Consolidating Security Notions in Hardware Masking

CHES 2019
Lauren De Meyer, Begül Bilgin, Oscar Reparaz
PROBLEM: SIDE-CHANNEL ANALYSIS
SOLUTION: MASKING
Adversary can probe up to $d$ intermediate values

"Ideal circuit": probes are exact and instantaneous and independent

Basis for many proofs in SCA
**Masking**

- Goal: no correlation between any $d$ wires and the secret
- Split sensitive intermediates into $d + 1$ shares
- $x = x_0 \boxplus x_1 \boxplus x_2 \Rightarrow y = F(x) = y_0 \boxplus y_1 \boxplus y_2$
EXTRA PROBLEM IN HW: GLITCHES!
GLITCH-EXTENDED PROBING MODEL [RBN+15]

- $d$ probes
- Assume a glitch on combinational logic $C_i$ can reveal any of its inputs
- $\rightarrow$ Includes worst-case glitch

Glitch-extended probe
\[ I(\quad ; \quad ) = 0 \]
\[ I(\quad ;\quad ) = 0 \]

- Simple
- Versatile
  - Probing/NI/SNI
  - Different models (with/without glitches, …)
  - Any type of masking (Boolean, multiplicative, arithmetic, …)
  - Non-uniformity possible
  - Information-theoretic vs practical security
  - Leakage functions (identity, Hamming, …)
THE STORY
CHES ’18: MULTIPLICATIVE MASKING

Boolean to Multiplicative → Local Inversion → Multiplicative to Boolean

δ(x)

Randomness recycling

Not Boolean masking

How To Verify?

| Probes     | Multiplicative to Boolean: $g_1, g_2, r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{18}, r_{19}, r_{20}, r_{21}, r_{22}, r_{23}, r_{24}, r_{25}, r_{26}, r_{27}, r_{28}, r_{29}, r_{30}, r_{31}, r_{32}, r_{33}, r_{34}, r_{35}, r_{36}, r_{37}, r_{38}, r_{39}, r_{40}, r_{41}, r_{42}, r_{43}, r_{44}, r_{45}, r_{46}, r_{47}, r_{48}, r_{49}, r_{50}, r_{51}, r_{52}, r_{53}, r_{54}, r_{55}, r_{56}, r_{57}, r_{58}, r_{59}, r_{60}, r_{61}, r_{62}, r_{63}, r_{64}, r_{65}, r_{66}, r_{67}, r_{68}, r_{69}, r_{70}, r_{71}, r_{72}, r_{73}, r_{74}, r_{75}, r_{76}, r_{77}, r_{78}, r_{79}, r_{80}, r_{81}, r_{82}, r_{83}, r_{84}, r_{85}, r_{86}, r_{87}, r_{88}, r_{89}, r_{90}, r_{91}, r_{92}, r_{93}, r_{94}, r_{95}, r_{96}, r_{97}, r_{98}, r_{99}, r_{100}, r_{101}, r_{102}, r_{103}, r_{104}, r_{105}, r_{106}, r_{107}, r_{108}, r_{109}, r_{110}, r_{111}, r_{112}, r_{113}, r_{114}, r_{115}, r_{116}, r_{117}, r_{118}, r_{119}, r_{120}, r_{121}, r_{122}, r_{123}, r_{124}, r_{125}, r_{126}, r_{127}, r_{128}, r_{129}, r_{130}, r_{131}, r_{132}, r_{133}, r_{134}, r_{135}, r_{136}, r_{137}, r_{138}, r_{139}, r_{140}, r_{141}, r_{142}, r_{143}, r_{144}, r_{145}, r_{146}, r_{147}, r_{148}, r_{149}, r_{150}, r_{151}, r_{152}, r_{153}, r_{154}, r_{155}, r_{156}, r_{157}, r_{158}, r_{159}, r_{160}, r_{161}, r_{162}, r_{163}, r_{164}, r_{165}, r_{166}, r_{167}, r_{168}, r_{169}, r_{170}, r_{171}, r_{172}, r_{173}, r_{174}, r_{175}, r_{176}, r_{177}, r_{178}, r_{179}, r_{180}, r_{181}, r_{182}, r_{183}, r_{184}, r_{185}, r_{186}, r_{187}, r_{188}, r_{189}, r_{190}, r_{191}, r_{192}, r_{193}, r_{194}, r_{195}, r_{196}, r_{197}, r_{198}, r_{199}, r_{200}, r_{201}, r_{202}, r_{203}, r_{204}, r_{205}, r_{206}, r_{207}, r_{208}, r_{209}, r_{210}, r_{211}, r_{212}, r_{213}, r_{214}, r_{215}, r_{216}, r_{217}, r_{218}, r_{219}, r_{220}, r_{221}, r_{222}, r_{223}, r_{224}, r_{225}, r_{226}, r_{227}, r_{228}, r_{229}, r_{230}, r_{231}, r_{232}, r_{233}, r_{234}, r_{235}, r_{236}, r_{237}, r_{238}, r_{239}, r_{240}, r_{241}, r_{242}, r_{243}, r_{244}, r_{245}, r_{246}, r_{247}, r_{248}, r_{249}, r_{250}, r_{251}, r_{252}, r_{253}, r_{254}, r_{255}, r_{256}, r_{257}, r_{258}, r_{259}, r_{260}, r_{261}, r_{262}, r_{263}, r_{264}, r_{265}, r_{266}, r_{267}, r_{268}, r_{269}, r_{270}, r_{271}, r_{272}, r_{273}, r_{274}, r_{275}, r_{276}, r_{277}, r_{278}, r_{279}, r_{280}, r_{281}, r_{282}, r_{283}, r_{284}, r_{285}, r_{286}, r_{287}, r_{288}, r_{289}, r_{290}, r_{291}, r_{292}, r_{293}, r_{294}, r_{295}, r_{296}, r_{297}, r_{298}, r_{299}, r_{300}, r_{301}, r_{302}, r_{303}, r_{304}, r_{305}, r_{306}, r_{307}, r_{308}, r_{309}, r_{310}, r_{311}, r_{312}, r_{313}, r_{314}, r_{315}, r_{316}, r_{317}, r_{318}, r_{319}, r_{320}, r_{321}, r_{322}, r_{323}, r_{324}, r_{325}, r_{326}, r_{327}, r_{328}, r_{329}, r_{330}, r_{331}, r_{332}, r_{333}, r_{334}, r_{335}, r_{336}, r_{337}, r_{338}, r_{339}, r_{340}, r_{341}, r_{342}, r_{343}, r_{344}, r_{345}, r_{346}, r_{347}, r_{348}, r_{349}, r_{350}, r_{351}, r_{352}, r_{353}, r_{354}, r_{355}, r_{356}, r_{357}, r_{358}, r_{359}, r_{360}, r_{361}, r_{362}, r_{363}, r_{364}, r_{365}, r_{366}, r_{367}, r_{368}, r_{369}, r_{370}, r_{371}, r_{372}, r_{373}, r_{374}, r_{375}, r_{376}, r_{377}, r_{378}, r_{379}, r_{380}, r_{381}, r_{382}, r_{383}, r_{384}, r_{385}, r_{386}, r_{387}, r_{388}, r_{389}, r_{390}, r_{391}, r_{392}, r_{393}, r_{394}, r_{395}, r_{396}, r_{397}, r_{398}, r_{399}, r_{400}, r_{401}, r_{402}, r_{403}, r_{404}, r_{405}, r_{406}, r_{407}, r_{408}, r_{409}, r_{410}, r_{411}, r_{412}, r_{413}, r_{414}, r_{415}, r_{416}, r_{417}, r_{418}, r_{419}, r_{420}, r_{421}, r_{422}, r_{423}, r_{424}}

<table>
<thead>
<tr>
<th>Simulation using</th>
</tr>
</thead>
</table>
**Tool from [Rep16]**

- Simulated traces of intermediates
- Random inputs
- $\rightarrow$ TVLA (t-test) to detect flaws
- Higher orders: combine probes (e.g. centered product)
- Only for software (no glitches 😞 )

```c
u8 m1 = nonzero_rnd() & bitmask;  add_timesample(m1);
u8 b1 = crypto_input_shares[0];  add_timesample(b1);
u8 b2 = crypto_input_shares[1];  add_timesample(b2);

u8 b1m1 = mult_log(b1, m1);      add_timesample(b1m1);
u8 b2m1 = mult_log(b2, m1);      add_timesample(b2m1);
u8 m2 = b1m1 ^ b2m1;             add_timesample(m2);
```

Idea: Glitch-extended probes

- Replace regular probes with glitch-extended probes
- → TVLA to detect flaws
- Higher orders: ?
IDEA: GLITCH-EXTENDED PROBES

- Higher orders: concatenate extended probes
- $\chi^2$ test to detect flaws

```c
u8 m1 = nonzero_rnd() & bitmask; add_timesample(m1);
u8 b1 = crypto_input_shares[0]; add_timesample(b1);
u8 b2 = crypto_input_shares[1]; add_timesample(b2);

u8 b1m1 = mult_log(b1, m1); add_timesample(b1, m1);
u8 b2m1 = mult_log(b2, m1); add_timesample(b2, m1);
u8 m2 = b1m1 ^ b2m1; add_timesample(b1m1, b2m1);
```
Essentially:

$$I( R ; x ) = 0$$
PROBING SECURITY WITH/WITHOUT GLITCHES
PROBING SECURITY [GM10]

Given $d$ wires $Q = (q_1, \ldots, q_d)$

$$I(Q; x) = 0$$
GLITCH-EXTENDED PROBING SECURITY

Given $d$ wires $Q = (q_1, ..., q_d)$
with glitch-extended probes $R = (R_1, ..., R_d)$

\[ I(R; x) = 0 \]
Threshold Implementations
**Threshold Implementations [NRS11]**

- **Non-Completeness**

  $x_0 \xrightarrow{f_0} y_0$
  $x_1 \xrightarrow{f_1} y_1$
  $x_2 \xrightarrow{f_2} y_2$

- **Uniformity**

  $\forall (x_0, x_1, x_2) \text{ s.t. } x_0 \oplus x_1 \oplus x_2 = x :$
  $\Pr[(x_0, x_1, x_2)|x] = p$

Threshold Implementations [NRS11]

- Non-Completeness

\[ f_0 \quad x_0 \rightarrow y_0 \]
\[ f_1 \quad x_1 \rightarrow y_1 \]
\[ f_2 \quad x_2 \rightarrow y_2 \]

- Uniformity

\[ \forall (x_0, x_1, x_2) \text{ s.t. } x_0 \oplus x_1 \oplus x_2 = x: \]
\[ \Pr[(x_0, x_1, x_2) | x] = p \]

1-Glitch Extended Probing Security
\[ I(\mathcal{R}; x) = 0 \]

(Not sufficient for higher-order probing security [RBN+15])

### Threshold Implementations [NRS11]

<table>
<thead>
<tr>
<th></th>
<th>Non-Completeness</th>
<th>Uniformity</th>
<th>$I(R; x) = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sufficient</td>
<td>🚫</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>Necessary</td>
<td>🟢</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>Efficient</td>
<td>🟢</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>Verification</td>
<td>🟢 [ANR18]</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>Multi-variate</td>
<td>🚫</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>Knowledge</td>
<td>🚫</td>
<td>🚫</td>
<td>🟢</td>
</tr>
<tr>
<td>required</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---


(Strong) Non-Interference
(Strong) Non-Interference [BBD+16]

- Notions introduced for composable security
- More efficient verification (MaskVerif [BBF+18])
- Based on simulatability:

  \[ |S| \leq |I| + |O| \] 
  \[ |S| \leq |I| \] 
  (NI) 
  (SNI)

- Implies t-probing security


(Strong) Non-Interference [BBD+16]

- Originally without glitches
- Extended by robust probing model [FGD+18]
- Unify with mutual information framework:

![Diagram of Gadget with inputs S, J, and output O]
**STRONG NON-INTERFERENCE [BBD+16]**

- Originally without glitches
- Extended by robust probing model [FGD+18]
- Unify with mutual information framework:

\[
I((I, O); x_S \mid x_{\bar{S}}) = 0
\]
**Strong Non-Interference** [BBD+16]

- Originally without glitches
- Extended by robust probing model [FGD+18]
- Unify with mutual information framework:

  \[ I((J, O); x_S \mid x_\overline{S}) = 0 \]

  - Example: output probes & SNI: \(|S| = 0 \Rightarrow I(O; x) = 0\)
(Strong) Non-Interference [BBD+16]

• Originally without glitches
• Extended by robust probing model [FGD+18]
• Unify with mutual information framework:

\[ I((J, O); x_{\bar{S}} | x_S) = 0 \]

- Example: output probes & SNI: \(|S| = 0 \Rightarrow I(O; x) = 0\)

• Glitches? → replace probes with glitch-extended probes
EXTENDING THE MODELS
BEYOND GLITCHES

• Gap between theory and practice
  o Coupling [DEM18]
  o CPU leaks [PV17]
  o …

• Robust Probing Model [FGD+18]

• In the same framework: \( I( ; ) = 0 \)
  o New probe definitions: X-extended probes
  o Same tools!!

ADVANTAGES

\[ I \left( ; \right) = 0 \]

• Simple
  o No difference uni-variate or multi-variate
  o No knowledge required on variables
• Any type of masking (Boolean, multiplicative, arithmetic, …)
• Non-uniformity possible (low entropy masking)
• Versatile
  o Probing/NI/SNI
  o Different models (X-extended probes)
  o Information-theoretic vs practical security (noiseless TVLA)
  o Leakage functions (identity, Hamming, …)
CONCLUSION

**consolidate**  *verb*

con·sol·i·date  |  \kən-sə-lə-dāt\  
*consolidated; consolidating*

**Definition of consolidate**

*transitive verb*

1  : to join together into one whole: **UNITE**
   // consolidate several small school districts

2  : to make firm or secure: **STRENGTHEN**
   // consolidate their hold on first place
   // He **consolidated** his position as head of the political party.

Thank You