TASK:

Develop a cost-effective solution for storing and manipulating information on a portable flash memory device and allowing secure, accurate, selective data transfer between this portable flash memory device and existing personal computers and computer networks.

Solution:

It was decided that the best means to display our safety features was through a medical health record application. Thus, introducing the SafeByte!

Obstacles Encountered

Having only a few team members with Computer Science knowledge, we had to overcome a slight lack of expertise. Also, just coming together for a consensus on certain issues or features sometimes posed a problem.

RESULTS

Accomplishments

We have designed a program that is:

- Portable
- Secure/Accurate
- Software and Data Authenticated
- Cost-effective
- Easy to use
- Capable of having multiple users
- Compatible (HIPAA)

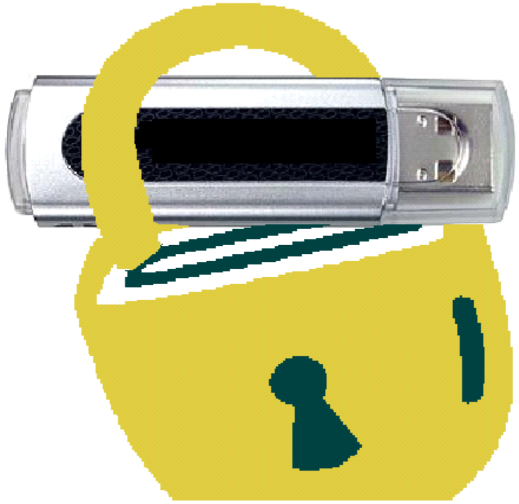
IN THE FUTURE:

- **Will be able to verify the identities of Doctors, etc. using a third party source (internet)**
- **Will be able to store wider range of Medical Records**

We would like to take this opportunity to thank our sponsor Dr. Brett Trockman. Also, we would like to thank Dr. Wai Gen Yee for his guidance and support throughout this project.

IPRO 304

SAFEBYTE: PORTABLE AND SECURE DATA STORAGE



TEAMWORK:

Application Design-

- Shaan Khan
- Dmitry Ratnikov
- Kanishk Sharma
- Pooja Oza

Application Security-

- Ikechi Emelogu
- Usman Abubakar

Physical Interface Security-

- Steven Banaska
- Mike Brenyo

THE TEAMS

APPLICATION SECURITY

Another security issue that was addressed is ensuring that the software that we loaded is authentic.

STEP ONE: Why is this important?

Authentic software is important for any software that stores and implements someone's private information. If the software itself is invaded, information that is being stored (in this case, highly sensitive medical information) can be accessed.

STEP TWO: Designing the solution

We searched for means to ensure the authenticity of software and determined that encryption was the best and easiest way to do so. Also, digital signatures are to be used as a means of encryption.

STEP THREE: Implementation

We used digital signatures in the following manner to ensure security:

- Software is passed through the signature generator and stored
- When software is to be authenticated, a new signature is made.
- Authenticity is achieved if the original signature matches the new signature

APPLICATION DESIGN

STEP ONE: Learn about EHR

Electronic Health Records are a revolutionary way of storing all of a patient's pertinent medical information in electronic format. We found that there are rules as to how these records need to be kept.

HIPAA:
The Health Insurance Portability and Accountability Act states that medical records need to be handled in an appropriate and private manner.

In our project, this was pertinent in that we are storing health records, so we must abide by the standards. Also, we have implemented levels of security for accessing and editing. Along with the rules, we had to include time stamps to show when data are altered along with who made the change.

STEP TWO: Design the Program

We researched many existing products for this part of the process. We found many features that we needed to include (pages of pertinent medical information) and also features that we wanted to avoid (easy access and security leaks). Afterwards, we wrote out our specifications.

STEP THREE: Build it

We started by designing the (Graphical User Interface: the way the program looks when it's run) and from there coded behind it to store information, manipulate it, and all the while, operating within different user access levels.

PHYSICAL INTERFACE SECURITY

PROBLEM:
The flash drive must not be able to attack the host computer.

- Disable Auto-run
- Identify the appropriate drive for a virus scan

STEP ONE: Identify attacks

Research was done to determine which kinds of attacks are common and associated with flash drives (CD emulation and auto-run).

STEP TWO: Design the solution

There are many methods available now to solve the issues associated with the aforementioned issues. We chose the two listed above.

STEP THREE: Implementation

We programmed our software to first edit the registry on the computer to prevent auto-run from loading any programs upon drive insertion. This prevents malicious software on the drive from infecting the computer. Also, our software scans the flash drive using a virus scan to determine the safety of the flash key.