A Refined Look at Bernstein's AES Side-Channel Analysis

Michael Neve
UCL Crypto Group
Place du Levant 3
1348, Louvain-la-Neuve, Belgium
mneve@dice.ucl.ac.be

Jean-Pierre Seifert
Intel Corporation
2111 NE 15th Avenue
97124 Hillsboro, Oregon, USA
jean-pierre.seifert@intel.com

Zhenghong Wang
Dept. of Electrical Engineering
Princeton University
08544 Princeton, NJ, USA
zhenghon@princeton.edu

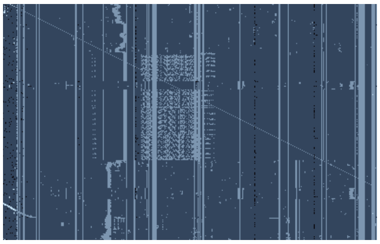
ABSTRACT

In a recent manuscript Dan Bernstein claims the successful extraction of an AES key from a network server through another client computer. His side-channel attack was actually the simplest conceivable timing analysis of AES. Although Bernstein gave no thorough analysis of his methodology or the underlying technique the paper contained the full C-source code. This was actually very useful to repeat, analyze and extend his experiments and technique. Our paper improves upon the work done by Bernstein in the following ways:

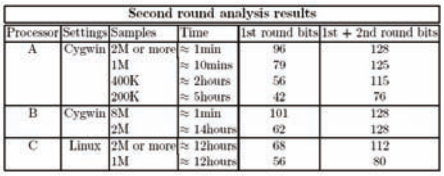
1) We present a thorough analysis of his used methodology hereby formally proving why and how his technique works. From this analysis we also derive a general limit on the number of derivable key bits through his technique which depends on the architecture of the underlying CPU.
2) We show the results of several important practical experiments. Those undermine first that the pure Bernstein technique cannot extract in practice all key bits even when the sample space is drastically increased. Second, they give evidence, that the Bernstein technique itself cannot be changed simply into a real remote side-channel analysis by just letting the client computer measuring himself the round trip times of his queries to the server.
3) Motivated by the above shortcomings of Bernstein's technique we improve upon his technique: while he uses only first round information, we show how to use this first round information to extract second round information. Thus, using a much lower number of samples this second round analysis allows for a direct full AES key recovery through simple overall timing measurements.

Our results aim at solving several fundamental open problems related to this kind of AES side-channel analysis:

I) What exactly makes this technique work? The following picture shows different cache line accesses when running AES-encryptions in a real environment – in the presence of many other parallel processes. It was made via a special software making the different cache accesses visible per cache line (x-axis) at different times (y-axis). The different S-Box accesses are clearly visible in the center. Among interesting features we can observe vertical lines of various width demonstrating continuous accesses to specific cache lines. Moreover, one also sees certain vertical stripes conflicting with the AES S-Box accesses generating cache-misses and causing longer encryption times when using those S-box entries. Now, the crucial observation here is the following: the exploitable different AES execution times, when averaged over many iterations are due to system-dependent cache-evictions. By varying over all possible values for the individual plaintext bytes, Bernstein's technique is implicitly searching for those cache evictions by the system. This leads then to small execution time differences for certain plaintext inputs.



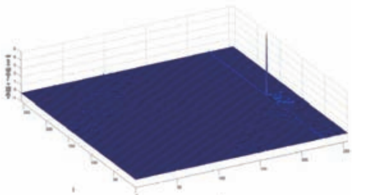
II) How many key-bits can it really extract? In order to prove the success of our 2nd AES round analysis, we tested it under several configurations which are summarized in Table 1.



Second round analysis results					
Processor	Settings	Samples	Time	1st round bits	1st + 2nd round bits
A	Cygwin	2M or more	≈ 1min	96	128
		1M	≈ 10mins	79	125
		400K	≈ 2hours	56	115
		200K	≈ 5hours	42	76
B	Cygwin	8M	≈ 1min	101	128
		2M	≈ 14hours	62	128
C	Linux	2M or more	≈ 12hours	68	112
		1M	≈ 12hours	56	80

Table 1. Improvements through a round 2 analysis of AES.

III) Is a truly remote AES key recovery through timing analysis really possible? Dan Bernstein uses two phases: a profiling phase first where the system under attack is analyzed with a known key and later an attacking phase with an unknown key. Finally, their correlation discloses several key bits. Because cache latencies are tiny compared to network delays, Bernstein's attack would require an extensive number of measurements to extract the cache behavior from the overall time. However, advanced attackers could increase the signal to noise ratio by submitting carefully chosen plaintexts. This comes from the following observation: the profiles for each byte are deeply linked to the position of the S-box tables loaded into the cache. Therefore, some bytes present similar profiles that can be combined to reduce the noise. Moreover, specially chosen plaintexts taking advantage of the S-boxes' location in the cache lead in some cases to a single peak as visible from the following picture. It shows the correlation between the attack phase and the profiling phase for one fixed byte. Using such chosen plain-text amplifications, combined with our second round extension, a real remote recovery eventually becomes possible in realistic times.



Keywords: *AES, Cache-state analysis, Computer Security, Information leakage, Side-channel analysis, S-box tables, Timing analysis.*

ASIACCS'06, March 21–24, 2006, Taipei, Taiwan.