Cache based side-channel attacks on DSA and ECDSA

Ritesh Goenka - 160050047
Abhishek Akkabathula - 160050074
Manoj Middepogu - 160050075
Sathvik Reddy Kollu - 160050077
Saiteja Nangunoori - 160050089

Indian Institute of Technology Bombay

August 7, 2019

Motivation

- Traditional cryptanalysis focuses on finding vulnerabilities in the mathematical properties and structure of the algorithms.
- Translating the cryptographic algorithms into real world software and hardware implementation presents new vulnerabilities. A class of such vulnerabilities arise when multiple users share some cache levels (eg. cloud environments), where the shared cache acts as a covert channel.
- In this project, we aim to study two cryptographic algorithms and their implementation, as well as explore, study and implement some cache-timing attacks.

Cache-Timing Techniques

Cache-timing attacks exploit timing information leaked from software implementations that run in variable time.

Evict+Time Technique

- *Trigger* an encryption of plaintext p in the target process
- *Evict* memory by accessing appropriate memory blocks
- *Trigger* a second encryption of p and time it

Prime+Probe Technique

- *Trigger* an encryption of plaintext p in the target process
- *Prime* by filling up the cache
- *Probe* by reading memory addresses and measure the time

Cache-Timing Techniques

Flush+Reload Technique

- To identify victim access to a shared memory line, flush the monitored memory line from the cache.
- Wait till the victim has time to access the memory line.
- Reload the memory line and measures the time it takes to load.

Flush+Reload vs (Evict+Time and Prime+Probe)

Unlike the earlier techniques that detects activity in cache sets, the Flush+Reload technique identifies access to memory lines, giving it a high resolution, high accuracy and high signal-to-noise ratio.

Digital Signature Algorithm (DSA)

- The algorithm uses the multiplicative group over a finite field. It is based on the computational intractability of the Discrete Logarithm Problem (DLP).
- Modular exponentiation is a central part of the algorithm. OpenSSL's implementation of the DSA uses the Sliding Window Exponentiation (SWE) algorithm for modular exponentiation.
- The SWE algorithm is exploited in [García, 2016] to retrieve and reconstruct portions of the private key using the Flush+Reload technique.

Elliptic Curve Digital Signature Algorithm (ECDSA)

- The algorithm uses the group of points on an elliptic curve. It is also based on the computational intractability of the Discrete Logarithm Problem (DLP).
- Scalar multiplication is a central part of the algorithm. OpenSSL's implementation of the ECDSA uses the Montgomery ladder algorithm for scalar multiplication on the elliptic curve.
- The Montgomery ladder algorithm is exploited in [Yarom, 2014] to retrieve and reconstruct portions of the private key using the Flush+Reload technique.

Outline for the attack on ECDSA using FLUSH+RELOAD

Input: Point P, scalar n, k bits
Output: Point nP
$R_0 \leftarrow \mathcal{O}$
$R_1 \leftarrow P$
for *i from k to* 0 **do**
  if $n_i = 0$ **then**
    $R_1 \leftarrow R_0 + R_1$
    $R_0 \leftarrow 2R_0$
  else
    $R_0 \leftarrow R_0 + R_1$
    $R_1 \leftarrow 2R_1$
  end
end

Algorithm 2: Montgomery ladder point scalar multiplication

References

Cesar Pereida García (2016)

Cache-Timing Techniques: Exploiting the DSA Algorithm

Conference Proceedings.

Yuval Yarom and Naomi Benger (2014)

Recovering OpenSSL ECDSA Nonces Using the FLUSH+RELOAD Cache Side-channel Attack

IACR Cryptology ePrint Archive, 2014, 140.