# Faster Evaluation of S-Boxes via Common Shares

J-S. Coron, A. Greuet, E. Prouff, R. Zeitoun

F.Rondepierre

CHES 2016

## AES

By definition: $S_{AES}(x) = A \cdot x^{254} + b \in \mathbb{F}_{2^8}[x]$

# S-Box Evaluation

## AES

By definition: $S_{AES}(x) = A \cdot x^{254} + b \in \mathbb{F}_{2^8}[x]$

## Other Blockciphers

DES S-Box Table

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 2 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 3 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

↓ Polynomial interpolation

$$S_{DES}(x) = \underbrace{a_{63}x^{63} + a_{62}x^{62} + \cdots + a_1 x + a_0}_{\text{compute with } +, \times, \cdot^2} \in \mathbb{F}_{2^6}[x]$$

# Security Context

## $t$-Probing Adversary

A $t$-probing adversary is allowed to know the exact value of at most $t$ intermediate results.

## $t$-Probing Adversary

A $t$-probing adversary is allowed to know the exact value of at most $t$ intermediate results.

- Adversary can access key values.
- Security is built to twhart limited adversaries.

## Secret Sharing/Masking

In order to thwart a $t$-probing adversary, each sensitive variable $x$ is split in $n = t + 1$ variables $(x_0, \ldots, x_t)$, such that:

$$x = x_0 \oplus x_1 \oplus \cdots \oplus x_t$$

- Variables $x_1, \ldots, x_t$ are by convention random **masks**.
- $x_0 = x \oplus \bigoplus_{i \geqslant 1} x_i$
- $X = (x_0, \ldots, x_t)$ is a **shared** representation of $x$.

## 1st Order Secure Multiplication

Let $A$, $B$ be two shared variables and say we want to compute $C = (c_0, c_1)$ such that $C$ is a sharing of $a \cdot b$:

$$\begin{aligned}(a \oplus a_1) \cdot (b \oplus b_1) &= a \cdot b \oplus a \cdot b_1 \oplus a_1 \cdot b \oplus a_1 \cdot b_1 \\ &= a \cdot b \oplus (a \oplus a_1) \cdot b_1 \oplus (b \oplus b_1) \cdot a_1 \oplus a_1 \cdot b_1 \\ a_0 \cdot b_0 &= a \cdot b \oplus a_0 \cdot b_1 \oplus b_0 \cdot a_1 \oplus a_1 \cdot b_1\end{aligned}$$

## 1st Order Secure Multiplication

Let $A$, $B$ be two shared variables and say we want to compute $C = (c_0, c_1)$ such that $C$ is a sharing of $a \cdot b$:

$$\begin{aligned}(a \oplus a_1) \cdot (b \oplus b_1) &= a \cdot b \oplus a \cdot b_1 \oplus a_1 \cdot b \oplus a_1 \cdot b_1 \\ &= a \cdot b \oplus (a \oplus a_1) \cdot b_1 \oplus (b \oplus b_1) \cdot a_1 \oplus a_1 \cdot b_1 \\ a_0 \cdot b_0 &= a \cdot b \oplus a_0 \cdot b_1 \oplus b_0 \cdot a_1 \oplus a_1 \cdot b_1\end{aligned}$$

- We would say $C(c_0, c_1)$:

$$\begin{aligned}c_0 &= a_0 \cdot b_0 \\ c_1 &= [(a_0 \cdot b_1) \oplus a_1 \cdot b_0] \oplus (a_1 \cdot b_1)\end{aligned}$$

## 1st Order Secure Multiplication

Let $A$, $B$ be two shared variables and say we want to compute $C = (c_0, c_1)$ such that $C$ is a sharing of $a \cdot b$:

$$\begin{aligned}(a \oplus a_1) \cdot (b \oplus b_1) &= a \cdot b \oplus a \cdot b_1 \oplus a_1 \cdot b \oplus a_1 \cdot b_1 \\ &= a \cdot b \oplus (a \oplus a_1) \cdot b_1 \oplus (b \oplus b_1) \cdot a_1 \oplus a_1 \cdot b_1 \\ a_0 \cdot b_0 &= a \cdot b \oplus a_0 \cdot b_1 \oplus b_0 \cdot a_1 \oplus a_1 \cdot b_1\end{aligned}$$

- Security needs an additional random $r$:

$$\begin{aligned}c_0 &= a_0 \cdot b_0 \oplus r \\ c_1 &= (a_1 \cdot b_1) \oplus [(a_0 \cdot b_1 \oplus r) \oplus a_1 \cdot b_0]\end{aligned}$$

## 1st Order Secure Multiplication

Let $A$, $B$ be two shared variables and say we want to compute $C = (c_0, c_1)$ such that $C$ is a sharing of $a \cdot b$:

$$(a \oplus a_1) \cdot (b \oplus b_1) = a \cdot b \oplus a \cdot b_1 \oplus a_1 \cdot b \oplus a_1 \cdot b_1$$
$$= a \cdot b \oplus (a \oplus a_1) \cdot b_1 \oplus (b \oplus b_1) \cdot a_1 \oplus a_1 \cdot b_1$$
$$a_0 \cdot b_0 = a \cdot b \oplus a_0 \cdot b_1 \oplus b_0 \cdot a_1 \oplus a_1 \cdot b_1$$

- Security needs an additional random $r$:

$$c_0 = a_0 \cdot b_0 \oplus r$$
$$c_1 = (a_1 \cdot b_1) \oplus [(a_0 \cdot b_1 \oplus r) \oplus a_1 \cdot b_0]$$

- Not secure if by construction we have $a_1 = b_1$

## Sequence of Secure Multiplications

Say we want to compute $E$, $F$ from $A, B, C, D$, such that:

$$E = A \cdot B$$
$$F = C \cdot D$$

## Sequence of Secure Multiplications

Say we want to compute $E$, $F$ from $A, B, C, D$, such that:

$$E = A \cdot B$$
$$F = C \cdot D$$

In a 1st order context, the paper deals with:

$$e_0 = a_0 \cdot b_0 \oplus r$$
$$e_1 = (a_1 \cdot b_1) \oplus [(a_0 \cdot b_1 \oplus r) \oplus a_1 \cdot b_0]$$
$$f_0 = c_0 \cdot d_0 \oplus r$$
$$f_1 = (c_1 \cdot d_1) \oplus [(c_0 \cdot d_1 \oplus r) \oplus c_1 \cdot d_0]$$

## Sequence of Secure Multiplications

Say we want to compute $E$, $F$ from $A, B, C, D$, such that:

$$E = A \cdot B$$
$$F = C \cdot D$$

In a 1st order context, the paper deals with:

$$e_0 = a_0 \cdot b_0 \oplus r$$
$$e_1 = (a_1 \cdot b_1) \oplus [(a_0 \cdot b_1 \oplus r) \oplus a_1 \cdot b_0]$$
$$f_0 = c_0 \cdot d_0 \oplus r$$
$$f_1 = (c_1 \cdot d_1) \oplus [(c_0 \cdot d_1 \oplus r) \oplus c_1 \cdot d_0]$$

# Common Shares

## Sequence of Secure Multiplications

Say we want to compute $E$, $F$ from $A, B, C, D$, such that:

$$E = A \cdot B$$
$$F = C \cdot D$$

In a 1st order context, we can have:

$$a_1 = c_1$$
$$b_1 = d_1$$

## Sequence of Secure Multiplications

Say we want to compute $E$, $F$ from $A, B, C, D$, such that:

$$E = A \cdot B$$
$$F = C \cdot D$$

The paper also extends the result to $t$-probing context:

$$a_i = c_i, \quad \frac{t+1}{2} \leqslant i \leqslant t$$
$$b_i = d_i, \quad \frac{t+1}{2} \leqslant i \leqslant t$$

## Optimality of sharing

Let $A$, $B$ be two shared variables, such that :

$$a_i = b_i, k \leqslant i \leqslant t$$

- If $k = 1$, then $a_0 \oplus b_0 = a \oplus b$

# Common Shares

## Optimality of sharing

Let $A$, $B$ be two shared variables, such that :

$$a_i = b_i, k \leqslant i \leqslant t$$

- If $k = 1$, then $a_0 \oplus b_0 = a \oplus b$
- If $k < \frac{t+1}{2}$, then $\bigoplus_{i<k} a_i \oplus b_i = a \oplus b$

## Optimality of sharing

Let $A$, $B$ be two shared variables, such that :

$$a_i = b_i, k \leqslant i \leqslant t$$

- If $k = 1$, then $a_0 \oplus b_0 = a \oplus b$
- If $k < \frac{t+1}{2}$, then $\bigoplus_{i<k} a_i \oplus b_i = a \oplus b$
- If $k \geqslant \frac{t+1}{2}$, then $\bigoplus_{i<k} a_i \oplus b_i$ requires more than $t$ probing

# Common Shares

**CommonShares**

**Input:** $A = (a_0, \ldots, a_t)$ shares of $a$ and $B$, shares of $b$
**Output:** $A' = (a'_0, \ldots, a'_t)$ shares of $a$ and $B'$, shares of $b$

```
for i = ⌈(t+1)/2⌉ to t do
    r_i ← F_{2^k}
    j ← i − (t+1)/2
    a'_i ← r_i, a'_j ← (a_j ⊕ r_i) ⊕ a_i
    b'_i ← r_i, b'_j ← (b_j ⊕ r_i) ⊕ b_i
end for
```

$$\textbf{for } i = \left\lceil \tfrac{t+1}{2} \right\rceil \textbf{ to } t \textbf{ do}$$
$$r_i \leftarrow \mathbb{F}_{2^k}$$
$$j \leftarrow i - \tfrac{t+1}{2}$$
$$a'_i \leftarrow r_i,\ a'_j \leftarrow (a_j \oplus r_i) \oplus a_i$$
$$b'_i \leftarrow r_i,\ b'_j \leftarrow (b_j \oplus r_i) \oplus b_i$$
$$\textbf{end for}$$

# Higher-Order Secure Multiplication

**SecMult**

**Input:** $A = (a_0, \ldots, a_t)$ shares of $a$ and $B$, shares of $b$
**Output:** $C$, shares of $a \cdot b$

**for** $i = 0$ **to** $t$ **do**
  $c_i \leftarrow a_i \cdot b_i$
**end for**
**for** $i = 0$ **to** $t$ **do**
  **for** $j = i + 1$ **to** $t$ **do**
    $r \leftarrow \mathbb{F}_{2^k}$
    $c_i \leftarrow c_i \oplus r$
    $c_j \leftarrow c_j \oplus [(a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i]$
  **end for**
**end for**

# Multiplications with Common Shares

## TwoMult

**Input:** $A, B, C, D$ shares of $a, b, c, d$, where $A, C$ (resp. $B, D$) have common shares

**Output:** $E, F$ shares of $a \cdot b, c \cdot d$

**for** $i = 0$ **to** $t$ **do**

$$e_i \leftarrow a_i \cdot b_i$$

$$f_i \leftarrow \begin{cases} c_i \cdot d_i & 0 \leqslant i \leqslant \left\lfloor \frac{t-1}{2} \right\rfloor \\ e_i = c_i \cdot d_i & \left\lceil \frac{t+1}{2} \right\rceil \leqslant i \leqslant t \end{cases}$$

**end for**

# Multiplications with Common Shares

## TwoMult

**Input:** $A, B, C, D$ shares of $a, b, c, d$, where $A, C$ (resp. $B, D$) have common shares

**Output:** $E, F$ shares of $a \cdot b, c \cdot d$

**for** $i = 0$ **to** $t$ **do**

$e_i \leftarrow a_i \cdot b_i$

$$f_i \leftarrow \begin{cases} c_i \cdot d_i & 0 \leqslant i \leqslant \left\lfloor \frac{t-1}{2} \right\rfloor \\ e_i = c_i \cdot d_i & \left\lceil \frac{t+1}{2} \right\rceil \leqslant i \leqslant t \end{cases}$$

**end for**

**for** $i = 0$ **to** $t$ **do**

**for** $j = i + 1$ **to** $t$ **do**

$r \leftarrow \mathbb{F}_{2^k}$ $\qquad$ $s \leftarrow \mathbb{F}_{2^k}$

$e_i \leftarrow e_i \oplus r$ $\qquad$ $f_i \leftarrow f_i \oplus s$

$e_j \leftarrow e_j \oplus [(a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i]$ $\qquad$ $f_j \leftarrow f_j \oplus [(c_i \cdot d_j \oplus s) \oplus c_j \cdot d_i]$

**end for**

**end for**

# Multiplications with Common Shares

## CommonMult

**Input:** $A, B, D$ shares of $a, b, d$, where $B, D$ have common shares
**Output:** $E, F$ shares of $a \cdot b, a \cdot d$

**for** $i = 0$ **to** $t$ **do**

$e_i \leftarrow a_i \cdot b_i$

$$f_i \leftarrow \begin{cases} a_i \cdot d_i & 0 \leqslant i \leqslant \left\lfloor \frac{t-1}{2} \right\rfloor \\ e_i & \left\lceil \frac{t+1}{2} \right\rceil \leqslant i \leqslant t \end{cases}$$

**end for**
**for** $i = 0$ **to** $t$ **do**
  **for** $j = i + 1$ **to** $t$ **do**

| | |
|---|---|
| $r \leftarrow \mathbb{F}_{2^k}$ | $s \leftarrow \mathbb{F}_{2^k}$ |
| $e_i \leftarrow e_i \oplus r$ | $f_i \leftarrow f_i \oplus s$ |
| $e_j \leftarrow e_j \oplus [(a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i]$ | $f_j \leftarrow f_j \oplus [(a_i \cdot d_j \oplus s) \oplus a_j \cdot d_i]$ |

  **end for**
**end for**

# Multiplications with Common Shares

**CommonMult**

**Input:** $A, B, D$ shares of $a, b, d$, where $B, D$ have common shares
**Output:** $E, F$ shares of $a \cdot b, a \cdot d$

**for** $i = 0$ **to** $t$ **do**

$e_i \leftarrow a_i \cdot b_i$

$$f_i \leftarrow \begin{cases} a_i \cdot d_i & 0 \leqslant i \leqslant \left\lfloor \frac{t-1}{2} \right\rfloor \\ e_i & \left\lceil \frac{t+1}{2} \right\rceil \leqslant i \leqslant t \end{cases}$$

**end for**

**for** $i = 0$ **to** $t$ **do**

**for** $j = i + 1$ **to** $t$ **do**

$r \leftarrow \mathbb{F}_{2^k}$ $\qquad$ $s \leftarrow \mathbb{F}_{2^k}$

$e_i \leftarrow e_i \oplus r$ $\qquad$ $f_i \leftarrow f_i \oplus s$

$e_j \leftarrow e_j \oplus [(a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i]$ $\qquad$ $f_j \leftarrow f_j \oplus [(a_i \cdot d_j \oplus s) \oplus a_j \cdot d_i]$

**end for**

**end for**

# Multiplications with Common Shares

## CommonMult

**Input:** $A, B, D$ shares of $a, b, d$, where $B, D$ have common shares
**Output:** $E, F$ shares of $a \cdot b, a \cdot d$

**for** $i = 0$ **to** $t$ **do**

$e_i \leftarrow a_i \cdot b_i$

$$f_i \leftarrow \begin{cases} a_i \cdot d_i & 0 \leqslant i \leqslant \left\lfloor \frac{t-1}{2} \right\rfloor \\ e_i & \left\lceil \frac{t+1}{2} \right\rceil \leqslant i \leqslant t \end{cases}$$

**end for**

**for** $i = 0$ **to** $t$ **do**

**for** $j = i + 1$ **to** $t$ **do**

$r \leftarrow \mathbb{F}_{2^k}$ $\quad\quad s \leftarrow \mathbb{F}_{2^k}$

$e_i \leftarrow e_i \oplus r$ $\quad\quad f_i \leftarrow f_i \oplus s$

$e_j \leftarrow e_j \oplus [(a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i]$ $\quad\quad f_j \leftarrow f_j \oplus [(a_i \cdot d_j \oplus s) \oplus a_j \cdot d_i]$

**end for**

**end for**

## Performances

| | | |
|---|---|---|
| SecMult | $(t+1)^2$ | |
| TwoMult | $2(t+1)^2$ | $-\left(\lfloor\frac{t+1}{2}\rfloor\right)^2$ |
| CommonMult | $2(t+1)^2$ | $-(t+1)\cdot\left(\lfloor\frac{t+1}{2}\rfloor\right)$ |
| $m$-Mult | $m(t+1)^2$ | $-(m-1)\left(\lfloor\frac{t+1}{2}\rfloor\right)^2$ |
| $m$-CommonMult | $m(t+1)^2$ | $-(m-1)(t+1)\cdot\left(\lfloor\frac{t+1}{2}\rfloor\right)$ |

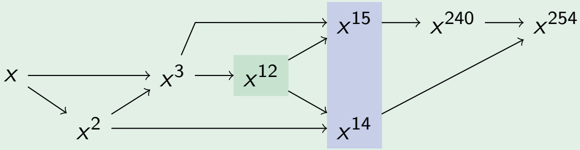
Table: Complexity Comparison of Secure Multiplications

## Security Proofs

- Security proven in the $t$-**SNI** model.
- The proof in this model ensures the security with only $t+1$ shares, instead of $2t+1$ shares in the original model.
- EasyCrypt verification tool on our AES S-box algorithm (thanks to S.Belaïd).

# Application to AES

## Possible evaluation of $x^{254}$ in $\mathbb{F}_{2^8}$



# Application to AES

## SecExp254

**Input:** A shared representation $X$ of $x$
**Output:** A shared representation Res of $x^{254} = x^{-1}$

$X_2 \leftarrow X^2$
$X \leftarrow \texttt{RefreshMask}(X)$
$X_3 \leftarrow \texttt{SecMult}(X_2, X)$
$X_{12} \leftarrow X_3^4$
$X_3 \leftarrow \texttt{RefreshMask}(X_3)$
$(X_{14}, X_{15}) \leftarrow \texttt{CommonMult}(X_{12}, X_2, X_3)$
$X_{240} \leftarrow X_{15}^{16}$
$\text{Res} \leftarrow \texttt{SecMult}(X_{240}, X_{14})$

# Performances on several S-Boxes

| | $k$ | $m$ | $N_{mult}$ | $N'_{mult}$ |
|---|---|---|---|---|
| AES | 8 | 16 | 4 | 2.8 |
| DES | 6 | 8 | 4 | 3.1 |
| PRESENT | 4 | 16 | 2 | 1.5 |
| SERPENT | 4 | 32 | 2 | 1.5 |
| CAMELLIA | 8 | 8 | 10 | 7.8 |
| CLEFIA | 8 | 8 | 10 | 7.8 |

Table: Equivalent number of multiplications $N'_{mult}$ for various block-ciphers, with $m$ $k$-bit S-Boxes.

## Conclusion

- General improvement for multiplications with $t$-**SNI** security.
- Core idea: improvements with common shared values.
  - The ratio between two multiplications and a `CommonMult` is $\frac{3}{4}$.
  - A sequence of $m$ multiplications has an equivalent cost of $\frac{3}{4}(m-1)+1$.
  - A sequence of $m$ `CommonMult` has an equivalent cost of $\frac{5}{8}(m-1)+1$.
- Implementation for AES S-Box evaluation.
- Theoretical gain for other block ciphers thanks to interpolation.