Magnetic Swipe Card System Security

A case study of the University of Maryland, College Park

Daniel Ramsbrock
dramsbro@umd.edu

Stepan Moskovchenko
stevenm86@gmail.com

Christopher Conroy
cconroy@gmail.com

Abstract

This paper provides a comprehensive security analysis of the Lenel magnetic swipe card system used at the University of Maryland at College Park. We first explore the cards and hardware components which comprise the system, and then present several plausible points and methods of attack on the system. We chose several of these attacks and demonstrated them using a $240 commercial card reader/writer and a customized unit powered by a microcontroller, which cost about $20 in parts. We developed the capability to read cards, write arbitrary data to cards, simulate card swipes through a reader using a flux reversal pattern generator, and "sniff" data from up to 16 live swipes using a single microcontroller which can be easily hidden in the reader's housing. We tested and successfully demonstrated these capabilities on the live Lenel system under the supervision of the university's Department of Public Safety. Based on our findings, we recommend that the university use neither social security nor university ID numbers on the cards, that it use magnetic card access only in low-security areas, and that it use a more sophisticated and secure system such as proximity smart cards for access to high-security areas. While the analysis and recommendations presented in this paper are aimed at the University of Maryland, building security professionals everywhere can use the material presented here to enhance the security of their own swipe card systems.

Table of Contents

Acknowledgments

First, we would like to thank Dr. Jonathan Katz in the Computer Science Department for being the primary advisor on this project. He taught an undergraduate class titled “CMSC414: Computer and Network Security” in the fall of 2004, and Daniel Ramsbrock was one of his students. One of the many topics that semester was magnetic-stripe based systems. When Mr. Ramsbrock approached Dr. Katz regarding undergraduate research, Dr. Katz immediately suggested investigating the university's swipe card system. He has been an invaluable resource during this project, providing expert advice and guidance on this sensitive project.

We would like to thank Dr. Timothy Horiuchi in the Electrical and Computer Engineering Department. He spent many hours in the lab working with Mr. Ramsbrock to develop an early version of the microcontroller software used to read and store swipe card information.

We would like to thank Mark McGuigan in the Department of Public Safety. He cooperated with us throughout this project, providing nearly all of the information in this paper relating to the Lenel system on campus as well as a sample Mercury Security MR-10 card reader.

We would like to thank Dr. James Purtilo for his assistance to Mr. Conroy in his initial investigation of the privacy issues of the swipe card system. The Maryland Public Information Act requests which provided a great deal of valuable information about the system were submitted as part of a class assignment for Dr. Purtilo's HONR239R class.

Finally, we would like to thank Austin Parker and Nathan Waisbrot for lending us their MAKStripe R2TAO reader/writer unit for the duration of this project.

Introduction and Motivation

Magnetic stripe card systems are widely used by many different organizations to provide both convenience and security. Hotels use them for room access, credit card companies use them for handling purchases, and college campuses use magnetic cards for both building access and electronic payments.

We are trusting these systems with hundreds of thousands of dollars worth of transactions and equipment. However, it is known among security professionals that magnetic stripe card systems have many inherent security problems and can be readily circumvented.

The goal of this research paper is to investigate just how easy it is to circumvent such a system, and based on this, to develop realistic and affordable recommendations for making the system more secure. We will use the University of Maryland's Lenel system as a case study, and our recommendations will be specific to this system. However, the general principles behind our investigation and recommendations will be useful to magnetic card system administrators in any setting.

The larger motivation for this research is the fact that members of the university community are trusting their swipe cards on a daily basis. They trust them to keep unauthorized people out of buildings, to prevent theft of equipment and information, and even to keep meal plan points and prepaid debit balances intact. Additionally, they are trusting the system to keep their social security numbers (SSNs) safe. These are currently stored unencrypted on the magnetic stripes, and are sent across campus whenever a card is swiped. Finally, all entry swipes and purchases are recorded and stored unencrypted for approximately three months, so members of the university have to trust that this data is being protected against theft and is not being used or sold improperly by the university.

Bibliographical Note: The vast majority of information contained in this paper comes from the authors' personal experiences based on experimentation and interaction with the swipe card system as student members of the university community. All technical details about the Lenel system, including information about the magnetic stripe and prox card hardware, come from Mark McGuigan, who is one of the Lenel system administrators at the Department of Public Safety. These pieces of information are not individually cited to prevent a cluttered and broken flow of text.

The University of Maryland's Access System

Uses of University Identification Cards

Every student, faculty, and staff member at the University of Maryland at College Park (UMD) is issued a university identification (ID) card. This card has the person's name, photo, signature, UID number, and issue date printed on the front. The UID is a 9-digit university-assigned number used for identification in place of an SSN. The card contains a holographic overlaminate with the UMD logo to discourage counterfeiting. On the back, it contains a magnetic stripe and a 14-digit bar code.

The ID cards at UMD are currently used for six primary purposes:

1. Photo Identification: The picture on the front is used to match a face to a name.
2. Building Access: The magnetic stripe is swiped through a reader to open doors.
3. Electronic Payment: The magnetic stripe is swiped to debit meal plans, TerpBucks, and TerpExpress accounts. TerpBucks come with certain meal plans, and unlike meal points can also be spent in on-campus convenience stores. TerpExpress is a prepaid debit plan which can be used to pay for nearly anything on campus: food, convenience store items, books and clothing at the campus bookstore, as well as printing and copies at the libraries.
4. Library: The bar code is scanned when checking out books. The 14-digit number is also entered into the computer when using UMD's digital library, which gives students off-campus access to premium web content such as full-text journals and periodicals.
5. Sports tickets: In order to receive free tickets for sports events, students sign into the online ticketing system using the 14-digit bar code number on the back of the card. The students then print tickets with individual bar codes, which are scanned with hand-held units during game admission. The name resulting from the bar code scan is compared to that printed on the front of the student ID card.
6. Other Services: Currently, the Campus Recreation Center (CRC) uses the magnetic stripe to verify student status before admitting customers. Non-students have to pay a fee to use the facilities. Intramural teams also swipe cards when taking attendance at practices and games. Any organization on campus can swipe the ID cards in order to verify student status and/or identity.

Information Stored on the Magnetic Stripe

The magnetic stripe on the back of the ID card contains 17 5-bit characters and has the following format:

S	123456789	5	0	000	E	L
start	SSN	check digit	issue code	facility code	end	LRC

Each digit is stored in 4-bit binary-coded decimal (BCD) format with a single parity bit for a total of 5 bits per digit. The check digit is a checksum over the SSN, and is discussed in detail below. It is ignored when cards are swiped, so access is still granted even with an invalid check digit. The issue code starts at zero and is increased by one every time the card is lost and replaced, thus deactivating the lost card. The three-digit facility code is used to distinguish between different systems, so that a card issued at UMD does not work in any other access system and vice versa. The code is strictly enforced by the UMD Lenel system, meaning that a card will not work if the

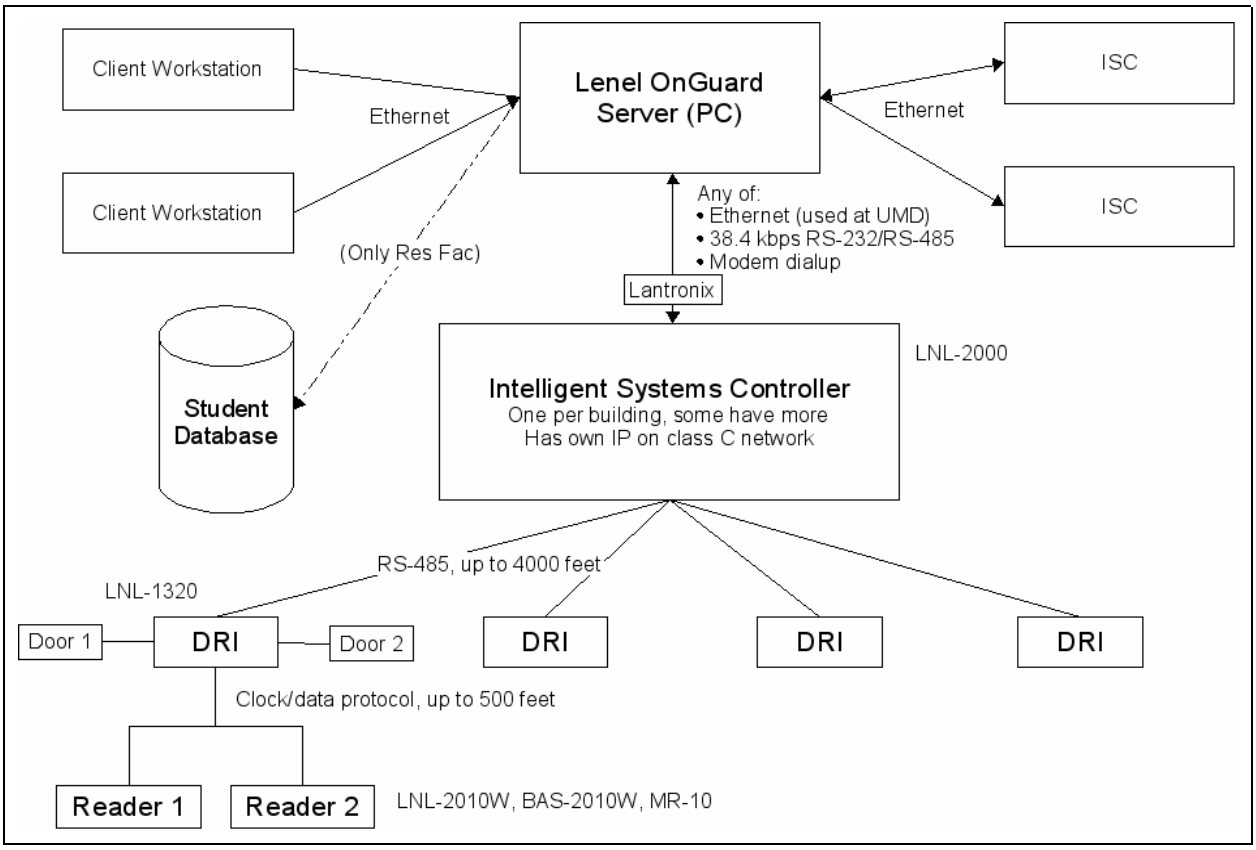
facility code is anything other than the expected value. The longitudinal redundancy check (LRC) is standard on magnetic stripe cards and is used to detect a bad swipe of the card.

There is a standard algorithm for a magnetic stripe checksum, specified under ISO/IEC 7812-1. This algorithm is commonly known as a Luhn checksum. However, the check digit on the UMD cards is not a Luhn check digit. After many attempts at trying to determine the check digit algorithm, we had just about given up. After some research about other checksum algorithms, we stumbled upon the proper check digit algorithm by pure chance. We found that the UID is verified by doubling every even-position digit of the UID, and then finding the overall digit sum. If adding the check digit to this sum results in a number divisible by ten, the UID and check digit combination is correct. We have verified that this algorithm does indeed work for every one of the dozen or so cards we checked.

Lenel Hardware

The swipe card system used by UMD is comprised of both hardware and software, and is provided by Lenel Systems International, Inc. In this section we provide an overview of the hardware and software components of the system, relying mainly on information from Mark McGuigan.

The diagram below shows how UMD's Lenel swipe card system is structured. Each of its components are discussed in more detail below.



The Lenel system configuration used at the University of Maryland

LNL-2010W Card Reader

The LNL-2010W is the basic reader used at nearly every door on campus (see picture on the right). It is pin-compatible with readers by many other manufacturers, such as the Mercury Security MR-10 and the Best Access Systems BAS-2010W. In many places on the UMD campus, the Lenel readers are replaced by these other pin-compatible readers. The LNL-2020W reader (also pictured on the right) includes a keypad, but this model is only used in about 50 locations on campus. All student access occurs without keypad codes since it would be cumbersome to maintain these numbers for all students.

Source: www.lenel.com

LNL-1320 Dual Reader Interface

The LNL-1320 Dual Reader Interface (DRI) is used on campus to connect two readers to their corresponding door locks and also to its Intelligent Systems Controller (ISC) described in the next section. The two readers connected to a single DRI can be at two separate doors up to 1,000 feet apart, since each has a maximum distance of 500 feet from the DRI. All of UMD's readers communicate with the DRI in Clock/Data mode.

LNL-2000 Intelligent Systems Controller

The LNL-2000 provides the link between the DRIs and the central Lenel OnGuard server described in the next section. An Intelligent Systems Controller (ISC) is also referred to as a "panel," and each building has at least one. UMD's ISCs are configured to communicate with the OnGuard server via Ethernet, using a class C local area network. The ISC communicates over Ethernet using a Lantronix MSS-100 serial-to-Ethernet interface device. There are also about 30 LNL-1000 ISCs installed on campus. These devices are functionally equivalent to the LNL-2000, but they do not support a secondary (backup) method of communication with the central server.

Lenel OnGuard Central Server

The OnGuard software on the central Lenel server is in constant communication with all of the panels (ISCs). It provides the interface between the panels and the users logged on for administrative/monitoring tasks (see next section). The server software maintains the status of each panel, as well as a database of all events in the system (swipe, door open/close, etc.). This information can be queried via the OnGuard client software.

Lenel OnGuard Client Software

The OnGuard client software can run on any Windows PC and provides administrative access to the Lenel system. The system is split into logical units called "segments," which correspond approximately to departments. A segment is essentially a collection of panels relevant to that department (for instance, all of the outside doors in their building, plus all doors to rooms belonging to the department). The OnGuard client software gives access to panels based on these segments. This prevents a member of the Philosophy department, for example, from accessing/changing information for members of the Computer Science department. Notice that not all members of a department have access to the OnGuard system. Typically one person from the department is designated as the Lenel administrator, and only that person gets the software

installed and a login for the OnGuard system. This person has complete control over all members of that department: he or she can add, change, and remove access to individual doors or entire panels.

Two Access Systems: Residential and Academic Buildings

There are two independent Lenel access systems installed at UMD. One is managed by the Department of Residential Facilities and includes all residential buildings on campus, such as the North Campus high-rise dorms, the South Campus suite-style dorms, and also South Campus Commons apartments. The other system is managed by the Department of Public Safety and includes all academic and administrative buildings on campus.

In this paper, we will focus almost exclusively on the academic and administrative buildings. The main reason for this is that we find it sufficiently easy to "tailgate" into residential buildings (walk in with another person who has swiped in legitimately). All residential buildings are required to have at least three-level entry security, which consists of two levels of card-swiping and one level of physical key access (two levels of physical key access in an apartment or suite: one for the entrance door, and one for the individual room door).

Every residential building has swipe card readers at the outside entrance doors. From there, students must clear another card reader, either at the doors to the hallways or inside the elevators. Both of these first two levels are easily circumvented by tailgating—students rarely ever challenge people who walk in behind them or even board the same elevator. The last level of security, however, is much harder to break. As long as students keep their dorm/apartment doors locked, an intruder would need either a copy of the key or sophisticated lock-picking equipment in order to actually get to students and/or valuable property.

We operate under the assumption that the problem of tailgating cannot be easily fixed by an upgraded swipe access system. Unless we install individual turnstile cages at every door, students will continue to let strangers in after them. However, tailgating by itself does not compromise the system—but unlocked doors or stolen keys do.

In the academic and administrative buildings, on the other hand, many important assets are protected only by one- or two-level swipe card security. This area is much more interesting from a security research point of view, since our findings and recommendations can improve security in this setting.

Student Database Connectivity

Currently, only the residential Lenel system is linked to the student database. The system is purged at the end of every semester and then preloaded with all resident information for the next semester. This ensures that the access is properly revoked each semester and that students can no longer get into buildings in which they lived previously.

The academic building Lenel system is not currently connected to the student and faculty database. People are added by hand, issue codes have to be incremented manually after a card is lost, and people have to be explicitly deleted from the system upon leaving the university. This has led to widespread problems with revocation, since departments are always quick to add new people to the system, but frequently forget to take people out when they graduate or otherwise

leave the department. All cards currently issued are set up to expire automatically after five years from the date of initial entry into the Lenel system, but many people stay at the university for considerably shorter times. This policy can also lead to problems for people who stay longer than five years: their cards may inexplicably stop working exactly five years after their first card was issued.

Compounding the revocation problem is the fact that the Lenel system does not support reporting by department, only by panel. Since panels can be accessed by multiple departments (especially panels controlling outside building doors), it is often hard to trace which department is responsible for a given user. If users cannot be reliably traced to departments, it is impossible to give departments lists of their users and ask them to remove inactive ones.

Planned Changes in May 2006

The university is currently working to eliminate SSNs from all campus applications which do not by law require it, including the Lenel system. This is called the Personal Identification Initiative, and it is being supported by the the Office of Data Administration and the Office of Information Technology (OIT), among others.

As a result of this initiative, the university will switch to newly formatted ID cards on May 28, 2006. The main difference with the new cards is that they no longer contain the SSN on the magnetic stripe, but rather the UID. This is a step forward in terms of preventing SSNs on student cards being used in off-campus identity theft, but we feel it is a step in the wrong direction since it makes on-campus identity theft trivial. We discuss this problem in detail below in the "Specific Security Implications" section.

In order to support the new ID cards, the Department of Public Safety is also making some important infrastructure changes: the student and faculty/staff databases will now be used to automatically keep the Lenel system up to date, rather than relying on manual updates. The advantage of using the database is mainly administrative since it results in less tedious work. However, it also addresses one important security concern: revocation of privileges for people who are no longer affiliated with the university is now handled automatically within several days, rather than having to wait until someone notices the discrepancy or the default five-year period runs out. Timely revocation is important to ensure security, especially in cases of disgruntled former employees or students who may wish to cause harm to their former supervisors, professors, or other members of the university.

We have found that while the changes in May 2006 do address two important security concerns, the use of the publicly available UID as the authenticator on the back of the card leaves large security holes and plenty of opportunity for on-campus identity theft. This is discussed in detail below in the section titled "Specific Security Implications."

Points and Methods of Attack

In this section we discuss the possible points and methods of attack in the UMD Lenel system. This is not meant to be an exhaustive list, but merely to give an idea of some feasible attacks.

Magnetic Stripe Cards

The three basic ways the cards can be attacked is through reading, copying, and creation of cards. Note that all of these attacks also apply to the new UID-based system, with the exception of the reading attack. This attack becomes a moot point under the new system since the UID number is also printed on the front of the card, so it would be much easier to simply look at the card and obtain the UID. Also note that only the reading and copying attacks require the attacker to have physical access to the card. The creation attack can be carried out without any physical access to the card.

Under the current system, simply reading someone's card can be considered an attack since it reveals that person's SSN to the attacker. Knowing someone's name and SSN is the basis for off-campus identity theft, so this is quite a serious problem. This attack does not require any specialized knowledge of electronics; all the attacker needs is a commercial magnetic card reader which attaches to a computer. These devices are available on the Internet for as little as $30 and are easy to set up and operate.

Another attack is the copying of an existing card. This can also be achieved without any electronics expertise—commercial card reader/writer combinations are also available on the Internet for about $240-300. These units allow someone to read in a card and then write out a copy of that card onto a blank card. As far as the system is concerned, this copy is identical to the original. The only difference is that the copy will not have the proper credentials printed on the front of the card, so it will not work for human identification purposes.

A more sophisticated attack on the cards is the creation of cards based purely on information (as opposed to having physical access to someone's card). The two pieces of information an attacker needs to re-create the magnetic stripe on someone's card is that person's SSN (or UID under the new system) and a rough idea of how many times he has lost his card (the issue code). In most cases, SSNs would be fairly easy to social-engineer, especially since many students are still used to utilizing them as student identification numbers. UIDs are public identifiers and are readily available, as discussed below. An attacker does not need to know the exact issue code since he can always start at a low number and keep incrementing it until it works. In the worst case, he will be successful in the current system after at most 10 attempts, since the issue code is a one-digit field (with possible values 0-9). Under the new system, the issue code is a two-digit field and could in theory reach up to 99. In practice, however, the issue code will be much lower than 99 or even nine, probably closer to two or three, so the attacker would need even less time to find the correct one

The most dangerous variation of the copying and creation attacks is to alter the data on an existing UMD ID card (for example, the attacker's own card). This preserves the authenticity of the front of the card but allows the attacker to put someone else's information on the back and get into buildings or make purchases in that person's name. The only way to detect this attack would be to look up the SSN (or UID) from the back of the card in the student database and compare it

to the name on the front. This is almost never done in practice—not by cashiers, and especially not at passive door readers.

Note that these attacks also imply that the current method of card revocation (incrementing the issue code) is not at all secure. If an attacker finds a lost (and deactivated) card, he can simply copy that card and start incrementing the issue code until the card works (most likely after incrementing it once).

Finally, there is the issue of using personal identification number (PIN) codes in addition to magnetic stripe cards. As mentioned above, Lenel readers with keypads are available, so it would be possible to require each student to type in his PIN code every time he swipes his card. This would indeed protect against the creation and copying attacks presented in this section, since the attacker would separately need to obtain the PIN associated with the card being copied. However, adding a PIN number does not protect against any of the three attacks below because it would also be transmitted unencrypted and could be just as easily sniffed as the number of the card. However, with hundreds of readers on campus, the cost of upgrading them all to include keypads would be prohibitive, and it would be much more sensible to switch to a more secure access control system altogether.

Connection Between Reader and DRI

The connection between the card reader and the dual reader interface module (DRI) is potentially vulnerable to both sniffing and data injection attacks. This connection is unencrypted and easily accessible by removing a single screw from the housing of the reader. Once an attacker has observed a few swipes, he can electronically simulate a valid swipe. This allows him to "swipe" arbitrary cards without making a physical card. This attack requires considerable electronics expertise and a small investment in the proper hardware (about $20).

Connection Between DRI and ISC

The unencrypted connection between the DRI and the intelligent systems controller (ISC) is similarly potentially vulnerable to sniffing and data injection attacks. However, since the wires for this connection are embedded in walls, ceilings, and wiring closets, these attacks would be hard to carry out in practice. Additionally, the more complex nature of the communication protocol between these two units would make this attack much harder in practice than the previous ones.

Connection Between ISC and OnGuard Server

The connection between the ISC and the central OnGuard server occurs over standard unencrypted Ethernet, so sniffing and data injection would again be fairly easy given physical access to the wires. However, these wires are also hidden in walls, ceilings, and wiring closets, making this the most difficult attack in practice. Also note that the Lenel system can be configured to use 128-bit encryption for this part of the system, so this attack would no longer work or at least become substantially harder (personal communication on April 13, 2005, with Craig Nick of Lenel Systems International). However, the UMD Lenel system is not currently configured to use this feature.

Chosen Methods of Attack

Based on the possible attacks presented above, we chose to implement a few which are realistic and relatively cheap. The first set of attacks can be carried out with a commercially available card reader/writer, and the second set requires customized hardware and software.

Attacks with a Commercial Card Reader/Writer

For these attacks, we used a MAKStripe R2TAO reader/writer unit which is available on the Internet for about $240. This unit is capable of reading all three tracks of both low- and high-coercivity (LoCo and HiCo) cards, and can also write to all three tracks of LoCo cards.

The simplest attack on the system is to read a UMD ID with this reader. This revealed to us the information stored on the magnetic stripe, as discussed above. Under the current system, this by itself is a powerful attack, since it reveals the SSN of the cardholder to the attacker. By swiping several of our friends' cards, including one set where someone had obtained a replacement card but later found the lost card, we were quickly able to figure out the meanings of most fields on the card. In the case of a replaced card, the only difference was that the issue code field had been incremented by one.

The software which comes with the MAKStripe unit allows both direct copying of cards and also custom-creation/editing of magnetic stripes. Given any UMD ID, we were easily able to copy it onto a LoCo card, such as those used for room access in hotels. Blank LoCo cards are also available in bulk on the Internet for several cents apiece. By editing the data on the stripe and experimenting with the values of the fields, we further confirmed our findings from above. We also discovered that the checksum field immediately following the SSN was not enforced: a copied card would still open doors successfully when this field was changed at random. The issue code and facility code, however, are strictly enforced. Any change in them results in the card being rejected.

As mentioned above, a more sophisticated version of this attack is to use an existing ID and alter the magnetic stripe on the back. However, since the UMD IDs are HiCo cards, we could not simply overwrite them with our MAKStripe unit. However, we were able to alter real UMD IDs in two ways: First, we copied the magnetic stripe of a valid ID onto an airplane boarding pass, which is thin and contains a LoCo magnetic stripe across the back. We then cut away the rest of the boarding pass, leaving only the magnetic stripe. We taped this stripe over the real magnetic stripe in the back of another UMD ID card, and it now swiped with the information from the ID we copied. Since cashiers do not usually turn cards over, this would be an easy way to charge a purchase to someone else's account.

Second, we used a commercial reader, a computer, an operational amplifier, and custom software to actually overwrite the HiCo magnetic stripe on an old UMD ID with the contents of another ID. This process required a high degree of electronics expertise, and it took a few tries to get it right. However, the end effect was the same as above, except without the telltale over-taping of the magnetic stripe. We now had a pristine-looking UMD ID which contained a SSN on the back not corresponding to the printed name on the front. No amount of physical examination can prove otherwise—the only remedy is to run the card through the student database and compare the resulting name to that on the card.

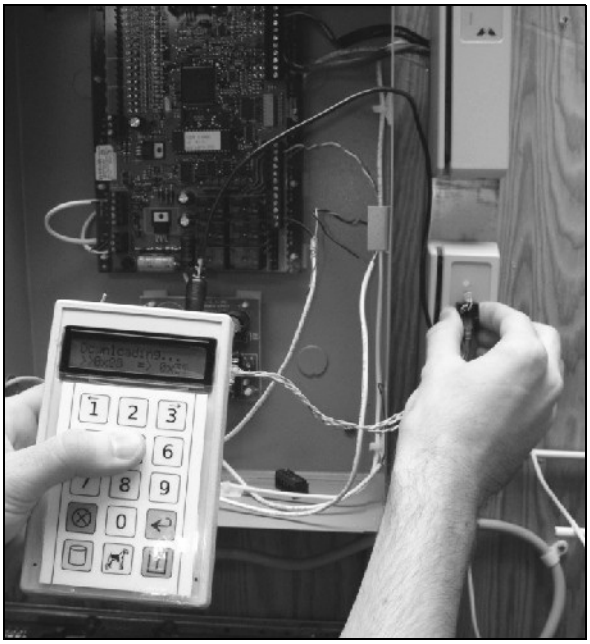
Attacks with Custom-Built Hardware and Software

Team member Stepan Moskovchenko designed and built a custom hand-held device to attack the swipe card system. It runs on a PIC16F877 microcontroller and allowed our team to perform more sophisticated attacks on the system. The total cost to build this device was about $20, but it did require a substantial amount of time and electronics expertise.

Manual data entry and transmission

The device has the ability to accept a modular plug from a standard card reader and decode the output (meaning it allows us to see what is stored on a card). This is nearly the same capability we had already achieved with the commercial reader, except this device allows us to interface directly with a commercial reader such as the LNL-2010W instead of having to rely on a reader which is designed to be connected to a computer. As mentioned above, most of the standard card readers are pin compatible with each other, meaning that our device can also read signals from readers by Mercury Security, Best Access Systems, and many other manufacturers.

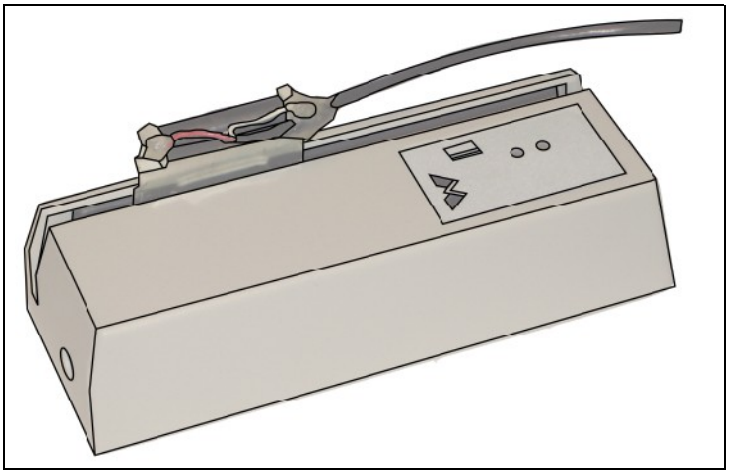
The device can also act as a flux reversal pattern (FRP) generator, producing the same magnetic signal pattern across a metal tab which would be created by a card being swiped. This metal tab is inserted into a card reader, and the device can then send arbitrary data to the reader. In other words, we can simulate the swiping of a card without actually making a card—all we need to know is the number on the card, which we key into to the device.



Data download from the skimmer

Finally, we developed a separate "skimmer" circuit which allows us to conceal a single microcontroller and an infrared (IR) transmitter inside the casing of a reader. The skimmer is programmed to record any swipes that pass through the reader, and it can store the last 16 swipes. After the 16th swipe, it overwrites the first one, and so on. The reader need only be removed from its wall-mount once when the skimmer is first installed. All future data downloads from the skimmer can be done via the IR transmitter, which is placed behind the empty lower LED slot facing the front of the reader. To download the data in the skimmer's memory, we insert the metal tab into the reader and send a special out-of-code bit string using the FRP generator. This string is only recognized by the skimmer, and is discarded by the reader as a bad swipe. Notice that an FRP generator is not

necessary for this operation: we could just as easily create a swipe card containing the special bit string. Upon receiving the special string, the skimmer starts dumping the contents of its memory over the IR transmitter, which can be picked up by an IR receiver attachment to the hand-held device. The IR receiver is held in front of the lower LED slot of the card reader for about 10 seconds while data is being transmitted. At this point, we can see all of the last 16 swipes on the screen of the device and could use this information to create cloned cards.

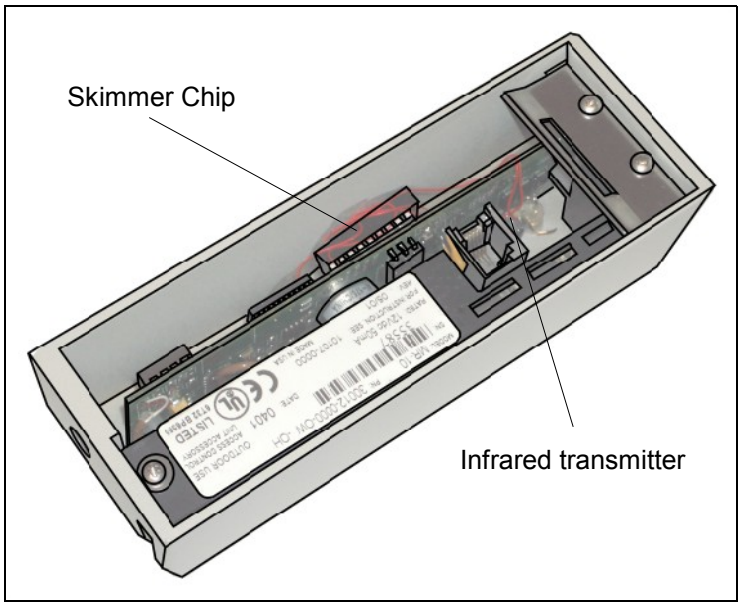


The FRP generator's metal tab inserted in a card reader

During the development of the device, we worked with the sample MR-10 reader provided by Mark McGuigan and also another pin-compatible unit purchased on the Internet. To verify that the device indeed works on a live Lenel system, we set up a demonstration at the Department of Public Safety where we tried our device on one of their installed test readers. This demonstration took place on February 24, 2006, at the Pocomoke Building under the supervision of Mark McGuigan and Major Jay Gruber. All features of the device, including the FRP generator and skimming chip, worked perfectly. The special bit string used to initiate a data download did indeed register as a bad swipe, and the skimmer left no other traces of its presence in the system.

Technical Details of the Custom-Built Hardware and Software

For a more detailed technical description of the custom-built hardware and software required to carry out the attacks above, please refer to Appendix A. Non-technical readers may safely skip that appendix and continue reading the paper here.



Breakaway view showing the back of the modified MR-10 card reader with the skimmer and infrared transmitter installed

Specific Security Implications

The attacks presented above have several specific security implications for the UMD campus community under the current system. All of these implications also apply directly to the new UID-based system, with the exception of the off-campus identity theft risk. We list this item first because we found it to be the most pressing issue, and we applaud the university administration for eliminating this threat by removing SSNs from the cards. However, we do not support the new UID-based system, as explained below.

Off-Campus Identity theft: As mentioned above, an attacker can easily read the unencrypted SSN on any UMD ID. He now has a valid name-SSN combination, which is the basis for full-scale identity theft. Assuming the attacker also has the ability to passively sniff live card swipes (as we describe above), this becomes even more of a problem because the attacker no longer needs to gain possession of the card in order to learn the SSN.

On-Campus Identity Theft: Under both the old and new system, it is fairly easy to assume another student's identity. All that is needed is the SSN/UID of the target and a rough idea of how many times he has lost his card. Using this information, the attacker can fabricate a card in the target's name using any of the techniques we discuss above.

We contend that using UIDs on the magnetic stripe creates a whole new set of security concerns. The UID is designed to be a public identifier—in fact, it is printed on the front of every UMD ID card. It is written on student papers/exams, used for posting of grades, and is even given out to student organizations in bulk by the Registrar's Office (for membership purposes). The fact that the university openly gives out UIDs further reinforces our claim that the UID was never designed to serve as an authenticator, but rather a public identifier that should have no real value to a potential identity thief.

However, using UIDs on the new magnetic stripes does exactly that: it relies on a public piece of information in order to authenticate the user to the card access system. It is now much easier to perform the creation attack described above. All the attacker needs is a person's UID and a rough idea of how many times the card has been lost, though that can easily be found by trial and error as well. There are many more ways to obtain a UID which were previously not feasible with SSNs. For example, an attacker can glance at the front of someone's ID to get a valid name-UID combination. Even worse, an attacker posing as an officer of a student organization might be able to obtain hundreds, if not thousands, of such valid combinations from the Registrar's Office by simply asking for them.

The three following implications are all related to on-campus identity theft, but we explore them below in more detail.

Theft of funds: We have mentioned several times throughout this paper how altered cards can be used to make purchases in other people's names, even when the front of the card is completely legitimate. The three main debit systems affected on campus are meal points, TerpBucks, and TerpExpress. Between these three systems, an attacker can use someone else's money to buy nearly anything available on campus, including expensive textbooks and UMD apparel.

Theft from academic buildings: Most academic buildings only have one or two levels of swipe card security between the outside world and many valuable assets. Outside doors to academic buildings are open during the day, leaving only one layer of security. Graduate labs, especially in the engineering and computer science buildings, contain equipment valued at many tens of thousands of dollars, and sometimes more. Furthermore, they contain large amounts of confidential research data, both in digital and hard-copy format. Limited protection combined with a large payoff make these labs attractive targets for thieves: with less than $300 worth of hardware, an attacker can easily bypass the swipe card system—all he needs is the SSN/UID of one of the graduate students who works at the lab. This can be obtained in many ways, including social engineering and by using the microcontroller-driven sniffing device we describe above. While the former method would at least in theory leave a trace (the student may remember a questionable character asking for his SSN/UID), the second method would leave no trace at all. Assuming that no fingerprints are found on the reader or the installed skimmer, and that no cameras were pointed at the reader at the time of skimmer installation, investigators would have no leads. Similarly, obtaining the target's UID from the Registrar's Office would leave little or no trace for investigators. The only lead at that point would be the SSN/UID used to swipe in, which could lead to an innocent person being accused of the crime.

Unauthorized entry into buildings: This is perhaps the most obvious consequence of a compromised access system. If the attacker can obtain the SSN/UID of a person with sufficient access privileges, such as a University Police officer or even a custodian or maintenance worker, he has basically free run of most buildings on campus. Even though the Lenel system provides the option of setting up lockout times when no people are allowed to enter the building at all, the UMD system does not use this feature for most campus buildings. The duplicated swipe card of a University Police officer would give an attacker access to nearly all outside building doors essentially 24 hours a day. Once inside, there are many things which a skilled thief can accomplish during the span of just a few hours in the middle of the night.

The ability to bypass the access card system at will presents many other opportunities for criminals, but we feel that these five are the most crucial issues.

Broader Privacy Issues

In the fall of 2004, University Registrar David Robb responded to a Maryland Public Information Act request that inquired about the records kept when cards are swiped, any privacy policy relating to such records, and any records of third party purchase or knowledge of the records. In Robb's response, he stated, “The ID card system neither collects nor stores any data about [card swipe] transactions.”

However, the university does keep such information for both the residential and academic swipe card systems. Denise Andrews, University Counsel, responded to a subsequent, more detailed request. According to Andrews' response, the university has no policy that outlines an access policy to the swipe databases, no policy for protecting this data, and no data retention policy.

If someone gains access to the swipe log data, he can exploit it in several malicious ways. First, he can track the purchase patterns of students in the dining halls and on-campus stores. This information, when paired with the log information from entry swipes at the Campus Recreation Center, provides a rough profile of the lifestyle choices which may be of concern to future employers and insurance agencies. Second, he can easily analyze the log information to determine patterns of movement for individuals or groups of interest. Stalkers, jealous friends, teachers, and parents could all find some very interesting information in a student's swipe log. Finally, if any part of the storage system is compromised—from the active database, to secondary storage, network connections between various parts of the system, and any removable storage media—then an attacker would be able to duplicate anyone's card and obtain access to arbitrary buildings and debit accounts.

Mark McGuigan is the only employee who has access to the archived information from his workstation. Nevertheless, this is an informal situation, and without a detailed security audit of the database server and its daily backup system, we can't rule out the possibility of rogue employees within the system who also have access to the log data. Moreover, the Department of Resident Life may have a much more open access system to their archive information. Mr. McGuigan purges archived logs older than three months onto a machine “behind a firewall in a facility with extremely limited proximity card access, CCTV monitoring, and restricted keying.” However, the university has never needed to access log data older than three months. After the information has served its purpose during these three months, it should be permanently destroyed due to its sensitive nature. At that point, it is simply a liability and there is no need to keep it available, even in an archival manner.

Recommendations

We have developed several specific recommendations for the UMD Department of Public Safety and the administration in general which will help alleviate some of the threats presented above.

Replace the SSN with randomly assigned number: We recommend that the university should not store the SSN on the card. This is a far too important piece of information to use on an easily readable medium such as a magnetic swipe card. In fact, there is no need for the SSN to be involved in this; all that is needed is a unique number assigned to each member of the university community. This number would be stored on the card and also in the access database, and by matching these two numbers, the Lenel system can determine the proper access profile.

As mentioned above, the university has recognized the dangerous nature of the SSN being stored on the card, and is preparing to switch the card access system from being SSN-based to being UID-based. However, due to our security concerns with the new UID cards, we recommend that the university use a completely separate, randomly assigned number on the magnetic stripe of the new UMD ID cards. Each number will uniquely identify a person, and there is no need for that person to even know what number is stored on his or her card. This number would be used by the Lenel system to link cards to access profiles. The mapping of these numbers to actual people should be stored in a central, secure location accessible only by the systems which need to map ID cards back to people (such as Dining Services and the Campus Recreation Center).

While this approach will not fix the problem of cards being copied or swipes being sniffed, it will protect against a variety of trivial attacks, such as those we describe above. It will also make the revocation process more secure: an attacker can no longer re-activate a lost card by simply incrementing the issue code—he would have to somehow guess the new random number.

Enable the tamper monitor feature of all card readers: The readers installed at UMD have an available feature called "tamper monitor." When this feature is enabled, the reader sends a constant stream of the same 8-bit code (01010100) repeated over and over to let the system know that it is connected and "alive." This allows the Lenel system to detect and log any reader disconnections. However, none of the installed readers at UMD currently take advantage of this feature, so we recommend that all newly installed readers have this feature enabled and that over time, installed readers are reconfigured as well. Having the tamper monitor feature enabled on all readers would make the undetected installation of a skimmer device much harder, if not impossible. In order to install the skimmer, the reader needs to be at least briefly disconnected from the system, either to physically solder the skimmer to the circuit board or to switch out the circuit board against one with a skimmer chip already installed. However short this disconnection may be, the Lenel system would be able to detect and log it. Upon inspection of the reader, it would be obvious that it has been modified, and the department could immediately replace it.

One theoretically feasible way to get around the tamper monitor is to use another reader which also has the tamper monitor enabled. The output of this reader could be temporarily spliced into the system while the circuit board is being replaced. However, this circumvention would require a very high degree of electronics expertise, and it is also not clear whether the system would not still be able to detect a third-party signal being spliced in due to slight timing differences. In summary, enabling the tamper monitor feature on all readers would make the skimmer attack we describe substantially harder to carry out without being detected.

Implement a security policy regarding live and stored swipe data: We recommend the university implement a formal security policy for the stored swipe data. A security system should not pose undue privacy threats for students, and a well-structured security policy will help to ensure that this data is not abused.

Since the only reason the university needs to archive data is to investigate crimes, this policy should formally declare that all log information be completely destroyed after it is three months old. After this generous time frame, the data ceases to be useful for investigating crime and only presents privacy concerns. We believe this is a reasonable recommendation because data in the academic system older than three months is already purged to a secure backup site and has never been required for an investigation.

Also, we recommend that the university formalize Mr. McGuigan's security practice for access to the archived information on the principle of "least privilege." Only one or two people in each department should have access to the logs due to the relative infrequency of crimes which necessitate examining the information. Ideally, any query on the log database would generate an access log of its own to be forwarded on to other university officials so that even these employees can be strongly discouraged from abusing this highly sensitive data.

Protect swipe data and its backups: While the data is stored, it should be encrypted using a strong cipher algorithm with digest hashes so that any unforeseen breeches of security by employees, building intruders, network intruders, or malicious software cannot compromise the integrity of the logged data. The primary purpose of the log is to assist in crime investigation; if the data is stored in a plain format, then an attacker could inject fake data to incriminate an innocent party in an investigation or erase relevant logs to protect a guilty party. Encrypting the data and using digest hashing such as SHA not only protects the privacy of students from a compromised system, it also protects the integrity of the data when it needs to be used for an investigation. Any backups made of this data should also be encrypted. The classic "low-tech" attack on a secure system is to physically steal one or more backup tapes. If these tapes are unencrypted and recent enough, the attacker essentially gains the same information from a backup tape as from hacking into the live system. Therefore, all backup tapes of this data should be encrypted in addition to being stored in a secure location.

Use magnetic stripe cards only for low-security areas: The card copying and sniffing issues are inherent to magnetic swipe card systems and cannot easily be fixed. Therefore, we recommend that magnetic stripe cards only be used in high-traffic, low-security areas. In fact, the UMD Lenel system was originally installed as a convenience method: it allowed remote opening and closing of all campus exterior doors on an automated schedule, rather than requiring custodians to make long rounds every morning and evening. It is important to view the magnetic swipe card system in exactly this way: a convenience solution for high-traffic areas where it would be impossible to manage a large number of physical keys. It works well to keep the casual passer-by out, but in high-traffic areas it does not even protect against intruders who tailgate in after legitimate users. A magnetic swipe card system is appropriate for applications such as after-hours outside door access to academic buildings and access to residential buildings (but only when combined with a third or fourth layer of physical key security as discussed above).

Use more secure access technologies for high-security areas: A magnetic swipe card system is not appropriate for low-traffic, high-security areas. The Department of Public Safety, which includes the University Police, has recognized this problem, and has switched to the potentially more secure proximity smart card (prox card) access system for its facilities. A prox card system has the advantage that the cards are capable of performing on-board processing, which allows for the implementation of a challenge-response protocol.

This kind of protocol allows for the reader to verify the identity of the card without actually having to transmit the secret number stored on the card. The reader transmits the challenge, a string of randomly generated bits. The card returns a response consisting of some cryptographic function of this challenge, using the internally stored secret number as the key. The reader knows this key and can compare the response to its own encryption of the challenge to verify the identity of the card. There are many variations on this protocol, but the important point is this: every time the same card is read, different data is transmitted by both the card and the reader. Given a strong enough cryptographic function and key, it is nearly impossible to use sniffed data to deduce the secret number on the card, which would be needed to successfully duplicate a card.

The prox card solution that is currently used by the Department of Public Safety consists of DuoProx II cards, which contain both a prox chip and a magnetic stripe. The magnetic stripe stores the same information as before and works in all regular readers on campus. The prox chip on the card is only used when entering areas protected by the prox system.

Unfortunately, this prox card system (consisting of HID Prox Pro readers and DuoProx II cards) does not use a challenge-response protocol and is therefore vulnerable to the same sort of sniffing and data injection attacks as the magnetic swipe card system. In fact, since prox cards use radio signals, they can be passively sniffed from a distance while being read by a reader—it would not be necessary in this case to physically install a sniffing device in the reader. Even worse, regular prox cards can be actively sniffed in the absence of a reader. All that would be necessary is to be within a couple of inches of a person's pocket/wallet where the prox card is located. An attacker would need to create a device that acts as a reader (or modify a commercial reader for this purpose). This device would be held close to the victim's pocket and would initiate the regular protocol of a card being processed, to which the card would respond by transmitting its stored number. The modified reader would be equipped with memory in order to store the number transmitted by the card. The attacker could now go back and use this number and a commercially available prox programmer to create a copy of the card. The victim would have no way of even knowing that the card has been compromised.

We recommend using a challenge-response prox card system (such as the HID iCLASS line of products) in places such as graduate labs, computer machine rooms, and private office suites. In fact, such a system should be used in any place where expensive equipment and/or sensitive information is stored. The cost of such a system in limited areas would be fairly low compared to the security benefit it would provide. Prox card readers can readily interface with the installed Lenel system, so the departments wishing to use this technology would only have to pay for the reader hardware upgrade ($200 per reader) as a one-time expense. There would be some ongoing cost involved since all department members would need to be issued prox cards, but this cost could either be shared between the department and each new member or directly passed on to the new member (in the form of a one-time “facility access fee” of a few dollars).

Conclusion

In our examination of the University of Maryland's Lenel system, we found it to be surprisingly straightforward and inexpensive to compromise the system. We determined that it is not feasible to use a magnetic swipe card system to protect high-security, low-traffic areas due to its inherent security problems. To address this problem, we devised six specific and realistic steps the university can take in order to enhance the security of the existing Lenel system at minimal cost. We have worked closely with the Department of Public Safety throughout this project, and we hope that it will carefully consider implementing our recommended changes.

Appendix A: Technical Details of the Custom-Built Hardware and Software

Low-Level Encoding

Like most magnetic stripe cards, the cards used with the Lenel system and the readers in questions are encoded using the ISO-7813 standard. Data on each track is recorded using two-frequency coherent-phase encoding, also known as Aiken Biphase. The track is divided into small, equally-sized units known as bit cells. A bit cell can represent the digits zero (0) or one (1). The value of a bit cell is specified by the number of magnetic flux reversals occurring within the cell. That is, the number of times the card flips magnetic polarity within the cell determines the value of the stored bit. A single flux reversal in a bit cell represents a zero and two consecutive flux reversals within a cell represent a one ("A Day in the Life of a Flux Reversal," by Count Zero, http://www.phrack.org/phrack/37/P37-06).

Card Layout

A typical magnetic strip has three tracks, but security applications generally use only the second track. This track stores numeric data using five-bit numeric characters. Each character consists of four data bits (least significant bit first) followed by one odd parity bit. This format allows for sixteen different characters, namely the decimal digits zero through nine, as well as special formatting characters used to separate fields and indicate track start/end. The track itself usually begins and ends with several zero bit cells. The sole purpose of these cells is to allow the reader to establish a time base for decoding the card. The string of zero bit cells lets the reader clock itself to the speed of the card to allow for proper decoding of the rest of the card. As the magnetic strip moves past the read head, the flux reversals induce electric currents within the head coil. The reader's processor then interprets the signal from the head and transmits the contents of the card to the Lenel system. We have found that it is possible to trick the reader into interpreting electromagnetic signals which did not originate from a card being swiped.

Our Electromagnetic Interface

To transmit arbitrary data to a card reader, we have designed a relatively simple device capable of imitating a moving magnetic strip. This device consists of a small piece of sheet metal soldered to the kind of electromagnet typically found in battery-powered clocks. This device is equipped with a small metal tab which is inserted into the card reader. Unlike a moving card, however, the electromagnetic interface is inserted into the card reader near the head and maintains stationary for the duration of the transmission. The principle of operation is similar to that of a transformer – a changing current applied to the electromagnet will generate a similar current in the reader's head through mutual induction. This effect can be used as the basis for a means of imitating a moving card. Now, all that is needed for transmitting a swipe is a way of generating the proper electric signals to drive the electromagnet.

Computer Audio Interface

We found that the simplest means of driving the electromagnet was by connecting it to a computer's audio output through a cheap operational amplifier (op-amp) chip. After studying various magnetic stripe (magstripe) specifications, we were able to create a simple command-line application for driving our interface. Given a numeric string, this application creates an audio file equivalent to the pattern of flux reversals on a magnetic strip carrying the same data. When the

audio file is played through the amplifier and the electromagnet, the card reader treats the incoming data as if it were coming from a moving card, and outputs the original numeric string typed into the computer. Although this is one possible way of transmitting data to a card reader, it is somewhat cumbersome in the fact that it requires a computer for playing back the signal.

Music Player Interface

Next, we tried creating a more portable version of the above setup. We transferred audio file generated by the command-line application to a portable music player. After several adjustments, we were able to successfully transmit a swipe signal to a card reader using just the music player, amplifier, and electromagnet. To test the setup further, we created audio file equivalents of our own cards. We were able to open doors in our residence halls by inserting the electromagnet into various card readers and playing back the audio files. Several adjustments had to be made to ensure reliable operation. The volume had to be turned to maximum and all frequency-domain enhancements such as bass boost had to be disabled to preserve the integrity of the peaks in the original signal.

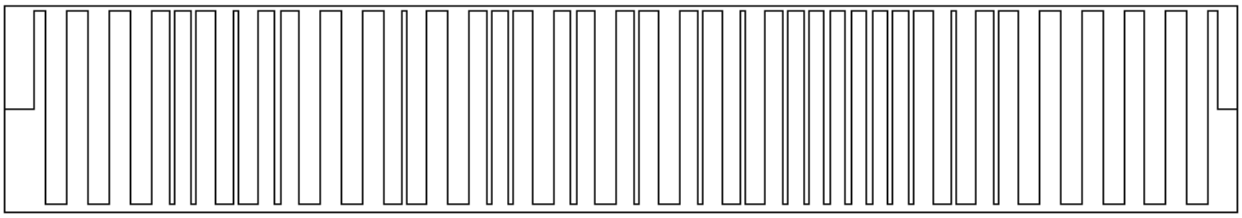
The Flux Reversal Pattern Generator

Thus far we had developed two ways of transmitting card data – by means of software on a laptop computer, and with a portable music player. The software version had the flexibility of being able to generate waveforms for any numeric string, but the involvement of a computer made it bulky. And although the setup with the portable music player was small and working fairly well, the possible cards it could imitate was limited to what files had been preloaded on the device. Thus, we developed the flux reversal pattern (FRP) generator to address the shortcomings of our previous designs.

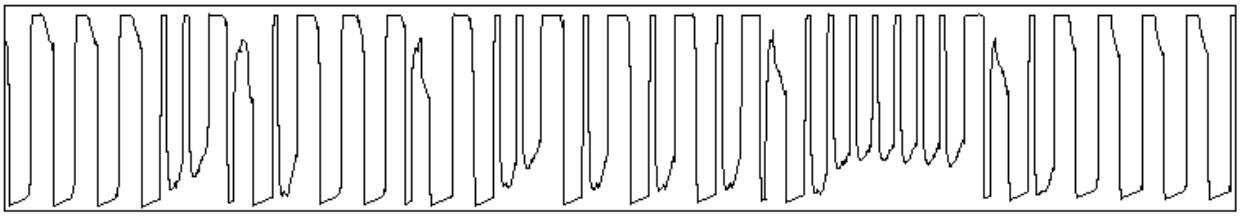
The flux reversal pattern generator is basically a combination of the two previous designs. This device is small, portable, and is able to output an electrical signal for driving the electromagnetic interface to produce a pattern of flux reversals for any arbitrary numeric sequence. The device is powered by an 8-bit microcontroller and is equipped with a keypad and LCD for user interaction. It is equipped with several ports – a connector for the electromagnetic interface, as well as a six-pin modular peripheral jack. Additionally, an in-circuit serial programming connector is hidden in the 9V battery compartment to allow for easy firmware upgrades.

FRP Generator Internals

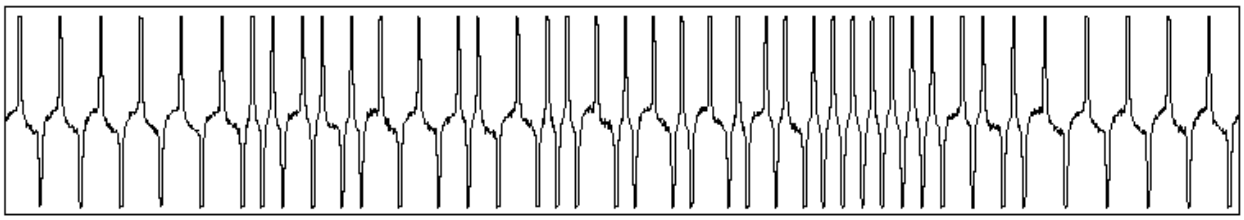
The FRP generator is powered by a microcontroller from Microchip Technology, namely the PIC16F877A. The microcontroller is based on a RISC core running at 4 MHz. This particular chip was chosen for its speed, availability (we were able to request several units through Microchip's free sample program) and high pin count. Although the task of generating card signals only calls for one I/O pin, the remaining pins were used for secondary functions such as controlling the LCD, reading from a matrix keypad, providing audible key feedback, and measuring battery voltage. Some of the extra I/O pins were connected to a modular jack embedded in the unit's casing. This allows the device to interface with a standard card reader, giving it the additional functionality of reading existing magstripe cards. These secondary subsystems are simple in their design and thus they will not be discussed here.



The theoretical ideal waveform of a sample string



The actual waveform output by our flux reversal pattern generator (FRP)



The actual waveform output by a card encoded on a professional Fargo encoder

Besides the microcontroller, the key subsystem of the FRP generator is the electromagnet driver circuit. This circuit is a simplified version of the amplifier which we used in our previous designs. Since card readers are only interested in waveform peaks and generally do not care about the shape of the waveform (see diagrams above), a complex digital-to-analog DAC circuit was not necessary to control the input to the amplifier. Thus, we were able to connect the amplifier circuit's input directly to an I/O pin on the microcontroller. The amplifier was powered directly from the 9V battery (as opposed to the microcontroller and LCD, which were powered by 5V through a 7805 voltage regulator). This was done to maximize the strength of the signal coming from the electromagnet, thereby greatly improving reliability. An interesting modification would be driving the electromagnet with an H-bridge circuit, but we have not attempted to do this as the existing circuit based on an op-amp already worked very well.

Card Encoder Design

Another way of transmitting data to a card reader is, of course, with a real magstripe card. We found it was relatively straightforward to encode LoCo as well as HiCo cards using the mechanism of an ordinary card reader. The MR-10 reader provided to us worked rather well for this purpose. We were able to disconnect the reader's circuit board from the head, and connect the head directly to the output of our amplifier. We then connected the amplifier's input to our computer's sound output. For the data source, we used our command-line application to generate the card signal, which we then repeatedly played back using commercial audio editing software. This arrangement repeatedly sent the audio signal directly to the reader's head. The amplifier in this case was required to boost the signal to the appropriate level, and to prevent damage to the computer's sound system in the event that we had made a mistake in the wiring. At this point, if we swiped a blank card at the proper moment, the data played back would be encoded on it. It took some amount of practice to learn how to properly swipe the card in sync with the sound

signal, but eventually we perfected our encoding technique. If we made a mistake, we were always able to pass a magnet over the card's magnetic strip to erase it, and try again. In the end, the signal readout of a homemade card closely resembled that of a card encoded using a professional encoder.

Skimmer Design

Any of these techniques alone would not pose a significant security risk without one important thing – the actual card data which needs to be transmitted to gain access to a certain building. While a brute-force attack is very much feasible on the temporary cards used for residential buildings, such an attack would be rather ineffective for academic buildings. A much more effective (and inconspicuous) attack we have developed is the skimming attack.

The skimming attack involves intercepting and storing the output of a building's card reader when a card is swiped by a student or staff member who has legitimate access to the building in question. After studying the card reader data sheets and communication protocol, it was fairly simple to program a PIC microcontroller to decode the output of the card reader. It took a few optimizations to efficiently store the data, as the clock rate of the card reader was rather high. Several other routines had to be implemented, such as stripping away the parity bits, repacking the bits to optimize storage, as well as reversing and realigning the data in the case that a card had been swiped backwards.

A very basic skimming circuit consists of just one component – a PIC16F628A microcontroller. This model was chosen for its small footprint and the fact that it does not require an external oscillator to operate. To carry out the skimming attack, one would have to program the microcontroller, obtain a card reader, and install the skimming chip. The installation was rather simple – the microcontroller need only to be connected to four points on the card reader's circuit board. Two of these provide power to the skimmer (we were able to get a regulated supply of 5V from a 7805 voltage regulator already present on the reader's circuit board). The other two connections tapped into the communication lines between the reader and the host system. A simple skimmer such as this can be installed by anyone with moderate soldering skills. The only difficulty we ran into was the fact that the circuit board was covered in a tough weather-proofing material, but this was easily scraped away.

Skimmer Data Retrieval

To retrieve the data from the skimmer, one would normally have to return to the installation site, remove the card reader from the building, disconnect the skimmer, and replace the card reader. It would then be possible to read the skimmer's internal EEPROM using any PIC programmer, or even with a read routine programmed into the skimming software. As straightforward as this is, it is still somewhat conspicuous to manipulate reader internals in open air. We have come up with a more intricate and much less conspicuous way of retrieving the skimmed data.

Infrared Data Retrieval

The card readers used at the university are equipped with two windows on the front membrane. These are intended for the two LEDs (red and green) on the circuit board, to indicate that access has been granted or denied. However, most of the newer model readers do not utilize both windows – they have a bi-color LED in the top window and nothing in the bottom one. We were

able to take advantage of the unused LED window as a means of inconspicuously retrieving data from the skimmer. We installed an infrared LED behind the second window and connected it to an unused I/O pin on the skimmer microcontroller. We programmed the skimmer to listen for a specially-designated bit string and upload the EEPROM contents through the infrared LED whenever this string is received. This bit string was crafted so as to not be decodeable by the Lenel system – several of the parity and LRC bits have deliberately been made incorrect. Additionally, we have added a menu option to the FRP generator to send this string through the electromagnetic interface and listen for incoming data from an IR receiver connected to the peripheral modular jack.

This configuration allows us to retrieve the card data accumulated in the skimmer without removing the card reader from the building. Removing the reader is now only required during initial skimmer installation. To download the stored data, the electromagnetic interface is inserted into the card reader and the IR receiver is held against the reader's lower LED window. A download command is then issued to the FRP generator, which transmits the specially-designated bit string to the skimmer. Upon recognizing this string, the skimmer syncs up with the FRP generator and sends its EEPROM contents through the infrared LED. The FRP generator receives the signals through its peripheral jack, stores them in memory, and displays the decoded card data on its LCD. The entire upload procedure takes less than 10 seconds. Infrared modulation is not required since the range required on the IR link is less than a centimeter. According to the results of our demo, the magic bit string is indeed unrecognizable by the Lenel system and is displayed in the swipe log as a card encoded in an invalid format.