2008

Software Implementation using Hardware-Based Verification for Secure Content Delivery

Andy Luse
Iowa State University

Anthony M. Townsend
Iowa State University, amt@iastate.edu

Kevin P. Scheibe
Iowa State University, kscheibe@iastate.edu

Software Implementation using Hardware-Based Verification for Secure Content Delivery

A. Luse[1], A.M. Townsend[2], and K. P. Scheibe[2]

[1]*Department of Electrical and Computer Engineering*
Iowa State University, Ames, IA 50011
Email: andyluse@iastate.edu

[2]*Department of Logistics, Operations, and Supply Chain Management*
Iowa State University, Ames, IA 50011
Email: amt@iastate.edu
Email: kscheibe@iastate.edu

Abstract
This paper presents a novel method for secure message transmission – the Software Implementation using Hardware-Based Verification for Secure Content Delivery (SIHBVSCD) method. This method incorporates a two-tier security protocol which allows messages to be verified at both the user level (coming from a particular user) and hardware level (originating from a particular machine) providing protection from espionage and/or clandestine manipulation of information. SIHBVSCD securely sets up a one-time symmetric key used for transmission, offering advantages over both the high theft/loss likelihood of smartcards and the inability of hardware-based verification for machines that do not contain hardware capable of remote attestation.

Keywords: Hardware verification, secure content delivery, authentication, encryption, public key cryptography

Introduction
In the context of information warfare, information is defined as the fundamental weapon which can be manipulated to the advantage of those wishing to influence events (Hutchinson & Warren, 2001). Thus, it is often critical to protect information contained in messages from both theft and/or manipulation (Denning, 1999). Encryption and authentication are both essential components of secure message transmission between parties. Encryption allows information to be transformed to make it unreadable to anyone not possessing the needed knowledge, typically a key value. While the message may still be intercepted, this encryption protects it from being read and understood by the outsider not privy to the key used. This fulfils one of three concerns of information warfare, namely, deception of opposing forces (Friman, 2001). Authentication allows two communicating entities to believe that they are communicating with each other and not with intruders (Burrows *et al*., 1990).

Traditionally, one or a combination of two encryption methods is used for secure communication. The simplest encryption method, symmetric key cryptography, allows two parties sharing a common key to communicate. The same key is used to both encrypt the message at the sender and decrypt the message at the recipient. While extremely efficient computationally, the drawback of this type of cryptography is that the key must be shared between both the sender and receiver. The parties must therefore share the key in advance over some other secure channel such as registered mail, private courier (Diffie & Hellman, 1976), or public-key cryptography.

Public-key cryptography solves the key sharing problem inherent with symmetric key cryptography. Public-key cryptography uses a two key approach for secure communication. Each user has a public key that is made available for encrypting a message and a private key that is used to decrypt the message. Once the message has been encrypted using the public key, only the matching private key can decrypt it. Two parties are allowed to setup a secure conversation using public keys where each party is ignorant of the private key used by the other party. Moreover, an eavesdropper cannot decipher the keys used based on the separate messages sent between the two parties (Diffie & Hellman, 1976). Since the process of public-key encryption is computationally intensive, the public-key exchange is typically used to exchange a secure symmetric key for the duration of the conversation, which allows for a faster exchange of information.

One inherent problem with public-key cryptography is the lack of assurance that the individual with a specified private key is truly whom they claim to be (Ellison & Schneier, 2000). One way a user's identity may be violated is if that user's private key, which makes up the public/private key pair, is compromised. Another user is then able to impersonate the true user using the compromised private key. This allows the attacker to masquerade as the true user from practically anywhere he or she can communicate with the deceived party.

To overcome the problem of key compromise, a trusted third party can provide verification and dissemination of public-key information for communication between parties (Needham & Schroeder, 1978). Requiring each member of a communication to register with a trusted third party, or Certificate Authority (CA), provides greater confidence in each member's authenticity. The CA operates as an authentication broker between the communicating parties. The trust established between each party and the CA is transferred from the CA to the other party.

Another problem with public-key cryptography, along with many other secure electronic transmission systems, is the lack of location-based authentication, which can appear under normal usage. Just as an attacker with a compromised private key can deceive opposition (Friman, 2001) and use this key from almost any machine which can connect with the communicating party, a valid user can do the same thing. While portability can be an important convenience for some users, it can become an issue with high security applications and content. High level security environments and applications require that not only must the user be identified digitally (using public-key cryptosystems and digital signatures) but also by the user's physical location. Finer-grained physical verification is needed to obviate the use of insecure hosts or locations for message delivery (e.g., the message can only go to "certified" locations, and not laptops, home machines, etc.). This is important in highly secure environments such as government and military installations, or in health care systems, where secure information must not be delivered to machines without the specified clearance. Public-key cryptography has no method for verifying a user based on the machine or hardware he or she is transmitting from. This challenge also cannot be solved by contemporary CA or third-party systems, which can only verify the identity of the users involved.

Smartcards provide one method for hardware-based user and message verification (Rankl & Effing, 2003). By incorporating a key that is in some way derived from a smartcard, the user is not only verified by a known secret (password) but also by a physical item (the smartcard itself). This helps to add an extra information warfare mechanism for secure transmission of data. A substantial disadvantage of smart cards, which may also be considered their main

advantage, is their size and portability. This small size makes these cards convenient to carry, but also more susceptible to loss and theft (Denning, 1999). During the first half of 2006 in the UK, banking fraud losses due to smart card loss/theft totalled £36.1m (APACS, 2006) or approximately $66.8 million. By themselves, smartcards do not provide adequate certainty that the user of a specific card is who that person claims to be.

Recently, the Trusted Computing Group (TCG) (2008) started a movement to develop open standards specifications for trusted computing components across multiple platforms. One area within their list of five key technology concepts is remote attestation. Remote attestation allows authorized parties to detect changes to a user's computer via the hardware generating a certificate attesting to its identity. While very useful for hardware-based verification, many legacy systems do not have such modules installed. Also, remote attestation depends on a secure operating system which helps to protect the system, so even if the hardware is installed, the necessary software may not be. These systems are therefore not viable in many corporate settings which require this level of security.

To address these limitations of current highly-secure systems, the Software Implementation using Hardware-Based Verification for Secure Content Delivery (SIHBVSCD) method is proposed that provides a more robust mechanism for both user and hardware-based security and verification for protection of information. The SIHBVSCD method incorporates a software implementation of hardware-based verification of the message based upon machine hardware characteristics while also incorporating traditional public-key methods. The SIHBVSCD method may be used to protect the critical infrastructure in high-security applications and environments where both user verification and hardware verification for the origination of the message is required and specialized hardware for remote attestation is not available (Pye & Warren, 2006).

This paper is organized as follows: In Section *SIHBVSCD Overview*, the SIHBVSCD method goals, parties involved, and a generalized overview of message transfer is presented. In Section *SIHBVSCD Explanation*, a precise definition of the SIHBVSCD method is given along with usage assumptions. In Section *Discussion*, design choices and possible attacks are discussed and in Section *Conclusion* concluding remarks and areas for future work are given.

SIHBVSCD Overview

Goals

The following is a list of goals for the SIHBVSCD method. SIHBVSCD's achievement of these goals is also included.

Goal 1: Secure transmission of data is obtained between the two communicating parties as well as with the CA. Secure here describes a transmission having the properties of confidentiality and integrity (Clark & Wilson, 1987).

Achievement: The use of the trusted CA as well as the incorporation of both symmetric and public-key cryptography help to insure the information in the message remains confidential between the sender and the recipient. Digital signatures assure the integrity of the message by verifying the sender of the message. This mitigates the chance of a man-in-the-middle attack.

Goal 2: Mutual authentication between parties is acquired.

Achievement: By verifying the identities of both parties using the trusted CA as well as both parties signing messages using their personal private key, both parties are authenticated with each other as well as with the CA.

Goal 3: User-based authentication is employed to verify the owner of a particular public key.

Achievement: The trusted CA authenticates the users based on a matching public/private key pair.

Goal 4: Hardware-based authentication allows the communication to be verified as originating from a particular machine.

Achievement: Using an agreed upon hardware identifier to sign the symmetric key used for the conversation allows the confirmation of the message as originating from a particular machine.

Goal 5: Likelihood of loss/theft is much less than with the use of smartcard technology.

Achievement: The use of hardware identifiers tied to a specific machine allow for improvements over smartcards which are easily lost or stolen.

Goal 6: Specialized hardware and secure operating system implementations are NOT required.

Achievement: Use of typical computer hardware and common security mechanisms built into operating systems allows for all systems to be utilized.

Parties involved

Three major parties are involved in message transfer incorporating the SIHBVSCD method: sender, receiver, and CA. The sender is the user or entity who wishes to communicate with another user or entity and initiates the message transfer. Alice will be used as the name of the initiating party. The receiver is the user or entity with whom the sender wishes to communicate. Bob will be used to delineate the receiver. The CA is the trusted third party who provides verification that the users with the provided public keys are registered as Alice and Bob respectively.

Generalized message transfer

In this section an overview of the communication used for SIHBVSCD setup is presented. In this scenario, Alice would like to communicate with Bob securely. An overview of the protocol exchange between Alice, Bob, and the CA is shown in Figure 1.

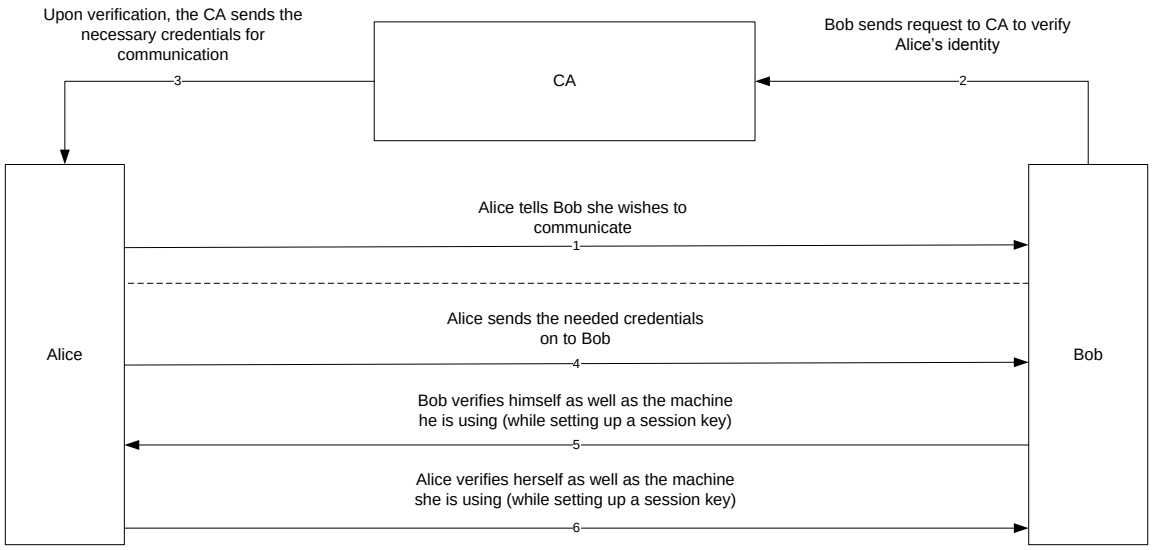

Figure 1: Overview of the conversation during the setup of message transfer using the SIHBVSCD method.

The setup begins with Alice notifying Bob that she would like to communicate with him (1). At this time, Bob can choose to pursue the setup of a communication with Alice or not. If he chooses to continue, Bob sends a message to the CA requesting verification of Alice's credentials (2).

The CA sends credentials to Alice which can be used to verify each user based upon the user's public key (3). The certificates holding the credentials for each user (including both the user's public key and the user's hardware identifier-derived public key) can then be used to establish a secure and trusted communication between Alice and Bob.

Once Alice has received the transmission from the CA, the remaining portion of the exchange takes place between Alice and Bob. Alice received a certificate with Bob's credentials from the CA to verify Bob's identity. Alice then sends the certificate from the CA containing her credentials on to Bob to verify her identity (4). Next, Bob verifies himself using his user-based credentials as well as the machine being used for the transmission using the machine-based credentials (5). At the same time, Bob begins the setup of a key which will be used as a one-time encrypting key for the message from Alice to Bob.

Alice can then verify Bob based on the user credentials he sent as well as verify the machine he is using based on the machine-based credentials sent. Alice then uses her own user-based and machine-based credentials to send a reply to Bob which contains her portion of the key setup which will be used for the message transfer (6). At this time, supposing Alice's credentials are verified by Bob, the session key can be used to pass the message between the two parties.

The SIHBVSCD method uses both public and symmetric key cryptography for secure transmission. The setup and message transfer between Alice, Bob, and the CA use traditional methods of secure message verification and transfer. Public-key cryptography is used to both verify the users to each other as well as for communication with the CA. Symmetric key cryptography is used in the final message delivery phase (not shown in Figure 1). A major

point is that the setup of a session's symmetric key is verified by two different digital signatures. Not only is the user verified, but also the machine which the user is employing to send or receive the message is verified based on a hardware identifier. This provides an extra layer of security and authentication for message delivery.

Note that the SIHBVSCD method used by Alice and Bob is also utilized between the CA and both Bob and Alice. This conversation setup is not shown in the included figures for brevity, but would occur in exactly the same manner as between Alice and Bob, only each user who is allowed to communicate with the CA would have a copy of the CA's public keys necessary for this communication setup.

Finally, the SIHBVSCD method assumes a real-time high-security environment. In environments where real-time security is not the highest priority or where constant connection to the CA is not possible, the system can allow users to download the public credentials needed to verify the user and the respective machine. Certificate Revocation Lists (CRLs) can then be disseminated by the CA at specific intervals in accordance with RFC 3280 (Housley *et al*., 2002).

SIHBVSCD Explanation

This section describes the details of the hardware-based verification method. Various notations are used here to describe the exact makeup of the SIHBVSCD method. To help with understanding and recollection, a list of notations used as well as their definitions has been included in Table 1.

Symbol	Symbol makeup (if applicable)	Definition
N_x		Nonce of X (see reference below)
P_x		Public key of X
S_x		Private (secret) key of X
$h(\ )$		Hash of
PID_x		Processor ID of X
PID_x^{PRIV}		Private (secret) key of X – derived from X's machine's processor ID
PID_x^{PUB}		Public PID (Processor ID) key of X – derived from X's machines's processor ID
CX	$(P_x, t), \{h(P_x, t)\}^{S_{CA}}$	Inner certificate for X (user-based ticket)
$CX++$	$(CX, PID_x^{PUB}), \{h(CX, PID_x^{PUB})\}^{S_{CA}}$	Entire certificate for X (user-based ticket plus hardware-based ticket)
K_{AB}		Session (symmetric) key between A and B
t		Timestamp
p		Large prime number
b		Base number
i_X		Secret integer of X
d_x		Derived number by X
dh_x	$N_A, N_B, A, B, p, b, d_x$	Diffie-Hellman information from X

Table 1: SIHBVSCD notations used

A more detailed diagram of the secure transmission setup process shown in Figure 1 is presented in Figure 2.

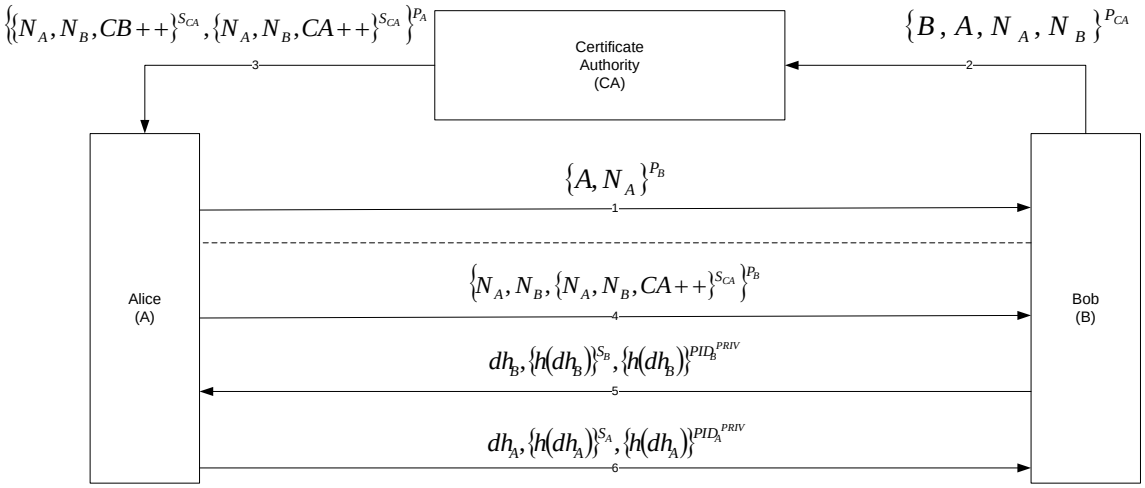

Figure 2: Detailed diagram of secure transmission setup of SIHBVSCD

SIHBVSCD definition

Alice (*A*) would like to send a message to Bob (*B*) securely. She sends Bob a transmission which contains both her identifier (*A*), telling Bob she claims to be Alice, and a nonce (N_A) to protect against replay attacks (Kehne *et al*., 1992, Neuman and Stubblebine, 1993). All this is encrypted using the public key of Bob (P_B), which can be located in a public directory, to ensure only the holder of Bob's secret key can decrypt the transmission.

$$\{A, N_A\}^{P_B} \tag{1}$$

Once Bob has received Alice's transmission, he forwards this information along with his own to the CA's authentication server. He adds his identifier (*B*) along with his own nonce (N_B). He encrypts this combined information using the public key of the CA (P_{CA}) to ensure privacy.

$$\{B, A, N_A, N_B\}^{P_{CA}} \tag{2}$$

The CA, upon receiving the transmission from Bob, constructs a transmission to allow secure communication setup between the two parties. This transmission consists of two certificates each with another embedded certificate. The embedded certificate (*CX*) consists of the public key for X, where X is one of the two parties (Alice and Bob). Also included in this embedded certificate is a timestamp (*t*) to aid in the protection against replay attacks (Denning & Sacco, 1981, Neuman & Ts'o, 1994). These two values are then hashed and this hash signed using the CA's secret key (S_{CA}). This hashed, signed value is appended to the certificate to verify that the certificate originated from the CA and was not modified along the way. This inner ticket serves as the user-based authentication mechanism for the SIHBVSCD method.

$$CX = (P_X, t), \{h(P_X, t)\}^{S_{CA}} \tag{3}$$

The first certificate (*CX*) is then embedded in a second ticket (*CX++*). This second ticket also includes the public key for the hardware identifier associated with X. The private/public key combination is derived from the specific hardware used. This implementation uses the processor ID (*PID*) which is a unique serial number which all processors possess. The embedded certificate and the public PID key ($PID_X{}^{PUB}$) are then hashed and this hashed value again signed using the CA's private key (S_{CA}).

$$CX++ = \left(CX, PID_X{}^{PUB}\right), \left\{h\left(CX, PID_X{}^{PUB}\right)\right\}^{S_{CA}} \tag{4}$$

The complete transmission from the CA to Alice includes two different subsections intended for Alice and Bob separately. These subsections include the combined certificate above (4) along with the nonces generated by both parties. Each user will get the other's credentials to allow each user to verify the other based both on their user credentials as well as their hardware credentials. So, Alice will extract the portion with Bob's information $\{N_A, N_B, CB++\}^{S_{CA}}$, and forward the second portion to Bob with her own information $\{N_A, N_B, CA++\}^{S_{CA}}$. Both of these subsections are signed using the CA's private key to verify that they have not been modified in transit. The entire transmission for Alice is encrypted using Alice's public key to ensure that only Alice can extract the information within.

$$\left\{\{N_A, N_B, CB++\}^{S_{CA}}, \{N_A, N_B, CA++\}^{S_{CA}}\right\}^{P_A} \tag{5}$$

After Alice has received the transmission from the CA and extracted the portion for her, she reassembles the portion intended for Bob in another packet. Along with the information from the CA, Alice also includes the original nonces. This information is then encrypted using Bob's public key.

$$\left\{N_A, N_B, \{N_A, N_B, CA++\}^{S_{CA}}\right\}^{P_B} \tag{6}$$

Bob extracts the information from the packet and verifies the nonces from Alice as well as the information sent from the CA. From here, Bob begins the process of setting up a unique symmetric key which will be used for the impending message transfer with Alice.

The Diffie-Hellman method provides the basis for the setup of the session key (Diffie and Hellman, 1976). First, Bob decides upon a base number (*b*). Next, he generates a prime number (*p*). This number should be a large prime number, or only have factors which are large and prime themselves. Bob then chooses a secret integer (i_B) which he uses ($b^{i_B} \bmod p$) to obtain his own derived number (d_B).

Once Bob has this information computed, he sends the same combination of information to Alice in three concatenated portions. This information includes the two nonces, the prime number (*p*), the base number (*b*), and the derived number (d_B).

$$dh_B = N_A, N_B, A, B, p, b, d_B \tag{7}$$

This information is then sent 3 times together. First, the information is sent out unencrypted (dh_B). Next, the same information is hashed and then signed using Bob's user-based private

key ($\{h(dh_B)\}^{S_B}$). Finally, the same information is again hashed and signed using Bob's hardware-based private PID key ($\{h(dh_B)\}^{PID_B{}^{PRIV}}$). The entire transmission looks as follows.

$$dh_B, \{h(dh_B)\}^{S_B}, \{h(dh_B)\}^{PID_B{}^{PRIV}} \quad (8)$$

When Bob's information is received, Alice is able to use the information she received about Bob in the certificate from the CA to verify the pieces of Bob's message. First, she is able to verify Bob based on his user-based credentials by decrypting the portion of the message signed with Bob's private key (S_B) using the CA supplied public key for Bob. Alice can then hash the unencrypted portion of the message (dh_B) and compare this to the hash contained in the portion signed using Bob's public key. Alice also verifies the machine Bob is using by employing the same method. She first decrypts the third portion of Bob's message using the public machine-based key for Bob provided by the CA ($PID_B{}^{PUB}$) and then compares this hashed value with the hash of the initial portion of the message.

Once Bob's information has been validated, Alice uses the base number (*b*) and the prime number (*p*) to compute her own derived number (d_A). She then sends the same information back to Bob using the same three-part repeating message, only, inserting her own information.

$$dh_A, \{h(dh_A)\}^{S_A}, \{h(dh_A)\}^{PID_A{}^{PRIV}} \quad (9)$$

Bob can then use the same methods described above to authenticate both Alice and the machine she is using.

Once both parties have this information, they can use the Diffie-Hellman technique to derive a common symmetric key for use during their message transfer. Alice computes

$$(d_B)^{i_A} \bmod p \quad (10)$$

while Bob computes

$$(d_A)^{i_B} \bmod p \quad (11)$$

both deriving the same symmetric session key to use for message transmission.

Usage assumptions

Six assumptions are made for proper use of the SIHBVSCD method.

Assumption 1: The Certificate Authority is trusted by both parties. This is expected for any secure and reliable certificate-based system.

Assumption 2: The CA is specially configured to support transactions with users based on the SIHBVSCD method. This requires the CA to construct and disseminate specialized certificates which contain information specific to the SIHBVSCD method.

Assumption 3: Both transmission participants each have a public/private key pair which has been registered with the CA. This is again expected in tandem with any certificate-based system. The user must have a method for obtaining a public/private key pair using a predefined algorithm set forth by the CA. The user then must setup a trust relationship with

the CA to allow the CA to disseminate the user's public key half of the public/private key pair to other trusted users.

Assumption 4: Each participant has a public/private key pair derived from the machine used to conduct the transmission. The information is then uploaded to the CA for dissemination. The private key information is not saved on the user's machine or by anyone besides the CA. The private key portion will be regenerated on the fly on the user's machine during message transmission setup to verify the user is on the machine and the key has not been compromised.

Assumption 5: The participants know the method used to derive the public/private key pair from the hardware identifier used. While this does not necessitate that the user actually know the algorithm to construct the key pair, they must have some method (the software program used for message transmission) to derive this key pair. This will be used during transmission when the user regenerates his or her private key.

Assumption 6: The operating system has been setup to restrict access to the hardware identifier used for hardware-based verification. Remote attestation utilizes a secure operating system which interacts with the hardware to verify that the user does not have access to this information outside the programs using it. While a complete implementation of a secure operating system is not needed for SIHBVSCD, some settings must be in place to keep this hardware information secret except for those programs (i.e. the SIHBVSCD program) which should have access to it. For example, this prototype used registry settings to block access to the PID except for the SIHBVSCD program.

Discussion

Choosing the hardware identifier

There are many different identifiers and combinations of identifiers which can be used for the hardware identifier. The proposed system utilizes the PID for the hardware identifier. Other examples include motherboard serial numbers, component numbers, etc. Electronic fingerprints have also been shown to be unique for a machine (Won & Keiswetter, 1998). This involves reading the electromagnetic output of the machine; while extremely hard to pilfer, the monetary costs of implementing hardware to read such signals typically outweigh the benefits except for in highly secure applications. Also, unlike human fingerprints, the electronic fingerprint of a machine may change with time as the components of the machine age.

Another possibility is to use a combination of hardware identifiers. For instance, a combination of the PID and the motherboard serial number could be used. This can provide extra security as it will be tougher for an attacker to modify multiple pieces of hardware on the machine.

User requirements

User machine requirements may add greater complexity to the overall deployment of the SIHBVSCD method. Today's work environment requires individuals to use a vast array of resources. Many users typically use multiple computing devices throughout the day, which could cause problems for the SIHBVSCD method. First, some of these devices may be utilized by multiple workers (multi-user machines, roaming users, etc.) which may require different users to utilize the same hardware identifier when sending communications. Second, users may use multiple devices for sending and receiving communications. The

SIHBVSCD method must therefore be modified to allow the CA to accommodate multiple hardware identifiers for a single user. Third, many people utilize portable devices such as laptops, PDAs and cell phones which may require message transfer. These devices may show some of the same vulnerabilities as smartcard technology.

Certificate authority requirements

The records held and certificates disseminated by the CA require unique construction. In addition to the typical information held within a certificate, the CA must also hold the public key which has been derived from the hardware identifier of the user. Protocol procedures must also be altered as the certificates disseminated to the communicating parties must contain the user's hardware-based public key along with typical certificate information.

Providing the proper credentials to the CA during initial validation poses a challenge. When a user first registers for a certificate, the CA must be provided with certain information about the user which is typically used to verify the user's authenticity. The SIHBVSCD method requires that the proper keys be derived from the specified hardware identifier which will be used for hardware-based verification without this information being seen by the intended user of the machine or saved in any way except for by the CA.

Attacks on SIHBVSCD

A great number of vulnerabilities exist within both symmetric and public-key cryptography, many of which can be applied to these portions of the SIHBVSCD method. The discussion in this section will be limited to those vulnerabilities which deal specifically with the hardware interaction of the SIHBVSCD method.

Hardware modification

Depending on the hardware identification method used, various types of hardware modification will compromise the SIHBVSCD method. For example, if the PID of the processor is used and a new processor replaces a defective one, this would not allow the user to send or receive messages. This information will help in the initial evaluation of a suitable hardware identifier.

Acquiring Hardware Information

The SIHBVSCD method is susceptible to local machine privilege escalation just like any other method. If a user is able to gain heightened privileges on a machine, then they can potentially garner information related to hardware identifiers such as the PID, motherboard serial number, etc.

Hardware mobility

Most users utilize mobile devices for their job. One of the stated goals of the SIHBVSCD method is to improve on tamper resistance vulnerabilities inherent in smartcard technology. If these mobile devices are utilized in secure communication, the smaller the device the more likely it is to be lost or stolen. A PDA could be almost as likely as a smartcard to get stolen or lost. While SIHBVSCD can be used to secure these devices (and to decertify them when reported lost) it is up to organizational users to determine whether or not it is appropriate to secure easily lost mobile devices.

Virtual Desktop

Many users make use of virtual desktop methods such as Remote Desktop to allow work on their primary machine without physically being present at this machine. This could allow the

user to retrieve SIHBVSCD encrypted/authenticated information from a machine which is not cleared to received this information. Also, this "backdoor" entrance could allow remote attackers to penetrate the system. The SIHBVSCD client could be configured to detect remote desktop-type controls and refuse to participate in authentication when they are in use. What is more likely however, is that organizations requiring SIHBVSCD-level security will lockdown any remote access capabilities.

Conclusion

This paper presents SIHBVSCD as a novel method for secure message transmission between two parties. The SIHBVSCD method provides increased transmission security by requiring a two-tier authentication process. The user is first verified by a user-based public/private key pair. Second, the user is verified by a machine-based public/private key pair, which is based on the PID of the machine. These two methods are used to setup a one-time symmetric key to transfer the message between the two parties.

Information warfare aims to protect information within the context of its specific use. SIHBVSCD protects information in messages exchanged between parties. The information is protected from both theft and manipulation. First, by encrypting the messages, the messages are protected from theft of information within the message and possible manipulation of the information for subsequent delivery to the intended party. Second, by tying the messages to both the user and machine, SIHBVSCD protects against theft and manipulation by others who do not possess the appropriate credentials and hardware to read the messages.

The intended use of the SIHBVSCD method is for high-security environments with critical infrastructures where both the user must be verified as well as the hardware which they are using to send or receive the message. This can help to ensure that even if a user's public/private key pair is compromised, an attacker would also need to be physically present at the user's machine to carry out an attack. Furthermore, machines which do not meet a specific security level will not be able to receive content which is above that security level.

The SIHBVSCD method improves on previous hardware-based authentication methods; first, by tying the hardware identifier to a more tamper-resistant portion of the machine, the chance for loss and theft is significantly reduced. Second, by utilizing a software mechanism which interacts with existing hardware, specialized hardware is not needed like with remote attestation. Moreover, the need for a secure operating system, as needed by remote attestation, is also not necessary with some slight modifications to the operating system setup.

Future work in the area could deal with greater testing of attacks against the SIHBVSCD method, and identifying and mitigating threats from internal users to the SIHBVSCD method.

References

APACS (2006) UK plastic card and online banking fraud losses, URL: http://www.apacs.org.uk/media_centre/press/06_07_11.html [Accessed: March 27, 2008].

Burrows, M., Abadi, M. and Needham, R. (1990) A logic of authentication. *ACM Transactions on Computer Systems*, **8**(1): 18-36.

Clark, D. D. and Wilson, D. R. (1987) A comparison of commercial and military computer security policies, *IEEE Symposium on Computer Security and Privacy*, Oakland, California, April 27-29, pp.184-194.

Denning, D. E. (1999) *Information Warfare and Security*, Addison Wesley, Reading, MA.

Denning, D. E.and Sacco, G. M. (1981) Timestamps in key distribution protocols, *Communications of the ACM,* **24**(8): 533-536.

Diffie, W. and Hellman, M. (1976) New directions in cryptography, *IEEE Transactions on Information Theory*, **22**(6): 644-654.

Ellison, C. and Schneier, B. (2000) Ten risks of PKI: What you're not being told about public key infrastructure, *Computer Security Journal,* **16**(1): 1-7.

Friman, H. (2001) A Systems View of Information Warfare, *Journal of Information Warfare,* **1**(1): 25-32.

Housley, R., Polk, W., Ford, W. and Solo, D. (2002) Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (RFC 3280), *Network Working Group - Standards Track*, URL: http://www.ietf.org/rfc/rfc3280.txt. [Accessed: March 27, 2008].

Hutchinson, W. and Warren, M. (2001) Principles of Information Warfare, *Journal of Information Warfare*, **1**(1): 1-6.

Kehne, A., Schonwalder, J. and Langendorfer, H. (1992) A nonce-based protocol for multiple authentications, *ACM SIGOPS Operating Systems Review*, **26**(4): 84-89.

Needham, R. M. and Schroeder, M. D. (1978) Using encryption for authentication in large networks of computers, *Communications of the ACM,* **21**(12): 993-999.

Neuman, B. C. and Stubblebine, S. G. (1993) A note on the use of timestamps as nonces, *ACM SIGOPS Operating Systems Review*, **27**(2): 10-14.

Neuman, B. C. and Ts'o, T. (1994) Kerberos: an authentication service for computer networks, *IEEE Communications Magazine*, **32**(9): 33-38.

Pye, G. and Warren, M. (2006) Modelling Critical Infrastructure Systems, *Journal of Information Warfare*, **6**(1): 41-53.

Rankl, W. and Effing, W. (2003) *Smart Card Handbook*, John Wiley & Sons, West Sussex, England.

TCG (2008) Trusted Computing Group, URL: https://www.trustedcomputinggroup.org/home [Accessed: April 14, 2008].

Won, I. J. and Keiswetter, D. (1998) Electromagnetic induction spectroscopy, *IEEE International Geoscience and Remote Sensing Symposium,* Seattle, Washington, July 6-10, pp 517-519.