to simulate q'_i.

Linear Gadget - Example

Let G a 3-shares gadget with 2 secret inputs $\mathbf{x}$ and $\mathbf{y}$.

$$\mathcal{Q} = \left(\begin{array}{l} q_1 = x_1 - r_1 \\ q_2 = x_2 + y_2 \\ q_3 = y_2 + r_2 \\ q_4 = x_3 + y_3 + r_1 - r_2 \end{array} \right)$$

Linear Gadget - Example

$$\mathcal{Q} = \begin{pmatrix} q_1 = x_1 - r_1 \\ q_2 = x_2 + y_2 \\ q_3 = y_2 + r_2 \\ q_4 = x_3 + y_3 + r_1 - r_2 \end{pmatrix}$$

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} + \underbrace{\begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & -1 \end{pmatrix}}_{\mathbf{S}} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

Linear Gadget - Example

$$\mathbf{S} = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & -1 \end{pmatrix} \longrightarrow \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \longrightarrow \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} = S'$$

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = M'$$

Linear Gadget - Example

$$S' = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \; M' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Linear Gadget - Example

$$S' = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \; M' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$Q' := \begin{pmatrix} q'_1 \\ q'_2 \\ q'_3 \\ q'_4 \end{pmatrix} = M' \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} + S'. \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

Linear Gadget - Example

$$S' = \begin{pmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \; M' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$\mathcal{Q}' := \begin{pmatrix} q'_1 \\ q'_2 \\ q'_3 \\ q'_4 \end{pmatrix} = M' \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} + S'. \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

$$\mathcal{Q}' = \begin{pmatrix} q'_1 = x_1 - r_1 \\ q'_2 = x_2 + y_2 \\ q'_3 = y_2 + r_2 \\ q'_4 = x_1 + x_3 + y_2 + y_3 \end{pmatrix}$$

x_1, x_2, x_3, y_2, y_3 are necessary to simulate $\mathcal{Q}'$ and $\mathcal{Q}$.

Non Linear Gadget

For a n-shares 2-to-m gadget. Gadget with probes q of the form :

$$q = R_3(F(R_1(\vec{x}, \vec{r_1}), R_2(\vec{y}, \vec{r_2})), \vec{r_3})$$

where F is any arithmetic circuit, R_i are linear arithmetic circuits and $\vec{r_i}$ are randoms.

- $\vec{r_3}$ is the **output random**.
- $\vec{r_1}, \vec{r_2}$ are the **input random**.

Non Linear Gadget - Steps

- Gaussian Step on output randoms.
- Factorization.
- Gaussian step on each factorized tuples.

Non Linear Gadget - Example

Let G a 3-share gadget with 2 secret inputs $\mathbf{x}$ and $\mathbf{y}$.

$$\mathcal{Q} = \left(\begin{array}{l} q_1 = x_1.y_1 + r_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{array} \right)$$

Non Linear Gadget - Example - First Step

$$\mathcal{Q} = \begin{pmatrix} q_1 = x_1.y_1 + r_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{pmatrix}$$

Gauss Steps on output randoms :

- q_2 does not have output randoms.
- Removes q_2 from $\mathcal{Q}$.

$$\mathcal{Q}' = \begin{pmatrix} q_1 = x_1.y_1 + r_1 \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{pmatrix}$$

Non Linear Gadget - Example - First Step

$$\mathcal{Q} = \left(\begin{array}{l} q_1 = x_1.y_1 + r_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{array} \right)$$

Gauss Steps on output randoms :

- q_2 does not have output randoms.
- Removes q_2 from $\mathcal{Q}$.

$$\mathcal{Q}' = \left(\begin{array}{l} q_1 = x_1.y_1 + r_1 \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{array} \right)$$

$$\mathcal{Q}_1 = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{M} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ x_1.y_1 \\ (x_1 + r_5).(y_3 + r_6) \\ (x_2 - r_3).y_1 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{pmatrix}}_{S} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

Non Linear Gadget - Example - First Step

$$\mathcal{Q} = \begin{pmatrix} q_1 = x_1.y_1 + r_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{pmatrix}$$

<u>Gauss Steps on output randoms :</u>

$$\mathcal{Q}_1 = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{M} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \\ x_1.y_1 \\ (x_1 + r_5).(y_3 + r_6) \\ (x_2 - r_3).y_1 \end{pmatrix} + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{pmatrix}}_{S} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$$

- Gaussian Step as in Linear Gadget.
- Obtains a new tuple of probes $\mathcal{Q}_2$
- Add q_2 to $\mathcal{Q}_2$.

$$\mathcal{Q}_2 = \begin{pmatrix} q'_1 = x_1.y_1 + r_1 \\ q'_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q'_4 = (x_2 - r_3).y_1 + (x_1 + r_5).(y_3 + r_6) + x_1.y_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \end{pmatrix}$$

Non Linear Gadget - Example - First Step

$$\mathcal{Q} = \left(\begin{array}{l} q_1 = x_1.y_1 + r_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \\ q_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q_4 = (x_2 - r_3).y_1 - r_1 - r_2 \end{array} \right)$$

$$\mathcal{Q}_2 = \left(\begin{array}{l} q'_1 = x_1.y_1 + r_1 \\ q'_3 = (x_1 + r_5).(y_3 + r_6) + r_2 \\ q'_4 = (x_2 - r_3).y_1 + (x_1 + r_5).(y_3 + r_6) + x_1.y_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \end{array} \right)$$

- q'_1 and q'_3 contains output randoms present nowhere else.
- Remove q'_1 and q'_3 from $\mathcal{Q}_2$.

$$\mathcal{Q}_3 = \left(\begin{array}{l} q'_4 = (x_2 - r_3).y_1 + (x_1 + r_5).(y_3 + r_6) + x_1.y_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \end{array} \right)$$

$\mathcal{Q}$ is a failure tuple iff $\mathcal{Q}_3$ is a failure tuple.

Non Linear Gadget - Example - Second Step

$$\mathcal{Q}_3 = \left(\begin{array}{l} q'_4 = (x_2 - r_3).y_1 + (x_1 + r_5).(y_3 + r_6) + x_1.y_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \end{array} \right)$$

Factorization :

- $\mathcal{Q}_x$ contains the factors containing the secret x.
- $\mathcal{Q}_y$ contains the factors containing the secret y.

$$\mathcal{Q}_x = \begin{pmatrix} q_{1,x} = (x_2 - r_3) \\ q_{2,x} = (x_1 + r_5) \\ q_{3,x} = x_1 \\ q_{4,x} = (x_3 + r_3) \end{pmatrix} \qquad \mathcal{Q}_y = \begin{pmatrix} q_{1,y} = y_1 \\ q_{2,y} = y_3 + r_6 \\ q_{3,y} = y_1 \\ q_{4,y} = y_2 + r_4 \end{pmatrix}$$

Non Linear Gadget - Example - Second Step

$$\mathcal{Q}_3 = \begin{pmatrix} q'_4 = (x_2 - r_3).y_1 + (x_1 + r_5).(y_3 + r_6) + x_1.y_1 \\ q_2 = (x_3 + r_3).(y_2 + r_4) \end{pmatrix}$$

Factorization :

- $\mathcal{Q}_x$ contains the factors containing the secret x.
- $\mathcal{Q}_y$ contains the factors containing the secret y.

$$\mathcal{Q}_x = \begin{pmatrix} q_{1,x} = (x_2 - r_3) \\ q_{2,x} = (x_1 + r_5) \\ q_{3,x} = x_1 \\ q_{4,x} = (x_3 + r_3) \end{pmatrix} \qquad \mathcal{Q}_y = \begin{pmatrix} q_{1,y} = y_1 \\ q_{2,y} = y_3 + r_6 \\ q_{3,y} = y_1 \\ q_{4,y} = y_2 + r_4 \end{pmatrix}$$

- The necessary input shares of x to perfectly simulate $\mathcal{Q}_3$ are the necessary input shares to perfectly simulate $\mathcal{Q}_x$.
- The necessary input shares of y to perfectly simulate $\mathcal{Q}_3$ are the necessary input shares to perfectly simulate $\mathcal{Q}_y$.

Non Linear Gadget - Example - Third Step

$$\mathcal{Q}_x = \begin{pmatrix} q_{1,x} = (x_2 - r_3) \\ q_{2,x} = (x_1 + r_5) \\ q_{3,x} = x_1 \\ q_{4,x} = (x_3 + r_3) \end{pmatrix} \qquad \mathcal{Q}_y = \begin{pmatrix} q_{1,y} = y_1 \\ q_{2,y} = y_3 + r_6 \\ q_{3,y} = y_1 \\ q_{4,y} = y_2 + r_4 \end{pmatrix}$$

- Gaussian Step on $\mathcal{Q}_x$ and $\mathcal{Q}_y$ as for linear gadget.

$$\mathcal{Q}'_x = \begin{pmatrix} q_{1,x} = (x_2 - r_3) \\ q_{2,x} = (x_1 + r_5) \\ q_{3,x} = x_1 \\ q_{4,x} = x_3 + x_2 \end{pmatrix} \qquad \mathcal{Q}'_y = \begin{pmatrix} q_{1,y} = y_1 \\ q_{2,y} = y_3 + r_6 \\ q_{3,y} = y_1 \\ q_{4,y} = y_2 + r_4 \end{pmatrix}$$

We need x_1, x_2, x_3, y_1 to perferctly simulate $\mathcal{Q}$.

Conclusion

- IronMaskArithmetic : Extension of IronMask on any arithmetic field $\mathbb{F}_p$.
- Different approaches to build all the failure tuples.
- Same approach to determine if a tuple is a failure tuple : the Gaussian Elimination Technique.
- Supports linear and non-linear gadgets.

Timing performance

	IronMask (Car 2)	IronMaskArithmetic (Car 3329)
Add_3_shares RP -c 4	0s	0s
Add_7_shares SNI -t 6	8s	4s
Copy_6_shares RPC -c 5 -t 2	1min 20s	1 min 9s
Mult_6_shares NI -t 5	1s	4min 7s
Mult-ref_5_shares RP -c 4	1s	10s