Side Channel Analysis of the SHA-3 Finalists

Michael Zohner, Michael Kasper, and Marc Stöttinger

{michael.zohner|michael.kasper|marc.stoettinger}@cased.de
Side Channel Analysis - Power Analysis

- Power Analysis is based on the dependency of the power consumption on the processed data
Differential Power Analysis (DPA)

Device processes $m \oplus \text{key}$

$\text{HW}(4 \oplus \text{key}) < \text{HW}(6 \oplus \text{key}) < \text{HW}(2 \oplus \text{key})$

$\Rightarrow \text{key} = 5$
Profiling Based Attacks

- First phase: profile the power consumption on a fully controllable device

- Second phase: compare profiles to power consumption of attacked device
Side Channel Attacks on MAC Functions
Side Channel Attacks on MAC Functions
Side Channel Attacks on MAC Functions
### Side Channel Attacks against the SHA-3 Finalists

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Attack Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE</td>
<td>MAC Forgery</td>
</tr>
<tr>
<td>Grøstl</td>
<td>MAC Forgery</td>
</tr>
<tr>
<td>JH</td>
<td>-</td>
</tr>
<tr>
<td>Keccak</td>
<td>-</td>
</tr>
<tr>
<td>Skein</td>
<td>-</td>
</tr>
</tbody>
</table>

Benoît et al. (DPA)
Background for this Work

- We use the same power consumption model as Benoît et al., namely the Hamming weight model

- We analyzed:
  - Grøstl-MAC (Envelope MAC)
  - JH-HMAC
  - Keccak-MAC (built in MAC function)
  - Skein-MAC (built in MAC function)

- The attacks were verified on:
  - ATMega 256-1 microcontroller (8 bit register)
  - AVR Cortex M3 (32 bit register)
Analysis of Grøstl

- Grøstl-MAC computes a MAC by hashing $\overline{K} \| \overline{M} \| K$
- The attack, suggested by Benoît et al., can be altered to fit Grøstl-MAC
- A successful DPA is able to recover the processed key, since the last key $K$ is processed with variable data
Two state values are needed for inner and outer hash function call

For each state value, two operations have to be exploited
Analysis of Keccak (1)

- Keccak-MAC hashes \((K \parallel M)\)
- First exploit the XOR between the bitrate and the message

The Sponge Construction based on a permutation \(f\)
Secondly exploit the XOR of the columns during θ until all values are known.

If the key is only few bits long, a key recovery is possible.
Analysis of Keccak (2)

- Secondly exploit the XOR of the columns during θ until all values are known.
- If the key is only few bits long, a key recovery is possible.
Secondly exploit the XOR of the columns during θ until all values are known.

If the key is only few bits long, a key recovery is possible.
• Secondly exploit the XOR of the columns during θ until all values are known.
• If the key is only few bits long, a key recovery is possible.
Secondly exploit the XOR of the columns during θ until all values are known.

If the key is only few bits long, a key recovery is possible.
Analysis of Skein

- Target the modular addition between the state value and the message
- Recover the key by dividing each 64 bit addition in eight 8 bit additions and attack them independently

Attacked UBI call

Split the 64 bit modular addition into 8 bit blocks and attack them independently
## Side Channel Attacks against the SHA-3 Finalists

<table>
<thead>
<tr>
<th></th>
<th>Benoît et al. (DPA)</th>
<th>This work (DPA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE</td>
<td>MAC Forgery</td>
<td>-</td>
</tr>
<tr>
<td>Grøstl</td>
<td>MAC Forgery</td>
<td>Key Recovery</td>
</tr>
<tr>
<td>JH</td>
<td>-</td>
<td>MAC Forgery</td>
</tr>
<tr>
<td>Keccak</td>
<td>-</td>
<td>MAC Forgery (Key Recovery)</td>
</tr>
<tr>
<td>Skein</td>
<td>-</td>
<td>MAC Forgery</td>
</tr>
</tbody>
</table>
Analysis of Grøstl

- Use algebraic side-channel analysis to recover the hashed message

1) Determine Hamming weights

- \( HW(m) = 4 \)
- \( HW(S(m)) = 3 \)
- \( HW(MX(m)) = 6 \)

2) Solve Equations

- \( m = 232 \)
1) Determine Hamming weight (1)

- Support Vector Machines (SVM) are used for binary classification
1) Determine Hamming weight (2)

- Profiling Hamming weights using Support Vector Machines
2) Solve Equations

- The variables in the equation system are composed of:
  - HW of the input
  - HW of the S-box input
  - HW of the S-box output
  - HW of the MixBytes output

- Inserting the HW of these variables for the first two rounds (200 HW) allows solving the system
## Side Channel Attacks against the SHA-3 Finalists

<table>
<thead>
<tr>
<th></th>
<th>Benoît et al. (DPA)</th>
<th>This work (DPA)</th>
<th>This work (Profiling)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAKE</td>
<td>MAC Forgery</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Grøstl</td>
<td>MAC Forgery</td>
<td>Key Recovery</td>
<td>Message Recovery</td>
</tr>
<tr>
<td>JH</td>
<td>-</td>
<td>MAC Forgery</td>
<td>-</td>
</tr>
<tr>
<td>Keccak</td>
<td>-</td>
<td>MAC Forgery (Key Recovery)</td>
<td>-</td>
</tr>
<tr>
<td>Skein</td>
<td>-</td>
<td>MAC Forgery</td>
<td>-</td>
</tr>
</tbody>
</table>
Remarks

- The side channel analysis was performed for the Hamming weight leakage model, an analysis using a more complex model, such as the Hamming distance model, is more difficult.

- Ranking the finalists in terms of side channel resistance is not possible since different implementations have different characteristics.

- A feasibility study of the algebraic side channel attack for all finalists still remains.
Questions