[[1]](#footnote-1)

Effectiveness of Biometric Security

George Turner
University of Colorado at Colorado Springs

*Abstract*—Since the dawn of electronic computing, there has been the need for computers to be able to accurately identify the people that use them. It is a particularly difficult challenge, as computers do not natively possess this ability like people or animals. From birth, we (human-beings) are capable of distinctly identifying other people, places, and things from one another. Even more impressive, we learn this behavior naturally, with very little or no training. The same cannot be said for computers.

In addition, human-beings are also capable of detecting intrusions and facsimiles. While this capability is not perfect, we (human-beings) can determine the difference between the intent. For example, consider the difference between a person using a key to unlock a door, and someone using a pick to trick the tumblers and unlock a door. You can easily see the difference, and recognize the intent of the action. Now, consider the point of view of the lock. It can only perceive that all tumblers have been moved into their appropriate positions. Once that has happened, the lock must grant access (or it wouldn’t be a very effective lock). This is also the same with computers. They do not have the means to perceive the intent of the access.

# INTRODUCTION

The primary reason that it is difficult for computers to identify people is that computers can only process digital information. As the vast majority of the natural world is free form (or analog), computer engineers are forced to design hardware to handle the measurement and conversion to binary. To make things more complicated, converters are generally unique to the data that they capture. 70 or so years ago, analog-to-digital converters took a good deal of time to develop and construct. Furthermore, the technology was not completely reliable, which only adds to the complexity of the comparison of the data.

Now, in the 21st century, analog-to-digital converters are everywhere. People use them every day, whether we know it or not. Just to name a few: televisions, cellular and hardwired telephones, thermometers, cameras, microphones, video game consoles, or anywhere a computer is involved. With all of these examples, it is very easy to say that analog-to-digital converters have become inexpensive, and easy to manufacture.

# Passwords

Let’s go back 50 years ago, when computers were starting to emerge into a world of multiple users with the need to protect their data. Only a few devices existed to capture analog data. The simplest and the most accurate, of the time, was the standard QWERTY keyboard. From this, the *password* was born. While the use of a *spoken* *password* has existed in various societies for thousands of years, this paper refers to *password* in its digital form. Its first introduction to the world was 1961, at MIT, with much the same interface as it is used today [1].



Figure 1: Classic Login prompt for website access.

It should be obvious, to see why MIT chose this particular device as a means to safe guard data. First, its use requires little to no training; most people, nowadays, already possess it. The keyboard is modeled after one of the most basic skills, the ability to read and write. Its ease of use has kept this technique of security around for a very long time, and it’s still the most popular.

Next, the number of possible *keys* for a user is limitless. A *key* is meant to symbolize the device that generally paired to a single *lock*, such that it can only be *unlocked* (accessed) by the paired device. To further add to its simplicity, *keys* can be created and/or stored with nothing more than the human mind. Another point in its favor is that the *key* can absolutely secure, assuming the user hasn’t shared their password with anyone and/or written it down. (Additionally, assuming there aren’t any telepaths running around, and/or assuming the user is capable of enduring countless hours of torture.)

So far, it sounds like the *password* technique is flawless. However, it has some drawbacks; the *key* is easy to steal (in its cleartext form). Once this has occurred, the only way to re-secure your data is to change all your *locks*, and that means creating a new password. (If this has ever happened to you, I’m sure you’re fully aware of how frustrating and tiresome this can be).

In the past, the *locks* were also vulnerable. Early versions, of Linux and Windows stored password data in cleartext form. This was very problematic as a hacker could easily gain access to the password file. Subsequently, with little effort, they could determine the password for your account. Over time this vulnerability has been reduced substantially. Operating systems have learned to secure passwords in encrypted form. Furthermore, in enterprise environments, passwords are stored on a firewalled central server.

While computer engineers can continue to make this process more secure, users have been known to leave themselves vulnerable to attack. According to the New York Times, one in five internet users use an extremely simple password to protect their accounts [2]. To illustrate this problem, the most popular password is “123456”. As you can see, this is the fatal flaw of the password technique; it is the end-user. Whether it’s from neglect, laziness, or just being jaded, computer users cannot be trusted with their own security.

# Identification Cards

Returning to 50 years ago, another means of user identification was devised, with a different technology. Rather than trying to decode and analyze the natural world, this technology relies on the digital nature of computers. *Keys* are left in a digital format, and given to users as “identification cards”. Generally speaking, an identification card is a piece of plastic that presents some of the following items: user’s name, user’s photo, issuing organization or an expiration date. However, all of this data is meant for the user’s sake, it is rarely, if ever, presented to the computer.



Figure 2: Example CAC (Common Access Card), United States Department of Defense's usage of the Smart Card technology.[3]

The data that is transferred to a computer is generally a pre-selected, and unique, random series of *ones and zeros*. This allows for the computer to distinctly identify the sequence, without the use of advanced algorithms or logic. For security, it should have little, if any, correlation to the user. Otherwise, the key could easily be reproduced from the printed information, and eventually be used to grant access to undesirable users. In addition, when present in the computer, as a binary stream, it will be meaningless to hackers; as the data won’t trace back to the individual that it is intended to represent.

The technology used to implement Identification Cards has varied over the years. The first of its kind was implemented using magnetic strip. Today many commercial applications exist for such technology, but it is dying as a means of secure identification. It is far too vulnerable with today's technology, because it does not have a way to protect the data from being read by unauthorized scanners.

Instead, *smart cards* are the most recent implementation of this design. Smart cards were designed to solve the fatal flaw of magnetic strips, such that they only can read the *key* if the paired scanner is authorized. Such a design removes the possibility of duplicating a single instance of an Identification Card. With this concept in mind, holders of smart cards can be uniquely identified by the computer that reads their card.

As you’re beginning to see, Identification Cards are very similar to the *password* concept. The binary data that is stored on the card is essentially a *password*. Where this concept differs, is that passwords are already secure, and unique before they are assigned to the end-user. This resolves one of the fatal flaws with passwords; ensuring that it is complex enough when transmitted to the *lock*; such that hackers cannot randomly reproduce your *key*.

Identification Cards also share one other aspect with the *password* approach. Once the *key* (or card) has been discovered by a hacker or other thief, it can by stolen. Once provided to a card reader, the computer still has no way determine the difference between an authorized user, and someone who stole the access card. In order to prevent this kind of access, identification cards are generally paired with a PIN (Personal Identification Number). While this is capable of slowing down a hacker, it is obviously not a bullet proof means of protection.

# Biometrics

One of the developments of the last decade, or so, has been the ability for computers to reliably scan various features of the human body. This process is better known as *biometrics*. With biometrics, computers can identify people very similar to the way that we (human-beings) identify others. Furthermore, this a much more accurate way of identification, as there is not a means to duplicate an individual, which is the major flaw of the previously mentioned security techniques.

The difficulty of biometrics, is the ability discern similar samples of data apart from dissimilar. The afore mentioned techniques relied upon the *key* data remaining constant. In other words, their *locks* were designed only to accept a single *key*. With biometrics, such a constraint cannot be used, in order to create a functional *lock*. For example, go outside, and take a picture of some object; save the picture, and perform the same thing tomorrow. Were you in the same spot? Were you in the same position? Was the lighting the same? Was the new picture an exact replica of the original? Obviously, all of the answers to these questions should be "no", as you can also see with Fig. 3. Each type of biometrics has its own techniques for breaking down this problem, such that a computer can make a reliable comparison between the samples.

|  |  |  |
| --- | --- | --- |
| \\192.168.1.100\public\finger prints\geo1.gif | \\192.168.1.100\public\finger prints\geo2.gif | \\192.168.1.100\public\finger prints\geo3.gif |

Figure 3: Scan's of the same finger, of the same person, on the same fingerprint scanner. Do they look the same to you?

# Finger Print Scanners

There has been a process used by governments, and police organizations for the last couple hundred years, known as finger printing. This is all possible due to the fact that human skin creates ridges for the purpose of enhanced sensory transmission, and to allow for better gripping of rough and/or smooth surfaces. A side effect of the creation of these ridges is that the pattern that the create is unique for every individual on the planet, and are more/less constant for their lifespan. To further prove their uniqueness, "even identical twins (who share their DNA) do not have identical fingerprints" [4]. With these facts in mind they are perfectly suited for providing identification of an individual.

Early designs for fingerprint scanners used optical technology very similar to today's digital cameras. As you can imagine, these devices can be easily fooled by simple photographs of a user's fingerprint. With this in mind, it's obvious that this eventually leads to the same flaw as *passwords* and *identification cards*, the ability to duplicate the *key* of a given user.

In order to resolve this flaw, engineers decided to attack the problem differently. The new implementation, uses capacitance to resolve a surface definition of the human finger. Based on the voltages registered by the numerous capacitors in such a scanner, a computer can determine the ridges and valleys of the sample. See Fig. 4 below. More so, this technique requires that the sample come from the human tissue. Plastic, latex, rubber, paper, or any other material do not have the same electrical resistance to generate this effect, nor the same voltage range that will be registered by the scanner.



Figure 4: Diagram explaining the capacitor interaction with the human finger [5].

Another advantage of this technology, is that it becomes easier to discern when hackers are breaking in. For example, remember the scenario described in the biometrics section; no matter how hard or well you try, you cannot reproduce the exact original image or sample. Intrusion software can be easily be written to monitor when repetitive identical samples are generated from the same scanner. This scenario cannot be used with *passwords* or *identification cards* are their functionality depends on precise matches.

There are still many concerns with this technology. First, is your fingerprint safe as a means of identification? Even though the scanners are safe, what about the databases that store your information. Today, our society has enough issues with identity theft. What will begin to happen, when criminals have access to a repository of fingerprints to be staged as forensic evidence at a crime?

Another concern is the lifetime of a fingerprint. Various professions have the ability to damage fingerprints, such as construction, hazardous waste disposal, or even professional athletes. Workers in these areas are encouraged to wear gloves, but accidents can and sometimes do happen. Sometimes, there is enough of the original print, that the computer can work around the missing data, see Figure 6. However, severe alterations can occur, as such the question becomes, "how do you update the computer to recognize you with a new set of fingerprints?"

# Operating Systems and Application Integration

Another remarkable aspect of this technology has been ubiquitous mainstream support. All of the major operating systems (Windows, Mac OS, and Ubuntu) support each of the various fingerprint scanners. This is truly a feat; as the computer technology world has been filled with rivals for quite some time. In recent memory, there have been wars between, Blu-ray vs. HD-DVD, iPhone vs. Android, Google vs. Bing, and Intel vs. AMD. Oddly enough, these battles are still being fought, with very little chance for resolution. These battles are so intense, such that businesses will reach a point, where they refuse to integrate or support their respective competitor's product.

To add to the OS support, SDKs (Software Development Kits) have also been provided for all of the major programming languages (Java, .NET (C#, Visual Basic, etc.), Pascal, and etc) [6]. The drivers to communicate to the scanner and algorithms to analyze provided data are already present within the SDK. Client code that access this API is not required to manage this functionality. This greatly reduces the amount of time to develop an application that can make use of this technology to secure their application.

# Conclusions

Throughout this paper, we have seen many different techniques used to identify the end-users of a computer. The older techniques to protect data are still secure and have relevance. Their only flaws are that people can harm themselves by having weak passwords, or are susceptible to very skilled hackers. Through use of biometrics, both of these problems can be averted. Unfortunately, it seems that there are some still unaddressed issues with biometrics, such as privacy, and duration of the master sample. As time progresses, I feel that these issues will be addressed, and eventually lead to biometrics as the new security standard.

# References

[1] *Password – Wikipedia, the free encyclopedia* (n.d.) [Online]. Available: <http://en.wikipedia.org/wiki/Password>

[2] *Simple Passwords Remain Popular, Despite Risk of Hacking – NYTimes.com* (n.d.) [Online]. Available: <http://www.nytimes.com/2010/01/21/technology/21password.html>

[3] *File:ExampleCAC.jpg – Wikipedia, the free encyclopedia* (n.d.) [Online]. Available: [http://en.wikipedia.org/wiki/File:ExampleCAC.jpg](http://en.wikipedia.org/wiki/Password)

[4] *Fingerprint– Wikipedia, the free encyclopedia* (n.d.) [Online]. Available: <http://en.wikipedia.org/wiki/Fingerprint>

[5] *HowStuffWorks "How Fingerprint Scanners Work"* (n.d.) [Online]. Available: http://computer.howstuffworks.com/fingerprint-scanner3.htm

[6] *VeriFinger fingerprint recognition technology, algorithm and SDK for PC and Web* (n.d.) [Online]. Available: http://www.neurotechnology.com/verifinger.html



Figure 6: Left side, image of scarred fingerprint. Right side, computers interpolation of the left image.

1. [↑](#footnote-ref-1)