6.857 Computer and Network Security **Fall Term, 1997**

Lecture 14 : October 21, 1997

Lecturer: Ron Rivest *Scribe: Lewis Girod*

1 Topics Covered

- Tamper Resistance
- Timing Attacks

Today's lecture covers the two papers in Handout 15, on Tamper Resistance and Timing Attacks. Unlike any of the attacks we have seen so far in this class, these papers illustrate different kinds of attacks on *implementations* of security systems.

The first paper we will cover documents the authors' experiences with breaking into "tamper-resistant" chips. In general, smartcards and other devices that rely on tamper-resistant hardware are vulnerable in many ways that are independent of the mathematics of cryptography. This becomes a real issue as smartcards gain acceptance in industry, for example at Florida State University or as part of a pay-TV service like DSS.

Cryptography requires secrets, such as the secret key held by a principal. If an attacker can discover the secret data by exploiting vulnerabilities in the implementation of a system, then as a rule the barriers imposed by hard-to-break cryptographic algorithms can be circumvented. Thus the maintenance of physical security over this secret data is an important part of a robust system.

Smartcards are a good example of a system that presents some interesting challenges in this area. In many cases, smart cards contain a secret encoded on a chip that implements a form of zero-knowledge proof. Smartcards are typically relatively readily available and are often distributed to a large user population. Typical applications include bank cards, debit cards, and access cards to buildings or pay-TV systems. If the embedded secret can be extracted, cards can be copied, allowing the holder of the duplicate to masquerade as a legitimate client, tenant, employee, etc. If the cards are distributed widely, it is impossible to maintain control over who has the opportunity to disassemble them, and "tamper-resistance" refers to the effort to make the secret difficult to recover from a card in the hands of an attacker.

What are the real risks? No security system is perfect, but in order to be cost-effective the security system must be secure enough to offset the potential costs, should the system be compromised. The credit card industry has performed this calculation in determining their policies and pricing structure. The level of security afforded by the mag stripe on a credit card is fairly low, but they have a number of other systems in place to identify fraudulent charges, and these mechanisms serve to keep their loss rate at an acceptable level.

The cell phone industry got into a great deal of trouble because their original protocol broadcasts plaintext ID information in the clear to the base station. It turned out to be very easy to copy an overheard ID number into another cell-phone, at which point the attacker's calls would be charged to the account associated with that ID. This practice is almost impossible to stop, and costs the cell-phone industry huge amounts of money in lost revenue and billing disputes. Newer protocols handle this ID information more securely.

2 Tamper Resistance

The first paper describes different ways of breaking into "tamper-resistant" chips. The overall message is that perfect tamper resistance is impossible. You must expect that someone will get past it, but a variety of techniques can make it more difficult and more expensive to extract the secret. The term "tamper-proof" seems to imply perfect tamper resistance, and is therefore not a good term.

The paper begins by defining three categories of attacker:

- Clever outsider: not well funded with incomplete knowledge of the devices
- Knowledgable insider: has access to detailed information about the design of the hardware
- Funded organizations: has significant funds, manpower, and lab equipment. May include insiders, and may be outside of legal reach.

When assessing the weaknesses of a particular system we assume that attackers can get unsupervised access to the devices themselves and any associated equipment. For example, in the case of a pay-TV access card, the attacker has several expired access cards, a non-expired card, and the set-top box that accepts them.

The paper describes several categories of attack in detail: non-invasive attacks, physical attacks, and advanced attacks.

2.1 Non-invasive Attacks

The typical smartcard consists of an 8-bit microprocessor, ROM, RAM, and an EEPROM, in a single chip. The secret key is stored in the EEPROM. The card has several inputs: clock, power, ground, reset, and several data lines. Non-invasive techniques try to exploit sensitivity to external and input conditions.

- Supply voltage: under-voltage, over-voltage, spikes
- Clock line: vary speeds, short clock cycles, etc.
- Reset line: trigger a reset and repeat an experiment
- Temperature: many components are temperature-sensitive

Often the EEPROM is protected by a security bit that is clear when the card comes from the factory and is set after the EEPROM is programmed. Once the security bit is set, it prevents access to the data in the EEPROM. Some microcontrollers can be tricked into clearing the security bit by raising the power to 0.5V less than the programming voltage. Another chip has an analog random number generator that consistently generates 1 values when given a low voltage.

In order to protect the cards from these kinds of attacks, card manufacturers have begun installing sensors that detect efforts to fiddle with the values and react by forcing a reset. However, in many cases the variations in actual environments tend to trigger these alarms and result in poor reliability.

One interesting attack uses power and clock transients to affect the execution of specific instructions, by adjusting the parameters of the transients to affect specific transistors. By causing specific instructions to be executed at specific times, a loop that outputs a specific memory range can be extended to dump the entire contents of memory. A countermeasure to this attack is the use of internal clock generators.

2.2 Physical Attacks

These attacks exploit physical characteristics of the device. Chips are composed of several layers of SiO_2 and deposited metals that form wires. Physical attacks often involve opening up the chip to figure out how it works or to detect specific voltages on the wires, and as a result many times involve physically damaging the device to get at the internals.

One simple physical attack involves focusing UV light on the part of the EEPROM containing the security bit, thus clearing it. A countermeasure to this attack is to include light detectors in the chip that disable the chip when triggered.

In many cases, a physical attack involves disassembling a chip to the point where probes can be attached to the components. The following series of steps works for many cards:

1. The plastic is cut away from the back of the chip with a knife.
2. Fuming nitric acid is used to remove the epoxy resin from the chip.
3. The passivation layer can be removed by probing with a needle that transmits ultrasonic vibrations, or with a laser. Probes can then be poked through the holes in the insulation and real-time values on the wires can be recovered.

2.3 Advanced Attacks

Advanced attackers often need to reverse-engineer the device in order to acheive their results. A recently developed technique cleanly etches away a single layer of a chip. By subsequently depositing a layer of gold and scanning the surface with an electron beam, the doping of various layers can be detected. By repeating this process for each layer in turn the entire contents of the chip can be derived. IBM used this technique to reverse-engineer the '386 in about 2 weeks, and recently used the same technique to reverse-engineer the Pentium.

Once the exact layout of the chip is known, the chip can be easily observed in operation. The voltage level of a specific feature on the chip can be read at a frequency of 25 MHz by shining a UV laser on a crystal of lithium niobate that is placed over the feature. This can be used to read the values of keys directly from the EEPROM output amplifiers.

In response to this attack, chip manufacturers use a "comformeal glue" to bond to the surface of the silicon and make it harder to remove the glue without damaging the surface of the chip. The Clipper chip implemented some additional ideas in tamper-resistance. A Clipper chip is manufactured using an unclassified mask that includes fusable links. These links are configured individually for each chip to install a classified encryption algorithm and secret key.

Another attack involves shining an IR laser through the back of a chip. Using this technique it is possible to probe the values of specific transistors. Focused Ion Beam machines, which are used by semiconductor manufacturers, can be used to cut wires, add wires, and change the doping of layers without physically taking the chip apart.

2.4 Advanced Protection

Protection against sophisticated attackers is practically impossible. To acheive protection with high certainty it requires physical guarding of the device and self-destruct capabilites to prevent loss of a cryptographic key. While the security devices used to lock out nuclear weapons use explosive charges to protect the key, many commercial security devices rely on similar but less drastic measures. Charged capacitors are included in some chips that can erase the EEPROM when sensors detect tampering. These designs can sometimes be subverted by cooling the device so that the erase circuitry fails to work, or by chopping the chip up faster than the circuitry can completely erase the key.

2.5 The Dallas 5000 Microcontroller

The Dallas 5000 series secure microcontroller is a CPU designed to be used with an external bank of RAM. In order to maintain security, a secret key stored within the CPU is used to encrypt both the address and data lines using for encryption a function of both the key and the address data. Thus, when a memory address is written the data written there is encrypted, and data read in from RAM is decrypted before it is used by the processor.

However, an attack called “cipher-instruction search” can discover the encryption function using inexpensive hardware. The attack experiments with feeding different combinations of wrong data back to the CPU until a specific 3 byte sequence corresponding to `MOV 90h, #xxh` is discovered. This instruction causes the processor to output the decrypted value of the third byte to parallel port P1 (address 90h). Once such a sequence is discovered, the complete mapping for the address of the third byte can be discovered by trying each possible value at that location and executing the three byte sequence. A similar search technique, this time for the sequence `NOP MOV 90h #xx` can reveal the complete mapping for the next byte. After those two addresses are mapped, the rest of the addresses can be discovered without needing to guess a working sequence.

Fixing the Dallas chip can be done in a number of ways:

- use longer instructions
- add an onboard cache and perform fetches in 8 bytes blocks
- include MAC bytes for blocks of memory

2.6 Lessons

One of the foremost lessons to be learned from these attacks is to treat claims about smartcards with skepticism. The engineering that goes into a security system must take imperfect security into account by minimizing the damage caused by a single breach of security. Finally, hostile testing is very important to the development process.

3 Timing Attacks

Our second paper describes timing attacks. Timing attacks are non-invasive attacks that rely on the variation in computation time required for the CPU on a smartcard to perform its secret calculation. Guarding against this attack is fairly easy; simply ensure that each response takes the same length of time. However, when this paper was released several smartcards were susceptible to this technique.

3.1 Attacking Diffie-Hellman

As an example we will consider a card that implements a zero-knowledge proof based on Diffie-Hellman. In this case there is a secret value x, and the card computes $y^x \bmod n$, given y. Typically a smartcard uses the standard approach to modular exponentiation, as shown previously in class. Essentially, a result register is initialized to 1 and for each bit b_i of x, if $b_i = 1$ then the result register R is multiplied by $y_i = y^{2^i} \bmod n$, where y_i is computed incrementally as $y_i = {y_{i-1}}^2$.

The attack is simplest to understand if there are some known unusual cases in which the multiplication algorithm is very slow. Suppose the attacker has knowledge of bits b_0 through b_{i-1} of the secret exponent, the value of y, and has knowledge of certain values for which multiplication is slow. Then the attacker can compute the result of the exponentiation of the first $i-1$ bits of x. If the attacker now chooses y such that the multiplication of the $(i-1)^{st}$ result with y^{2^i} is very slow, then the i^{th} bit is set only if the total calculation time is much greater than the time required to compute the first $i-1$ bits.

However, wide variations in timing are not required for this attack to succeed. In fact, using many trials with random y values, even slight variations in timing can be exploited in this way to detect correlations that reveal the bits in the secret exponent. Because these correlations disappear after a single bit is chosen incorrectly, errors are quickly corrected. Extracting the bits of an exponent may require as few as 2000 trials.

3.2 Factoring RSA Keys

Timing attacks can also be used to factor RSA keys. If the Chinese Remainder Theorem is not used, a similar attack to the Diffie-Hellman attack given above can be used. If the CRT is used to perform the exponentiation, then a slightly different technique must be used.

In the RSA cryptosystem, using CRT, the public key n is formed from two secret primes p and q, and the card performs the exponentiation $C^d \bmod n$, where C is a cyphertext chosen by the attacker. Using the CRT, the card performs the following calculations:

$$\begin{aligned}
&C_1 = C \bmod p\\
&C_2 = C \bmod q\\
&M_1 = {C_1}^{d_1} \bmod p\\
&M_2 = {C_2}^{d_2} \bmod q\\
&M = M_1 a_1 + M_2 a_2 \bmod n
\end{aligned}$$

Because the modular reduction operations performed in the first two steps do not run in constant time, the attacker can tell if a chosen C value is greater or less than p or q. If $C < \min(p,q)$, then those operations will take no time, but otherwise p or q will need to be subtracted from C at least once.