Accountable Security Architectures for Protecting Telemetry Data

Jeffrey R. Kalibjian
CounterSign Software, Inc.

ABSTRACT

Today there are many security solutions available which can facilitate both protection and sharing of telemetry data. While the technologies behind these solutions are maturing [1] [2] [3], most products lack a consistent and coherent paradigm for enforcing who is able to access the secured data, what is done with it, and insuring it can be recovered if the person who secured it is disabled.

KEY WORDS

Computer Security, Event Auditing, Security Policy, Public Key Infrastructure (PKI), Key Recovery, Access Control.

INTRODUCTION

Accountability and security usually go hand in hand. Unfortunately, in the security marketplace today, there is much more emphasis on protection than controlling and tracking resources that have been secured. After reviewing the commonly used information security paradigms and the important accountability concepts of auditing, security policy enforcement, and key recovery, an example will be given of a security architecture embracing these accountability concepts that might be particularly suited to protecting telemetry data hosted in an open network environment.

SECURITY STATUS QUO

The information security marketplace is heavily reliant on symmetric key and dual asymmetric key technologies for securing information. Symmetric key algorithms [4] use the same key to encrypt and decrypt data; while in dual asymmetric key algorithms [4] when one (of the dual keys) is used to encrypt, the other can be used to decrypt. Because of the computational complexity of dual asymmetric key algorithms, the market place has adopted a hybrid approach for securing information. Symmetric key algorithms are used to bulk encrypt large amounts of data; while dual asymmetric key technology is used to secure those symmetric keys used for bulk encryption.

In the dual asymmetric key paradigm (otherwise known as Public Key Infrastructure, PKI), one key is called a private key, while the other is called a public key. As their names imply, the private key is kept private ----- that is, it is only utilized by the owner; while the public key is made public ----- that is, it is made freely available to anyone who might wish to securely communicate with the public key owner. Someone wishing to secure information for a particular individual simply generates a symmetric key, uses that key to encrypt the message; then encrypts the symmetric key with the recipient’s public key. The recipient uses his private key to decrypt the symmetric key; then uses the symmetric key to decrypt the message. Fortunately, users of secure applications rarely have to worry about such low level details

IDENTITY CRISIS?

There is a problem with the public/private key paradigm. How does one know that a public key really belongs to an intended recipient? To solve this problem, the user’s public key is typically embedded in something called a X.509 certificate [7]. The certificate contains a user’s public key and other important information about the user, like e-mail address, company name, and serial number. The certificate is digitally signed (meaning it is cryptographically hashed and the hash result is encrypted with the signer’s private key). The signed hash is put in the certificate as well, and this is known as a “digital signature” over the certificate. Thus, anytime a third party wishes to send secure information to a user, he or she may validate the signature on the certificate, then use the public key in the certificate as needed.

Certificate signers are known as Certificate Authorities, and they freely distribute their public keys so users may easily validate certificates they sign. Observe that the method of obtaining the Certificate Authority’s public key is critical, since it must be implicitly trusted. Certificates can be revoked by Certificate Authorities by distributing revocation lists, that indicate the serial numbers of certificates they have signed which are no longer valid.

REALITY CHECK

While the cryptographic foundations for information security are rock solid, as are the actual protocols used to exchange secure information, the implementation of accountability concepts in software applications supporting security has been lacking. There are generally four primary accountability concepts that should be implemented when information is to be secured. The first is audit. Any action taken with secured information should be documented. The second is security policy definition and enforcement. Not all individuals have a right to view, use or modify certain types of secured information. The third is key recovery. No one entity should hold exclusive rights to secured data in a corporate environment. The fourth is privilege revocation and modification. Changes in security policy or digital credentials must take effect immediately.

AUDIT

Information audit is fundamental to security practice. Once information is secured, there is an implicit value being placed on the data being protected. Consequently any subsequent action taken with that secured data (who decrypts it, who has modified it) must be known. Analogies would be a bank account. Once an account is opened, detailed records are kept on modifications to the account (deposits, withdrawals), because the account has value. Another example is phone service, once a phone service account is opened detailed records are kept regarding use (who is called, who called in, time, location, etc.).

While there are currently no “official” standards for audit practice in the information security field, there are rudimentary notions of good procedure which might include noting some of the following when secure data is accessed:

- User Viewing Data
- IP Address of User
- Date and Time
Application Using Data
Changes to Data

It is important to note that audit must be an integrated part of the security application. That is, auditing as a service, has little meaning, because the auditing assertions could be false. When the auditing mechanism is integrated into the application, there can be no dispute that recorded events are actually occurring.

SECURITY POLICY

Too often security privileges end up being an “all” or “none” situation. In some cases this might be appropriate (e.g. access to the front door of an office building), and in other cases it might not (e.g. access to corporate financial information). Generally what one “needs to know” relates to their position in a company or the project they are working on. Thus, it is important that access privilege to secured information be implemented in a hierarchical or group mechanism.

Today access control, when implemented, is usually accomplished by creating an access control list for data. Individuals who have a right to access the data are placed on the list, and the list is always checked when the data is to be accessed. More sophisticated applications allow definition of hierarchy that might reflect a company structure (e.g. engineering, executive, finance), then a user is granted access to data only when his role in the company confirmed.

In all cases the actual identification of the individual is done by evaluation of the user’s X.509 certificate. In fact, the certificate may include entities known as extensions, which might also contain information about access privilege. However, it should be noted that extensions are somewhat static. That is, the certificate must be re-created when privileges change; whereas, access control lists are more dynamic, since they can be quickly changed.

KEY RECOVERY

Since information is vital to the functioning of a company or organization, there must be guarantees that any company information can be accessed by anyone having the appropriate privilege. Several scenarios are common:

What happens to data secured by a person who becomes disabled?
What happens to data secured by a person who leaves the company?
What if secured data needs to be examined by law enforcement personnel?
What if data secured by “old” keys needs to be accessed?
What if a person forgets the password protecting their private key?

Typically these problems are addressed by implementing a key recovery strategy. User’s private keys are archived in a fashion where they can be recovered, by some extreme level of effort, in an emergency situation. The key concept is “extreme level of effort.” The archiving mechanism must be secure, such that no one person could initiate the recovery in a rogue fashion and impersonate an individual. Instead, the recovery process should require the consent of a few people to insure the intent of the recovery is legitimate.

While key recovery is often thought of a measure “of last resort,” in reality it will probably be used very often. This is because, typically, Certificate Authorities, recommend that a user get new public/private key pairs every year. Thus, it is possible over a thirty year career in a company, a user may accumulate
up to thirty key pairs. If a user wishes to access old data that was encrypted several years in the past, he or she may have to initiate a key recovery process, if they have not carefully maintained their old private keys.

**PRIVILIGE REVOCATION**

A very key part of accountability in information security is the ability to quickly modify and enforce changes in access policy. When an employee’s role in a company changes, those changes need to be reflected in their access policy immediately. This usually takes the form of a complete revocation of privilege or some change in the type of data the employee is able to access.

Today complete revocation of privilege usually takes the form of certificate revocation. Certificate Authorities periodically issue revocation lists which detail serial numbers of certificates that are no longer valid. Access privileges can be modified by changing group membership or access control lists. When access information is actually placed as an extension in a X.509 certificate, the certificate must be modified and re-issued.

**AN ACCOUNTABILITY ARCHITECTURE FOR MANAGING TELEMETRY DATA**

An accountable and secure architecture for managing telemetry data would contain the following elements:

- PKI technology for authentication and data protection
- Central server control of data access and modification
- Central server control of certificate issuance/key escrow
- Central server control of audit
- Central server control of access privilege

Each company program or project producing telemetry would create group access lists. These lists would indicate user access rights to the secured telemetry data. For instance, mechanical engineers maybe given access to rate (gyro) or temperature data; while optical engineers maybe only given access to image data. When the telemetry data was received, it would be secured by the central server. The data access rights and the (encrypted key) to decrypt the data would also associated with the data.

Engineers wishing to access secured data would first have to present their X.509 certificates to the central server. After validating the user’s certificate, the central server would check the user’s current access privileges. If the engineer was on the access list for the data he or she requested, the data would be returned to the user, along with an ephemeral key to initially decrypt the data. Any time the engineer wished to access additional data, or even data that he had downloaded to his machine, he would have to go back to the central server for a re-authentication and a re-evaluation of access privileges. This approach insures that the central server will have an instantaneous capability to modify access privileges or revoke access completely, it also insures that a complete audit trail will be in place for who accessed and modified data.

Observe that the central server concept is in direct contrast to traditional PKI deployments, which are typically deployed in a de-centralized fashion. That is, privilege revocation can only be accomplished by a Certificate Authority issuing an incremental certificate revocation list. Since the time between issuance of revocation lists is out of an organizations control, the ability to instantaneously revoke privilege is made very difficult. This concept of associating access information with data, along with the key to decrypt data; but, controlling access of this information through a centralized server was pioneered by
the TriStrata Corporation [8]. Further notice that key recovery is made very simple, by the centralized server controlling the keys used to secure the telemetry information.

CONCLUSION

While great progress has been made in information security technology, turnkey security applications typically lack accountability features that are necessary for protection of high sensitivity items like telemetry data. An architecture was described incorporating the accountability features of audit, key recovery, access control and instantaneous privilege revocation that would be suitable for protecting a sensitive telemetry data product in an opened network environment.

REFERENCES


