ALMTEK *P.O. Box 6425, San Rafael, CA 94903 (415) 336-6173*

Fortress Mail©

Overview

The need for secure communications has existed for a long time. In the contemporary world this need has been filled, in a large part, by encrypted messaging and chat programs.

Extensively vetted and peer reviewed encryption algorithms are widely available in the public domain. Many working in this field assume that these algorithms cannot be broken even by those organizations with extensive resources. Although no one can actually prove that this assumption is correct, the author accepts that it is and the design of the device described in this document is based on this assumption.

Unfortunately, in the contemporary environment the existence of strong encryption does not ensure that an adversary cannot read the underlying encrypted messages. This is true because at the end points of the communications channel the message must exist in human readable form. This may (or most likely will) occur in an environment that can be compromised.

The desktops, laptops, and smart phones that are used as the human interface devices for communications are dependent on complex operating systems that are a major point of vulnerability. This vulnerability may occur as a result of undiscovered design errors that are inevitable in systems of this complexity, or through deliberate inclusion of “back doors” by designers in an often ill thought out effort to try to prevent pernicious use of a product.

The exploitation of these vulnerabilities is widespread. Production of virus and Trojan horse programs that can capture unencrypted text from an unsuspecting user has turned into a cottage industry. Organizations with extensive resources are rumored to have vast libraries of such programs that can exploit any hardware, operating system, and application program combination. Even relatively small players have developed impressive capabilities due, in no small part, to the large number of vulnerabilities in the target systems.

The second assumption made in designing the secure communication device is that all existing commercially available end user devices are not only vulnerable to the attack described above, but also that they are likely to have already been infected.

The answer to the problem of providing truly secure communications in the contemporary environment lies in a technique sometimes known as “Sand Boxing”. The name refers to a child who plays only in his or her own sand box and severely limits exposure to outsiders. In the context of interest, this technique translates into creating a hardware device which is designed from the ground up to be free of exploitable vulnerabilities at the time of manufacture and limiting data transfers into and out of the device in a way such that the integrity of the device can’t be compromised.

Fortress Mail

Fortress Mail© is a hardware device which implements the security strategy described above. Not only does the hardware and firmware design of Fortress Mail provide a secure environment, but also the human interface elements of the implementation encourage the end user to follow protocols that minimize operational vulnerabilities.

Figure 1 shows the general appearance of the device.

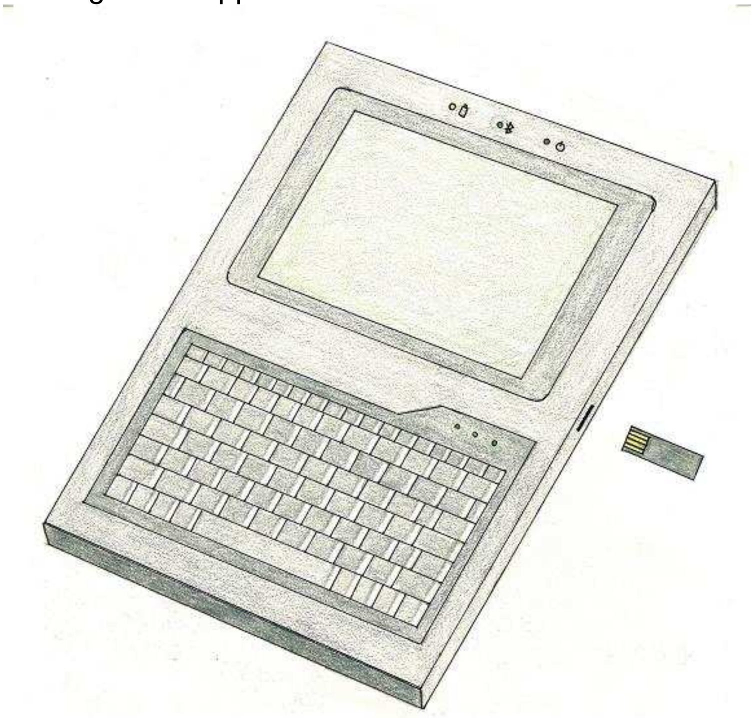

Figure 1

The device communicates with the outside world through a Bluetooth link. The encrypted messages are formatted as printable ASCII characters. Fortress Mail can be paired with and communicate with any Bluetooth capable device using a simple application which will be available for Windows, Mac OS X, Android, IOS, and Linux. ASCII formatted encrypted messages are the ONLY data that can traverse the link and a secure proprietary application layer protocol is used. Once the encrypted text has been transferred to the host system, it can be cut and pasted for transmission as a standard e-mail message over an insecure link.

The device screen displays both encrypted and unencrypted text. A message can exist in an unencrypted form only if it is being displayed on the screen. Only encrypted messages can be transferred into or out of the device. To save an existing message it must be transferred out of the device in an encrypted form so that it can then be stored safely in an insecure environment.

A spell checker is provided for messages that are composed on the device. The device is powered by rechargeable lithium ion batteries. For security reasons, Fortress Mail cannot be charged and operate at the same time.

Key Management

In a prototypical cryptographic system a message (plaintext) and a password (key) are operated on by a mathematical formula (cryptographic algorithm). The computation results in the production of an encrypted message (cyphertext) which may then be transmitted over an insecure communications channel. On the receiving end of the channel the intended recipient of the message reverses the process to retrieve the original plaintext using a related decryption algorithm and a decryption key.

In classic cryptosystems the encryption key and the decryption key are the same. This implies that the communicating parties must find a secure way to share the key before any communication can take place. Some systems use different but mathematically related keys for encryption and decryption. Fortress Mail uses the same key on both ends of the exchange.

In a well designed cryptosystem, detailed knowledge of the algorithm does not allow a would be code breaker to read a message. That is, lack of knowledge of the key is sufficient to foil any code breaking attack.

There is a belief held by many working in the field that any password that can be memorized by a human is not sufficiently secure. Fortress Mail uses numeric keys to encrypt messages. A key of 384 binary digits (bits) is used. This is the equivalent of a 116 digit decimal number.

The user employs the device to fill a hardware key module with random numbers. Each key module has a capacity of 65,536 bits or 19,728 decimal digits. The 384 bit message keys are automatically and randomly selected from the key module. An index number indicating which bits are used is included in the message so that the message can be decrypted by anyone possessing a copy of a given key module.

Keys are generated in the following manner: A blank key module is inserted into the device and “Create Key” is selected on the screen. A window will then appear on the screen and the user is instructed to touch 16 random spots within the screen area. The screen coordinates of the spots where the user touches the screen are stored. Since the window is 500 X 300 pixels in size there are 150,000 different coordinate pairs that can be selected. Sixteen selections give $150000^{16} \approx 10^{82}$ possible combinations. These numbers are used to generate the 65,536 bits which are stored in the key module.

Once a key module has been generated, it can be duplicated or loaded into the device to encrypt or decrypt messages.

Project Status

A fully functional prototype of the device has been constructed and is available to demonstrate all aspects of the design.

The next step is to design and build the production version of the device.