Robust Codes for Fault Attack Resistant Cryptographic Hardware*

Konrad J. Kulikowski, Mark G. Karpovsky, Alexander Taubin

Reliable Computing Laboratory
reliable.bu.edu
SideChannelAttacks.com
Boston University, USA

*this work was partially supported by a grant from OmniBaseLogic Inc.
Concurrent Error Detection

![Diagram showing concurrent error detection](image)

- **Input**
- **Original Device**
- **Predictor**
- **Extended Output**
- **Output**
- **Error**

Symbols:
- $x$
- $k$
- $v = Px$
- $r$

**Redundant Hardware**
Linear Code Limitations

• Detection depends on error multiplicity and error distributions. Cannot be predicted for an attack.
• Large differences in probabilities of detection for different classes of errors:

![Graph showing detection rates for different error multiplicities.](image)

Linear Duplication (k=r=7)
Robust Code

Probability of Missing an error $e$

\[ Q(e) = \frac{| \{ w \mid w \in C, w \oplus e \in C \} |}{|C|} = \text{Constant}, \quad e \neq 0 \]

Robust Code Construction

\[ C_v = \{ (x, v) \mid x \in GF(2^k), v = (Px)^{-1} \in GF(2^r) \} \]

Error Masking Condition

\[ (P(x \oplus e_x))^{-1} = (Px)^{-1} \oplus e_v \]
### Robust Codes (inversion based)

<table>
<thead>
<tr>
<th>Prob. of detection</th>
<th>Linear</th>
<th>Robust (r is odd)</th>
<th>Robust (r is even)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$2^k$</td>
<td>$2^{k-r}$</td>
<td>$2^{k-r}$</td>
</tr>
<tr>
<td>1</td>
<td>$2^n - 2^k$</td>
<td>$2^{n-1} + 2^{k-1} - 2^{k-r}$</td>
<td>$2^{n-1} + 2^{k-1} - 2^{k-r} + 2^k - 2^{k-r}$</td>
</tr>
<tr>
<td>$1 - 2^{-r+1}$</td>
<td>0</td>
<td>$2^{n-1} - 2^{k-1}$</td>
<td>$2^{n-1} - 2^{k-1} - 2(2^k - 2^{k-r})$</td>
</tr>
<tr>
<td>$1 - 2^{-r+2}$</td>
<td>0</td>
<td>0</td>
<td>$2^k - 2^{k-r}$</td>
</tr>
</tbody>
</table>
Robust Codes

If $\|e\|=1$

$$\max Q(e) = 0 \quad \text{for linear and Robust}$$

If $\|e\|=1$

$$\max Q(e) = \{0, 1\} \quad \text{for linear}$$

$$\max Q(e) = \{0, 2^{-k+1}, 2^{-k+2}\} \quad \text{for Robust}$$
Robust Detection

\[ V = \{(x, x) \mid x \in GF(2^k)\} \]

\[ C_v = \{(x, v) \mid x, v \in GF(2^k), v = (x)^{-1}\} \]
Robust Detection, Data Dependence

Maximum probability of missing a repeating error after $M$ messages ($k=r$)

$$\max Q(e) = 2^{-r+1}, M = 1$$

$$\max Q(e) = 2^{-r}, M = 2$$

$$\max Q(e) = 0, M = 3$$
Robust Codes

• With Robust protection it is difficult to inject errors in a encryption device (DFA attacks) which are not detected since error detection depends not only on the errors (as it does for linear codes), but also on the message (output of the device) which depends on the secret key.
Application to Hardware

Redundant Hardware needed for robustness

\[ v = (Px)^{-1} \]

Input

Original Device

Predictor

Extended Output

Output

Error

Redundant Hardware
Overhear Reduction

Division of one large inversion into \( t \) smaller inversions
Signature Splitting and Detection

$t=1$ (robust)

$t=2$

$t=4$

$t=8$ (linear)
Reduction of Overhead

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=1 (linear)</td>
<td>50%</td>
</tr>
<tr>
<td>t=4 (robust)</td>
<td>53%</td>
</tr>
<tr>
<td>t=8 (robust)</td>
<td>58%</td>
</tr>
<tr>
<td>t=16 (robust)</td>
<td>72%</td>
</tr>
<tr>
<td>t=32 (robust)</td>
<td>80%</td>
</tr>
</tbody>
</table>

Reduction of hardware overhead for a FPGA implementation of AES-128 with where k=128 r=32.

Splitting of signatures allows for a robustness/hardware tradeoff.
References


Conclusions

• The protection provided by linear error detecting codes is not uniform and is not suitable for cryptographic hardware which is susceptible to fault attacks.

• We presented a method of protection based on nonlinear systematic robust codes which can provide for uniform protection against all errors thus drastically reducing the probability that an attacker will be able to inject an undetected error.

• We also presented an optimization which allows for a tradeoff between the level of robustness and area overhead.