VIRAL ATTACKS ON THE DoD COMMON ACCESS CARD (CAC) [1]

Partha Dasgupta, Karmvir Chatha, and Sandeep K. S. Gupta
Department of Computer Sc. & Eng.
Arizona State University, Tempe AZ
{partha, karam.chatha, sandeep.gupta}@asu.edu

ABSTRACT

The DoD CAC (Common Access Card) is a PKI-enabled smartcard that provides the following functions: Authentication, Data Integrity, Confidentiality and Non-repudiation. Since the private key of the client certificates are stored in the card, and this key cannot be extracted from the card, it provides a high degree of security even when the card is used on a untrusted workstation (or point of sale).

This paper shows that using a DoD CAC on a untrusted workstation can allow a variety of attacks to be performed by malicious software. These attacks range from simple PIN phishing, to more serious attacks such as signatures on unauthorized transactions, authentication of users without consent, unauthorized secure access to SSL enabled web servers as well as remote usage of the DoD CAC by attackers. We also show the root cause of such problems is the lack of a secure I/O channel between the user and the card and outline steps that can be taken to ensure such a channel is available making the documented attacks not feasible.

INTRODUCTION

Authentication, Data Integrity, Confidentiality and Non-repudiation are the core functionality needed for a variety of official and personal functions that a human being performs [Lop01, MBP03]. The applications range from electronic account management to building entry, from e-mail to financial transactions, from identity management to confidential information storage.

As is well known, the current system of using usernames and passwords do not work in a secure environment. The shared secret schemes that form the basis of such authentication systems are much too prone to attacks such as phishing, spoofing, eavesdropping, simple social engineering and data theft. Shared secret based multi-factor authentication schemes are considered to be better, but they lack the features of data-integrity and non-repudiation (actions are not secured by signatures and hashes).

It is also well known that system utilizing public key systems and certificates are well suited for all the above functions, provided there is a secure method of ensuring the private key remains private. The DoD Common Access Card (CAC) is a particular implementation of a Public Key Infrastructure (PKI) based solution that provides all the above functionality [Rig03, NISTPKI05, LiMi03, Th84, RFC2704, NIHPKI97].

As an important component of its Defense-in-Depth strategy, DoD is moving away from vulnerability-prone user name password based access control to a hardware token, certificate based access control. Several well-known vulnerabilities in password based access control is bringing insufficiently protected computers on the unclassified but sensitive Internet Protocol Router Network (NIPRNet). **Homeland Security Presidential Directive** – 12 (HSPD-12) (Subject: *Policy for a Common Identification Standard for Federal Employees and Contractors*) [HSPD-12] mandates all federal government agencies to begin utilizing strong and reliable form of identification for physical and local access to federal buildings and computer systems that is *"(a) is issued based on sound criteria for verifying an individual employee's identity; (b) is strongly resistant to identity fraud, tampering, counterfeiting, and terrorist exploitation; (c) can be rapidly authenticated electronically; and (d) is issued only by providers whose reliability has been established by an official accreditation process*." [HSPD-12]. An underlying goal is to protect the computer systems against viruses and prevent the resulting costly clean-up effort (saving army-wide IT

[1] This research is partially supported by grants from the National Science Foundation (CNS-0617671) and The Consortium for Embedded Systems.

personnel time and lowering operating budget) and consequently enhancing system availability and operational readiness.

The public key operations used in the CAC are the basis of its security (along with physical tamper resistance and portability). It is assumed that the CAC will provide secure access and authentication even when the host that it is connected to, cannot be fully trusted. In this paper, we show that this is not indeed the case, and a set of viral attacks can be launched against the CAC when there is some malicious software runs on the host computer. We also discuss methods of protecting against such attacks.

THE DoD CAC

The DoD has adopted an X.509 standard based hardware token called the DoD Common Access Card (CAC). The DoD CAC is a non-tamperable, credit card sized smartcard that can be plugged into a reader connected to a client PC. The card stores the user's DoD PKI certificates and public/private keys. The card has on-board computing power and can store certificates, provide PKI based challenge response processing and perform digital signature operations. The card features a 32-bit RISC processor and embedded software that can generate public-private key pairs on the card and perform PKI operations *without exposing the secret keys to the host machine*. The card is secured to a particular human via one or more PINs (could be biometric) to protect against theft.

The CAC card is being heavily deployed, about 3.5 million have been shipped to individuals, and the final total is expected to be about 4.5 million. The cards will be used by DoD affiliated personnel for physical security applications such as building entry and for web enabled applications such as digitally signed email and documents. *The host based software for the card can be expanded to provide support for consumer computing such as login, single sign on, secure signed credit card charges, and so on.*

As an example the DoD Card can be used to log on (e.g. to Active Directory on Windows platform) instead of using the standard username and password method. This DoD CAC based login procedure is more secure than simply using username and password since it employs two-factor authentication: DoD CAC ("something the user has") and a 6-8 digit PIN ("something the user knows").

The details of the login procedure are provided in the Army CAC Cryptographic Logon Implementation Plan [CCL]. This plan applies to only Window-based workstations (referred to as client PC below) that are networked to the NIPRNet with personnel having access authority using a Defense Enrollment Eligibility Reporting System (DEERS)/Real-time Automated Personnel Identification System (RAPIDS)-issues CAC containing 3 X.509 digital certificates. This procedure is expected to be implemented by June 2006.

The CCL procedure consists of inserting the CAC in a reader attached to a client PC (or Workstation/Thin Client) and providing a PIN when prompted by the PC. The PIN in essence "unlocks" the CAC and gives the client PC access to the user's X.509 certificate *ucert* (using user's public key K_{pub}) and user's private key (K_{pri}).

In particular, the client PC forwards the user's logon request (which consists of among other things *ucert* and an authenticator *auth* digitally signed using K_{pri}) to the KDC which does the following: 1) verify the certification chain/path of the user's certificate to ensure that it can be trusted. This involves verifying the *ucert* has not expired and contacting the Certificate Revocation List (CRL) Distribution Point (DP) (for DoD the CRLs are located at a public LDAP location operated by DISA); 2) verify the digital signature on the *auth* using K_{pub} from *ucert* – this establishes that the request originated from the user; 3) obtains the user's account information, e.g. from active directory, based on the Principal Name specified in the Subject Alternative Name field in *ucert*. For DoD CAC issues by DEERS, the principal name information is the EDI-PI@mil in the CAC; and 4) constructs a TGT and session ticket using the user's account information and sends it to the client PC encrypted using Kpub. The client PC learns the KDC's secret key by decrypting the TGT using Kpri. Further, it validates that the reply is from trusted KDC using a procedure similar to that used by KDC in step 1 above.

TRUSTED AND UNTRUSTED COMPONENTS

The CAC is a trusted component, that is in the threat model for DoD applications, the CAC is assumed to be untamperable and immune to software attacks. The CAC stores the private keys of all certificates issued to the user and the CAC performs PKI operation (challenge-response, signatures and so on) when requested to do so. The operations are done in the processor inside the CAC and hence the keys are never made visible to the host computer connected to the CAC. This ensures security of the credentials of a user.

The important question is whether the host computers running the CAC application are trusted or not. If we assume that all the computers that a user inserts a CAC into are secure and trusted, then we can argue that there is no real reason to have a processor on the CAC. The CAC could reveal the credentials to the host computer, which then can perform the PKI operations. In addition, we can use many symmetric key operations for signatures and authentication based off of a multifactor password, making the deployment and manufacture of the CAC much simpler. *In fact the DoD CAC design pre-supposed that the workstations are not secure*.

Assuming the trusted nature of every host computer (or terminal, or point of sale) is naïve. According to an article published in the USA Today on June 13th 2006, Microsoft has revealed that 62% of all Windows computers are infected with at least one "backdoor Trojan". Hence, for example, when a DoD personnel inserts a CAC into his or her home PC, it is quite possible that the home PC has a viral infection inside it. Hence the home PC is not to be trusted. Now when the DoD personnel uses the CAC to perform email signatures or authenticated SSL connections, he or she is open to a host of nefarious attacks that we discuss in the next section.

The CAC *seems* to be safe even when there are vial agents on the host computer. This appearance comes from the assumption that since the private key for the user certificates are stored securely in the CAC, and the CAC never exposes them, an attacker will never be able to get access to the private key. This is indeed true.

However, the CAC is not secure from a host of attacks where the attacker does not need to physically obtain the private key. The attacker uses the CAC functionality and the fact that the CAC has the embedded private key, to its advantage.

ATTACKING THE CAC

We now assume that a valid, untampered CAC belonging to Alice is inserted into the reader attached to a computer that has attack software loaded on to the computer. The attack software, of course, can be planted via the backdoor Trojan that most computers are found to have.

1] PIN phishing

The simplest attack that can be launched against the CAC is PIN phishing. As Alice enters the PIN on the computer, to unlock the CAC, the attack software reads the PIN from the keyboard and transmits it to a hacker site, along with the serial number of the CAC. Now the attacker can build databases of CAC pins, which can be very useful if someone steals a CAC and looks up the PIN in the database, essentially compromising the CAC.

2] False authentication

Alice inserts her CAC and unlocks it using the PIN. Now while Alice is idle, the attack software connects to a secure site posing to be Alice. The secure site sends the host computer a challenge, to ensure this site is using Alice's CAC. The attack software sends the challenge to the CAC, gets a correct response and transmits the response to the secure site. Now the attack software is logged in, as Alice, without Alice's knowledge and can perform any function Alice can.

3] Fraudulent Signatures

Alice uses the CAC to sign her email. Attack software can also send email and get the CAC to sign it without Alice knowing. Thus Alice will be unable to repudiate any fraudulent email sent by the attack software using her identity. This attack also works for digitally singed financial transaction. For example, Alice buys $10 of merchandise form Amazon and Amazon requests a signed charge slip for $10 made out to Amazon, and the CAC is used to sign this charge slip. Attack software can make the CAC sign charge slips to fraudulent merchants, in any amount, without Alice knowing and thus defrauding Alice of money.

4] Remote Control of CAC

Alice's CAC, in essence, after it has been unlocked, can be used by any software agent, anywhere on the Internet, that uses attack software as the proxy. Since the CAC performs challenge response and signature operations upon requests

from the host, the attack software on the host can send such requests to the CAC on behalf of any malicious user anywhere.

5] SSL Hijacking and Data Theft

Let us assume, no attacks are currently underway. Alice uses the CAC to securely log into her bank account. The CAC provides the authentication to the bank server and Alice is actually logged on, upon her request. Now the attack software can perform two attacks:

1. All of the data displayed on the hosts screen from the banking site can be scraped by the attacker and sent to unauthorized parties
2. Transaction such as withdrawals from the bank account can be performed by the attack software, tunneling requests via the SSL connection and not displaying the results – thus Alice cannot find out what transactions are being done by the attacker.

SECURE I/O AND ATTACK RESILIENCE

The attacks described above are a set of attacks that are the result of the lack of secure I/O between a user and the CAC hardware. For example, the PIN phishing attack works as the PIN is exposed to the host machine. If there was a method by which the CAC would obtain the PIN without the host in the middle, then this attack would not be successful.

Similarly, the other attacks work as the CAC has no way of securely informing the user as to what functions it is performing and on what data. This may seem to be possible to solve by additional host based software, but it is not.

Flawed solution for Authentication

Suppose the host software is designed such that when a secure site requires authentication from the CAC then a pop-up box appears on the screen that displays the name of the site and asks the user for permission before the challenge is sent to the CAC. This would prevent any authentication information to be sent to any site without the human user's permission. However this solution does not work, as it is possible for the attacker to suppress the pop-up and to send an OK click to the software that tried to put the pop-up.

The bottom line is that when the CAC needs human input, as well as when it needs to inform that human about its inner workings, there needs to be an I/O channel that is not tamperable by untrusted software on a host computer. *This is called secure I/O*.

Implementation of Secure I/O

The challenge of a secure CAC design is to incorporate not only PKI but the secure I/O subsystem in a manner that does not compromise the form factor, connectivity and portability of the CAC. We identify several approaches to the problem:

Cellular phones

Assuming that every human possesses a cellular phone, we can use a cell-phone to a secure site communication channel to simulate the secure I/O with some degree of assurance. Firstly, when a CAC needs a PIN, it opens a SSL channel to a known authentication server (the CAC now has to have the capability of terminating SSL connections). The CAC then requests the Authentication server to call the user's cell phone, obtain the PIN from the cell phone via DTMF signaling and then send it to the CAC via SSL connection. In this manner, the authentication server can get the PIN but has to be trusted not to store or reveal it. The CAC could also be integrated with the cell phone.

Separate Hardware Channel

The CAC could have a separate wired channel to a special I/O device that can handle the user inputs, and thus enable to the CAC to directly communication with the user. Alternatively this separate channel could be a Bluetooth cannel to a handheld keyboard and display. Bluetooth of course introduces its own set of security risks.

SSL Connections

The CAC could use a SSL channel to a particular I/O device that is trusted. This is a difficult to deploy solution. What hardware device would we use for the secure end-point? It could be some software on the host machine that is run in some attested mode and can be trusted to correctly process input and output. However, ensuring that some other software cannot grab/spoof the actual screen displays may not be possible.

On Card Display and Keypad

One of the solutions that is attractive to the authors is the integration of a simple text display and a

keypad on the CAC itself, hardwired to the CAC processor. Many credit-card sized calculators have such display/keypads. This of course requires a modification of the physical design of the CAC.

HUMAN INTERFACE AND SECURE I/O

The secure I/O ensures that what the CAC performs is approved by a human. This is the "human in the loop" strategy that is essential to secure operation of the CAC. The user prompts are displayed on the secure output device and user enters responses on the secure input device. The steps are:

1. **PIN Entry**: To unlock the CAC a PIN is needed, and this is entered securely and only available to the CAC.
2. **Authentication**: When the CAC is asked for a response to a challenge, it displays the name of the site it is authenticating itself to, and the user has to enter "yes".
3. **Signatures**: When a document is being signed by the CAC, the essential parts of the document (e.g. amount of charge, name of receipient, etc) are displayed on the output and the user has to enter "yes"
4. **Certificate Updates**: Any stored certificate updates, root certificates and such, displays the fingerprint of the certificate and the user affirms after checking the fingerprint from a public site.

The above descriptions are brief but illustrate the importance of securely communicating crucial information to a human user, in a non-interceptable manner, to ensure the CAC is not attacked. In the current design, the CAC is too vulnerable for secure usage and is prone to many attacks that can seriously undermine its usefulness.

CONCLUSIONS

This paper showed that using a DoD CAC on a untrusted workstation can allow a variety of attacks to be performed by malicious software. These attacks range from simple PIN phishing, to more serious attacks such as signatures on unauthorized transactions, authentication of users without consent, unauthorized secure access to SSL enabled web servers as well as remote usage of the DoD CAC by attackers. It was shown that the root cause of such problems is the *lack of a* secure I/O channel between the user and the card The paper outlined steps that can be taken to ensure that such a secure I/O channel is available making the documented attacks not feasible.

REFRENCES

[CCL] K. Watkins, "Army CAC Cryptographic Logon Implementation Plan, CCL – Providing Certificate-Based Access Control to Unclassified Army Networks", Submitted By Information Assurance CAC/PKI Division, March 2006.

[HSPD-12] G. W. Bush, Homeland Security Presidential Directive/Hspd-12 http://www.whitehouse.gov/news/releases/2004/08/20040827-8.html

[KY+01] T-W Kwon, C-S You, W-S Heo, Y-K Kang and J-R Choi, "Two implementations methods of a 1024-bit RSA cryptoprocessor based on modified montgomery algorithm", IEEE International Symposium on Circuits and Systems (ISCAS), pp 650-653, May 2001.

[LiMi03] N. Li and J. C. Mitchell, ``RT: A Role-Based Trust-Management Framework," in The Third DARPA Information Survivability Conference and Exposition(DISCEX III).Washington DC, April 2003.

[Lop01] Lynn M. LoPucki,, "Human Identification Theory and the Identity Theft Problem" . Texas Law Review, Vol. 80, pp. 89-134, 2001 http://ssrn.com/abstract=263213

[MBP03] Marco Casassa Mont, Pete Bramhall and Joe Pato, "On Adaptive Identity Management: The Next generation of Identity Management Technologies", HP Labs 2003 Technical Reports, 2003 http://www.hpl.hp.com/techreports/2003/HPL-2003-149.pdf

[NIHPKI97] NIH, "NIH PKI Policy White Paper – draft", August 1997, http://www.alw.nih.gov/pki/docs/nihpolicy-whitepaper/NIH-white-paper-draft.txt

[NISTPKI05] National Institute of Standards and Technology (NIST), “NIST PKI Program”, http://csrc.nist.gov/pki/

[NISTPKI97] National Institute of Standards and Technology, “Public Key Infrastructure Technology”, July 1997, http://csrc.nist.gov/publications/nistbul/itl97-07.txt

[RFC2704] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, “The Keynote Trust Management System Verison 2”.http://www.cis.upenn.edu/~keynote/Papers/rfc2704.txt

[Rig03] Alisa Riggs, “Ensuring Common Access”, Soldiers Magazine, October 2003, http://www.findarticles.com/p/articles/mi_m0OXU/is_10_58/ai_108838068

[Th84] Ken Thompson, Reflections on Trusting Trust, Communication of the ACM, Vol. 27, No. 8, August 1984, pp. 761-763.

[Wie00] Micheal J. Wiener, “Secure Roaming with Software Tokens”, September 2000, http://csrc.nist.gov/pki/twg/Archive/y2000/presentations/twg-00-32.pdf