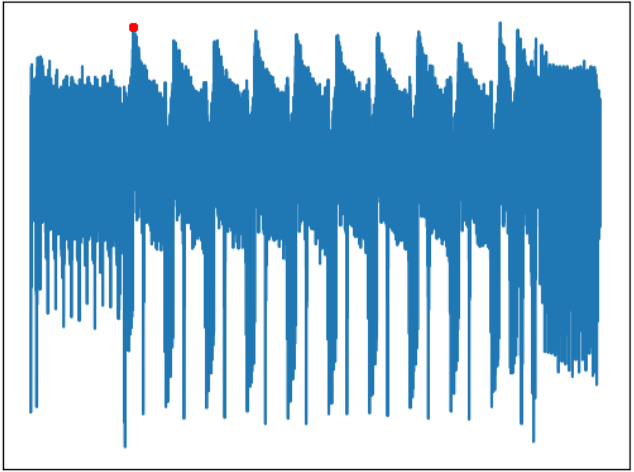
Towards Globally Optimized Masking: From Low Randomness to Low Noise Rate

Gaëtan Cassiers François-Xavier Standaert

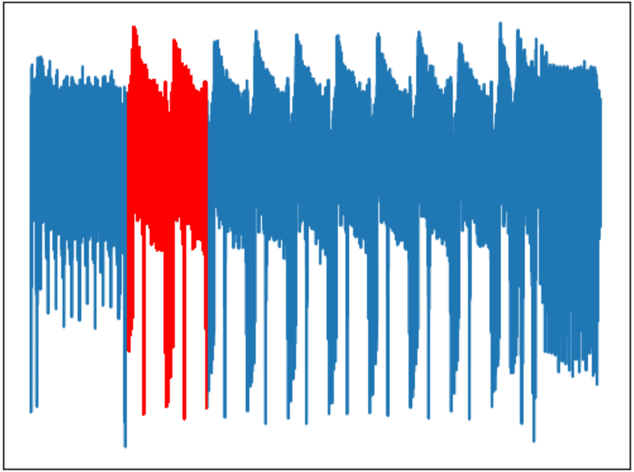
CHES 2019

The threat of horizontal attacks



DPA

The threat of horizontal attacks



Horizontal attack

Just mask it !

Masking a sensitive bit $\boldsymbol{x}$:

$$\boldsymbol{x} = \underbrace{x_0 \oplus \cdots \oplus x_{d-2}}_{\text{random}} \oplus x_{d-1}$$

and compute only on sharing $(x_0, \ldots, x_{d-1})$.

Just mask it !

Masking a sensitive bit x:

$$x = \underbrace{x_0 \oplus \cdots \oplus x_{d-2}}_{\text{random}} \oplus x_{d-1}$$

and compute only on sharing $(x_0, \ldots, x_{d-1})$.

Secure in the probing model at order t.

What about horizontal attacks ?

[BCPZ16]: Qualitative analysis and countermeasure.

Contributions

Horizontal attacks against masking:

- Quantitative (heuristic-based) analysis
- Automated tool

Countermeasures:

- Improved masked multiplication gadget

Outline

Outline

Masked AND gate

([ISW03]-like) AND gadget: $z = x \otimes y$

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} x_0 \otimes y_0 & \oplus & (x_0 \otimes y_1 \oplus r_0) & \oplus & (x_0 \otimes y_2 \oplus r_1) \\ (x_1 \otimes y_0 \oplus r_0) & \oplus & x_1 \otimes y_1 & \oplus & (x_1 \otimes y_2 \oplus r_2) \\ (x_2 \otimes y_0 \oplus r_1) & \oplus & (x_2 \otimes y_1 \oplus r_2) & \oplus & x_2 \otimes y_2 \end{pmatrix}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \xrightarrow[\substack{x_{i,j}=x_i \\ y_{i,j}=y_i}]{\texttt{MatGen}} \begin{bmatrix} (x_{0,0}, y_{0,0}) & (x_{0,1}, y_{1,0}) & (x_{0,2}, y_{2,0}) \\ (x_{1,0}, y_{0,1}) & (x_{1,1}, y_{1,1}) & (x_{1,2}, y_{2,1}) \\ (x_{2,0}, y_{0,2}) & (x_{2,1}, y_{1,2}) & (x_{2,2}, y_{2,2}) \end{bmatrix} \cdots$$

$$\cdots \xrightarrow[\alpha_{i,j}=x_{i,j} \otimes y_{i,j}]{\texttt{Product}} \begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} \end{bmatrix} \xrightarrow[\bigoplus_j \alpha_{i,j} \oplus r_{i,j}]{\texttt{Compression}} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}$$

Other Masked AND gates

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \xrightarrow{\texttt{MatGen}} \begin{bmatrix} (x_{0,0}, y_{0,0}) & (x_{0,1}, y_{1,0}) & (x_{0,2}, y_{2,0}) \\ (x_{1,0}, y_{0,1}) & (x_{1,1}, y_{1,1}) & (x_{1,2}, y_{2,1}) \\ (x_{2,0}, y_{0,2}) & (x_{2,1}, y_{1,2}) & (x_{2,2}, y_{2,2}) \end{bmatrix} \xrightarrow{\texttt{Prod.}} \begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} & \alpha_{0,2} \\ \alpha_{1,0} & \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,0} & \alpha_{2,1} & \alpha_{2,2} \end{bmatrix} \xrightarrow{\texttt{Comp.}} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix}$$

`Compression`

- Reduced randomness requirement [BBP+16].

`Product`

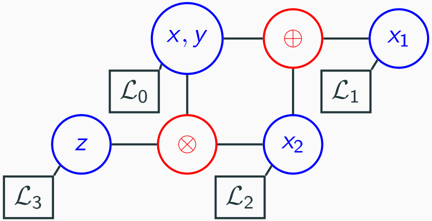
- Other security property for composition (PINI) [CS18].

`MatGen`

- Security against **horizontal attacks** [BCPZ16].

Outline

Factor graph & Belief propagation



$x = \mathrm{g}(y) \qquad x = x_1 \oplus x_2 \qquad z = x_2 \otimes y$

$\mathcal{L}$: intrinsic information from leakage trace.

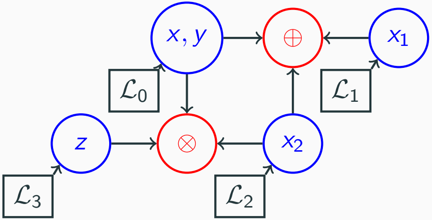
Belief propagation (BP): estimate x, y given information about x, y, x_1, x_2, z.

Alternate propagation of distribution estimates (beliefs):

Variable nodes $\rightleftharpoons$ Function nodes

Basis for Soft-Analytical Side-Channel Attacks (SASCA).

Factor graph & Belief propagation



$x = g(y) \qquad x = x_1 \oplus x_2 \qquad z = x_2 \otimes y$

$\mathcal{L}$: intrinsic information from leakage trace.

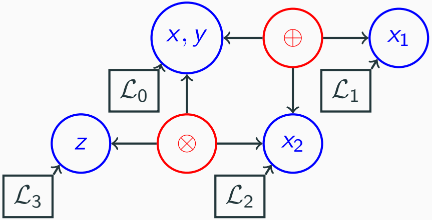
Belief propagation (BP): estimate x, y given information about x, y, x_1, x_2, z.

Alternate propagation of distribution estimates (beliefs):

Variable nodes $\rightleftharpoons$ Function nodes

Basis for Soft-Analytical Side-Channel Attacks (SASCA).

Factor graph & Belief propagation



$x = \mathrm{g}(y) \qquad x = x_1 \oplus x_2 \qquad z = x_2 \otimes y$

$\mathcal{L}$: intrinsic information from leakage trace.

Belief propagation (BP): estimate x, y given information about x, y, x_1, x_2, z.

Alternate propagation of distribution estimates (beliefs):

Variable nodes $\rightleftharpoons$ Function nodes

Basis for Soft-Analytical Side-Channel Attacks (SASCA).

Local Random Probing Model (LRPM)

ϵ-Random probing model

Observe for each variable x:

$$\begin{cases} x & \text{with probability } \epsilon \\ \perp & \text{with probability } 1-\epsilon \end{cases}$$

Local random probing model [GGS18]:

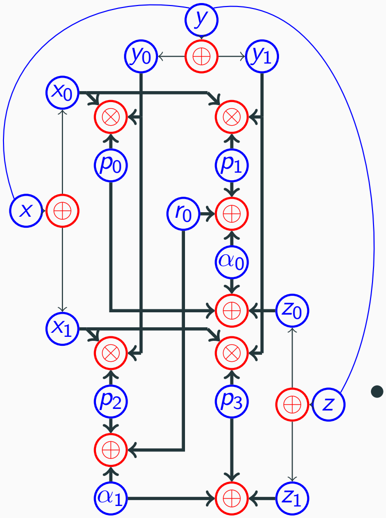
Random probing model adversary using BP.

Computing bounds in the LRPM:

Adaptation of BP to estimate mutual information (MI) instead of distributions.

- Input: *noise level* as observation MI on manipulated variables.
- Result: *security level* as MI on sensitive target variables.

LRPM example: Multiplication gadget



$$p_0 = x_0 \otimes y_0 \qquad \alpha_0 = p_1 \oplus r_0$$

$$p_1 = x_0 \otimes y_1 \qquad \alpha_1 = p_2 \oplus r_0$$

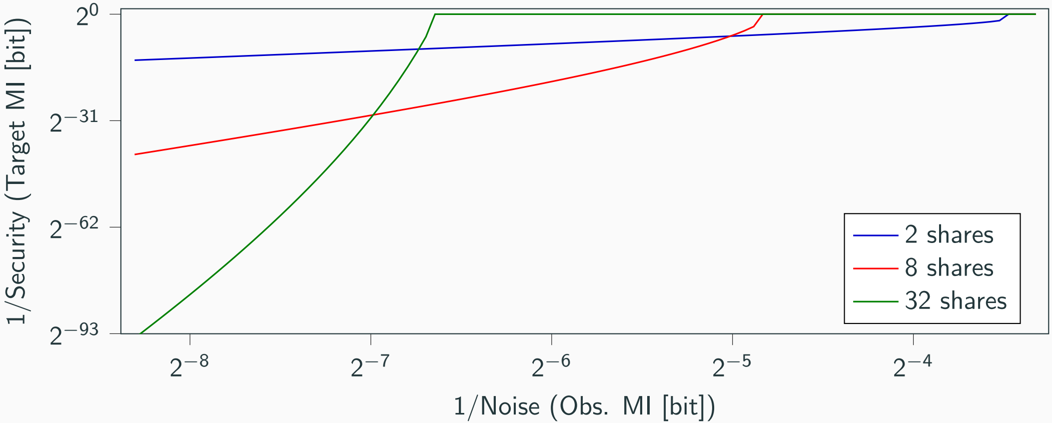
$$p_2 = x_1 \otimes y_0 \qquad z_0 = p_0 \oplus \alpha_0$$

$$p_3 = x_1 \otimes y_1 \qquad z_1 = p_3 \oplus \alpha_1$$

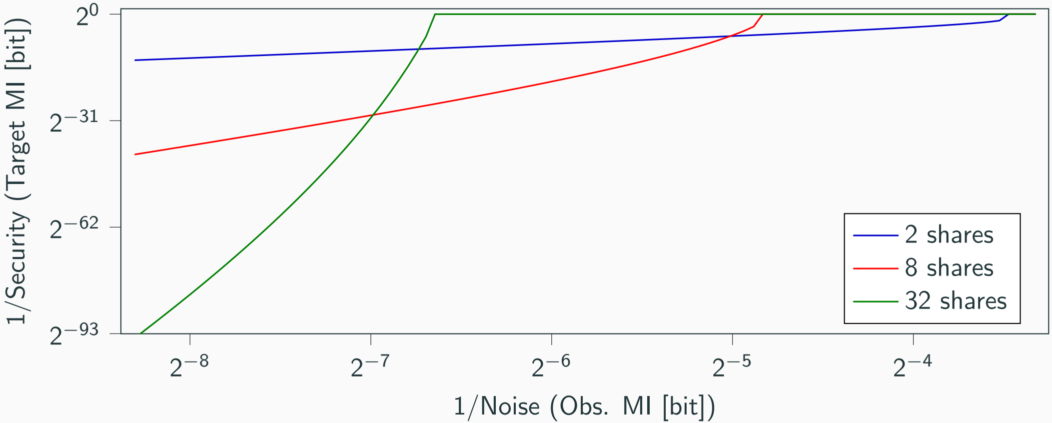
- Every operand/result computation leaks (sources of thick edges)

Outline

LRPM bound: [ISW03] masked AND

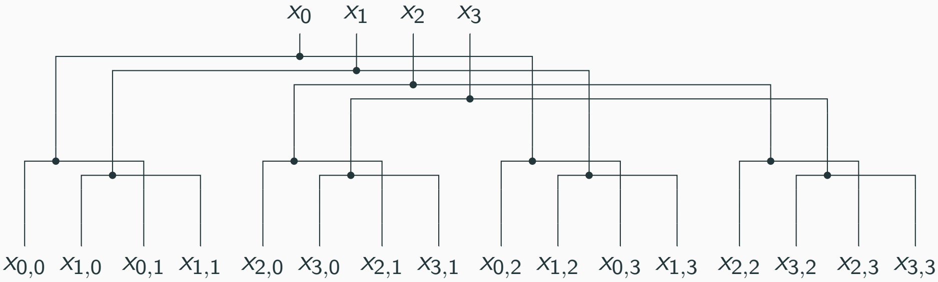


LRPM bound: [ISW03] masked AND



- Required noise increases with #shares
- More shares may be worse.

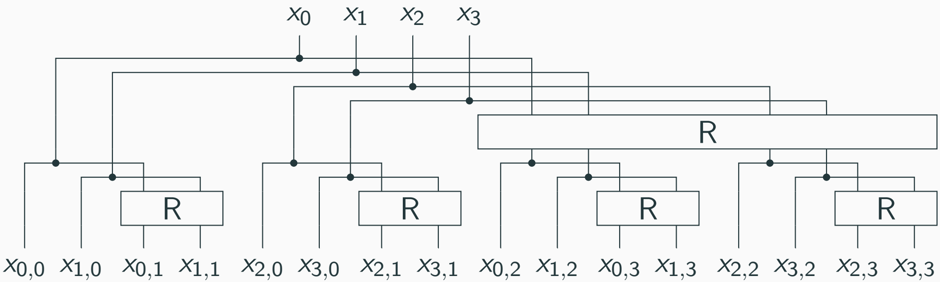
Reducing shares re-use: `MatGen` (I)



[ISW03]

- Simplest strategy
- Maximal efficiency
- Each input share used d times

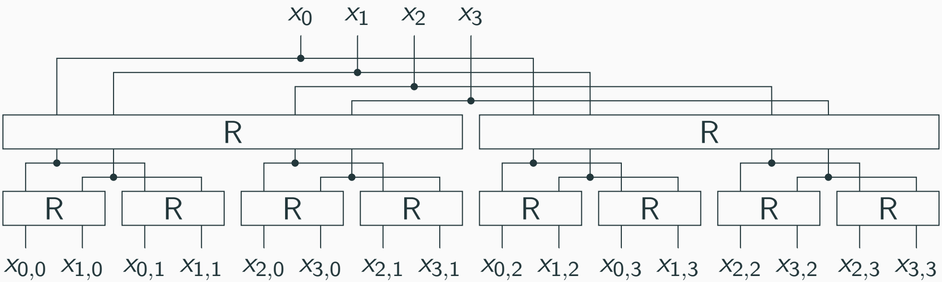
Reducing shares re-use: `MatGen` (II)



[BCPZ116]

- Add refreshing before shares multiplication
- Each input share used $\log d$ times

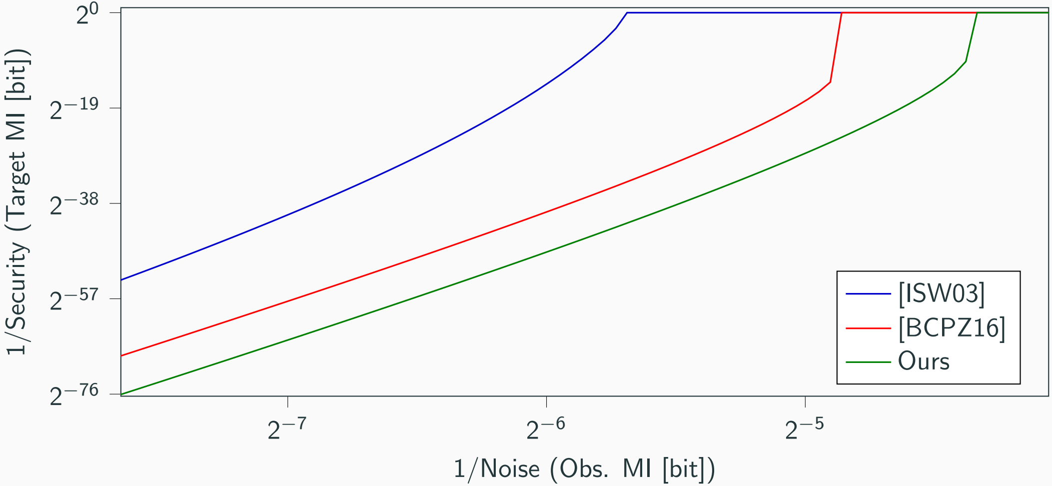
Reducing shares re-use: `MatGen` (III)



This work: many cheap refreshings

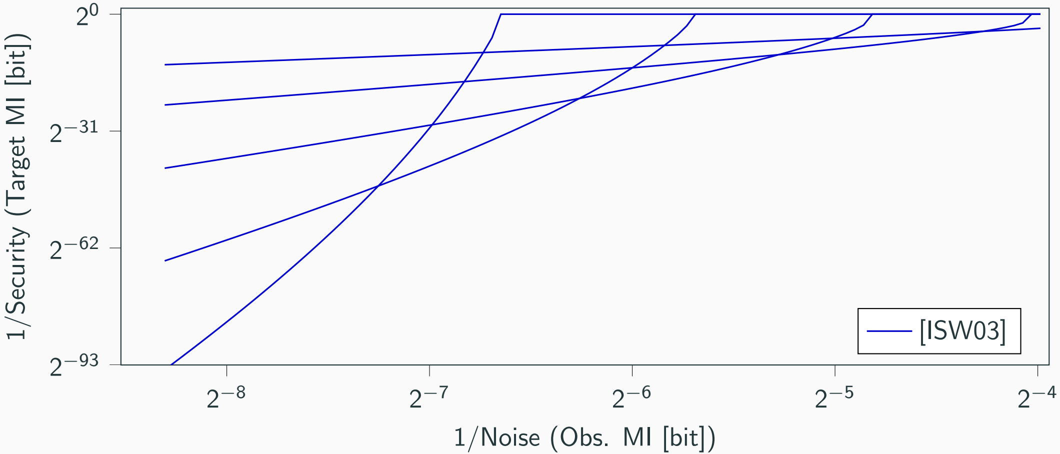
- 2 refresh gadgets per layer
- Each input share used 3 times

AND implementations: Gadget comparison



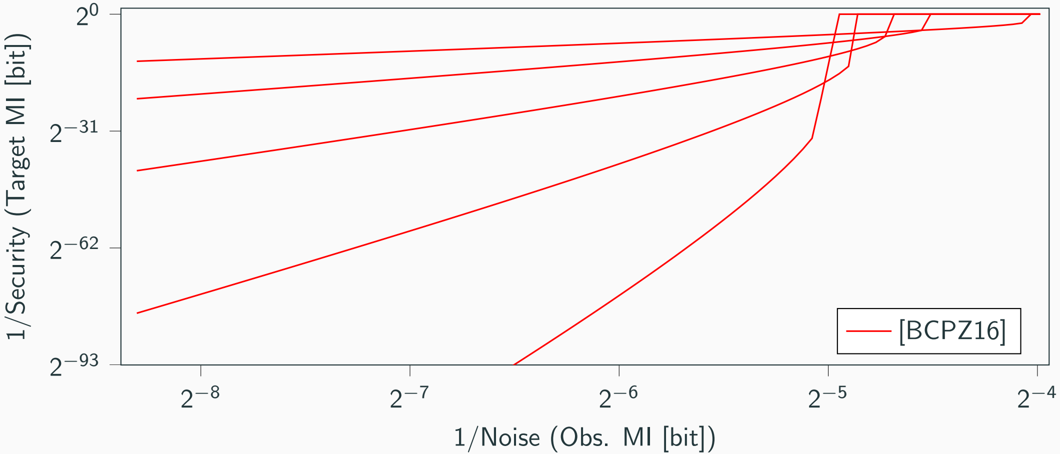
$d = 16$

AND implementation: [ISW03]



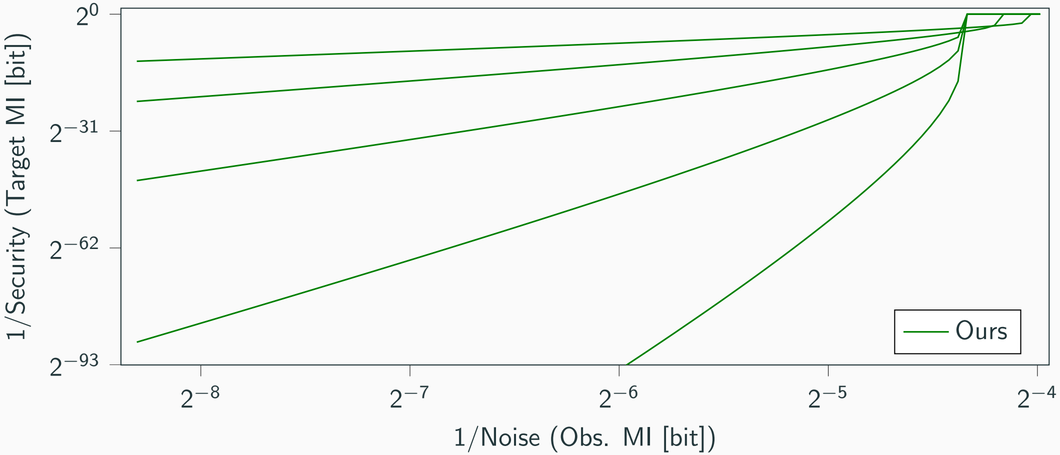
- $d = 2, \ldots, 32$
- Noise rate: $1/d$

AND implementations: [BCPZ16]



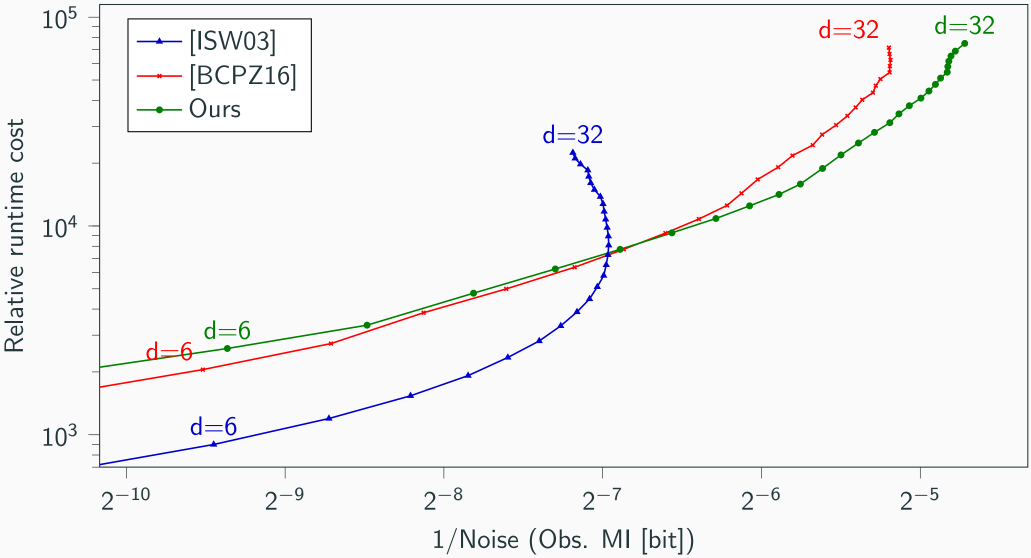
- $d = 2, \ldots, 32$
- Noise rate: $1/\log(d)$

AND implementations: Ours

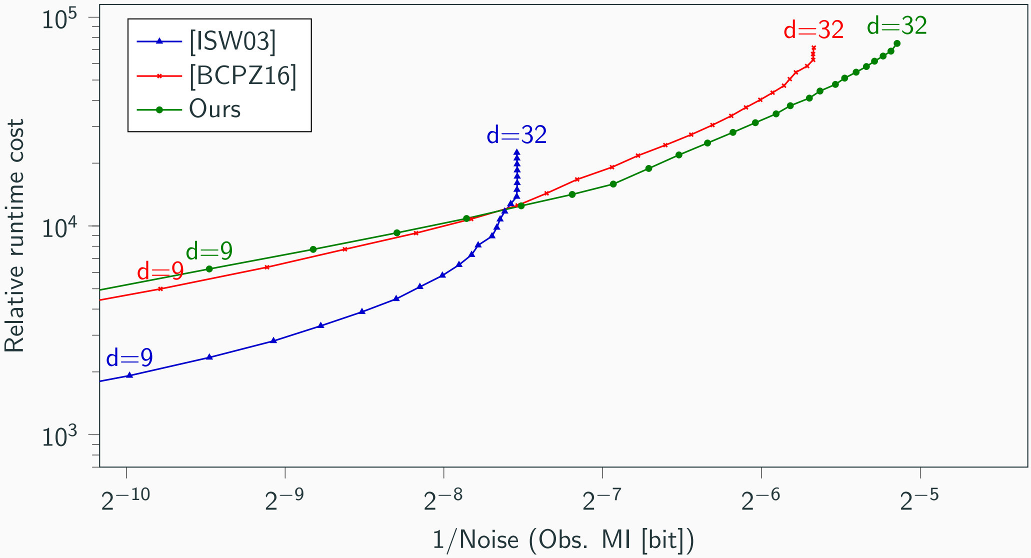


- $d = 2, \ldots, 32$
- Noise rate: 1

Cost: 40 bit security



Cost: 60 bit security



Conclusion

Horizontal attacks:

- Qualitative $\rightarrow$ Automated quantitative analysis.
- New multiplication gadget: conjectured $\mathcal{O}(1)$ noise rate (in $\mathbb{F}_2$).
- Randomness+Computations vs Noise: implementer's trade-off.

Conclusion

Horizontal attacks:

- Qualitative $\rightarrow$ Automated quantitative analysis.
- New multiplication gadget: conjectured $\mathcal{O}(1)$ noise rate (in $\mathbb{F}_2$).
- Randomness+Computations vs Noise: implementer's trade-off.

Not presented here:

- New composable (PINI) gadget reducing randomness.

Conclusion

Horizontal attacks:

- Qualitative $\rightarrow$ Automated quantitative analysis.
- New multiplication gadget: conjectured $\mathcal{O}(1)$ noise rate (in $\mathbb{F}_2$).
- Randomness$+$Computations vs Noise: implementer's trade-off.

Not presented here:

- New composable (PINI) gadget reducing randomness.

Thank you!