Exploring the Resilience of Some Lightweight Ciphers Against Profiled Single Trace Attacks

Valentina Banciu Elisabeth Oswald Carolyn Whitnall

University of Bristol, Department of Computer Science
Merchant Venturers Building, Woodland Road, BS8 1UB, Bristol, UK
{valentina.banciu, elisabeth.oswald, carolyn.whitnall}@bristol.ac.uk

COSADE 2015

Contents

1. Overview of Single Trace Attacks (STA)
2. Overview of Ciphers
3. Implementation and Results

Overview of STA

STA are profiled attacks aimed at key recovery using a single trace.

STA consist of two phases:

1. extracting the side-channel information from traces (i.e., profiling)
2. exploiting the available leakage in order to recover the secret key

In this talk, we focus on the *exploitation* phase.

Overview of STA

STA attacks:

1. involve directly interpreting power consumption measurements
2. exploit key-dependent differences (patterns) within a trace

General assumptions:

1. precise knowledge about the targeted implementation
2. (identical) training device available

'Classification' of STA attacks:

1. Enumeration-based attacks
2. Solver-aided attacks: ASCA, TASCA, Gröbner basis

In this talk, we focus on *enumeration-based attacks*.

Overview of STA

STA attacks crucially rely on the ability to extract side-channel information from traces.

Overview of STA

STA attacks crucially rely on the ability to extract side-channel information from traces.

Assume the 8-bit Hamming weight (HW) leakage model.

Overview of STA

STA attacks crucially rely on the ability to extract side-channel information from traces.

Assume the 8-bit Hamming weight (HW) leakage model.

1. If $w = \mathrm{HW}(v)$ (the HW of an intermediate value) is known, $|\mathrm{PossibleValues}(v)| = \binom{8}{w}$

Overview of STA

STA attacks crucially rely on the ability to extract side-channel information from traces.

Assume the 8-bit Hamming weight (HW) leakage model.

1. If $w = \mathrm{HW}(v)$ (the HW of an intermediate value) is known, $|\mathrm{PossibleValues}(v)| = \binom{8}{w}$
2. In practice, $w \in S = \{w_1, w_2, \ldots w_s\}$ (uncertainty about measurements due to noise) and thus $|\mathrm{PossibleValues}(v)| = \sum_i \binom{8}{w_i}$

Overview of STA

STA attacks crucially rely on the ability to extract side-channel information from traces.

Assume the 8-bit Hamming weight (HW) leakage model.

1. If $w = \mathrm{HW}(v)$ (the HW of an intermediate value) is known, $|\mathrm{PossibleValues}(v)| = \binom{8}{w}$
2. In practice, $w \in S = \{w_1, w_2, \ldots w_s\}$ (uncertainty about measurements due to noise) and thus $|\mathrm{PossibleValues}(v)| = \sum_i \binom{8}{w_i}$

In a nutshell:

1. STA attacks target multiple intermediate values (i.e., subkeys)
2. leakage corresponding to each intermediate value is represented as a set (currently: $|S| = 5$)
3. the attack closely follows the encryption function

Motivation

Why these ciphers?

Motivation

Why these ciphers?

1. AES and PRESENT have been standardized
   KLEIN and LED share features with AES, respectively PRESENT
2. Publicly available 8-bit implementations:
   1. `http://perso.uclouvain.be/fstandae/lightweight_ciphers/`
   2. `http://led.crypto.sg/software`

Motivation

Why these ciphers?

1. AES and PRESENT have been standardized
   KLEIN and LED share features with AES, respectively PRESENT
2. Publicly available 8-bit implementations:
   1. http://perso.uclouvain.be/fstandae/lightweight_ciphers/
   2. http://led.crypto.sg/software

Why STA?

Motivation

Why these ciphers?

1. AES and PRESENT have been standardized
   KLEIN and LED share features with AES, respectively PRESENT
2. Publicly available 8-bit implementations:
   1. `http://perso.uclouvain.be/fstandae/lightweight_ciphers/`
   2. `http://led.crypto.sg/software`

Why STA?

1. realistic attack scenario
2. robust attacks w.r.t. noise tolerance

Overview of Ciphers

Table: Overview of cipher characteristics

	Key size	**Block size**	**# rounds**	**Existing key schedule?**
AES	128	128	11	yes
KLEIN	64	64	12	yes
PRESENT	80	64	32	yes
LED	64	64	8	no

Overview of Ciphers

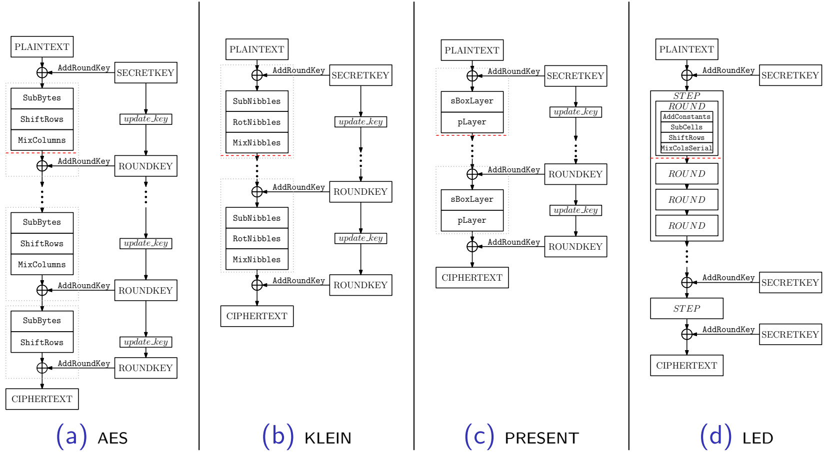

Figure: Overview of encryption algorithms

Overview of Ciphers

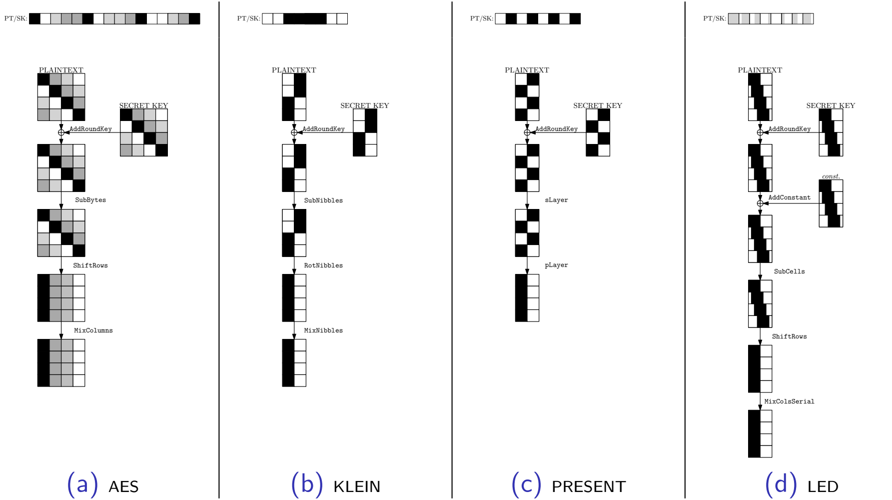

Figure: The first encryption round. The byte mixing layer acts on a 4-byte ‘block’

Assessing the Vulnerability to STA

Because of the diffusion properties of the byte-mixing layer, enumeration-based STA attacks target only the first encryption round.

Assessing the Vulnerability to STA

Because of the diffusion properties of the byte-mixing layer, enumeration-based STA attacks target only the first encryption round.

The key addition and substitution layers can be attacked using pre-computed lookup tables, i.e. 1-byte subkeys.
Let $ByteSet_i, i = 1 \ldots 4$ be the key candidates that match the key addition and substitution leaks of a 'block'.

Assessing the Vulnerability to STA

Because of the diffusion properties of the byte-mixing layer, enumeration-based STA attacks target only the first encryption round.

The key addition and substitution layers can be attacked using pre-computed lookup tables, i.e. 1-byte subkeys.
Let $ByteSet_i, i = 1 \ldots 4$ be the key candidates that match the key addition and substitution leaks of a ‘block’.

STA attacks boil down to targeting the byte mixing layer, i.e. 4-byte subkeys.
We have generated 100 plaintext and secret key pairs and simulated encryption and leakage using the cipher suite.
The reported results are averaged out over this set.

Assessing the Vulnerability to STA

Algorithm 1 `MixColumns` (used by AES and KLEIN)

Input: in_1, in_2, in_3, in_4

Output: $out_1, out_2, out_3, out_4$

$Tmp \leftarrow in_1 \oplus in_2 \oplus in_3 \oplus in_4$;

for $i = 1 \rightarrow 4$ **do**

  $Tm \leftarrow in_i \oplus in_{i+1}$;

  $Tm \leftarrow \texttt{xtime}(Tm)$;

  $out_i \leftarrow in_i \oplus Tm \oplus Tmp$;

end for

Assessing the Vulnerability to STA

Algorithm 1 `MixColumns` (used by AES and KLEIN)

Input: in_1, in_2, in_3, in_4
Output: $out_1, out_2, out_3, out_4$

$Tmp \leftarrow in_1 \oplus in_2 \oplus in_3 \oplus in_4$;
for $i = 1 \rightarrow 4$ **do**
  $Tm \leftarrow in_i \oplus in_{i+1}$;
  $Tm \leftarrow \texttt{xtime}(Tm)$;
  $out_i \leftarrow in_i \oplus Tm \oplus Tmp$;
end for

`MixColumns` will leak at most **17** intermediate values

Assessing the Vulnerability to STA

`pLayer` will leak at most **12** intermediate values
`MixColumnsSerial` will leak at most **32** intermediate values

Assessing the Vulnerability to STA

`pLayer` will leak at most **12** intermediate values
`MixColumnsSerial` will leak at most **32** intermediate values

Table: Size of the attack surface (i.e., number of leaked intermediate values) corresponding to the diffusion layer

	AES	**KLEIN**	**PRESENT**	**LED**
'Basic'	4	4	4	4
'Maximum'	21	21	12	32

Attacking the Encryption Round

Algorithm 2 Previous attack strategy.

$\mathit{ReducedKeySpace} = \emptyset$
for all $K_1 \in \mathit{ByteSet}_1$ **do**
    for all $K_2 \in \mathit{ByteSet}_2$ **do**
        for all $K_3 \in \mathit{ByteSet}_3$ **do**
            for all $K_4 \in \mathit{ByteSet}_4$ **do**
                if $[K_1, K_2, K_3, K_4]$ matches the byte mixing leaks **then**
                    append $[K_1, K_2, K_3, K_4]$ to $\mathit{ReducedKeySpace}$
                end if
            end for
        end for
    end for
end for
return $\mathit{ReducedKeySpace}$

Attacking the Encryption Round

Algorithm 2 Current attack strategy.

1: $\mathit{ReducedKeySpace} = \mathit{ByteSet}_1 \times \mathit{ByteSet}_2 \times \mathit{ByteSet}_3 \times \mathit{ByteSet}_4$
2: filter out ‘rows’ that do not match the byte mixing leaks
3: return *ReducedKeySpace*

Attacking the Encryption Round

Why is the current attack strategy better?

Attacking the Encryption Round

Why is the current attack strategy better?

1. running time: under 5 minutes
2. success rate: 100%

Attacking the Encryption Round: Results

Table: Reduced key space when targeting the encryption function

Cipher \ Setsize	HW model 1	2	3	4	5	HD model 1	2	3	4	5
AES	3	2^{10}	2^{20}	2^{23}	2^{25}	30	2^{15}	2^{22}	2^{25}	2^{26}
KLEIN	3	2^{9}	2^{12}	2^{18}	2^{23}	90	2^{15}	2^{22}	2^{24}	2^{26}
PRESENT	23	2^{11}	2^{19}	2^{23}	2^{25}	60	2^{15}	2^{22}	2^{24}	2^{25}
LED	2	2^{10}	2^{18}	2^{21}	2^{24}	35	2^{16}	2^{21}	2^{23}	2^{25}

(a) Targeting the 'basic' attack surface

Attacking the Encryption Round: Results

Table: Reduced key space when targeting the encryption function

	HW model					HD model				
Cipher \ **Setsize**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
AES	1	1	2^{10}	2^{18}	2^{24}	1	1	2^{12}	2^{19}	2^{24}
KLEIN	1	1	2^{7}	2^{12}	2^{20}	1	1	2^{9}	2^{14}	2^{21}
PRESENT	1	1	2^{8}	2^{12}	2^{20}	1	1	2^{10}	2^{13}	2^{22}
LED	1	1	2^{5}	2^{11}	2^{19}	1	1	2^{7}	2^{13}	2^{20}

(a) Targeting the ‘maximum’ attack surface

Attacking the Encryption Round

The size of the reduced subkey space depends on:

1. the set size
2. the number of statistically independent intermediates

and less so on the specific cipher particularities.

Attacking the Key Expansion

The key expansion algorithms are substantially different w.r.t. their diffusion properties.

1. AES: target $1 \ldots 5$ consecutive round keys
2. KLEIN: target $1 \ldots 12$ (i.e., all) consecutive round keys
3. PRESENT: target 32 (i.e., all) round keys (minimal differences between round keys)
4. LED: no key expansion, uses the same key for all rounds

Attacking the Key Expansion

The key expansion algorithms are substantially different w.r.t. their diffusion properties.

1. AES: target $1 \dots 5$ consecutive round keys
2. KLEIN: target $1 \dots 12$ (i.e., all) consecutive round keys
3. PRESENT: target 32 (i.e., all) round keys (minimal differences between round keys)
4. LED: no key expansion, uses the same key for all rounds

Here, we only present the methodology for the KLEIN key expansion attack.
We report results for all ciphers. We are targeting the full key and no longer a 4-byte 'block'.

Attacking the Key Expansion

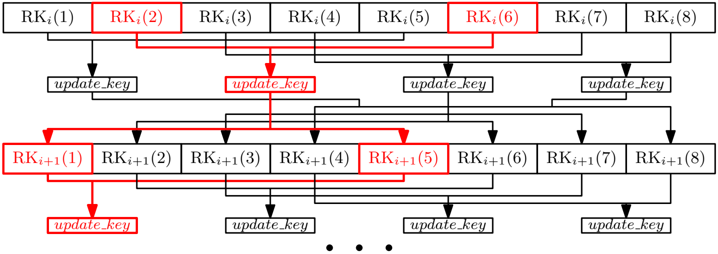

Figure: Targeting the KLEIN key expansion

It is possible to target 2-byte subkeys and use leakages from as many rounds as available.

Attacking the Key Expansion: Results

Table: Reduced key space when targeting the key expansion

	HW model					HD model				
Setsize / **# RK**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
1	2^{58}	2^{74}	2^{95}	2^{106}	2^{115}	2^{60}	2^{75}	2^{99}	2^{107}	2^{118}
5	10	2^{15}	2^{35}	2^{58}	n.a.	30	2^{17}	2^{37}	2^{55}	n.a.

(a) AES (128-bit key)

Attacking the Key Expansion: Results

Table: Reduced key space when targeting the key expansion

	HW model					HD model				
# RK \ Setsize	1	2	3	4	5	1	2	3	4	5
1	2^{35}	2^{45}	2^{50}	2^{57}	2^{60}	2^{40}	2^{48}	2^{55}	2^{57}	2^{61}
6	2^{8}	2^{15}	2^{35}	2^{45}	2^{55}	2^{12}	2^{21}	2^{37}	2^{49}	2^{57}
12	1	2^{4}	2^{20}	2^{32}	2^{45}	1	2^{4}	2^{22}	2^{37}	2^{50}

(a) KLEIN (64-bit key)

Attacking the Key Expansion: Results

Table: Reduced key space when targeting the key expansion

	HW model					HD model				
# RK \ **Setsize**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
31	2^{10}	2^{16}	2^{45}	2^{60}	2^{73}	2^{14}	2^{16}	2^{45}	2^{60}	2^{73}

(a) PRESENT (80-bit key)

Attacking the Key Expansion

The attack outcome is influenced by:

1. the set size
2. the number of statistically independent intermediates
3. the diffusion rate

Conclusion

1. We have compared various ciphers w.r.t. their vulnerability against profiled single trace attacks.
2. We found that mainly two factors influence the attack success:
    1. the diffusion properties of a cipher
    2. the number of intermediate values occur in a concrete implementation (i.e., the attack surface)
3. Furthermore, particularly light key schedule algorithms are 'easy' targets

Conclusion

1. We have compared various ciphers w.r.t. their vulnerability against profiled single trace attacks.
2. We found that mainly two factors influence the attack success:
    1. the diffusion properties of a cipher
    2. the number of intermediate values occur in a concrete implementation (i.e., the attack surface)
3. Furthermore, particularly light key schedule algorithms are ‘easy’ targets

Thank you for your attention!

Related Work

Attacks on the AES encryption round

1. Valentina Banciu and Elisabeth Oswald. **Pragmatism vs. Elegance: Comparing Two Approaches to Simple Power Attacks on AES**. In COSADE, pages 29–40. Springer, 2014.
2. Shize Guo, Xinjie Zhao, Fan Zhang, Tao Wang, Zhijie Jerry Shi, François-Xavier Standaert, and Chujiao Ma. **Exploiting the Incomplete Diffusion Feature: A Specialized Analytical Side-Channel Attack Against the AES and Its Application to Microcontroller Implementations**. IEEE Transactions on Information Forensics and Security, 9(6):999–1014, 2014.

Related Work

Attacks on the AES key schedule

1. Stefan Mangard. **A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion**. In ICISC 2002, pages 343–358. Springer, 2003.
2. Joel VanLaven, Mark Brehob, and Kevin J Compton. **A Computationally Feasible SPA Attack on AES via Optimized Search**. In Security and Privacy in the Age of Ubiquitous Computing, pages 577–588. Springer, 2005.

Related Work

Side-channel information extraction for STA

1. Valentina Banciu, Elisabeth Oswald, and Carolyn Whitnall. **Reliable Information Extraction for Single Trace Attacks**. IACR ePrint Archive, 2015:45, 2015.