

Requirements for PassFree Devices

Cheyenne Software, Inc.

Table of Contents

Privacy.....	1
Security.....	2
Decentralization.....	3
Communication.....	4
Power.....	4
Form.....	4
Simplicity.....	5
Extended Features.....	5
Printing Digital Signatures.....	5
Pass-thru Access Control or Encryption Onion Layers.....	5
Expiring Cryptography Keys.....	6

Privacy

The user device must not emit any sensitive information about its owner. The best way to implement this may be to avoid storing any sensitive information about the owner inside the device. Generally security systems already have information about their users and need only associate a specific human being with a record they already have.

For example, a thumbprint-based device could store the thumbprint signature and the user's private key, and nothing else. It may even be possible to avoid storing the thumbprint signature by making it the pass phrase for the user's private key. This way, even if the device is stolen, the private key will not be usable because the thumbprint signature will be required to unlock it yet the attacker will not be able to extract the thumbprint from the electronic device. The user should frequently clean the surface of the device to avoid any oil stains that could allow an attacker to reproduce the thumbprint physically.

The authentication system must not be required to store sensitive information about its users. The only information that is required for

operation is a list of users' public keys, in order to provide a yes/no answer to a device asking if a given key is known.

Other information stored along with the users' public key can be managed according to the policies of the organization using the authentication system.

The user device must not emit any information at all about its owner (including echoing information in encrypted form) without express authorization of the owner. To alert its owner that authentication has been requested, the device may flash a light for a brief period of time, such as five seconds, vibrate, or play a recorded sound.

The user device should not emit any sounds at all unless the user has physical interface option to mute them. For human interface purposes, the reader should be the one to emit confirmation, warning, and error sounds. The obvious exception to this is blind people who will require their user device to emit sounds or vibrations.

Security

The device must not allow external devices to overwrite any part of its essential memory. It would be most secure to prevent this physically instead of by logic only. Physical prevention implies using read-only memory to store the user's private information. The next best thing is to require a wire connection in order to write, meaning the user will be physically aware that something may have the ability to write to the device if it is connected and does not need to fear any device that is not connected to his by wire.

Using read-only memory to store the user's information means that the registration must be done in person – user provides the thumbprint, and the company representative produces the read-only memory chip on the spot, and uses it to create the user device. It would be good to weld or melt the device shut. If the user's name changes (marrying women, etc) then the user must visit a representative to get a new user device.

Welding shut means that in order to fool a user into using a device with false information, the attacker must steal the user device, read the thumbprint signature from the read-only memory using a pirate device, and replicate the production process in order to produce a new user

device identical to the stolen one. Or, the attacker must be able to re-seal the user device well enough to be unnoticed.

Decentralization

It must be possible to set up a local authentication system that can use existing and pre-configured user devices so that users do not have to submit their device to reconfiguration when being invited to private locations.

In order to set up ad-hoc authentication systems, the device should routinely transmit its public key along with other information. This supports simplicity by avoiding the need for special protocols for requesting the public key, and allows security managers to create an ad-hoc authentication list with public keys ready for immediate use.

An ad-hoc web of trust can be created if user devices are able to sign each other's keys. A separate memory space should be allocated for trust signatures and after successful authentication they may be transmitted along with the signed name and public key. The process to accept a trust signature should involve consent from the user of the device (possibly a specific amber and green light flashing pattern prompting a thumbprint acceptance). The simplicity of the user devices means that the signing process must involve a third device that acts as a control panel, displaying to the users their trust signatures and guiding them through the process. As external writable data, the trust signatures must be regarded as disposable. Users may keep backup copies of trust signatures on their personal computers and use trust management software to select which trust keys to write to their user device at any time. Alternatively, the trust signing may be implemented as an accessory that does not require a thumbprint authentication: the user only needs to waive it near the power source and press a button (to prevent someone from covertly reading the user's identity by supplying power near the user's pocket) and the accessory will automatically power on and transmit its contents.

Communication

The user device should use one rugged communication technology and “add-ons” should be available to translate between the user device and any other communication technology.

A wireless technology should be used so that moving parts are not required and friction is not created when the user device is communicating with a reader. Readers can then have USB, radio, or any other kind of output to communicate with the system into which they are integrated.

Power

The user device must require minimal power for operation and be rechargeable. The device should accept wireless power transmission when available for operation and recharging.

Some lock devices should also accept wireless power transmission, such as lock devices used on rarely used or outside storage containers that do not have a permanent electricity source nearby or wish to conserve power by employing client-powered locks. To use such locks, the key device must transmit power or the user must carry a separate power transmitter.

Form

The user device should have a form that is easy to hold while pressing the thumb onto the thumbprint reader. The user device may also be designed using another biometric or a PIN instead of a thumbprint. The standard car remote keychain shape may be a good start because it is easy to press between the index finger and the thumb.

The user device should have a form that allows easy storage by a variety of personal security containers such as key chains, necklaces or amulets, utility belts, and pockets.

The shape of the thumbprint reader should make obvious the required thumb orientation (if any).

The form should allow for replaceable external parts such as thumb pad covers, waterproof cases, and chain attachments (ring or key chain).

Simplicity

Usage of the device must be simple.

Current recommendation: three lights colored red, amber, and green. Red indicates that power is available. The brightness of the red light should reflect the amount of power stored in the battery. Amber indicates that a request for authentication has been received. Flashing green indicates that the user's thumbprint has been authenticated. Solid green indicates that the authentication has been successfully transmitted to the reader. All lights should turn off after a brief period of idleness or after a successful transaction (for example, two seconds of solid green and then all off).

In order to set up ad-hoc authentication systems, the device should routinely transmit its public key along with other information. This supports simplicity by avoiding the need for special protocols for requesting the public key, which prevents user interface complexity (no additional lights or buttons required).

Extended Features

Printing Digital Signatures

To support use case of additional authentication for notary and electronically signing paper documents, a printable digital signature format can be employed, such as extended barcode technology that prints compactly on paper yet provides enough digital space to print the electronic signature block, so that it can be easily read by an electronic scanner and passed to an encryption device for verification.

Pass-thru Access Control or Encryption Onion Layers

To support use case of access control device replaces traditional credit card reader: the lock-key communication protocol must allow the lock to provide its own public key to the key device as well as the public key to which it prefers that the key device encrypt the communication. This will allow the key devices to confirm it's still talking to the same lock but that the lock will be passing the information thru to another access control device.

Expiring Cryptography Keys

Because the user's public key will end up on a variety of databases, and will eventually be accessible to people who try to access it (by stealing a database, or by installing a rogue lock in a public place to just initiate the protocol and record people's public keys and then drop the communication) it must be assumed that from the moment of generation, a user's key-pair has a useful life limited by the computing power available to the bad guys to derive the private key and by the strength of the algorithms currently in use. A device infrastructure that doesn't make it easy for people to change keys will eventually cause a situation where many keys are at risk of compromise and likely to become a problem because users will inevitably take the risk instead of struggle with a difficult change/upgrade process, both technically and administratively. So the key devices and the locks all need to support changing their key-pairs.

User devices can either be made resistant to new keys (see the security section above) or with a function on them that creates a new key pair and then signs it with the current keys. Then each service that maintains a public key database must support a key update where the user initiates an update, then the server sends a request encrypted to the original key asking for the newly signed key, then the user answers with the key device. Then each service would automatically associate the new key with all of the old key's records. Upon creating the new key, the key device will automatically start using it to answer all challenges, but it will keep the old key in its memory until it expires. If a lock denies it access with the new key, the key device can automatically try again with the older key (until that older key expires). Locks should note this and make it easy for their administrators to change the ACL to list the user's new key instead of the expiring key.

For more information about improved electronic access control using PassFree, contact:

Cheyenne Software, Inc.
(800) 935-9637
someone@mypassfree.com