

Use Cases for PassFree

Improved Electronic Access Control

Cheyenne Software, Inc.

Table of Contents

Case #1a: Authentication before using a credit card in retail store	1
Case #1b: Authentication before using a credit card in retail store, and second authentication by the credit company.....	4
Case #2a: Key device registration at CC company and transaction signing with merchant or separate device.....	5
Case #2b: Access control device replaces traditional credit card reader and provides customer's personal payment options from bank	8
Case #3: Additional authentication for notary public	9
Case #4: Replacement for mechanical lock and key.....	10
Case #5: Integrated replacement for Computer Access Cards	11
Case #6: Replacement for using SSN as an authentication token; SSN reverts to being just a number.....	11
Case #7: Key-signing	11
Case #8: Signing Electronic Documents	13
Case #9: Signing Electronic Forms.....	13
Case #10: Decrypting emails.....	14
Case #11: Website Login.....	14
Case #12: Securing Electronic Voting	15
Case #13: Securing Medical Records.....	15

Case #1a: Authentication before using a credit card in retail store

Concept: An identity-verification center compares electronically submitted credit card numbers and user names to registered numbers and names on file and reports electronically whether the key-pair that signed the submitted credit card number matches the public key assigned to that credit card number in its files. A merchant would query the center for identity verification prior to submitting a customer's payment information to the existing credit payment centers. No changes are required to the credit payment centers and therefore we are not in competition with that business. The credit companies may elect to compete for identity verification as well by offering an integrated solution similar to one of the other use cases (where the public key information links to payment information and user only has to select a payment method). The user must register his/her credit card numbers and public key on the independent identity verification center's website prior to attempting authentication with this method.

Prequel:

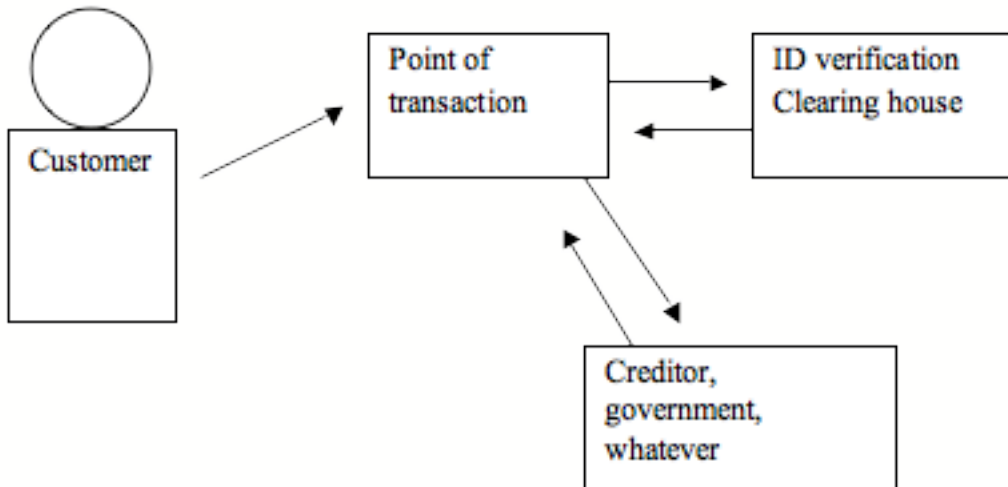
1. User obtains credit card and key device
2. User visits independent authentication center's website and purchases public key registration
 - a. Authentication center presents electronic form to user containing fields for name, credit card number, CC expiration, CC PIN (3 or 4 digits printed on card), billing address, telephone number, and email address
 - b. User completes form and signs it using private key on key device (computer hardware required either at home or at local service center, see usage case #8, signing electronic forms)
 - c. Authentication center charges credit card for service and to confirm the number
 - d. Authentication center conducts credit check, local background check, email confirmation, telephone confirmation, and postal mail confirmation (asynchronous)
 - e. Authentication center registers user's credit card number (but not the rest of the credit card information) and the user's public key in a database.
 - f. Authentication center notifies the user that the credit card has been registered
3. User receives notification that credit card and public key have been registered

Scenario:

1. User walks into store
2. User selects items to purchase and gives credit card to cashier
3. Cashier totals the user's purchase amount and enters this amount into the access control device using a keypad or automatically from the register
4. Cashier enters credit card number into the lock device using a keypad or by swiping it
5. Access control device is powered on
6. User's key device sends its public key as a ready-to-receive signal
7. Access control device declares a lock having the cashier's number (12, 13, 14, etc) and the user's total purchase amount, and the only action being to authorize the purchase;
8. Customer presses thumb onto customer's own key device
 - a. Key device checks thumbprint against internal signature, and if matching it will allow use of the private key for authenticating to the access control device
 - b. Key device sends answer to the lock device
9. Lock device queries independent credit card authentication service using the signed credit card number and total purchase amount as well as the current date/time
 - a. Independent authentication service queries database to verify credit card number and public key are associated and that the signature matches the public key
 - b. Independent authentication service returns a yes/no answer without any further details about the user, but only if the current date/time in the signed

input is within the last 15 minutes, in order to prevent replay attacks; the service returns yes if and only if the card number and public key match and the signature is valid and the time in the request is within the last 15 minutes

- c. Cashier lock device displays response and emits appropriate wire signal ; response includes an authentication receipt number
10. Cashier now has one form of authentication and can employ a second traditional form as required by local policy (checking license id with credit card name, for example)
11. Cashier proceeds to charge customer's credit card using existing system
12. Cashier acknowledges payment



Script:

Customer: I want to buy this sweater, here's my credit card.

Point of transaction to ID people: Here's a public key and a card number (signed by the private key), is this really the public key for this credit card?

ID verification clearing house: Yes it is (and your verification receipt is # 1234)

Point of transaction to creditor: Here's a card number, I want to charge \$25 to it

Creditor: Approved (based on their independent rules)

The verification receipt number would be like our version of a credit authorization number... if someone has a question about the transaction, we look up the date and time and this number to find our records for the transaction they are asking about.

Privacy: No information about the user is disclosed; in order to query the database, a

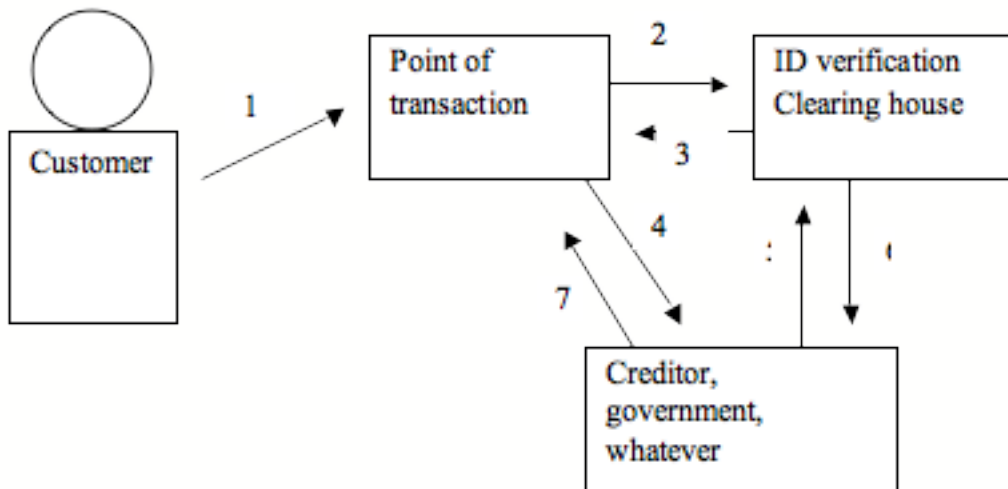
merchant must have the user's credit card number and current date/time signed by the user's private key, as well as the user's public key advertised from the key device. If the database is stolen from the authentication center, the thief will have only credit card numbers and public keys – by itself not useful enough to fake online purchases but it may be possible to then correlate public keys to people names. For this reason, this use case is not as good as use case #2 where the credit card company is directly involved. However, it does support repudiation for victims of credit card fraud, because if they use the key device for purchases then they have an easy way to separate legitimate purchases from fraudulent purchases. Obviously in order to make that practical, enough merchants have to adopt the technology so that customers can make most purchases using the extra authentication.

Case #1b: Authentication before using a credit card in retail store, and second authentication by the credit company

Concept: Same as #1a but with the additional step that the cashier sends an authentication receipt number, received from the independent authentication center, to the credit card company along with the request for credit card charge authorization. The credit card company contacts the independent authentication center and asks whether an authentication request with the given number was approved within the last 15 minutes, and uses the answer in determining whether to authorize the transaction.

Prequel is the same as use case #1a.

Scenario is the same but in step 11, "Cashier proceeds to charge customer's credit card using existing system", cashier also transmits the authentication receipt number received in step 9c, "cashier lock device displays response and emits appropriate wire signal". The credit card company can then check the authentication response number with the independent authentication center.



Script:

The extra step is that the point of transaction also forwards our verification receipt # to the creditor along with the charge request, and the creditor contacts us to ask if we recently issued a receipt #1234 for a credit card #6789, and we say either no or “yes, in the last 15 min.” We don’t say exactly when because that would be giving out info about our customers.

Case #2a: Key device registration at CC company and transaction signing with merchant or separate device

Concept: Credit card company allows users to register their key devices and declare they want the CC company to only authorize transactions that are signed or that were preceded by a separate key device signature (user or bank can set expiration... I suggest about 5 minutes) ; signed transactions can be done wherever the merchant or website supports the technology (our point-of-sale equipment for merchants or website plugins for e-commerce), and preceding signatures can be done by the user himself if he carries another device for contacting the CC company independently to pre-authorize transactions. This provides 100% credit card fraud protection to the user because it will not be possible for anyone else to charge the user's credit card with the user's info without having the user's key device (private key) when the user has enabled protection at the credit card company website.

How it works, setup:

1. user purchases a smart phone – iPhone or Blackberry for example
2. user logs on to credit card website and clicks a link to register his key device
3. website provides a download link to an application for the user's smart phone (our software, customized for this use case)
4. the application will be pre-configured with the user's credit card website from where it was downloaded, so no further setup will be necessary

How it works, regular usage:

5. whenever the user wants to make a purchase with his credit card, either at a store or online, he just opens this application first on his smart phone and presses a button. the smart phone will use its data network to contact the credit card website and sign a credit card authorization form (all automatic, nothing required from the user other than the initial button press) ; the website will then enable the user's credit card to be used for the configured time period ... 5 minutes or so.

Options: users with credit cards at multiple companies that offer the same feature would get a separate pre-configured application for each one. so it would have a different icon and it would be clear which credit card he's "turning on" for 5 minutes. or we could make the application configurable by the user, to add and remove credit card accounts.

Privacy: Credit card information is still vulnerable from the credit card itself as always but can no longer be used fraudulently when the user enables 100% protection at his bank's website. No credit card information is stored on the user's smart phone, it's still just public/private key-pairs for us.

Advantages: This credit card fraud protection may be really attractive to smaller credit card companies, who can offer this feature to differentiate themselves. I expect all the credit companies will want to do this when enough users demand it, so we can get into the market and prove the idea with a smaller company if the big ones won't see us initially. Even the first user will have 100% protection, no need to wait for infrastructure or widespread deployment besides user and his credit card company. Over time merchants will install the cryptography-capable point-of-sale devices and users will not have to carry the separate pre-authorization device with them when conducting their day-to-day business. This may be the quickest path to real protection from credit card fraud by making it so credit card information by itself will not allow criminals to conduct

transactions with someone else's name.

An announcement on Apple Inc.'s website on January 22nd, 2008 said:

Apple reports best quarterly revenue and earnings in its history
Announcing financial results for its fiscal 2008 first quarter, which ended December 29, 2007, Apple today posted revenue of \$9.6 billion and net quarterly profit of \$1.58 billion, or \$1.76 per diluted share. These results compare to revenue of \$7.1 billion and net quarterly profit of \$1 billion, or \$1.14 per diluted share, in the year-ago quarter. In attaining its highest revenue and earnings in company history, Apple shipped 2,319,000 Macs, a 44% unit growth and 47% revenue growth over the year ago quarter; sold 22,121,000 iPods, representing five percent unit growth and 17 percent revenue growth over the year-ago quarter; and sold 2,315,000 iPhones in the quarter.

So in 2008 there were over 2 million (probably 3 or 4) iPhones out there. RIM has sold a total of 20 million Blackberries by 2008 and has more than 11 million Blackberry service subscribers. So that's a lot of millions of people who would already have the hardware to enjoy 100% credit card fraud protection. And if someone doesn't yet, I think what we have might be a pretty compelling reason to get a smart phone...

So what happens if you are in a place that doesn't have cell service (or abroad, where your phone doesn't work)? I guess if you're planning a trip to a place outside your cell phone coverage area you might want to go to your bank's website and disable the 100% protection temporarily by un-checking the box. But you'd want to turn it on immediately after coming home because trips abroad and other irregular events pose the highest risk for credit card details being stolen.

Or you could rent a local smart phone during your trip, get online from somewhere during your first day to register it (with its own private key -- not your original!) , then use that one to pre-authorize your transactions until you're ready to travel home. Carry cash or travelers checks during travel days to cover you between using your own smart phone and using the rental.

But I don't know if there are smart phone rentals here or in Europe. Probably "no" everywhere else in the world. Don't travel to Nigeria if you're worried about fraud...

Case #2b: Access control device replaces traditional credit card reader and provides customer's personal payment options from bank

Concept: Users do not have to use a traditional credit card. They use only their key device to make purchases. Users must register their key device on their bank's credit card website in order to link their public key to their payment information. Users may register multiple credit cards to the same public key and assign labels to them for displaying on their key device in order to select a specific card number to use for each purchase. There are two possible deployments of this concept: either the credit card companies adopt the technology and use it as a drop-in replacement for existing readers, or a new company is formed to directly compete with credit card companies and provide a payment gateway to merchants – a “PGP key company” instead of a “credit card company” – so this new company would be a drop-in replacement for all existing credit card companies and their infrastructure. PayPal may be interested in this in order to increase its prominence and compete directly with the credit card companies.

Prequel: Customer registers payment information and public key on payment authorization website (either credit card company or competing payment company). Merchants configure their access control devices to with the public key of their assigned authorization server.

Scenario:

1. Customer walks into store
2. Customer selects items to purchase
3. Cash register transmits total purchase amount to the access control device (can use same protocol as for the regular credit card reader, to provide drop-in replacement capability)
4. The machine sends a challenge to near-by keys including the store name, register number, and total purchase amount + the access control device's own public key (because access control device must later verify that the amount is correct before forwarding to the authorization center – don't want rogue key devices to sign amounts lower than the cashier's total)
5. The customer's key displays the name of the lock (store name + register number + purchase amount)
6. Customer presses authentication button (or thumbprint, depending on key model)
7. Key transmits signed and encrypted answer to the lock + a copy of the key device's public key
8. The lock sends the signed answer (re-encrypted to the authorization server's public key) and the user's public key to the authorization server. The authorization server looks up the user's account based on the provided public key (user must register public key at his/her bank's website before using).
 - a. If an account is not found, of course the transaction is rejected.

- b. If the account matches and contains only one registered credit card, the authorization backend sends an authorization number to the machine for transmittal to the store register (same as for the regular swiper).
 - c. If there is more than one credit card registered, the authorization center sends a list of credit card numbers (last 4 digits only) encrypted to the key's public key and the machine relays these to the key. The key displays the list to the user and the user selects a credit card to use. This selection is encrypted to the authorization center's public key and is relayed through the access control device to the authorization center. The authorization center then verifies the selection and if all is ok then it sends the authorization number to the access control device for transmittal to the store register (same as for the traditional credit card reader).
9. Cashier acknowledges payment

This case puts the device as a much-desired security upgrade to credit card companies, who can buy the components and use them as drop-in side-by-side alternatives to regular swiping.

Privacy: User name and payment information are known to the user and to the credit card company and are NOT disclosed to any third party. Payment options are NOT stored on user device or shared with any entity besides the user and his/her bank, eliminating the possibility of en-route compromise. The authorization center is the same as the one that processes credit cards already. Merchants cannot implement man-in-the-middle attacks because the authorization server requires the user's public key to look up account information, and then encrypts payment options to that public key. If the merchant tries to substitute his own key (in order to intercept account information) he will fail because the authorization server will not find the customer's payment info without the customer's key. The payment data is then encrypted to the customer's key so it cannot be read except by the customer's key device, which will receive in the signed/encrypted message the authorization server's public key, and will sign/encrypt the response with that key so that the reply cannot be intercepted.

Requirement: 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.

Case #3: Additional authentication for notary public

Concept: a notary public can accept stand-alone electronic signatures or perform additional authentication with an external authentication service (see use case #1a but instead of registering user's credit card, the user registers his/her name with the public key) by submitting the user's name as appears on user's ID to the user's key device to sign, and then submitting the signed name to the authentication service.

Prequel:

1. User registers name as appears on user's identification card (driving license) on the authentication service website with user's public key

Scenario:

1. User approaches notary public with valid identification card
2. Notary swipes identification card on his own computer to provide information to notary's own lock device (alternatively, notary can type in the user's name as it appears on the id card)
3. Notary's lock device declares itself with an action to sign user's own name
4. User signs name with key device
5. Notary records stand-alone signature in his records
6. Optional: Notary can then send the signature to the authentication service, which will look up the user's public key and then compare the signed name to the names associated with that key in the database. If there is a match, an approval is returned.

Privacy: User name is NOT disclosed by authentication service, but must already be provided by user / notary in order for user to sign his/her name.

Requirement: to support electronic signatures on paper documents (this is optional, because without it the notary can just keep an electronic record of them, or worst case just print out the electronic signature and attach to document, and it can be typed in and verified at any time) we may want to employ some sort of 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.

Case #4: Replacement for mechanical lock and key

Any lock on a door or container can be replaced with this authentication system. In this scenario, an external authentication agency (us) is not required. We can provide the service to people who prefer to use us as experts, but it should be possible for people to set up their own localized authentication center. In this way, the technology can spread in small steps and it is useful to individual people even if the rest of the population doesn't use it.

A door lock. The electronic lock is installed in the door of a house or car or room. The lock does not require an external face so the door is smooth except for the handle. The owner of the car approaches, reaches for the key device (either a stand-alone device or a smart phone or PDA with the key software) and presses a button to unlock the door. The door authenticates the request by looking up the user's public key in a list of authorized users (the owner, his spouse, and his teenage son) that was defined previously, recognizes

the owner, and unlocks the door.

Privacy: No information about the user is disclosed.

Case #5: Integrated replacement for Computer Access Cards

Replacement for magnetic or electronic Computer Access Cards. Current infrastructure can stay in place. Only the CAC reader and driver software needs to be replaced. For applications that require more than one authorized user of the same certificate, CACs are convenient because you can hand yours to someone else.

Privacy: No information about the user is disclosed. System must be able to look up the public key in a database in order to obtain any other needed information about the user.

Case #6: Replacement for using SSN as an authentication token; SSN reverts to being just a number

Concept: government can install an authentication server as in use case #1a, where the public keys are associated with the social security number. Again as in case #1a, queries must be in the form of a social security number signed by the user's key. If the public key is in the database and the signed SSN matches the SSN in the database, an approval is returned. Otherwise a rejection is returned. Either way no information about the user or any record in the database is disclosed which was not already present in the query.

Case #7: Key-signing

At a key signing party (minimum two users, but the snacks and drinks are optional), members bring their key devices and identification. The key device models suitable for this usage case have a display so the owner can view other key devices nearby and, on his or her own display, inspect their public key fingerprints and other public information.

The procedure would be like this:

1. Everyone sends an email signed with their key device to the party host
2. The host adds the public key of each email received to an access control device
3. The host adds the public key of each email received to a key ring (or exports the key ring from the access control device) and emails the key ring to every

- registered participant
4. The host prints out a list of every public key received, along with its fingerprint and owner name
 5. Each participant adds the key ring received by email to his or her key device as non-trusted keys
 6. To gain entry to the party, every participant must authenticate to the lock with his or her key device – gaining entry proves that the holder of the key device is the person who registered for the party (unless the lock has been rigged to allow impostors); after entry, each participant places his or her key device in “party mode”, where it is looking for keys as well as locks, and for each key or lock selected it displays the fingerprint, a checkbox for verifying the fingerprint, and a checkbox for verifying the identification.
 7. Participants socialize at the party with as many people as possible; at each meeting of two people, they select each other on their key devices and verify the key fingerprints and each other’s identity. After establishing trust that they are who they say they are, they use their devices to sign each other’s keys.
 8. Upon leaving the party, participants again authenticate to the lock. At every authentication, key devices transmit their public key as well as signatures, so at this second authentication, the lock also records the signatures.
 9. The party host compiles all the public keys with signatures into a key ring (or exports it from the access control device), signs it and sends it out to participants.

This use case is a probe – it may turn out to be a bad idea, and even if it’s a decent idea it may be a low priority for implementing required features to realize it. However, after a second look it appears to be critical for implementing the document-signing feature because it requires user key devices to verify a web of trust and that means they need to be able to download new keys and key signatures to their key-ring – which is what the key-signing use case suggests. The download need not be direct – there isn’t a need for the key device to accept connections directly. To exchange keys, two key devices can both upload their keys with signatures to a lock and then download updates from the lock. This would happen as described in this use case.

Maybe positioning this case as a party is not appropriate. Maybe the use case should be for two people in private, signing each others keys, and then if someone wants to make a procedure for a party (such as in the reference below) they can do it with or without the devices. But I think it would be useful for just two people to be able to sign each other’s keys, as an ad-hoc infrastructure building activity.

For reference, instructions for such a party without hardware:

http://cryptnet.net/fdp/crypto/keysigning_party/en/extra/annc-example.html

<http://www.nylug.org/keys/>

<http://www.chaosreigns.com/code/sig2dot/>

Requirement: the key device must be able to download signatures from trusted locks (or any lock with user confirmation), and the user’s key device must be able to show the user the fingerprint of his or her own key in hexadecimal and in PGP odd/even keywords in

order to read or show them to a person with whom the public keys are being exchanged and signed.

Case #8: Signing Electronic Documents

A person may use a lock connected to computer by USB to sign an electronic document on the computer. Verifying web of trust path is optional but user's key device might show the "sign" button in a different color based on the trust path: green means the path is complete and verified, amber means the path is incomplete or unverified, and red means that the key of the lock or other signers has expired or has been revoked.

As an optional step, the signed document can then be signed by a notary and stamped with the current date and time. The notary can be there in person (to also verify photo id etc) or can be online.

Requirement: To support web of trust verification by the user's key device, the key device must be able to download signatures from trusted locks (or any lock with user confirmation), and the user's key device must be able to show the user the fingerprint of his or her own key in hexadecimal and in PGP odd/even keywords in order to read or show them to a person with whom the public keys are being exchanged and signed.

Case #9: Signing Electronic Forms

This is similar to case 8 (signing electronic documents) but is done online. At any website, a user may be presented with a standard HTML form. The form contains an additional input element marked as the form signature, with possible supporting elements to indicate algorithm and format info. The additional input elements may be rendered by the browser however is most appropriate for the user's system. Upon submitting the form, the browser will compile the form elements for submission and then prompt the user for a signature.

Computers having access control device hardware can communicate with the user's key device and obtain a signature by declaring a lock having the label of the web page or form name and the only action being to sign (since encryption is done by the SSL without needing the keys). The user can then act on the key device to sign. The lock would automatically approve the action since it's a one-way submission.

Computers without access control device hardware can show a popup containing the compiled form information and an input area for the signature. The user can then use the computer's PGP services to sign that form and paste the signed form into the input area. PGP Desktop users would find this very easy since it has a keyboard and menu shortcut to sign any highlighted text.

Case #10: Decrypting emails

Users of this invention can receive encrypted emails anywhere – they log on to the computer, download the encrypted email from the internet, then their email reader should try to use a keyring on the computer to decrypt it and when it does not find one, it can ask the user whether the user has a key device capable of decrypting the email, if the user clicks yes then the email reader will communicate with the USB access control device to create a “lock” that the user's key device can then connect to and request a list of actions, the list will include “decrypt” and when the user selects this action, the lock will provide the encrypted material to the key.

There are two ways to handle the decrypted material, and we'll probably have to select one or design the system so both can be used:

First, the key can KEEP the decrypted material, displaying it to the user on its own monitor. The key would NOT send any decrypted material to the lock/computer. This is advantageous for the super paranoid who may not trust the computer system they are using, and who have a key device capable of adequately displaying the document they have decrypted. This method may impose a practical limit on the size of documents that can be decrypted and displayed, or allow only documents that can be viewed in a streaming mode, such as pages of a PDF or webpage, slides of a presentation, etc, because of the memory and CPU power available to the key device.

Second, the key can SEND the decrypted material back to the lock/computer. The material may be either the actual decrypted document, or only the decrypted symmetrical encryption key which can then be used by the computer to decrypt the actual message. In practice, I expect it would be the symmetrical encryption key because that allows messages to be encrypted/decrypted faster by the computer, and to be encrypted to multiple recipients using PGP, etc. This method assumes that the user will not choose to decrypt messages when using a non-trusted computer, because then the decrypted contents will be available to malicious software on that system.

Some customers may prefer one method over the other, and this calls for a varied product line that meets the very specific needs and preferences of its users.

Case #11: Website Login

Any website can implement a simple and secure login using the key device and the browser plugin, in the same way that the authentication center of use case #1A obtains the user's electronic signature.

This is also very similar to the electronic signature use case #9.

This is just a slightly different use case because it covers the login phase of a website and can be developed as a library in several languages (Perl, C, Java, Python, Ruby, ...) which can then be installed by a website and used as a drop-in replacement or side-by-side alternative to their traditional login options. It's also a separate use case because it represents an entire new arena of convenience for the user... it would be great to be able to login to all my website accounts with my key device instead of having to remember or keep a database of passwords for all the sites.

For example, a website might have a traditional username/password box with a link below it labeled "login using your personal key". When the user clicks the link, the website displays the signature request page containing a challenge, which the browser plugin intercepts. The browser plugin then creates a new virtual lock on the computer's lock device to represent this electronic form and asks the user to press the button on his/her key device if s/he would like to sign the form and login to the website. The user can press the key or cancel, and the browser notifies the website of the action. The website can then authenticate the user and create a regular session (as it would with a username/password) to use from that point on with the browser.

Case #12: Securing Electronic Voting

At the voting booth, voters can be authenticated using their private key devices and a clearinghouse for verifying social security numbers.

Case #13: Securing Medical Records

At a hospital, administrators, doctors, and nurses can be routinely added and removed from being able to access patient files.

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

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