Automatic Synthesis of Fault Attack Resistant Cipher Implementations

Chester Rebeiro
IIT Madras
Side Channel Analysis

- Timing channels
- Power consumption
- Electro magnetic radiation
- Fault injection
Preventing Side Channel Attacks is Difficult

- Overheads
- Platform / Compiler Specific
- Programming Language Specific

Naïve implementations can have significant size or performance overheads, which may introduce further vulnerabilities.

Countermeasures need to be tuned based on:
- Device (IoT devices to server)
- Compilers (VHDL, Assembly, Java)

Cannot be implemented by your average Joe.

Automatic Synthesis of protected implementations
Block Cipher Specification Language

1. `h begin`
2. `h lookups`
3. `SBOX : { 0x63, 0x7c, 0x77, 0x7b, ·· · }`
4. `KEY : { 0x54, 0x68, 0x61, 0x74, ·· · }`
5. `h /lookups`
6. `h operations`
7. `h func`
8. `h MUL2 ( a )`
9. `h h : { a : RS ( a, 7 ) }`
10. `h n : { h : MUL ( h, 0x1b0 ) }`
11. `h m : { ( n, t ) : XOR ( n, t ) }`
12. `ret m`
13. `h /func`
14. `·· · · · · · ·`
15. `h /operations`
16. `·· · · · · · ·`
17. `h F2 i h nonlinear`
18. `h F2[1] : { F1[1] : LKUP ( F1[1], SBOX ) }`
19. `·· · · · · · ·`
20. `h F2[16] : { F1[16] : LKUP ( F1[16], SBOX ) }`
21. `·· · · · · · ·`
22. `h F3 i h linear`
24. `·· · · · · · ·`
26. `·· · · · · · ·`
27. `h F4 i h linear`
29. `·· · · · · · ·`
30. `h F4[16] : { F3[13], F3[14], F3[15], F3[16] : XOR ( XOR ( MUL2 ( F3[16], MUL3 ( F3[23] ) ) , XOR ( F3[14], F3[15] ) ) ) }`
31. `·· · · · · · ·`
32. `·· · · · · · ·`
33. `h end`

HTML like language that captures basic functionality of the cipher
- operations
- information flow

Just need one per cipher
Platform independent
Programming language independent
Synthesis Tools

**Synthesis tools**

---

**Side-Channel Evaluation & Synthesis**

---

**Side-channel secure**

---

**Executable / design**

---

**Compiler**

---

**Source Code**

(Java / Assembly / RTL)
Automatic Synthesis of Fault Attack Resistant Block Cipher Implementations
Fault Injection in Block Ciphers

Physical Fault Injection

Memory

Instructions

Logical Fault Effect

Fault Propagation

Random single byte fault injection

- Glitch in clock
- Optical glitch in power

Target device

Attack Outcome

Secret Key

sub r4, #23

cmp r3, #0

load r1, #key

out r9, r0

mul r2, r1, r4

out r9, r1

add r0, r1, r2
Differential Fault Attacks

ENCRYPTION

ciphertext

ANALYSIS

FAULT INJECTION

plaintext

ENCRIPTION

faulty ciphertext

faulty ciphertext

ENCRIPTION

ANALYSIS
A Simple AES Fault Attack

Many more stronger fault attacks are possible
Fault attackers look to solve equations in one of the following form

\[ S^{-1}(y \oplus k) \oplus S^{-1}(y' \oplus k) = d \]

where \( x \) and \( x' \) are known and \( k \) and \( k' \) are unknown.

Iterate over all possible values of \( k \) and identify those that satisfy the above equations.

The number of solutions depends on the properties of the S-box.
Central Idea in DFA

If multiple equations of this form are found, then the complexity is reduced considerably

\[ S^{-1}(y_1 \oplus k_1) \oplus S^{-1}(y'_1 \oplus k_1) = g_1(\delta) \]
\[ S^{-1}(y_2 \oplus k_2) \oplus S^{-1}(y'_2 \oplus k_2) = g_2(\delta) \]
\[ \vdots \]
\[ S^{-1}(y_n \oplus k_n) \oplus S^{-1}(y'_n \oplus k_n) = g_n(\delta) \]

where \( g_1(\delta), g_2(\delta), \ldots, g_n(\delta) \) are linear functions in

Only key tuples \((k_1, k_2, \ldots, k_n)\) that satisfy all \(n\) equations are potential candidates

Assuming \( \delta \) is a byte, we can recover \(n\) bytes of key with a complexity of \(2^8\)
**summarize**

**Inject Fault**

**Solve Equations**

Online complexity: \#faults are needed to retrieve the key

Offline complexity: Search space for finding the keys

\[
S^{-1}(y_1 \oplus k_1) \oplus S^{-1}(y_1' \oplus k_1) = g_1(\delta) \\
S^{-1}(y_2 \oplus k_2) \oplus S^{-1}(y_2' \oplus k_2) = g_2(\delta) \\
\vdots \quad \vdots \quad \vdots \quad \vdots \\
S^{-1}(y_n \oplus k_n) \oplus S^{-1}(y_n' \oplus k_n) = g_n(\delta)
\]

More the better

Involving sbox

Linear functions
### Differential Fault Attacks on AES

<table>
<thead>
<tr>
<th>Fault Injected</th>
<th>#faults (online complexity)</th>
<th>Offline complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>in first round</td>
<td>requires 128 faults</td>
<td>NIL</td>
</tr>
<tr>
<td>in last round</td>
<td>requires 128 faults</td>
<td>NIL</td>
</tr>
<tr>
<td>in 9(^{th}) round</td>
<td>requires 4 faults (each fault derives 32 bits of key)</td>
<td>256</td>
</tr>
<tr>
<td>in 8(^{th}) round</td>
<td>Requires 1 fault</td>
<td>256</td>
</tr>
<tr>
<td>all other rounds</td>
<td>Not exploitable</td>
<td>N/A</td>
</tr>
</tbody>
</table>

A majority of the faults are unexploitable

Naïve countermeasures do not consider the online or offline attack complexity requirements, thus significant overheads
SAFARI
Automatic Synthesis of Fault Attack Resistant Block Cipher Implementations
XFC: Exploitable Fault Characterization

- XFC
- Countermeasure Addition to Specification (CAS)
- Ranked list of fault locations with corresponding offline complexity
- BCS
- Synthesis
- Security
- Block Cipher Specification (BCS)
- Software Implementation
- RTL Implementation

DAC'17
Fault Propagation Phase

A Typical Block Cipher
Color the fault affected part.
Propagate and color as follows.

1. When passing through a linear layer, retain same color
2. When passing through a non-linear layer, change color
1. If two bytes of different colors are combined, change the color.

Same colors are linearly correlated
Different colors are not correlated
Key Determination Phase

\[ S^{-1}(y_1 \oplus k_1) \oplus S^{-1}(y_1' \oplus k_1) = g_1( ) \]
\[ S^{-1}(y_2 \oplus k_2) \oplus S^{-1}(y_2' \oplus k_2) = g_2( ) \]

For every possible value of determine \( k_1, k_2 \) that satisfy the equations.

The complexity is \( 2^4 \); the possible values can take.
Key Determination Phase

\[ S^{-1}(y_3 \oplus k_3) \oplus S^{-1}(y_3' \oplus k_3) = g_3(\ ) \]
\[ S^{-1}(y_4 \oplus k_4) \oplus S^{-1}(y_4' \oplus k_4) = g_4(\ ) \]

Can be used to determine \((k_3, k_4)\)
Key Determination Phase

We can thus determine keys \((k_1, k_2, k_3, k_4)\) and, based on the S-Box properties, we can computer the offline complexity.

Continue this process upwards toward the location of the fault eventually getting a result as follows:

Fault at location \(x\) gives keys \((k_1, k_2, k_3, k_4)\) with an offline complexity of
XFC: Exploitable Fault Characterization
Countermeasure Addition to Specification

Block Cipher Specification (BCS) → Countermeasure Addition to Specification (CAS) → Synthesis

Synthesis → XFC

Ranked list of fault locations with corresponding offline complexity

Security → Countermeasure Addition to Specification (CAS) → BCS

BCS → XFC

Software Implementation → RTL Implementation
Countermeasure Addition to Specification

Based on the functions of BCS analyse countermeasures.

List of Vulnerable Locations

Choose countermeasure with performance overhead

Countermeasures with security and performance overheads

For each type of countermeasure, evaluate security achieved, and performance overheads.
## Security Analysis

If the probability of injecting a fault $f$ is $p_f$, then the probability with which an attack is successful ($p_s$) is defined as:

$$p_s = \prod_{i=1}^{55} p_f = p_f^{55}.$$ 

## Performance Overhead

- Overhead is estimated in terms of time and area from $BCS^\ast$.
- For Software synthesis, number of extra clock cycles required is calculated.
- If the target device is known then we know the number of clock cycles taken by each of the primitive operations. Hence total time can be estimated.
- For Hardware synthesis overhead estimated through number of extra gates used.
## Countermeasure Evaluation

### For Software

<table>
<thead>
<tr>
<th>Countermeasures</th>
<th>Security</th>
<th>Function</th>
<th>Memory Overhead (in bytes)</th>
<th>$T_e$</th>
<th>Memory → $T_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy</td>
<td>1/255</td>
<td>ARK</td>
<td>64</td>
<td>96</td>
<td>6144</td>
</tr>
<tr>
<td></td>
<td>1/255</td>
<td>SB</td>
<td>256</td>
<td>64</td>
<td>16384</td>
</tr>
<tr>
<td></td>
<td>1/255</td>
<td>MC</td>
<td>-</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>1 Bit Parity</td>
<td>1/2</td>
<td>ARK</td>
<td>-</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>1/2</td>
<td>SB</td>
<td>32</td>
<td>256</td>
<td>8192</td>
</tr>
<tr>
<td></td>
<td>63/255</td>
<td>MC</td>
<td>-</td>
<td>448</td>
<td>448</td>
</tr>
<tr>
<td>2 Bit Parity</td>
<td>1/4</td>
<td>ARK</td>
<td>-</td>
<td>288</td>
<td>228</td>
</tr>
<tr>
<td></td>
<td>1/4</td>
<td>SB</td>
<td>64</td>
<td>272</td>
<td>17409</td>
</tr>
<tr>
<td></td>
<td>15/255</td>
<td>MC</td>
<td>-</td>
<td>960</td>
<td>960</td>
</tr>
<tr>
<td>4 Bit Parity</td>
<td>1/16</td>
<td>ARK</td>
<td>-</td>
<td>192</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>1/16</td>
<td>SB</td>
<td>128</td>
<td>224</td>
<td>28672</td>
</tr>
<tr>
<td></td>
<td>1/255</td>
<td>MC</td>
<td>-</td>
<td>976</td>
<td>976</td>
</tr>
</tbody>
</table>
## Countermeasure Evaluation

### For Hardware

<table>
<thead>
<tr>
<th>Countermeasure</th>
<th>Security</th>
<th>Function</th>
<th>XOR</th>
<th>AND</th>
<th>OR</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Redundancy</strong></td>
<td>1/ 255</td>
<td>ARK</td>
<td>256</td>
<td>0</td>
<td>128</td>
<td>16, 8 Bit Register</td>
</tr>
<tr>
<td></td>
<td>1/ 255</td>
<td>SB</td>
<td>128</td>
<td>0</td>
<td>128</td>
<td>8 Bit 256 → 1 MUX</td>
</tr>
<tr>
<td></td>
<td>1/ 255</td>
<td>MC</td>
<td>768</td>
<td>0</td>
<td>128</td>
<td>-</td>
</tr>
<tr>
<td><strong>1 Bit Parity</strong></td>
<td>1/ 2</td>
<td>ARC</td>
<td>128</td>
<td>16</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1/ 2</td>
<td>SB</td>
<td>128</td>
<td>16</td>
<td>15</td>
<td>8 Bit 16 → 1 MUX</td>
</tr>
<tr>
<td></td>
<td>63/ 255</td>
<td>MC</td>
<td>176</td>
<td>16</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td><strong>2 Bit Parity</strong></td>
<td>1/ 4</td>
<td>ARC</td>
<td>128</td>
<td>32</td>
<td>31</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1/ 4</td>
<td>SB</td>
<td>128</td>
<td>32</td>
<td>31</td>
<td>8 Bit 32 → 1 MUX</td>
</tr>
<tr>
<td></td>
<td>15/ 255</td>
<td>MC</td>
<td>432</td>
<td>32</td>
<td>31</td>
<td>-</td>
</tr>
<tr>
<td><strong>4 Bit Parity</strong></td>
<td>1/ 16</td>
<td>ARK</td>
<td>128</td>
<td>64</td>
<td>63</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1/ 26</td>
<td>SB</td>
<td>128</td>
<td>64</td>
<td>63</td>
<td>8 Bit 64 → 1 MUX</td>
</tr>
<tr>
<td></td>
<td>1/ 255</td>
<td>MC</td>
<td>608</td>
<td>64</td>
<td>63</td>
<td>-</td>
</tr>
</tbody>
</table>
Countermeasure Addition to Specification

Based on the functions of BCS analyse countermeasures

List of Vulnerable Locations

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Filter locations based on Security

filtered locations

Based on the functions of BCS analyse countermeasures with security and performance overheads

Security

XFC

List of Vulnerable Locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations

Security

BCS

Choose countermeasure with given security and minimum performance overhead

Apply chosen countermeasures to the filtered locations
Modified Block Cipher Specification

\[
\begin{align*}
&h F_2 i \ h \ \text{nonlinear} \ i \ h \ \text{SUBBYTE} \ i \ h \\
&h F_2[1] : \{ \ F_1[1] : \text{LKUP} ( F_1[1], \text{SBOX} ) \} i \\
&h F_2[2] : \{ \ F_1[2] : \text{LKUP} ( F_1[2], \text{SBOX} ) \} i \\
&\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \qu...
Synthesis

Block Cipher Specification (BCS) → Countermeasure Addition to Specification (CAS) → Synthesis

Synthesis → XFC

XFC → Ranked list of fault locations with corresponding offline complexity

Ranked list of fault locations with corresponding offline complexity → Software Implementation

Software Implementation → RTL Implementation

RTL Implementation → Synthesis

Synthesis → Security

Security → Block Cipher Specification (BCS)
```c
uint8_t F1[16], F2[16], F2d[16];
uint8_t SBOX[256] = {S1, S2, ··· , S256};

F2[0] = SBOX[F1[0]];
F2[1] = SBOX[F1[1]];
·· · · ·
F2[15] = SBOX[F1[15]];

F2d[0] = SBOX[F1[0]];
F2d[1] = SBOX[F1[1]];
·· · · ·
F2d[15] = SBOX[F1[15]];

if (F2[0] == F2d[0] && ··· && F2[15] == F2d[15])
{
    continue;
}
else
{
    exit(0);
}
```

```verilog
wire[127 : 0] F1, F2, F2d;
wire[15 : 0] F2c;
SubByte SB1 (F1[127 : 120], F2[127 : 120]);
SubByte SB2 (F1[119 : 112], F2[119 : 112]);
·· · · ·
SubByte SB16 (F1[7 : 0], F2[7 : 0]);
SubByte DSB1 (F1[127 : 120], F2d[127 : 120]);
SubByte DSB2 (F1[119 : 112], F2d[119 : 112]);
·· · · ·
SubByte DSB16 (F1[7 : 0], F2d[7 : 0]);

F2c[15] = ECMP(F2[127 : 120], F2d[127 : 120]);
F2c[14] = ECMP(F2[119 : 112], F2d[119 : 112]);
·· · · ·
F2c[0] = ECMP(F2[7 : 0], F2d[7 : 0]);
```
Results (Software)

Lower is better
Results (Hardware)

Lower is better
Conclusions

• First attempt to automatically synthesize fault attack resistant block cipher implementations
  – User configurable security
  – Reduced overheads
  – any block cipher, any device, any programming language

• Covers a wide range of ciphers and countermeasures
• Block Cipher Specification Language

• Limitations:
  – In software especially, the generated code is not very efficient for high performance processors.