**Vulnerabilities in NFC**

The market is showing an increasing interest in contactless cards due to their ability to facilitate quick and effortless operations. These cards can be waved in close proximity to a reader in many cases without needing to be removed from a wallet or pocket. Furthermore, contactless communication nature ensures that the card is not subjected to the usual wear and tear effect that affects contact smart cards. As a result, contactless cards are becoming more prevalent and being employed in various applications, including but not limited to access control for secure facilities and environments, electronic identification documents, payment cards, and electronic tickets.

The wireless communication used in contactless cards makes them susceptible to security vulnerabilities. Specifically, the messages exchanged between the card and reader may be intercepted or the card content could be surreptitiously skimmed by establishing contactless communication with its chip. If the card is compromised, its content could be used to create a cloned card. To prevent such attacks, contactless card communications are usually encrypted, and access control mechanisms are implemented to restrict access to the chip content and functionalities. For instance, a code such as a Personal Identification Number (PIN) may be requested by the system with which the card is interacting, and the cardholder must provide it for access to be granted.

**Denial of Service attacks**

A DoS attack on NFC devices or readers can render them unavailable to intended users. One DoS attack scenario involves using a jamming device to disrupt communications between two NFC devices. This interference can destroy transmitted data and cause DoS. Continuous detection of jamming attacks can be used as a solution to this scenario. NFC devices can check the radio frequency field while transmitting and stop data transmission when a jamming attack is detected.

Another type of Denial of Service (DoS) attack that aims to destroy the trust relationship between customers and service providers. The attack involves a malicious actor creating a tag that causes an NFC mobile phone to crash after scanning. The attacker then places the malicious tag on top of the service provider tag. When a customer visits the service provider and scans the tag using their NFC mobile phone, the phone crashes. Since the malicious tag looks like a normal tag, it cannot be linked to the phone crash incident. This attack can destroy the trust relationship between customers and the service provider.

Another scenario of denial-of-service attack is simply touching the device with an empty tag causes it to generate an error message, which can easily occupy the device and make it unavailable. However, the use of an NFC switch can help to prevent this attack scenario. Although the drawback of this solution is that the user must manually turn the NFC functionality on and off each time they need to scan.

**Relay Attack**

A contactless card can be vulnerable to a relay attack, which involves tricking a reader into believing that it is in close proximity to the card when it is not. To carry out this type of attack, two devices are required: a proxy and a mole. The proxy needs to be situated near the reader, while the mole must be positioned near the target card. Both devices must be equipped with an interface that complies with contactless card standards to communicate with the reader and the card, respectively. A communication channel must be established between the two devices. The proxy forwards commands from the reader to the card through the mole, and vice versa for the card's responses. By presenting the proxy to the reader, the attacker can pretend to be the cardholder of the target card, thus gaining access to the privileges and rights granted by that card.

In a relay attack, the reader communicates directly with the legitimate contactless card, thereby circumventing some of the security measures used to protect such cards. For example, if encrypted communication is used, the secure channel setup and management are handled directly by the original parties, and the attacker does not have to play any role in this process. Similarly, if challenge-response mechanisms are employed to ensure the authenticity of the card, the attacker can bypass this security measure. Access control mechanisms can be an effective defense against relay attacks. If the attacker handling the proxy is unaware of the card code (such as a PIN) required by the system, they will be unable to complete the transaction. However, the effectiveness of access control mechanisms depends on their design and implementation, which are not always foolproof. In fact, some vulnerabilities may still exist that could be exploited by attackers.

**Non-Invasive And Algorithm Implementation Attacks**

Non-invasive attacks pose no physical harm to the contactless card and are not specific to a particular card. Once an attacker has developed an attack for a specific processor and software version, they can quickly replicate it on other cards of the same type. The equipment used in these attacks can often be disguised as a normal smartcard reader. Non-invasive attacks are particularly concerning in some applications for two reasons. First, the card owner may not realize that their secret keys have been stolen, so the compromised keys are unlikely to be revoked before they are exploited. Second, non-invasive attacks can scale easily, as the necessary equipment is usually cheap to reproduce and update.

* **Timing Analysis Attacks**

Timing attacks rely on measuring the time it takes for a system to perform operations and can reveal information about secret keys. For example, an attacker might use precise timing measurements to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems. If a system is vulnerable, the attack can be relatively simple and may only require knowledge of ciphertext.

Different inputs may cause cryptosystems to take slightly different amounts of time to process. This can be due to performance optimizations, conditional statements, processor instructions, and other factors. Performance characteristics typically depend on both the encryption key and the input data (e.g., plaintext or ciphertext).

Attackers can use timing measurements to find the entire secret key of vulnerable systems. This can be particularly dangerous because the owner of the system may not even realize that their key has been compromised.

* **Side Channel attack**

Side channel attacks (SCAs) are different from regular cryptographic attacks because they exploit side channels like power consumption, voltage fluctuations, temperature, or sound to extract secret information from a system. These side channels can be accessed by placing an antenna, magnetic probe, or other sensor close to the device or system. There are two primary types of SCAs: simple power analysis (SPA), which uses direct observations of power or electromagnetic measurements to extract secret information, and differential power analysis (DPA).

Diagram

Description automatically generated

To start with, it is crucial to evaluate the possibility of exploiting power analysis attacks on the card key. Power analysis attacks are a type of side-channel attack that can be used to extract sensitive information, such as secret keys or plaintext data, by analyzing the power consumption of a device during cryptographic operations.

In power analysis attacks, an attacker typically performs repetitive cryptographic computations using a fixed secret key and varying plaintexts. This enables them to collect a significant number of side-channel signals, such as power consumption measurements, that can be analyzed to reveal the secret key or other sensitive information.

In the case of a transit card, when a session key is generated, the cryptosystem operates with the targeted card key and a random number as input plaintext. This means that an attacker who can intercept the communication between the card and the reader can potentially obtain the session key by performing power analysis attacks on the card key.

Moreover, as the attacker sends the recharge command to the card, the cryptosystem carries out cryptographic operations, which further increases the opportunities for an attacker to obtain side-channel signals. By analyzing these signals, an attacker can mount side-channel analysis attacks and potentially extract the card key or other sensitive information.

Therefore, it is essential to consider the potential threat of power analysis attacks when designing and implementing security measures for transit cards and other similar systems that use cryptographic operations to protect sensitive information. By taking appropriate measures to mitigate the risk of power analysis attacks, such as using countermeasures like masking or randomizing the power consumption during cryptographic operations, the security of these systems can be significantly improved.

**Out of Bounds Read/Write Attack**

The out-of-bounds write vulnerability is a type of software vulnerability where the software writes data beyond the intended buffer limits, either before the beginning or past the end of the buffer. This can lead to various consequences, such as data corruption, program crashes, or even code execution. Out-of-bounds write vulnerabilities are frequently exploited by attackers to execute arbitrary code or crash a program. Sometimes, the overwriting of critical program data structures, such as heap control blocks, can occur, which can result in a denial-of-service attack.

This type of software vulnerability arises when the software writes data outside of the intended buffer space, leading to data corruption. Attackers can take advantage of out-of-bounds writes to modify program execution flow, besides overwriting function pointers and return addresses on the stack. It's crucial to note that this vulnerability can have severe consequences, as it can enable attackers to gain unauthorized access to systems, steal sensitive data, or disrupt the normal functioning of programs. As such, it's essential for software developers to take the necessary steps to prevent, detect and mitigate out-of-bounds write vulnerabilities in their code.

* **Buffer Overflow attack**

An attacker can exploit a buffer overflow vulnerability to gain access to an organization's IT systems by injecting additional code into a program, which sends new instructions to the system. In some cases, attackers may deliberately input data that cannot be stored by the buffer, thus allowing them to overwrite memory locations that store executable code with malicious code. This can give the attacker full control over the program and potentially the entire system.

Diagram

Description automatically generated

Buffer overflows have been a known vulnerability for some time now, but the fact that they are present in the firmware of Near Field Communication (NFC) readers is particularly concerning. When a buffer overflow attack occurs, the attacker aims to manipulate the function of a privileged program in order to take control of it. If the program has enough privileges, the attacker can then take control of the entire host system. It's worth noting that this type of attack is usually targeted at root programs, but not always. After successfully compromising the system, the attacker may immediately execute code that grants them root access, such as "exec(sh)", but there are other methods they may use as well. As such, the presence of buffer overflows in NFC readers presents a significant security risk that must be addressed.

Buffer overflow vulnerabilities can be particularly damaging in web applications, as attackers can use them to corrupt the execution stack and execute arbitrary code. These vulnerabilities can exist in both application servers and web servers, especially in web applications that use libraries like graphics libraries. Custom web application codes can also be vulnerable to buffer overflows. While these codes may receive less scrutiny from security teams, they can be more challenging for hackers to discover and exploit. Therefore, it's crucial to ensure that proper security measures are in place to detect and mitigate buffer overflow vulnerabilities in web applications.

**Man in the Middle Attack**

An attacker can carry out a man-in-the-middle attack on Calypso systems by exploiting a weak protocol design. This allows the attacker to recharge a card at a point of service without the terminal's knowledge. The protocol involves the reader sending a signal called Signature Hi to the card to indicate that the transaction on the reader side is complete. The card then commits the transaction on its side and sends the signal Signature Low to the reader if the transaction is successful. The reader commits the data on its side when it receives Signature Low but cancels the transaction if it doesn't receive the signal. The attacker can prevent Signature Low from reaching the reader, causing the reader to cancel previous transactions while the card is charged, and the new ticket number is stored in the card's memory.

Diagram

Description automatically generated

Another type of man-in-the-middle attack involves exploiting the plain messages that are sent at the start of a communication. A smart card (ex-OPUS; a transit card) tickets remain valid for two hours from their first use, so an attacker can manipulate the messages related to the last three usages of the card. By changing the time/date information in these messages, the attacker can deceive the reader into believing that an invalid ticket is still valid or that a valid ticket has expired. This is a significant attack that can allow the attacker to use the same ticket for an extended period, regardless of its actual validity.

**Skimming Attack**

The secure element has two modes - external and internal.

* External mode requires smart card chips in NFC devices to emulate a tag. An external reader can access the secure element but cannot distinguish between a smart card and an NFC device with a secure element.
* Internal mode allows the host controller to access the secure element, which can be read and altered by running applications on the handset's host controller. This allows users to remotely manage the information in the secure element through online connections, also known as Over The Air (OTA) management. For example, when using NFC for ticketing, a regular smart card can be used, and the tickets or money can be stored remotely online in the secure element.

The secure element provides an index of applications for both memory cards and processor cards. However, this makes it vulnerable to third-party players since other applications in the secure element are exposed. This issue is not limited to NFC technology but also affects the wider smart card industry.

**Eavesdropping Attack**

Although the communication range of NFC devices is limited to a few centimeters, eavesdropping attacks are still possible. The attacker's proximity to the target device depends on factors such as equipment used, location of the attacker, and communication mode. Passive mode is more difficult to eavesdrop on because the target device draws its power from the electromagnetic field generated by the active device. However, an attacker with sufficient knowledge and equipment, can still capture NFC communication.

There are three ways in which an attacker can launch an information leakage attack on an OPUS card.

The first method involves stealing the unique serial number of the card through Eavesdropping, skimming, or simply looking at the back of the card. This unique ID can be used to track and even identify the card holder by correlating it with other information associated with the card holder such as their purchase history, residential or work address, and daily commute schedule. Although there may be better ways to track or identify a person, this method is still a privacy-violating feature that can be carried out at a distance and in an automatic way without the user's knowledge. Attackers can use this information to create a customer profile for targeted advertising or track the card holder for any other reason.

The second Eavesdropping method involves estimating the time when the OPUS card was purchased by taking advantage of the fact that unique serial numbers are generated sequentially. By collecting many OPUS card IDs, it is possible to deduce the month and year of purchase with ease. With more precise methods, an attacker could reduce this time slot to a day or even less.

The third method involves sniffing or skimming information about the last three usages of the card, such as the bus line or metro station used, through communication with a legitimate reader or by using a Proxmark3 in the vicinity of the card. This information can be used to deduce the daily commute schedule of the card holder and their home or work neighborhood.

In summary, the use of contactless communication in OPUS cards can lead to information leakage through various methods, and attackers can use this information to track or identify the card holder, deduce their purchase history, and daily commute schedule. It is essential to implement proper security measures to prevent such attacks and protect the privacy of OPUS card users.

**References**

1. Yassaei, M. (2013). Security and privacy analysis of Radio Frequency Identification systems.
2. Alrawais, A. (2020). Security Issues in Near Field Communications (NFC). *International Journal of Advanced Computer Science and Applications*, *11*(11).
3. [www.blackhat.com/docs/asia-17/materials/asia-17-Kim-Breaking-Korea-Transit-Card-With-Side-Channel-Attack-Unauthorized-Recharging-wp.pdf](http://www.blackhat.com/docs/asia-17/materials/asia-17-Kim-Breaking-Korea-Transit-Card-With-Side-Channel-Attack-Unauthorized-Recharging-wp.pdf)
4. Y.C. Lee, Smart-card-loss-attack and Improvement of Hsiang et al.’s Authentication Scheme, Journal of Applied Research and Technology
5. <https://www.helpnetsecurity.com/2018/08/13/vulnerabilities-smart-card-drivers/#:~:text=Most%20of%20the%20vulnerabilities%20he,be%20exploited%20via%20malicious%20smartcards>
6. <https://www.nytimes.com/2002/05/13/business/vulnerability-is-discovered-in-security-for-smart-cards.html>
7. <https://www.rfwireless-world.com/Tutorials/NFC-Near-Field-Communication-tutorial.html>
8. <https://www.sciencedirect.com/science/article/pii/S1874548219300642>
9. <https://www.computerweekly.com/ehandbook/Smart-cards-security-risks>
10. <https://www.st.com/content/st_com/en/support/learning/essentials-and-insights/connectivity/nfc.html#:~:text=What%20is%20NFC%3F-,Background%20on%20NFC%20Technology,up%20to%2010%20centimeters%20apart>
11. *A method for resynchronizing a random clock on smart cards* ,By Didier Moyart and Régis Bevan, Oberthur Card Systems
12. *A Timing Attack on RC5*, By Helena Handschuhand Howard M. Heys
13. An Overview of Smart Card Security, BySiu-cheung Charles Chan