15 Sep 2019

SecurePi

Secure Internet Messaging

Overview and Use

Purpose

Establishes a secure and purposely ambiguous communications pathway between any two or more individuals, so that they may communicate with relative anonymity to third parties exceeding that of email. The system is simple to install and simple to use.

Physical Overview

This system is divided into three components:

- The message storage site is a publicly accessable website for storage of the encrypted message log. The user never communicates directly with the storage website, nor does he/she need to be aware of its presence or location. All communication with this website is done by the SecurePi Processor. Insomuch as the system provides end-to-end encryption, any reveal by the message storage site provides no useful data - encrypted messages arrive, are stored, and are sent without modification. The storage site has no knowledge whatsoever of message encryption/decryption methodology.
- The SecurePi Processor is a small stand-alone self-contained single-board computer, which is connected to the user's local network. It appears as a web-server to the user and as a client to the top-level website. All encryption software is contained within this unit. Other than its self-directed metered I/O to/from the top-level website, it is undetectable and inaccessable to anyone looking inward from the internet. This unit is the only portion of the system the user needs to physically install. There is no software installation required. Having all the cryptographic software located within this unit, prevents the security uncertainty of having any software on the user's computer.
- The user's computer is the lowest-level component of the system. The user will run his/her choice of web browser and logs on to the intermediate unit. He/she will respond to it as he/she would with any other interactive website. We must presume, and the user must assure, that no one can intercept messages between the user's computer and the SecurePi unit.

Functional overview

The SecurePi Processor was designed to be simple and intuitive to operate. The user installs the SecurePi processor onto his/her local network by making just two connections:

- Power provided by a common USB charger AC power adapter, and
- A local network connection using an ethernet cable.

No other connections are needed. Once powered, allow one minute for the unit to boot up. Then, the user may open his browser, type into the address line “rpi3”, and view the opening page of the

SecurePi. From here the user can just follow the instructions on each screen.

The SecurePi acts as a message handler, and an encryption machine. The unit interfaces the user's red-side (plaintext) to/from the user's computer, via the local network, and to the black-side (ciphertext) to/from the external internet. No red-side plaintext is ever exposed to the external internet.

The operating system, SecurePi software, and all security key material are all contained on one micro-SD card. This small (roughly 3/8” x 5/8”) card defines the functionality and security of the system. Should the physical security of the system become in jepardy, simply removing the micro-SD card secures the system. Better that the card be destroyed than be compromised - a revised card with updated keys can be installed later. All system and key updates are is performed by replacement of the one micro-SD card.

This system has amnesia. It never inquires, stores, or indicates any user's identity or IP address.

The SecurePi provides end-to-end encryption security. The system provides “broadcast” security, in that, each user has one transmit key and one or more receive keys. A user must have a receive key for each person from whom he needs to receive a message. Ordinarily, keys are exchanged between user pairs, but a system could be set up for one user to broadcast to many with or without a return path. Once posted, a message is available for all to read for 48 hours. Where message security is unimportant, plain text may be used by using the OrderWire function.

User Instructions

Presumptions

> The user's computer is power on, is freely able to access the internet via a local network that has at least one open internet port.

Initate operation

- Open your computer's browser.
- On the browser's address bar, enter the SecurePi's local address ('rpi3').
- You should immediately see the SecurePi main screen. From here, you can read or send messages.

Read an encrypted message:

- Select the [Select + Decrypt Incoming Msg] tab.
- Click [Find Incoming Msg].
- Find, select, and copy(^c) the message of interest. Note that messages only stay alive for 48 hours.
- Paste the message into a box.
- Click [Decrypt Msg]
- Go to the [Read Plaintext Message] tab.
- Click [Read Plaintext]
- Read your plaintext message.

15 Sep 2019

Send an encrypted message:

- Select the [Compose + Encrypt Message] tab.
- Compose or paste outgoing plaintext message.
- Enter password and click [Encrypt Msg]
- Select the [Send Encrypted Message] tab.
- Click [View Encrypted Msg]
- Verify the encrypted message leaks no plaintext, then enter password and click [Send].

Send a Non-encrypted (plaintext) message:

- Select the [OrderWire] tab.
- Compose or paste an outgoing plaintext message.
- Enter password and click [Send Non-Encrypted Msg]

Whenever a message is sent, a confirmation screen verifies the message was sent. The confirmation displays three items:

- the assigned message number,
- the posting date-time, and
- a reminder that the message will be deleted from the system in 48 hours.

The message number should be passed to the receiving party per an arranged method (a fixed schedule, phone call, SMS text, email, etc.) An incorrect password invokes, an "Entry Denied" warning, and no message is posted.

Plaintext Message Format/Content

To promote interesting uses, there are very few limits on the message. At this time, there are no limits on message size. (That could be a fatal flaw!) The screen for message entry is 81 columns wide. Continuing past that limit will wrap the screen, but it does not wrap the data posted to message center. (You must use an occasional carriage-return or your entire Log entry will be on just one very-long line!) SecurePi adds nothing to the posted message, including spaces, line feeds or carriage returns, as some encryption programs may be intolerant of ciphertext changes.

You are free to directly type your message into the message box on the screen or cut-and-paste it from elsewhere (like from Word, Notepad, or any text editor). The plaintext should consist solely of upper/lower case letters, numbers, and normal punctuation. It should contain no control/unprintable characters.

Ciphertext Message Format

The encrypted message automatically conforms to a format of 5-letter groups, 5 groups per line, and however many lines that are required. All letters are capitals. This format allows clarity of communication where alternate communication methods are allowable. Avoiding small letters, unprintable characters, numbers (dodging the number 0 to letter O ambiguity), makes hand writing/communicating a ciphertext message much easier and more reliable.

Incoming Message Format (Read All Messages)

At the top of the page, preceding the messages is:

- the present date-time,
- a reminder that messages older than 48 hours old have been deleted, and
- the last message number deleted (all prior messages will have been deleted).

Following this preamble is every message posted within the last 48 hours. Each message is preceded by its message number, and posting date-time. To facilitate a copy/paste into a decryption program, the message is contained within ■■ pairs. That is, the message starts in column 1 on the line immediately following the start-of-message indicator, "■■". The "end-of-message" indicator, "■■<<+", is on the line immediately after the last character of the message. (Any blank lines seen between the start-of-message and end-of-message indicators are part of the contained message.)

Each message is separated by one blank line.

Example of Log Format:

Wed, 3-MAY-2017, 19:02:45 z.
Messages auto-delete after 48 hours.
Message 1493665165 and prior deleted.

```
msg:1493743444:
Tue, 2-MAY-2017, 16:44:04 z.
■■
WINCB DSTNP QXMZX KDLAW DSOFG
HTISP LZMBW PAKSJ LCOER YZMCM
VNOIJ HENKO PQMAN CKLCV BFGQP
ETRUB XZMZN BCIOD OIOEU IQNDK
UIQAL CFEUB SPIOC JSOFD PEPLZ
■■<<+

msg:1493744122:
Tue, 2-MAY-20177, 16:55:22 z.
■■
This is an example message that I recently
submitted using the OrderWire function. It
obviously is not encrypted.
■■<<+
```

15 Sep 2019

Security Analysis

Comms Security: All plaintext communications between the user's computer and the SecurePi is via the local network. It is the user's responsibility to assure that the local network is inaccesssible to others during the period of use. Messaging between the SecurePi and the message storage website is always encrypted, so no extraordinary precautions are necessary.

Message Security: The SecurePi uses One-Time-Pad (OTP) encryption. There is no algorithm or "code" to break, hence, as long as the SecurePi is physically secure (i.e. the key files are secure) the crypto system is secure. The major complaint of crypto designers considering OTP systems is that it is difficult to distribute key material and the quantity of key material required can exceed the distribution channel. That is largely true when used to encrypt huge data files - for our messaging system sufficient key material is on hand to encrypt thousands of messages. For our messaging use, the distribution is by micro-SD card replacement. Micro-SD cards are very small and relatively easy to ship securely. Each micro-SD card contains enough key material to encrypt a total of one-million characters before requiring replacement.

Where message security is unimportant, a plaintext message may be posted (OrderWire), still maintaining anonymity.

Password: The password used to enter a message is unique to each user, serving to restrict use to one individual. The password also reduces the probability of a third party overloading the system, creating a denial of service for legitimate users. No password is ever transmitted, as it may lead to revealing the user's identity.

Addressee: There are no user-readable To/From headings. The site does not inherently know, therefore can never reveal the sender nor the recipient.

It is essential that the sender and recipient agree upon a method of recognizing each other's messages. Using this pre-arranged method is the only method a recipient can recognize a message is for them. When a user reads the incoming messages, all messages in the system are down-loaded, whether destined to a particular user or not. Hence, an observer cannot determine whether any, all, or no messages are addressed to the reader.

Message Available Announcement

One suggestion is to note the assigned message number and pass that number to the recipient, via some unrelated method, like via text message. The recipient then gets a "heads-up" that he should go to the SecurePi and find his message. Where security is utmost and no passing of a message number is possible, the recipient is implied as only he possesses the knowledge to decrypt the message.

Identity by IP: Realize that a real-time observer may determine (by IP address) who and when a user connects to with the message storage center. The observer may infer (by the quantity of data sent) that a message was sent, but can never be sure that a message was ever delivered.

Message Deletion: No one, including the recipient, is allowed to delete a message. If a recipient could delete a message after reading, an observer might correctly infer that the message was received, and determine which message went to whom. Messages over 48 hours old are automatically deleted by the system.