

Timing-Based Side Channel Attacks

Mohammad Nilforoush

Bradley Department of Electrical and Computer Engineering, Virginia Tech

In May of 2002, the Advanced Encryption Standard, developed by Joan Daemen and Vincent Rijmen, was adopted by the NIST for use by the U.S. government. In most implementations, this algorithm is optimized by incorporating many of the calculations involved in the encryption into pre-calculated lookup tables. However, Daniel Bernstein showed that the use of lookup tables makes these implementations vulnerable to timing-based side channel attacks. This means that an attacker can extract information about the encryption key by observing the time it takes to perform a number of encryptions using different input data.

The fact that this algorithm, thought to be secure because of the difficulty of breaking the encryption mathematically, is vulnerable to an advanced form of breaking cryptographic software based on execution time, poses a threat to the security of the private and proprietary data protected using this algorithm. In a time when protecting personal information is becoming increasingly important to guard against crimes like identity theft, the threat posed by such vulnerability is even more relevant. Thus, it is important to understand how these types of attacks on cryptographic software work as well as the extent to which an attacker could practically use them to launch a successful attack. Knowing this information would make it possible to develop effective and practical countermeasure to such attacks.

The purpose of this project was to gain a more detailed understanding of what causes a particular implementation of AES to be vulnerable to timing attacks, as well as to implement an attack on AES. This project was an attempt to guess an encryption key using the fact that modern computers have different types of memories and the time it takes to access a piece of data depends on the type of memory it is stored in. This type of attack is possible because the data accessed during part of the encryption depends only on the data to be encrypted and the encryption key. By finding out what data was accessed during the encryption and working backwards, the encryption key can be discovered.

The attack used in this experiment was implemented in two ways. When the implementation that assumed full knowledge of where the accessed data was stored in memory was used, information on every byte of the encryption key was recovered on the first guess after about 110 to 150 encryptions. However, when no such knowledge was assumed, information about only 13 out of the 16 key bytes could be recovered, even when as many as 3,000 encryptions were performed. These experiments demonstrated that information about the encryption key can be discovered using a timing attack, however, the practicality of these attacks is limited. Because in practice an attacker does not have the information used in the first implementation, they would have to perform the attack in a manner similar to second implementation, which was unable to extract information about every key byte. Thus, it would be relatively easy to implement AES in a way that would make it more difficult for an attacker to succeed.

This research was completed to meet part of the inHonors degree requirements. It was performed under the direction of Dr. Schaumont in the Secure Embedded Systems group in the Bradley Department of Electrical and Computer Engineering.