Security and Privacy in Mobile Payment Systems

Erin McBride, Andrés Molina-Markham and Kevin Fu

August 24, 2011

Abstract

Near-field communication (NFC) is a radio-based protocol defined in ISO-14443. Financial institutions started issuing NFC-based smartcards (or PICCs) years ago as a replacement for cards that rely on magnetic stripes for data transmission. Adoption of this new technology has been surprisingly fast for such large organizations [4]. If there exist security flaws for this finance-critical technology, the economic impact would be tremendous.

The availability of NFC-enabled consumer devices has recently surged with the release of the Android Nexus S phone. This phone includes an NXP PN65N NFC modem chip, which can easily function as a card reader (or PCD) and, under the right conditions, be able to emulate a smartcard.

The purpose of this research is two-fold: create the tools necessary to conduct a survey of the security and privacy-protection features of bank PICCs, and to find security flaws present in bank PICCs.

Acknowledgements

Special thanks to Olga Korobova for help in the exploration of Android's source, as well as quality assurance and user-interface redesign of implemented applications. Also to Andrés Molina-Markham and Prof. Kevin Fu, for their mentorship, project ideas and direction. Lastly, to the Distributed Research Experiences for Undergraduates (DREU) program, ran by the Computer Research Association's Committee on the Status of Women in Computing Research (CRA-W) and the Coalition to Diversify Computing (CDC).

1 Bank Card Privacy Survey

The majority of banks base their smartcards on the EuroPay, MasterCard and Visa (EMV) standard. While EMV was originally designed for smartcards that use contacts, it was ported with minimal modification to use NFC. The EMV specification describes the behavior of both bank PICCs and point-of-sale (POS) terminals.

The core of the EMV protocol is based on the transmission of Application Protocol Data Units (APDUs). The PCD pushes a Command APDU to the PICC, and then the PICC computes the response and pushes a Response APDU to the PCD. Most of the APDUs sent between the two are transmitted in plaintext. Cryptographic security is only employed in the authorization phases of a transaction.

There are two categories of authentication methods: online and offline. Online authentication requires the POS terminal to be networked with the card's issuer. Offline authentication allows authentication to be performed solely between the PICC and PCD, and uses public-key signature systems.

In many implementations there exist safeguards to deactivate the PICC in the event of a brute-force attack on the authentication system. We focused instead on the availability of private information and the viability of a relay attack, as proposed by Andres Molina.

Discovering static, private information from bank PICCs was a matter of implementing the first few phases of the EMV protocol in the Nexus S. These phases all precede the authorization phase, so a full mock-transaction isn't necessary. We completed this arm of the project within a month.

The first phase is Application Selection, where the POS terminal attempts to create a list of candidate applications available on the PICC. This can be performed with just two or three Command-APDUs per candidate application, depending on the method used.

The phase that follows consists solely of Read Record Commands, where the PCD queries the PICC for all declared file-like objects and saves them. The interpretation of this data requires some effort, as most are Tag-Length-Value objects built on a multitude of primitive data formats.

The amount of plaintext information that can be retrieved at this phase is surprising. Some implementations revealed the cardholder's name, activa-

tion and expiry dates for the bank card, a number that references a bank account, and even the data available on magnetic-stripe cards. In these cases, it's possible to create a functional magnetic-stripe card [3].

2 Relay Attack Viability

Implementing a relay attack between NFC devices has been done before [1], but not with the Android Nexus S. There are three main communication channels in this relay attack: the NFC connection between PICC and the first phone, the Internet connection between the two phones, and another NFC connection between the second phone and the POS terminal. We implemented those first two connections fairly quickly, as they were somewhat trivial. The third connection, card emulation, proved too difficult to complete in the given time frame. We settled on simulating a POS terminal in the second phone rather than have it talk to actual terminal.

We attempted to get the Nexus S to emulate a smartcard. We spent two weeks trying successfully edit, compile, and run the Android 2.3.4 source, to no avail. Even had we the time to successfully modify the source to unlock features not currently available, we would still have to overcome even more challenging problems at the hardware level.

2.1 Card Emulation Hurdles

While the hardware of the Nexus S is capable of emulating both NFC-A and NFC-B cards, there are significant barriers in place to prevent user applications from gaining access to this feature.

The phone uses the PN65N chip from NXP, which is a combination of the PN544 chip and a SmartMX secure element. There are two ways to make this chip behave like a smart card (PICC):

- Through the secure element.
- With a UICC (aka SIM card) that supports the Single Wire Protocol (SWP).

The SmartMX secure element runs the Java Card OS, which is essentially a very small JRE. On top of this sits the GlobalPlatform Card Specification, which allows for the management of multiple software card applets. New applets can't be installed on the secure element without knowing the secret keys the manufacturer configured the GlobalPlatform card manager with. If these keys were known, it would be possible to install a simple relay-style applet that forwarded APDUs to and from the Android OS as they were received.

Enabling card emulation through SWP is also problematic, as such a card won't be able to communicate with Android's application space.

Aside from discovering the secure element's secret keys, there are two solutions:

- Replace the PN65N with another in which the secret keys are known.
- Attach the SWP pin on the PN65N to a pin accessible by the Android OS rather than than to a UICC.
- Attach an external USB NFC modem capable of card emulation.

The second solution may be the most difficult. The SWP protocol is well defined in TS 102 613. The NFC chip sends information along the single wire by modulating current, and the UICC does the same by modulating the voltage. If a full-duplex I/O pin exists somewhere on the Nexus S that is accessible by the OS, it may be possible to re-purpose it to interface with the PN65N. Of course, such a change would require modification of Android's source code.

While the third solution will probably work, it defeats the intent of using consumer hardware. Android 2.3.4, which can be run on the Nexus S, has optional support for the Android Open Accessory platform. This allows the phone to behave as a host to a USB device. Such a device might be an NFC modem capable of card emulation.

2.2 Distance-bounding Protocol Weakness in NFC

One major limitation to relay attacks are connection time-outs and distance-bounding protocols. It takes time to process incoming traffic, modify it as necessary, and send it along. Even if traffic was passed through without modification, the increased distance between the two target systems necessarily increases latency. Usually time-outs are implemented in systems without security in mind; typically with the purpose of ensuring responsiveness. No matter the intent, such time restrictions put upper bounds on the distance between two devices and how much computation can be performed on through-traffic in real-time.

ISO 14443-4 defines timing restrictions for communications between a proximity integrated-circuit card (PICC) and a proximity coupling device

(PCD). The Request Guard Time and Frame Guard Time are lower bounds for communication, so they can be ignored for in the context of a relay attack. The Startup Frame Guard Time (SFGT) imposes a maximal limit of 4949 ms to a PCD's response to a PICC's Answer to Select (ATS). The Frame Waiting Time (FWT) can range from 302 s to 4949 ms, and determines the minimum time between two consecutive frames.

The value of the FWT is sent by the PICC to the PCD during the ATS phase of the Activation Sequence, in the TB(1) byte. It is possible to modify the TB(1) byte in transit since it is sent in plaintext and is unsigned, but it still can not exceed 4949 ms.

If the relay attack system was smart, the Activation and Deactivation sequences would be handled by the proxy reader and proxy card without being sent over the network between the two. This would bypass the SFGT limitation.

There's a way to get around the FWT limit in the PICC-to-PCD direction. A PICC can request more time (up to 292 seconds) to respond by sending a S(WTX) message, and these messages can be chained in order to get the PCD to wait indefinitely. Since PICCs are mass produced and cheap, manufacturers may cut corners and solely rely on the PCD to enforce the FWT limit. If this is the case, the FWT limit can be bypassed completely.

An NFC relay attack that implemented both ideas could extend the range between a PICC and PCD indefinitely, and have considerable time to process traffic before forwarding the data. However, timing restrictions implemented at the application layer can be sufficient to prevent this type of attack.

3 Conclusion

While the protocol contacless bank cards use to communicate with POS terminals provides no inherent security, application-level cryptographic methods appear to be adequate to safeguard the transaction process. Bank cards continue to keep security as a top priority with successive iterations of the technology, but the tendency to ensure backward-compatability with older cards will almost ensure the existence of a vulnerable but small population. Additionally, it will be difficult to prevent relay attacks without significantly changing the existing protocol or technology.

References

[1] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. "Practical NFC Peer-to-Peer Relay Attack using Mobile Phones." Royal Holloway University of London.

[2] Steven J. Murdoch, Saar Drimer, Ross Anderson, and Mike Bond. "Chip and PIN is Broken." University of Cambridge.

[3] Dan Balaban. "Contactless Bank Cards Forecasted To Continue Strong Growth in 2011" NFC Times. http://www.nfctimes.com/news/contactless-bank-cards-forecasted-continue-strong-growth-2011 Cited Jan Aug 22 2011.