New instantiations of the CRYPTO 2017 masking schemes

Pierre Karpman
Joint work with Dan Roche

Université Grenoble Alpes

GT GRACE — Palaiseau
2018–11–08

Masking schemes for finite field multiplication

Proving security

New instantiations of the schemes from CRYPTO 2017

Conclusion

Masking schemes for finite field multiplication

Proving security

New instantiations of the schemes from CRYPTO 2017

Conclusion

The context

Context: Crypto implementation on observable devices

Objective: secure finite-field multiplication w/ leakage

- Implement $(a, b) \mapsto c = a \times b$, $a, b, c \in \mathbb{K}$
  - Used in non-linear ops in sym. crypto (e.g. S-boxes)
  - Input/outputs usually secret!
- Problem: computations leak information
  - When computing $c = a \times b$, one may learn the values a, b, c
- $\rightsquigarrow$ Need a way to compute a product w/o leaking (too much) the operands & the result

Basic idea

- 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 neither of a, b, c can be (easily) recovered

Attack model; d-Privacy (ISW, 2003)

Given a randomized *circuit* $\mathcal{C}$ computing $(c_0, \ldots, c_d) = (a_0, \ldots, a_d) \times (b_0, \ldots, b_d)$, an attacker:

- May *probe* the values $p_0, \ldots, p_{t-1}$ of $t \leq d$ wires of $\mathcal{C}$
- Succeeds if the distribution of $\mathcal{F}(p_0, \ldots, p_{t-1})$ depends on a (or b, or c), for some $\mathcal{F}$
- Equivalently, fails if the distribution is the same for all $\mathcal{F}$ and all values of a (and b, and c)

Examples:

- Probing $a_0, \ldots, a_{d-1}$ does not lead to a successful attack
- Probing $a_0 \times (b_0 + \ldots + b_d)$ leads to an attack

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 $c_i = a_i \sum_{j=0}^{d} b_j$?
- Problem: any single c_i reveals information about the inputs
- One solution (Ishai, Sahai & Wagner, 2003): rerandomize using fresh randomness
  - For instance (for $d = 3$):
  - $c_0 = a_0 b_0 + r_{0,1} + r_{0,2} + r_{0,3}$
  - $c_1 = a_1 b_1 + (r_{0,1} + a_0 b_1 + a_1 b_0) + r_{1,2} + r_{1,3}$
  - $c_2 = a_2 b_2 + (r_{0,2} + a_0 b_2 + a_2 b_0) + (r_{1,2} + a_1 b_2 + a_2 b_1) + r_{2,3}$
  - $c_3 =$
    $a_3 b_3 + (r_{0,3} + a_0 b_3 + a_3 b_0) + (r_{1,3} + a_1 b_3 + a_3 b_1) + (r_{2,3} + a_2 b_3 + a_3 b_2)$
- Scheduling of the operations is important (impacts the probes available to the adversary), hence the $(\cdot)$s

Masking complexity

- ISW provides a practical solution for masking a multiplication
- But the cost is quadratic in d. More precisely, d-privacy requires:
  - $2d(d+1)$ sums
  - $(d+1)^2$ products
  - $d(d+1)/2$ fresh random masks
- Decreasing the cost/overhead of masking is a major problem
  - Use block ciphers that need few multiplications (e.g. ZORRO, Gérard et al., 2013 (broken))
  - Amortize the cost of masking several mult. (e.g. Coron et al., 2016)
  - Decrease the cost of masking a single mult. (e.g. Belaïd et al., 2016, 2017)

Schemes from CRYPTO 2017

Two schemes introduced by Belaïd et al. (2017):

- "Alg. 4", with linear bilinear multiplication complexity, requiring:
  - $9d^2 + d$ sums
  - $2d^2$ linear products
  - $2d + 1$ products
  - $2d^2 + d(d-1)/2$ fresh random masks
- "Alg. 5", with linear randomness complexity, requiring:
  - $2d(d+1)$ sums
  - $d(d+1)$ linear products
  - $(d+1)^2$ products
  - d fresh random masks

Focus on Alg. 4

This scheme uses shares of three kinds:

- $c_0 := \left(a_0 + \sum_{i=1}^{d}(r_i + a_i)\right) \cdot \left(b_0 + \sum_{i=1}^{d}(s_i + b_i)\right)$;
- $c_i := -r_i \cdot \left(b_0 + \sum_{j=1}^{d}(\delta_{i,j}s_j + b_j)\right)$, $1 \le i \le d$;
- $c_{i+d} := -s_i \cdot \left(a_0 + \sum_{j=1}^{d}(\gamma_{i,j}r_j + a_j)\right)$, $1 \le i \le d$.

With:

- $\gamma = \left(\gamma_{i,j}\right) \in \mathbb{K}^{d\times d}$
- $\delta = \left(\delta_{i,j}\right) \in \mathbb{K}^{d\times d}$ s.t. $\gamma + \delta$ is the all-one matrix

(Plus an additional post-processing, not studied here)

Instantiation issues

Problem: finding γ so that the scheme is *secure* is hard. Belaïd et al.:

- Found an explicit γ for $d = 2$ over $\mathbb{F}_{2^2}$ (and other larger fields)
- Proved (non-constructively) the existence of good γ at order d over $\mathbb{F}_q$ when $q > \mathcal{O}(d)^{d+1}$

Our results: we give constructions/examples for:

- $d = 3$ over $\mathbb{F}_{2^k}$, $k \geq 3$
- $d = 4$ over $\mathbb{F}_{2^k}$, $5 \leq k \leq 16$
- $d = 5$ over $\mathbb{F}_{2^k}$, $10 \leq k \leq 16$
- $d = 6$ over $\mathbb{F}_{2^k}$, $15 \leq k \leq 16$

Masking schemes for finite field multiplication

Proving security

New instantiations of the schemes from CRYPTO 2017

Conclusion

What's a good γ anyways?

Recall that to attack Alg. 4, one wants to:

1. Select d probes $p_0, \ldots, p_{d-1}$ of intermediate values
2. Find $\mathcal{F}$ s.t. the distribution of $\mathcal{F}(p_0, \ldots, p_{d-1})$ depends on a (say)

In Alg. 4, the possible probes (relating to a) are:

- a_i, r_i, $a_i + r_i$, $\gamma_{j,i} r_i$, $a_i + \gamma_{j,i} r_i$, for $0 \leq i \leq d$, $1 \leq j \leq d$
- $a_0 + \sum_{i=1}^{k}(a_i + r_i)$, $1 \leq k \leq d$
- $a_0 + \sum_{i=1}^{k}(a_i + \gamma_{j,i} r_i)$, $1 \leq k \leq d$, $1 \leq j \leq d$

Claim: it is sufficient to only consider $\mathcal{F}$s that are linear combinations of the p_is (cf. Belaïd et al., 2017)

Attack sets

One sub-objective: decide if a set of probes P leads to an attack

- One must find x_is s.t. $\pi = \sum x_i p_i = \sum y_i a_i + \sum z_i r_i$ with $y_i \neq 0$, $z_i = 0$ for all i
  - If π "includes an r_i" or "misses an a_i", then it is uniform
- For each probe, consider indicator matrices of its a_is and r_is
- For instance, the probe $a_0 + a_1 + \gamma_{1,1} r_1$ $(d = 2) \rightsquigarrow \begin{pmatrix}1 & 1 & 0\end{pmatrix}^t$, $\begin{pmatrix}0 & \gamma_{1,1} & 0\end{pmatrix}^t$
- Gather all such matrices in larger matrices $\mathbf{L}_P$ and $\mathbf{M}_P^\gamma$
- There is an attack iff. $\exists u \in \ker \mathbf{M}_P^\gamma$ s.t. $\mathbf{L}_P u$ is of full weight

“All at once”

For the entire scheme Alg. 4:

- Form the matrices $\mathbf{L}$, $\mathbf{M}^\gamma$ of all $\mathcal{O}(d^2)$ possible probes
- $\mathbf{L}$ includes blocks $\mathbf{I}_d$, $\mathbf{T}_d$ (all-one upper triangular)
- $\mathbf{M}^\gamma$ includes blocks $\mathbf{I}_d$, $\mathbf{T}_d$, $\mathbf{D}_{\gamma,j} = \begin{pmatrix}\gamma_{j,1} & \cdots & \gamma_{j,d}\end{pmatrix} \mathbf{I}_d$, $\mathbf{T}_{\gamma,j} = \begin{pmatrix}\gamma_{j,1} & \cdots & \gamma_{j,d}\end{pmatrix} \mathbf{T}_d$
- Find if there is a u of weight $\leq d$ s.t. $\mathbf{M}^\gamma u = \vec{0}$, $\mathbf{L}u$ is of weight $d+1$

Proving security: back to subcases

So far, haven't found a way to tackle $\mathbf{L}$ and $\mathbf{M}^\gamma$ globally :(So, to prove security for a given γ:

- Look at all submatrices $\mathbf{L}_P$ and $\mathbf{M}^\gamma_P$ for d probes P
- For each:
  1. Compute a basis $\mathbf{B}$ of the (right) kernel of $\mathbf{M}^\gamma_P$
  2. There is an attack with P iff. $\mathbf{N}_P = \mathbf{L}_P\mathbf{B}$ has no all-zero row

  - If $\mathbf{N}_P$ has a zero row, then no linear combination of probes depends on all a_is and cancels all r_is
  - If $\mathbf{N}_P$ has no zero row, there is at least one linear combination of probes that depends on all a_is and cancels all r_is
    - By a combinatorial argument, as long as $\#\mathbb{K} > d$ (e.g. use Schwartz-Zippel-DeMillo-Lipton)

Testing optimizations

The previous algorithms allows to test the security of an instance by checking $\approx \binom{d^2}{d}$ submatrices $\mathbf{L}_P$, $\mathbf{M}_P^\gamma$. Some optims:

- Do early-abort
- Check "critical cases" first
- Don't check stupid choices for P
- Use batch kernel computations

Masking schemes for finite field multiplication

Proving security

New instantiations of the schemes from CRYPTO 2017

Conclusion

Finding secure instantiations

The testing algorithm can be used to find secure instantiations:

1. Draw γ (δ) at random
2. Check that there is no attack

It works, but we can do better by picking super-regular/MDS γs (δs). Observations:

- If $\dim \ker \mathbf{M}_P^\gamma = 0$, then no attack is possible w/ probes P
  - Try to pick γ s.t. $\mathbf{M}_P^\gamma$ is invertible for many Ps
- Many $\mathbf{M}_P^\gamma$'s are made of submatrices of γ
  - All invertible, if γ is MDS

MDS precondition: small cases

- For $d = 1, 2$, it is sufficient for γ, δ to be MDS for the scheme to be secure
- For $d = 3$, one must additionally check that no matrix of the form

$$\begin{pmatrix} \gamma_{i,1} & \gamma_{j,1} & \gamma_{k,1} \\ \gamma_{i,2} & \gamma_{j,2} & \gamma_{k,2} \\ \gamma_{i,3} & \gamma_{j,3} & 0 \end{pmatrix}, i \neq j \neq k,$$

  is singular
  - Not systematically ensured by the MDS property
  - Can be solved symbolically

MDS precondition: larger cases; enforcement

- For $d \geq 4$, not feasible (?) to enforce invertibility of all $\mathbf{M}_P^\gamma$
- But MDS γs are still more likely to be secure than non-MDS ones
  - E.g. w/ Pr 0.063 instead of 0.030 for $d = 4$ over $\mathbb{F}_{2^8}$
- Problem: how to ensure that *both* γ and δ are MDS?
  - Use a (generalized) Cauchy construction $\big(x_{i,j} = c_i d_j/(x_i - y_j)\big)$, viz. $\big(\gamma_{i,j} = x_i/(x_i - y_j)\big)$
  - Then $\delta_{i,j} = 1 - x_i/(x_i - y_j) = -y_j/(x_i - y_j)$, so δ is Cauchy and then MDS

Masking schemes for finite field multiplication

Proving security

New instantiations of the schemes from CRYPTO 2017

Conclusion

The end?

- We found more instances of the (two) masking schemes of CRYPTO 2017, at larger orders
- Still only reaching $d = 4$ over "useful" fields such as $\mathbb{F}_{2^8}$

Future work: try to generalize by substituting $x \mapsto \gamma_{j,i}x$, $\gamma_{j,i} \in \mathbb{F}_{2^n}$ by $x \mapsto \Gamma_{j,i}x$, $\Gamma_{j,i} \in \mathrm{GL}(n, \mathbb{F}_2)$

- Tremendously more candidates for $\Gamma_{i,j}$ (e.g. $\approx 2^{62}$ instead of 256 for $n = 8$)
- Security condition should be similarly easy to define & test (TBC)
- MDS (Cauchy) constructions generalize well over block matrices (under some moderate constraints) (Li et al., 2018)

Source

For more details:

- https://eprint.iacr.org/2018/492