Fast verification of masking schemes in characteristic two

Nicolas Bordes
Joint work with Pierre Karpman

Université Grenoble Alpes, France

Eurocrypt 2021
2021–10–19

Masking schemes for finite field multiplication

Probing security model

Computationally checking security in $\mathbb{F}_2$

Results

Masking schemes for finite field multiplication

Probing security model

Computationally checking security in $\mathbb{F}_2$

Results

The context

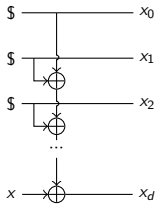
Context: Crypto implementations on observable devices

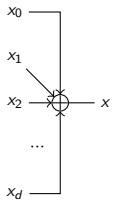
Objective: secure finite-field multiplication with leakage

- Implement $(a, b) \mapsto c = a \times b$, a, b, $c \in \mathbb{K}$
  - Used in non-linear ops in symmetric crypto (e.g. S-boxes)
  - Inputs/output usually secret!
- Problem: computations leak information

Basic idea: masking

- Split a, b, c into *shares* (i.e. use a secret-sharing scheme)
  - Typically simple and additive:
    $x = \sum_{i=0}^{d} x_i$, $x_{0,\ldots,d-1} \xleftarrow{\$} \mathbb{K}$, $x_d = x - \sum_{i=0}^{d-1} x_i$

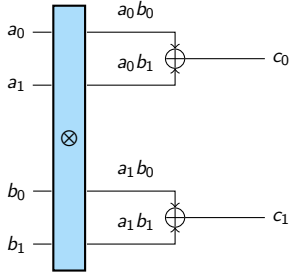




- Compute the operation over the shared operands; obtain a shared result
- Ensure that no information can (easily) be gained on a, b or c

First attempt

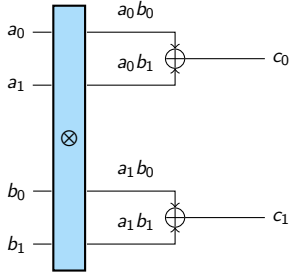
- We want to compute $c = \sum_k c_k = \sum_i a_i \times \sum_j b_j = \sum_{i,j} a_i b_j$
- So maybe define $c_k = \sum_{j=0}^{d} a_k b_j$?



- Problem: any single c_k reveals information about b
- This strategy is intuitively not secure.

First attempt

- We want to compute $c = \sum_k c_k = \sum_i a_i \times \sum_j b_j = \sum_{i,j} a_i b_j$
- So maybe define $c_k = \sum_{j=0}^{d} a_k b_j = a_k \times b$?



- Problem: any single c_k reveals information about b
- This strategy is intuitively not secure.

Masking schemes for finite field multiplication

Probing security model

Computationally checking security in $\mathbb{F}_2$

Results

Quick defs.

Gadget

A gadget for a function f is a (randomized) circuit C on (additively) shared inputs/outputs $\boldsymbol{x}_i$, $\boldsymbol{y}_j$ s.t. for every set of coins $\mathcal{R}$, $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) \leftarrow\!\shortmid C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \mathcal{R})$ satisfies:

$$\left(\sum_{j=1}^{v} \boldsymbol{y}_{1,j}, \ldots, \sum_{j=1}^{v} \boldsymbol{y}_{m,j}\right) = f\left(\sum_{j=1}^{u} \boldsymbol{x}_{1,j}, \ldots, \sum_{j=1}^{u} \boldsymbol{x}_{m,j}\right)$$

Probe

A probe on C maps a wire to the value it takes in a run of the gadget

d-privacy

Introduced by Ishai, Sahai & Wagner (2003)

- If every set of $t \leq d$ probes is independent from the unmasked values on which the gadget is evaluated $\rightsquigarrow$ d-privacy

ISW (2003) designed a d-private masking scheme with quadratic complexity in d

- $2d(d+1)$ sums
- $(d+1)^2$ products
- $d(d+1)/2$ additional random masks

Secure multiplication gadgets

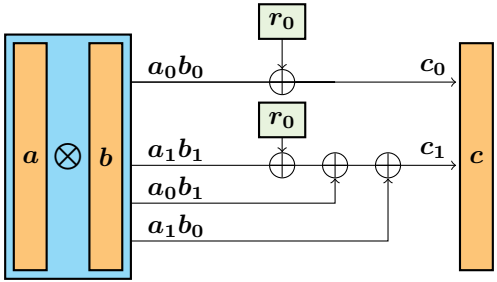


Figure: Multiplication gadget at order $d = 1$ from ISW, 2003.

Secure multiplication gadgets

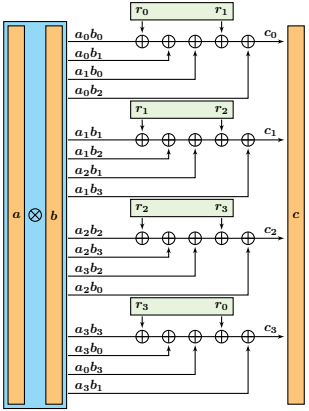


Figure: Multiplication gadget at order $d = 3$ from Barthe et al., 2017.

A composable security model based on simulatability

- d-privacy is not composable.
- Barthe et al. (2016) introduced composable models called *(strong) non-interference* based on simulatability

Simulatability

A given set of probes $\mathcal{P} := \{p_1, p_2, \ldots\}$ on C is t-simulatable if for fixed input sharings $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots)$, all the distributions induced on $\mathcal{P}$ by $\mathcal{R}$ can be simulated with the knowledge of $\leq t$ $\boldsymbol{x}_{1,i}$s; $\leq t$ $\boldsymbol{x}_{2,i}$s; etc.

d-(Strong) Non-Interference

d-Non-Interference (d-NI)

C is d-NI iff. any set of at most d probes is d-simulatable

d-Strong Non-Interference (d-SNI)

C is d-SNI iff. any set of at most $d_1 + d_2 \leq d$ probes is d_1-simulatable, where d_2 probes are on the output wires only

These security models are useful because:

- d-SNI $\Longrightarrow$ d-NI $\Longrightarrow$ d-private;
- under some independence hypotheses, SNI$\circ$NI = SNI.

Masking schemes for finite field multiplication

Probing security model

Computationally checking security in $\mathbb{F}_2$

Results

How to check a masking scheme?

To efficiently check if a masking scheme is d-NI, in absence of a generic proof, we want:

- An easy-to-check condition for simulatability of subsets of probes for a scheme defined over $\mathbb{F}_2$
- An efficient enumeration of all subsets of probes
- (Not shown here) A set of probes to check as small as possible
- (Not shown here) Extension to check d-SNI
- (Not shown here) Extension to check in the more hardware-oriented robust probing model

A linear condition on bilinear probes

Condition 3.2 (Belaïd et al., 2017)

A set of bilinear probes $\mathcal{P} = \{p_1, \ldots, p_\ell\}$ on a gadget C for a function $f : \mathbb{K}^2 \to \mathbb{K}$ satisfies Cond. 3.2 iff. $\exists\, \boldsymbol{\lambda} \in \mathbb{K}^\ell$, $\boldsymbol{M} \in \mathbb{K}^{(d+1)\times(d+1)}$, $\boldsymbol{\mu}$, $\boldsymbol{\nu} \in \mathbb{K}^{d+1}$, and $\tau \in \mathbb{K}$ s.t. $\sum_{i=1}^{\ell} \boldsymbol{\lambda}_i p_i = \boldsymbol{a}^t \boldsymbol{M} \boldsymbol{b} + \boldsymbol{a}^t \boldsymbol{\mu} + \boldsymbol{b}^t \boldsymbol{\nu} + \tau$ and all the rows of the block matrix $\begin{pmatrix} \boldsymbol{M} & \boldsymbol{\mu} \end{pmatrix}$ or all the columns of the block matrix $\begin{pmatrix} \boldsymbol{M} \\ \boldsymbol{\nu}^t \end{pmatrix}$ are non-zero

- No $\boldsymbol{r}$-dependency: cannot be simulated with a uniform distribution
- No zero rows/columns: full functional dependence on the $d+1$ shares of $\boldsymbol{a}$ or $\boldsymbol{b}$

Theorem for simulatability

The previous condition is useful to analyse the security of a gadget

Theorem (Belaïd et al., 2017)

If $\mathcal{P}$ satisfies Cond. 3.2, then it is not d-simulatable
If $\mathcal{P}$ is not d-simulatable *and* $\#\mathbb{K} > d + 1$, then it satisfies Cond 3.2

Corollary(Belaïd et al., 2017)

If $\#\mathbb{K} > d + 1$ and no set of $\leq d$ probes on C satisfies Cond. 3.2, then it is d-NI

A **new** linear condition on bilinear probes

Condition 12

A set of bilinear probes $\mathcal{P} = \{p_1, \ldots, p_\ell\}$ on a gadget C for a function $f : \mathbb{K}^2 \to \mathbb{K}$ satisfies Cond. 12 iff. $\exists \boldsymbol{\lambda} \in \mathbb{K}^\ell$, $\boldsymbol{M} \in \mathbb{K}^{(d+1)\times(d+1)}$, $\boldsymbol{\mu}$, $\boldsymbol{\nu} \in \mathbb{K}^{d+1}$, and $\tau \in \mathbb{K}$ s.t. $\sum_{i=1}^{\ell} \boldsymbol{\lambda}_i p_i = \boldsymbol{a}^t \boldsymbol{M} \boldsymbol{b} + \boldsymbol{a}^t \boldsymbol{\mu} + \boldsymbol{b}^t \boldsymbol{\nu} + \tau$ and *the block matrix* $\begin{pmatrix} \boldsymbol{M} & \boldsymbol{\mu} \end{pmatrix}$ *has* **at least** $\ell + 1$ *non-zero rows or the block matrix* $\begin{pmatrix} \boldsymbol{M} \\ \boldsymbol{\nu}^t \end{pmatrix}$ *has* **at least** $\ell + 1$ *non-zero columns.*

- No $\boldsymbol{r}$-dependency: cannot be simulated with a uniform distribution
- At least $\ell + 1$ non-zero rows/columns: functional dependence on more shares of $\boldsymbol{a}$ or $\boldsymbol{b}$ than the number of probes

New theorem for simulatability

The previous condition is useful to analyse the security of a gadget

Theorem

If $\mathcal{P}$ of ℓ probes satisfies Condition 12, then it is not ℓ-simulatable.
If $\mathcal{P}$ is not d-simulatable, then there exists $\mathcal{P}' \subseteq \mathcal{P}$ that satisfies Condition 12.

Corollary

If no set of $\leq d$ probes on C satisfies Condition 12, then it is d-NI

Efficient software implementation

Check Condition 12 for all subset of $\leq d$ probes.
To do it efficiently:

- Rewrite the condition in terms of weight of indicator matrices;
- Use vectorization to compute the matrices & weights;
- Use combination Gray codes for efficient enumeration;

$\rightsquigarrow$ Peak performance (@2.60 GHz) of $\approx 2^{27.5}$ checks/s
$\rightsquigarrow$ Easy parallelization

Masking schemes for finite field multiplication

Probing security model

Computationally checking security in $\mathbb{F}_2$

Results

Overview of our contributions (model and tool)

- New condition for d-(S)NI security over small fields
- New algorithm to check d-(S)NI over $\mathbb{F}_2$
- An efficient implementation
- Improved verification performance by 3 orders of magnitude
  - e.g. 8-SNI verif.: 13 days (4 threads, maskVerif, Barthe et. al. 2019) vs < 10 minutes (1 thread, our software)

Overview of our contributions (applications)

- NI/SNI verif. of concrete masking schemes up to order $d = 11$ (previously verified only to order 7)
- Disproving conjecture of Barthe et. al. on the security of generic transformation of NI gadgets into SNI gadgets
- Slightly better SNI masking schemes for high-order
  - e.g. at order $d = 7$, 17% less additional random masks (r_i)

References

- Full version: https://eprint.iacr.org/2019/1165
  (with additional examples and figures)
- Implementation: https://github.com/NicsTr/binary_masking