TU Graz

Towards More Practical Time-Driven Cache Attacks

Raphael Spreitzer (Graz University of Technology, Austria)
Benoît Gérard (DGA-MI, France)

Heraklion, 30th June 2014

Introduction to Cache Attacks

- Advanced Encryption Standard (AES)
  - Four round transformations
  - Software implementations employ T-tables
  - $\mathbf{T}[\mathbf{s}_i = \mathbf{p}_i \oplus \mathbf{k}_i]$
- CPU caches
  - Data cannot be accessed in constant time
- $\Rightarrow$ Cache attacks exploit these timing variations

Bernstein's Cache-Timing Attack [Ber05]

- Study phase
  - Encrypt $\mathbf{P}$ under a known key $\mathbf{K}$
- Attack phase
  - Encrypt $\widetilde{\mathbf{P}}$ under an unknown key $\widetilde{\mathbf{K}}$
- Correlation phase
  - Similar timing profile if pairs satisfy

$$\widetilde{\mathbf{p}}_i \oplus \widetilde{\mathbf{k}}_i = \mathbf{p}_i \oplus \mathbf{k}_i$$

$$\widetilde{\mathbf{k}}_i = \mathbf{p}_i \oplus \mathbf{k}_i \oplus \widetilde{\mathbf{p}}_i$$

- Key-search phase

Recent Investigations and Motivation

- ARM processors: still $\sim$ 60 bits to be search exhaustively [WHS12, SP13]
- How to improve the attack?
- Divide and conquer
  - Divide: gather leaking information
  - Conquer: exploit the gathered information
- Improve the attack by focusing on both phases

Divide Part

Study phase & attack phase

1) Attacking different key-chunk sizes

2) Minimum timing information [AE13]

Attacking Different Key-Chunk Sizes (1/2)

Bernstein attacked single bytes

- Attack 1-byte chunks: $n_{kc} = 16$, $s_{kc} = 256$

Different key-chunk sizes

- Attack 4-bit chunks: $n_{kc} = 32$, $s_{kc} = 16$
- Attack 2-byte chunks: $n_{kc} = 8$, $s_{kc} = 256^2$
- Attacking larger key chunks should reduce the noise

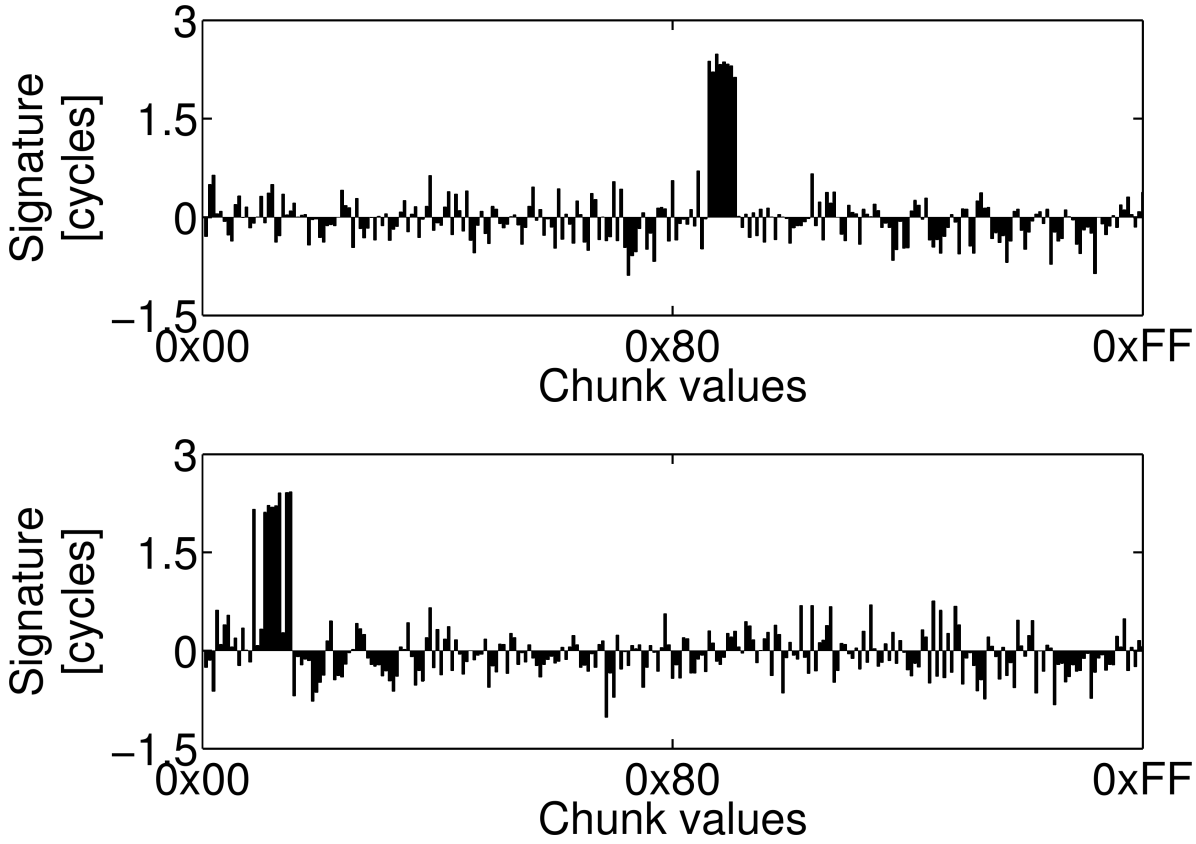
Attacking Different Key-Chunk Sizes (2/2)

Pitfalls?

- Memory requirements (8-byte elements $\mathbf{t}[n_{kc}][s_{kc}]$)
  - Attacking 1-byte chunks: 32 KB
  - Attacking 2-byte chunks: 4 MB
  - Attacking 4-byte chunks: 128 GB
- Number of measurement samples
  - Let $N = 2^{28}$ be the number of encrypted plaintexts
  - Each possible value b of a specific chunk is encrypted $\sim \frac{N}{s_{kc}}$
  - 1-byte chunks: $\sim 10^6$
  - 2-byte chunks: 4096

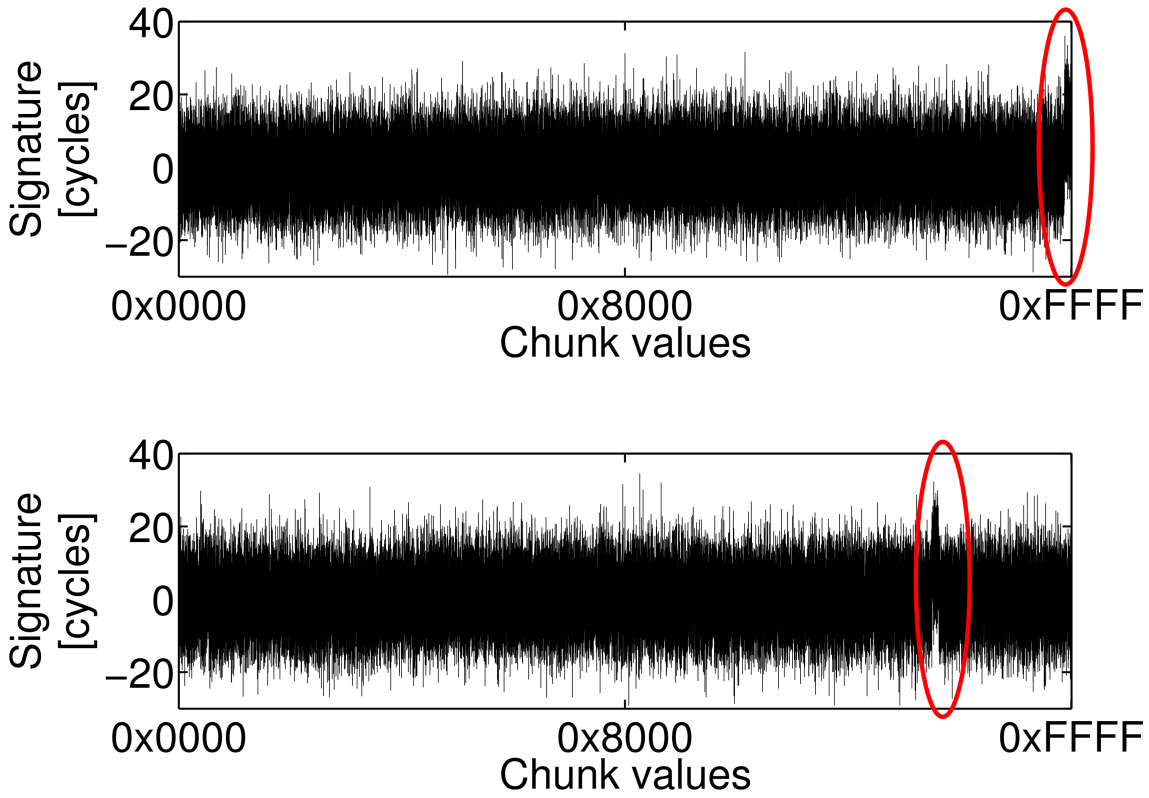
Practical Results (1/2)

- Attacking 1-byte chunks on Samsung Galaxy SII



Practical Results (2/2)

- Attacking 2-byte chunks on Samsung Galaxy SII



Divide Part

Study phase & attack phase

1) Attacking different key-chunk sizes

2) Minimum timing information [AE13]

Minimum Timing Information [AE13]

- Gather minimum encryption time
- Only noise increases the encryption time
- Improvement on Pentium processors

Our observations

- $\Rightarrow$ cache misses also increase the encryption time
- Misses potential useful information
- Does not work for ARM processors

Conquer part

Correlation phase & key-search phase

How to recover the full key?

Recovering the Full Key from Sub Keys

Threshold Approach [Ber05]

- Fix threshold
- Consider potential key bytes above this threshold
- Iterate over all sets of sub keys
- Complexity determined by product of cardinalities
- Disadvantages
  - Key might not be found
  - Ordering of sub keys is not exploited

Recovering the Full Key from Sub Keys

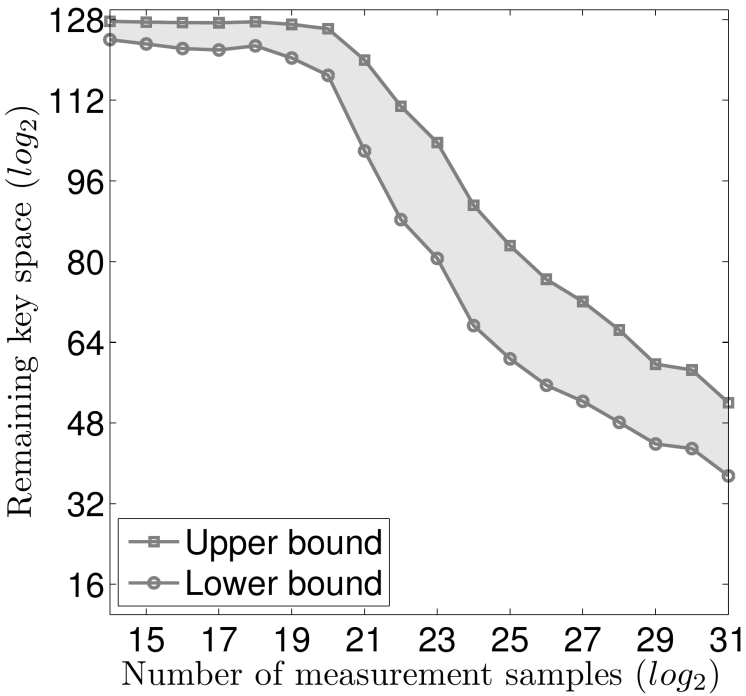
Optimal Key-Enumeration Approach [VCGRS12]

- Combination function to compute “global score”
- Test full keys in decreasing order of the score
- Improvement?

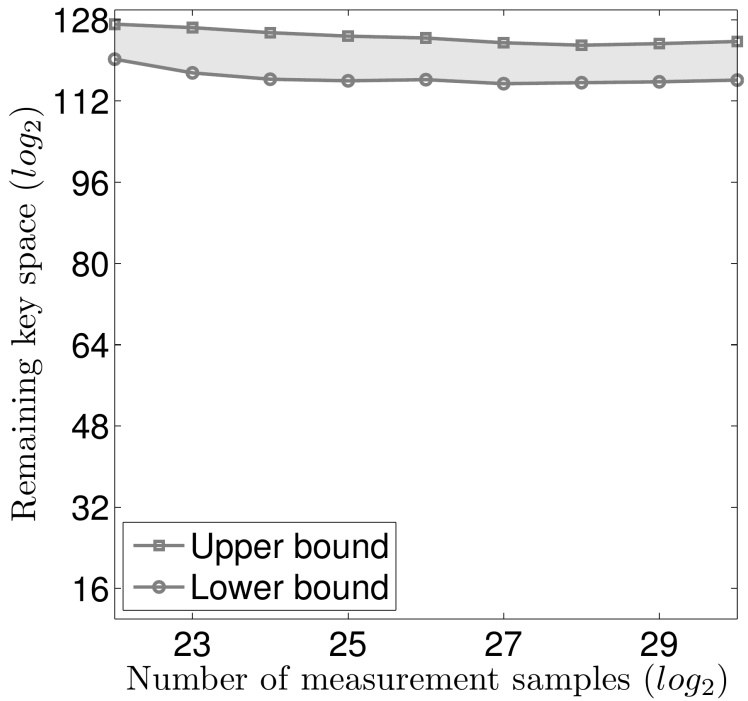
Run	Threshold	Optimal enumeration
1	64 bits	36.6 – 44.9 bits
2	74 bits	36.5 – 45.6 bits

Practical Results

Rank evolution



1-byte chunks



2-byte chunks

Conclusion

- Investigated potential improvements
  - Divide part
    - Best choice on mobile devices: attack 1-byte chunks
    - Minimum encryption time does not work
  - Conquer part
    - Optimal key-enumeration algorithm
- $\Rightarrow$ optimal key-enumeration algorithm significantly reduces the key-search complexity

Bibliography I

[AE13] Hassan Aly and Mohammed ElGayyar.

Attacking AES Using Bernstein's Attack on Modern Processors.

In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors, *AFRICACRYPT*, volume 7918 of *Lecture Notes in Computer Science*, pages 127–139. Springer, 2013.

[Ber05] Daniel J. Bernstein.

Cache-timing attacks on AES.

Available online at `http://cr.yp.to/antiforgery/cachetiming-20050414.pdf`, 2005.

[SP13] Raphael Spreitzer and Thomas Plos.

On the Applicability of Time-Driven Cache Attacks on Mobile Devices.

In Javier Lopez, Xinyi Huang, and Ravi Sandhu, editors, *Network and System Security*, volume 7873 of *Lecture Notes in Computer Science*, pages 656–662. Springer Berlin Heidelberg, 2013.

Bibliography II

[VCGRS12] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert.

An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks.

In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *LNCS*, pages 390–406. Springer, 2012.

[WHS12] Michael Weiß, Benedikt Heinz, and Frederic Stumpf.

A Cache Timing Attack on AES in Virtualization Environments.

In Angelos D. Keromytis, editor, *Financial Cryptography*, volume 7397 of *LNCS*, pages 314–328. Springer, 2012.