Attack Methods to Steal Digital Secrets

White Paper

Kerry Maletsky, Security IC Senior Product Line Director

All cryptographic systems depend on keeping secrets. Such secrets include passwords, secret keys, and private keys, but it is all the same; namely, a collection of bits to which access needs to be carefully restricted. Everyone has heard of the many ways to exploit software bugs or malware to attack a system, but really there is just only one way to retrieve the keys. This paper describes some very aggressive attack methods that can be used to get those keys out of small or large digital systems.

Crypto Background

The value of a secret is that it differentiates one digital system from another. A manufacturer may include secret keys, private keys, or certificates within a system when it is shipped, but in many cases, commercial digital systems are shipped as exact duplicates of one another in terms of both hardware and software. The different digital secrets that get written into the device when the product is brought online by the end customer and that is what creates the bases of system's security.

It should be clear that keys form the basis of the security mode for any system. The notion of security being based upon the secret keys is known as Kerckhoff's Principle, where a cryptosystem should be secure even if everything about the system (such as algorithms) except the secret key is public knowledge.

The opposite of this notion is called "security through obscurity", where the algorithm or protocol is kept secret. Security through obscurity has repeatedly been proven to be weak. Proprietary algorithms used to create obscurity have a long history that has led to the overall impression by modern cryptographic experts that they are fragile. Because it is often difficult to change a system design once it is in production, these weaknesses persist for a long time. Please refer to the following for more on this subject:

- **Transit Cards**
  Chapter 4, "Practical Attacks on the MIFARE Classic"
  http://www.doc.ic.ac.uk/~mgv98/MIFARE_files/report.pdf(1)
- **Common Systems Once Used for Garage Door Openers and Automobile Locks**
  "Cryptanalysis of the KeeLoq Block Cipher" (2)
- **Successful Attacks on the Originally-Secret GSM Encryption Algorithms**
  "GSM Sniffing" (3)

Modern Cryptographic Attacks

Most system designers now use industry-standard proven algorithms like RSA, ECC, SHA and AES. Since these are generally strong algorithms, the attacks tend to be focused on obtaining the secret key. One method of finding a key is to guess all possible values it could have. When guessing a human-entered password, this is often known as a "dictionary attack" since one method is to try all words, names, or combinations in a list. Strong cryptographic keys are never taken from a list, but from a random number generator is used to generate the key, so that guessing the value will take on the order of 2^(n-1) iterations, on average where n is the number of bits in the key. If n is large enough, guessing is impossible.

There are more efficient attacks exist on many algorithms than guessing, so key sizes are often much larger than expected. The website http://www.keylength.com/en(4) contains a nice comparison of key the combination of key lengths and algorithms. For example as an approximation, 128 bit AES keys are currently considered to be similar in strength to 3072 bit RSA keys or 256 bit Elliptic Curve keys.

The value of an attack on a single key varies depending on the system design, algorithm, and protected value within the system. This is a topic for another paper, however, it is generally true that the loss of a single key leads to the loss of others and thus a general degradation of the overall system security.

Many of the attacks addressed below are attempts to change the complexity of the attack from 2^(n-1) iterations of a method into just n iterations of some method. Putting that another way, if a method exists to find one bit of a key and that method can be extended to find the other bits of the key one at a time then that cryptosystem can be broken. The most conservative designers strive to prevent the loss of any bit of a key.

Hardware Attack Methods

There are many ways to attack the security of a system and the following is a list of just some of the most common physical methods for extracting keys from digital systems. All of these attacks are widely known and practiced. The goal of most hardware security designers is to raise the bar so the amount of time and money needed within the hacking community to break the system becomes prohibitive.

Fault Injection

Fault Injection attack methods generally involve inducing the system to operate incorrectly. By either injecting faults over many iterations or carefully choosing the faulting operation the attacker can manipulate the system to operate in way that allows the attacker to gain useful insight into the system.

Most modern digital systems are designed to operate properly over the stated datasheet's conditions of temperature, clock rate, voltage, timing, and other things. One class of fault-attack is to intentionally violate the datasheet to induce a failure. These attacks have been around for a long time and can be very successful, in part because the verification of modern semiconductors is typically focused on operation with the specification's window only.

These types of attacks do not need to be destructive. Heat or cold temperatures can be applied to the whole system or just selectively, the regulator's output can be overdriven to a non-specified voltage, additional clock edges can be injected 'between' the existing ones, and as other procedures can be done.

The "Fault-Based Attack of RSA Authentication" [5] article documents an attack on the OpenSSL RSA implementation that is achieved by running the processor at a voltage slightly below its lower specification limit. Please refer to "The Sorcerer's Apprentice Guide to Fault Attacks" [6] for a broad summary of fault attacks. The title describes its contents very well.

Another common attack methodology is to create a glitch in the power supply or to remove the power at an opportune time (i.e. tearing). These attacks are relatively easy to execute because they may not even require opening the case of the system. Just placing an electronic switch in series with the battery or power supply input can do the trick. Cookbooks to implement these attacks are widely available on the web.

An entirely different class of fault injection attacks are conducted through the normal communication path but with unexpected data contents being injected. Certainly, the most common target of such attacks is the software itself such as stack overflow for example. There are numerous examples of situations where the system can be induced to put the hardware in an undocumented, improper, or untested state where it may reveal a secret.

An example of the latter is the recent "Rowhammer" attack published by Google, "Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges" [7]. In this instance, the attackers were able to induce memory changes in adjacent rows of a DRAM through execution of carefully chosen application software. The changes in the memory allowed the attackers to have unauthorized supervisor access to the processor.

Lasers can also be used to inject a fault at a particular point and at a particular time into an integrated circuit. By carefully timing laser burst at the time when the device makes a security related decision (such as to accept or reject a command) the attacker can induce a change in a circuit element to reverse the computed decision.

Automated laser-based fault injection systems are available from multiple suppliers for use in the smart card and other industries. Usually these require destructive processing of the target integrated circuit target.

Timing Analysis

The timing analysis attack method generally operates by analyzing the time it takes to perform all or part of an operation. The operation might be a successful cryptographic component or it might involve the time it

takes to report an error. There are well-known attacks on RSA which work in this manner, please refer to "Timing-based Attacks on RSA Keys" [8] for a good example.

A simple example is where an attacker picks a random number as the proposed key and sending a message to the system protected by that key. The system might check one bit at a time and reject the message on the first failing bit. The attacker inverts the first bit of the proposed key and then tries again. If the rejection takes a little longer this time, the attacker will know the value of the first bit and can then go on to the second bit and so forth.

Certain algorithms are susceptible to classes of attacks in which the step in the overall protocol in which an error occurs is sufficient to weaken the key. It is relatively difficult to implement software in which the entire protocol sequence is time-independent, so local analysis of timing can benefit the attacker.

In some situations, an attacker over a network can measure the time taken to reject a bad/random packet. By carefully selecting subsequent packets sent to the system, the attacker can determine the key bit-by-bit.

Side Channel Analysis

Side channel analysis attack method generally operate by analyzing the side channel information emitted from the device. This often this takes the form of analysis of the transient system power consumption signature. This is often called "power analysis".

In a simple example, a processor might do an iterative math operation on each bit of a key. The first operation to be performed might depend on the "first" bit of the key, then the key is shifted and the loop is repeated. By simply observing the distance between the power spikes indicated by the loop opcode execution, one can determine if the value of the bit was a one or a zero from the outside.

More sophisticated versions of power analysis attacks are performed by building mathematical models of the power signatures taken over a number of executions to determine the key value even if care is taken to ensure that the loop programming is constant over time.

In its simplest form, a small resistor can be inserted in series with the power supply to the device such as a battery, charger, or external supply. An oscilloscope or other data-capture tool can be used to determine both the magnitude and temporal location of various current peaks. This data is then analyzed on a computer to determine the relationship between the measurements and data values being operated upon within the system.

It is not always necessary to directly measure the current consumed by a device. Many, if not most systems emit some form of radiated information related to the power consumed by the device.

There are published articles documenting successful attacks on cryptographic keys by analyzing the acoustic signatures of systems. Components within the system vibrate in a different way depending on the heat stress associated with the power consumed. Refer to "RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis" [9] for a successful method of recovering RSA keys from standard devices. Acoustic signature attacks do not require physical contact with the device.

Please refer to "The Temperature Side Channel and Heating Fault Attacks"[10] article for an example of a successful attack on an RSA key via temperature emissions measured on a device. Included in this paper are references to work done by analyzing the temperature of the air exiting a cooling fan of a system, as well as other methods.

There are a number of professional tools available to facilitate power analysis attacks in particular. Recently, a Kickstarter campaign was launched to make such attacks more available and at a lower price. Please refer to "ChipWhisperer-Lite: A New Era of Hardware Security Research" [11]. Such attacks are widely studied in academia and there are many documents available on the web. Generally the system is unaware that a power analysis attack is underway, and these attacks tend to be non-destructive.

Probing

The probing attack method centers on attacking an integrated circuit containing a secret by physically probing the circuit itself.

In some cases, the secrets are stored in nonvolatile memories, either EEPROM or Flash that are soldered to the board. Since the datasheets and functionality of most memory devices are well known, it is usually easy to read and/or write values to these devices and read the secrets directly.

It is usually relatively easy to locate the secret keys within the memory itself, despite the obfuscation that is often applied. Some methods include comparing the contents of two identical systems or looking for areas of memory for which the contents are not in well-defined formats like media files or operating programs for the local microprocessor.

If the secret is stored in an integrated circuit *other* than a standard memory then the silicon itself can usually be probed with microprobes (needles). Microprobe systems are widely used in the semiconductor industry to develop and debug production devices. They are not particularly expensive and can be found in most college laboratories, on the used market.

Most modern integrated circuits have two insulating layers between an attacker and the circuit elements on the chip:

- Plastic package of the device
- Passivation layer over the silicon itself

Fuming nitric acid, which is available at chemical supply stores, can easily etch away the epoxy that forms most integrated circuit packages. Depending upon the type of passivation used, various solvents or etch methods are available to strip that layer away, exposing the metallization layers which can then be measured.

It is usually possible to repeat an activity on most systems, so only a single probe may be necessary. The attacker successively probes each individual output of the on-chip memory or computation block with the same input stimulus applied. In this manner the key can be retrieved one bit at a time.

More sophisticated machinery can also be used to mount these attacks. Lasers can cut or burn away individual traces on a chip. E-beam probers do not require physical contact with the traces on the device to determine their state. Focused Ion Beam (FIB) machines can be used to effectively re-wire a device to change its operation. Please refer to “Mondex's Pilot System Broken” [12] article for an example of a successful attack using these kinds of methods. These attacks tend to be destructive.

Hardware Security Devices

Most general purpose MCU, MPU, SOC, and memory devices are designed to focus on performance, functionality, and cost above other considerations. So, it is not always practical to incorporate defenses against the above kinds of attacks in such devices. There are typically certain security features built into such standard products, however, they are generally not designed to defend against all assertive attacks. For instance, it is common in flash-based MCU devices to include a lock bit which prevents reading portions of the memory (such as those containing program code and/or secrets) as an example. It is often relatively easy to defeat these lock bits with one or more of the attacks noted earlier.

Dedicated hardware security devices are designed first and foremost with these attacks in mind and only secondarily focus on performance. Significant industrial and academic research has shown that with very careful design, testing, and independent validation that integrated circuits can be developed that raise the attack difficulty to a very high level. Such integrated circuits are available at various cost points depending on the functionality required and can be used within a range of systems from the smallest to the largest.

The Atmel® ATECC508A CryptoAuthentication™ and the Atmel AT97SC3205 Trusted Platform Module (TPM) are cost-optimized crypto element integrated circuit devices designed for small and medium complexity systems. Each of these crypto devices are designed to defend against the attacks described herein as well as others. Both use protected hardware-based key storage, advanced cryptographic engines, and countermeasures to keep the secret keys secret. Remember that modern cryptographic security is directly related to how well the keys are protected. Hardware key storage bests software key storage, and protected hardware key storage is the strongest hardware key storage there is. Please see www.atmel.com for more information on these and other security devices available from Atmel.

References

1. Wee Hon Tan. "*Practical Attacks on the MIFARE Classic*," http://www.doc.ic.ac.uk/~mgv98/MIFARE_files/report.pdf. Chapter 4, 2009.
2. Andrey Bogdanov. "*Cryptanalysis of the KeeLoq block cipher*." https://eprint.iacr.org/2007/055. 2007.
3. Karsten Nohl and Sylvain Munaut. "*GSM Sniffing,*" http://events.ccc.de/congress/2010/Fahrplan/attachments/1783_101228.27C3.GSM-Sniffing.Nohl_Munaut.pdf
4. BlueKrypt. "*Cryptographic Key Length Recommendation*," www.keylength.com/en, 2015.
5. Andrea Pellegrini, Valeria Bertacco and Todd Austin. "*Fault-Based Attack of RSA Authentication*," http://web.eecs.umich.edu/~taustin/papers/DATE10-rsa.pdf
6. Hagai Bar-El, Hamid Choukri, David Naccache, Michael Tunstall, and Claire Whelan. "*The Sorcerer's Apprentice Guide to Fault Attacks*," https://eprint.iacr.org/2004/100.pdf.
7. Mark Seaborn and Thomas Dullien. "*Exploiting the DRAM rowhammer bug to gain kernel privileges,*" http://googleprojectzero.blogspot.hk/2015/03/exploiting-dram-rowhammer-bug-to-gain.html. 2015.
8. OpenSSL Security Advisory. "*Timing-based attacks on RSA keys*," www.openssl.org/news/secadv_20030317.txt. 2003.
9. Daniel Genkin, Adi Shamir, and Eran Tromer. "*RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*," www.tau.ac.il/~tromer/acoustic/, 2013.
10. Michael Hutter and Jőrn-Marc Schmidt. "*The Temperature Side Channel and Heating Fault Attacks*," https://eprint.iacr.org/2014/190.pdf.
11. Colin O'Flynn. "*ChipWhisperer-Lite: A New Era of Hardware Security Research*," www.kickstarter.com/projects/coflynn/chipwhisperer-lite-a-new-era-of-hardware-security, 2015.
12. Anonymous. "Mondex's Pilot System Broken," http://cryptome.org/jya/mondex-hack.htm. 1997.

Editor’s Notes About Atmel Corporation

Atmel Corporation (Nasdaq: ATML) is a worldwide leader in the design and manufacture of microcontrollers, capacitive touch solutions, advanced logic, mixed-signal, nonvolatile memory and radio frequency (RF) components. Leveraging one of the industry's broadest intellectual property (IP) technology portfolios, Atmel® provides the electronics industry with complete system solutions focused on industrial, consumer, security, communications, computing and automotive markets.

Today, microcontrollers are just about everywhere, powering an expansive array of digital devices. Many are calling this the era of The Internet of Things, a highly intelligent, connected world where Internet-enabled devices will outnumber people. Atmel is pleased to be at the heart of this movement, developing innovative technologies that fuel machine-to-machine (M2M) communication and the “industrial Internet.”

Further information can be obtained from the Atmel website at www.atmel.com.

Contact: Kerry Maletsky, Security IC Senior Product Line Director
1150 E Cheyenne Mountain Blvd
Colorado Springs, CO 80906
United States
T: (+1)(719) 540-1848
Kerry.Maletsky@atmel.com

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**