A Global PMI for Electronic Content Distribution

Carlisle Adams and Robert Zuccherato

Entrust Technologies, 750 Heron Road
Ottawa, Ontario, Canada K1V 1A7
{cadams,robert.zuccherato}@entrust.com

Abstract. The paper describes a novel application of a Privilege Management Infrastructure (PMI) to enforce copyright protection in electronic content distribution. The PMI is "global" in nature and thus permits customers to gain access to content on any appropriate device. The use of a PMI also allows delegation of access to content. A unique key encrypting scheme provides increased security over other methods of protecting electronic content.

1 Introduction

The distribution of electronic content via the Internet is becoming more and more common. Electronic content includes such objects as electronic documents (e.g., PDF, Word documents), music (e.g., MP3), video (e.g., MPEG), and games. There are presently problems with this method of distribution. The most significant of these is that once the content has been downloaded it can easily be copied and re-distributed. Thus, enforcing copyright protection is difficult, particularly if the customer wishes to download the content for use when off-line or for use on multiple machines/devices. In addition, it is difficult to provide customers fine-grained access to content (e.g., it is difficult to allow individual customers to buy one article from a magazine), to reliably identify customers, and to allow customers to further delegate access to content in a controlled manner, when required. This proposal attempts to address these problems by taking advantage of some current solutions for authentication using a Public Key Infrastructure (PKI) (see [3] for an overview of PKI) as well as some new ideas using attribute certificates [9].

In order to provide a concrete example, this paper will describe a sample application that distributes PDF versions of magazine articles using a PDF viewer. Generalizations to other forms of electronic content should be relatively straightforward.

While this solution, like most solutions for content distribution, does not prevent unauthorized distribution by determined and malicious legitimate customers (especially in a software environment), it does prevent the typical user from doing so, while still allowing online or off-line use, access from different devices, further delegation and fine-grained access control.

D.R. Stinson and S. Tavares (Eds.): SAC 2000, LNCS 2012, pp. 158–168, 2001.

2 What Is a PMI?

Within a Public Key Infrastructure a public key is bound to a user's identity through the use of a certificate. For example, in an X.509 [6,7] based PKI, a Certification Authority (CA) will verify the identity of the user and that he/she actually has the private key associated with the claimed public key. If the test of identity and Proof-of-Possession pass, then the CA will place the public key along with the user's identity in an X.509 public key certificate and sign this certificate using it's private key. When any other entity wishes to verify a signature of or encrypt data for the user, he/she first verifies the signature on the certificate using the CA's public key. If the signature verifies then the public key contained in the certificate can be used for the desired purpose. Thus, end entities need only trust the CA's public key, typically achieved through some out-of-band method [2], in order to validate the certificates of other entities in the PKI. If an end entity trusts a CA and has obtained the CA's public key in a way that guarantees its authenticity, the CA's public key is known as the CA's root key.

A Privilege Management Infrastructure (PMI) [7] is similar to a PKI, except that instead of using public-key certificates to bind a user's identity to a public key, an attribute certificate is used to bind an identity to certain rights or privileges. An Attribute Authority (AA) that wishes to grant a user certain privileges will codify the privileges (usually represented by an attribute-value pair) and place them in an attribute certificate with the user's identity. The AA then signs the attribute certificate using its private key. When the user wishes to use those privileges to gain access to a protected resource he/she presents the attribute certificate to the entity controlling access (the "gatekeeper"). The gatekeeper will then authenticate the user and verify the signature on the attribute certificate using the AA's root public key. The gatekeeper must have already established trust in the AA's public key (again, typically achieved through some out-of-band mechanism). If the signature verifies and the attribute certificate contains the required attribute, the user is allowed access to the protected resource. In our example, the PDF viewer will act as the gatekeeper.

In a PKI, a CA can certify the public key of another subordinate CA, thus allowing end-entities that trust one CA to validate certificates of the subscribers in another CA domain. Similarly, an AA can grant privileges to another AA, thus allowing gatekeepers who trust one AA to accept attribute certificates issued by another AA. The gatekeeper must now verify that the intermediate AA has, in fact, been delegated authority to grant this privilege by the trusted AA. This process is referred to as delegation in [9]. We will also adopt that terminology.

3 The Idea

The idea is that a root Attribute Authority for a large Privilege Management Infrastructure (PMI) would control access to individual pieces of electronic content. Each PDF viewer, for example, would have the root key for this PMI embedded within it and access to the document would not be granted unless a

valid attribute certificate for the customer within that PMI existed. In addition, each viewer would require an embedded CA root key for a PKI to authenticate users and also a master symmetric key for obtaining access to content.

Thus, the content creator would encrypt each piece of electronic content. For example, the magazine could make articles from each issue available for purchase. Customers would authenticate themselves to the magazine website and pay for and download the articles desired. When the user wished to read the downloaded articles, the viewer would first require authentication of the customer and, if a valid attribute certificate existed, decrypt and display the article.

The advantage of using this method of distributing electronic content instead of just encrypting the content for each customer using, for example, CMS [5] or PKCS #7 [10] is that if the content is encrypted directly for the customer, he/she can simply decrypt and distribute the pirated content. If the proposed method were used however, the PDF viewer, for example, would only decrypt the content upon authentication of the customer and could make the plaintext difficult to obtain (e.g. would not write the content to disk).

There is, however, at least one potential problem with this scheme (see Section 5.1 below). This proposal will only make it more difficult for most legitimate customers to illegally gain access to or copy and distribute electronic content. Determined individuals (i.e., those with the ability to analyze executable code or those with access to the internal workings and components of their computer, device, etc.) will still be able to do bad things. Unfortunately, it appears that this will always be a property of e-content distribution schemes since at some point the plaintext content must appear somewhere on the customer's machine. If someone has the ability to analyze how the plaintext was obtained or to gain direct access to the plaintext as it is being displayed, they will always able to compromise the system.

4 The Architecture

This section describes the proposed Privilege Management Infrastructure as well as the accompanying Public Key Infrastructure for authentication.

4.1 The PMI

In this architecture, the root key of the PMI is embedded in the PDF viewer (or the appropriate viewer for the type of content). It is envisioned that this root could be the root of a global PMI similar to the PKI roots that exist in web browsers. The root Attribute Authority would then issue an attribute certificate to the PDF viewer manufacturer indicating that it was authorized to produce PDF documents to be displayed by the viewer and that this privilege could be delegated. In a similar way, the root Attribute Authority could issue attribute certificates to any manufacturer of electronic content viewers. The PDF viewer manufacturer would then issue an attribute certificate to the magazine publisher indicating that it was authorized to produce PDF documents (i.e., that it could

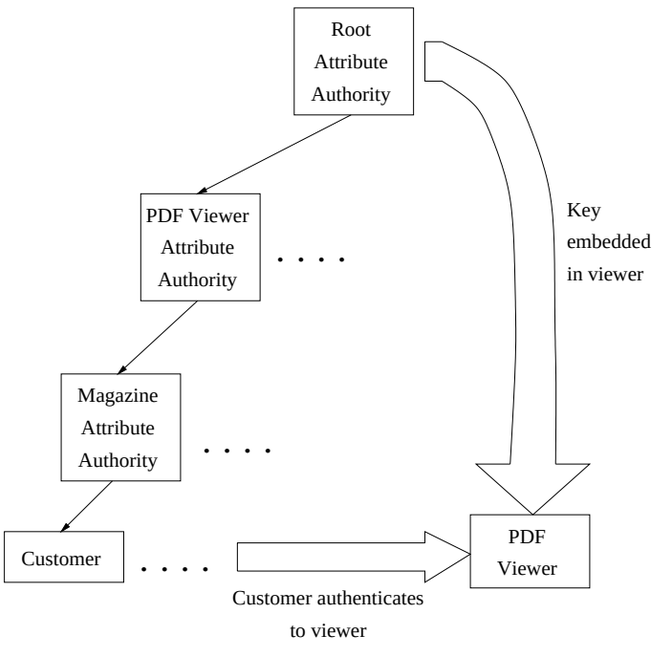

Fig. 1. The PMI architecture

authorize customers to have access to the encrypted content). The utility of having these layers of Attribute Authorities will be discussed further in Section 6.3. The need for a PMI, and in particular attribute certificates, will be described in Section 5.4.

Upon the purchase of an article, the customer would authenticate to the magazine publisher and provide it with a customer symmetric key. The magazine publisher would then issue an attribute certificate to the customer indicating that the customer was authorized to view the particular article and also including the content symmetric key used to encrypt the article, encrypted with the master symmetric key and then encrypted with the customer symmetric key. (The purpose of doubly encrypting the content symmetric key will be discussed further in Section 5.) The customer's attribute certificate could also place restrictions on when or how the content is to be viewed and may or may not allow further delegation. For example, a university library may subscribe to the magazine and then provide access to all of its students. The encrypted article including the customer's, the magazine publisher's and the PDF viewer manufacturer's attribute certificates would be delivered to the customer.

When the customer wishes to read the article, the viewer would authenticate and receive the customer symmetric key from the customer, and also verify the validity of the customer's, the magazine publisher's and the PDF viewer manufacturer's attribute certificates using its embedded PMI root key. If the

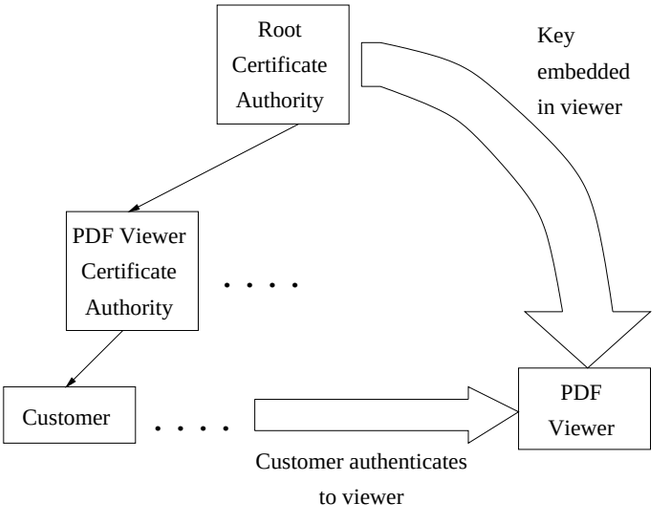

Fig. 2. The PKI architecture

complete attribute certificate chain validated, the customer symmetric key and the master symmetric key would be used to decrypt the content symmetric key that would be used to decrypt the article, which would be displayed to the user.

In this way, only legitimate customers that had paid for the article could view it.

4.2 The PKI

There are a number of possible PKI architectures that are compatible with the proposed PMI architecture. In fact any method of authenticating the customer could be used instead of a PKI. Here we describe one possible architecture.

Since the viewer must be able to authenticate legitimate customers, it must have the root of a PKI embedded within it. Then the root CA could certify the PDF viewer manufacturer's CA, which would in turn certify the customer (as well as, in this example, the magazine publisher).

4.3 How It Could Work

For example, when the customer downloads the PMI-enabled PDF viewer, he/she would also be enrolled in the PDF viewer manufacturer's PKI domain. (Alternatively, the customer could already belong to a PKI that could be chained to the PDF viewer manufacturer's PKI domain.) Then, when the customer wishes to purchase a magazine article he/she first authenticates him/herself to the magazine publisher using standard Internet authentication techniques (SSL/TLS [4] or SPKM [1], for example), provides it with the customer symmetric key over

the established session, pays for the content and obtains the content and appropriate attribute certificates. When he/she wishes to read the article, the PDF viewer would log the customer into their PKI identity (if he/she was not already logged in), authenticate the customer's identity (using, for example, techniques in ISO/IEC 9798-3 [8]) and obtain the customer symmetric key from the customer in order to determine whether or not to allow access to the content.

Thus, the sequence of events becomes:

1. Customer downloads the PDF viewer.
2. Customer enrolls in PDF viewer manufacturer's PKI. (If not already enrolled in a PKI.)
3. Customer authenticates to the magazine publisher.
4. Customer pays for the article and provides the magazine publisher with the customer symmetric key.
5. Magazine publisher encrypts the article with the content symmetric key. (Could be done in advance.)
6. Magazine publisher encrypts the content symmetric key with the master symmetric key to produce an encrypted symmetric key. (Could be done in advance.)
7. Magazine publisher encrypts the encrypted symmetric key with the customer symmetric key to produce a doubly encrypted symmetric key.
8. Magazine publisher Attribute Authority creates an attribute certificate for the customer containing the doubly encrypted symmetric key.
9. Magazine publisher sends the encrypted article and the attribute certificates to the customer.
10. Customer authenticates to the PDF viewer and provides the customer symmetric key.
11. The PDF viewer validates the attribute certificate.
12. The PDF viewer decrypts the content symmetric key using the customer symmetric key and the master symmetric key.
13. The PDF viewer decrypts the content using the content symmetric key.
14. The PDF viewer displays the content.

5 Security Issues

5.1 Encrypting the Content

Each piece of content would be encrypted with a unique content symmetric key. This encryption would only have to be performed once for each piece of content. The content symmetric key would then be encrypted with a master symmetric key and a customer symmetric key and placed within each customer's attribute certificate. The master symmetric key would be embedded in each viewer to allow decryption of the content symmetric key and thus, the content. This key should also be different for each type of viewer (i.e. for each type of e-content). The customer symmetric key, which would also be required to obtain access to the content, should be stored securely for the legitimate customer in such a way

that it is portable to different machines/devices. A simple solution is to store it within the customer's Personal Security Environment (PSE) [2]. A PSE is a software or token based secure storage for the customer's private keys and other sensitive information. Most PSEs can be moved from one device to another.

In this solution knowledge of both the master symmetric key and the customer symmetric key is required to gain access to the content. Thus, both must be securely protected. If the customer symmetric key is stored in the customer's PSE, it should only be available to the legitimate customer. The master symmetric key must be embedded in each viewer, however, and thus must not be readily available by analysis of the viewer executable. This could be accomplished by implementing a function whose sole purpose is to decrypt keys encrypted with this master symmetric key value. In other words, the plaintext key need not appear in memory and need not be passed into a general purpose decryption algorithm. A function could be used that would only perform decryption with the given key. The key then wouldn't need to actually appear in memory since bit operations could be used to optimize and obfuscate decryption with this code. Even so, the master symmetric key could become available to determined adversaries. However, unless they have the cooperation of a legitimate customer in order to acquire a customer symmetric key, they are no further ahead. Thus, determined legitimate customers may be able to get access to the plaintext, but this may not be preventable (see Section 5.2).

Note that if the viewer is implemented in secure hardware, then it is highly unlikely that the master symmetric key could be obtained and thus the solution described in this paper is very secure. For this reason this solution is more applicable to hardware implementations.

In addition, to support encryption the master symmetric key would have to be kept in a secure location so as not to be compromised. A Trusted Third Party (e.g. the root CA or root AA) could keep this key in secure hardware and encrypt content symmetric keys for content creators. The content creators (e.g. the magazine publisher) would authenticate themselves to the Trusted Third Party and present their attribute certificate indicating that they are legitimate creators. They could then provide the content symmetric key to the Trusted Third Party and it would be encrypted using the master symmetric key. Again, this operation would only need to be performed once for each piece of content. This encrypted key would then be encrypted again for each customer using the customer symmetric key.

The content, master and customer symmetric keys could, in fact, all be asymmetric keys. However, it is recommended that symmetric key cryptography be used for these keys, to allow for more efficient operations at the server.

5.2 Making Plaintext Unavailable

Using the other methods described in this document to restrict access to electronic content will not be successful if the decrypted content is somehow made available or stored on the user's disk, allowing copying of the content and unauthorized distribution. Thus, viewers should keep the plaintext in memory. How-

ever, even then, determined individuals could certainly read the content by scanning memory. Therefore, again, determined legitimate customers may be able to gain access to the plaintext content.

In some applications the disclosing of plaintext content may not be undesirable for certain customers. In these cases, the customer's attribute certificate could indicate whether or not the decrypted content should be made easily available to them.

5.3 Further Delegation

One of the advantages of this scheme is that it allows further delegation of privilege to access the electronic content. Let us consider the example of a university library that wishes to grant access to magazine articles to its students. The library has an attribute certificate containing its unique identifier, an indication of the privilege to view the content, and the content symmetric key encrypted with the master symmetric key and also the library's customer symmetric key. In order to delegate access, the attribute certificate must also contain an indication that the library is in fact allowed to delegate access.

When it does wish to delegate access to the magazine articles, the library will create an attribute certificate for each student to which access will be granted. The new attribute certificates will contain an identifier for the student to which access is granted, an indication of the privilege to view the content and the content symmetric key. The content symmetric key will be encrypted with the master symmetric key and the student's customer symmetric key. The library can produce this encrypted key by taking the doubly encrypted key out of its attribute certificate, decrypting it using its own customer symmetric key (leaving the singly encrypted key) and then encrypting it with the student's customer symmetric key.

The library must obtain the student's customer symmetric key in order to place the properly encrypted content symmetric key in the attribute certificate. Thus, it may make sense in these circumstances (and, in fact, any situation where the AA cannot be trusted with the customer symmetric key) for the customer to produce different symmetric keys for each application.

5.4 Why Attribute Certificates?

One may be tempted to not use attribute certificates at all in this type of scheme. Shouldn't the presence of the content symmetric key encrypted with both the master and customer symmetric keys be enough evidence that the customer had been granted access to the e-content?

Unfortunately, this is not the case. Since the outer encrypting of the content symmetric key is performed using the customer symmetric key, a malicious customer could very easily remove this encryption and encrypt it with any other symmetric key, thus easily delegating access. This would be undesirable. Attribute certificates eliminate this security weakness by placing this doubly encrypted key inside a signed object that cannot be created by the customer.

Note that it is not feasible in a large scale environment to reverse the order of encrypting so that the outer encrypting is performed by the master symmetric key. This change would require that the content symmetric key must be encrypted with both the customer and master symmetric key each time a customer was granted access. This would mean that the master symmetric key must be kept on-line which will decrease efficiency and could make the key vulnerable to attackers which attempt to break into the server in which it resides. With the present scheme, the content symmetric key need only be encrypted with the mater symmetric key once, and then encrypted with a customer symmetric key each time a customer is granted access.

6 Other Issues

6.1 Anonymous DNs

A PKI could be used for authentication of customers. However, it is possible that some customers would not want their name or other vital information to appear in a widely available certificate. For such environments, it is recommended that anonymous DNs be used. In this example, the PDF viewer manufacturer's CA may be required to keep a database linking the anonymous DNs with actual identity information. Also, naming rules must be enforced so that each customer receives a unique DN within this PKI. It may also be desirable for customers to have different certificates (and DNs) for each viewer for which he/she is registered.

6.2 Certificate Rollover

In many cases it would be undesirable if a customer bought a song and after 6 months he/she couldn't use it because his/her public key certificate had expired. There are two possible solutions to this problem. One solution is to make customer certificates very long-lived (e.g. 10, 20 years). However, issuing long-lived public key certificates to end entities is discouraged in most environments for security reasons.

A second solution is to make the key short-lived (e.g. 6 months or a year) and require that every few months users must re-connect to the Internet to contact the PDF viewer manufacturer's CA and obtain a new public key certificate. A warning would have to be displayed when certificate expiry is approaching which advises customers of this requirement. This has the disadvantage that people who remain off-line for extended periods of time lose access to all of their electronic content. In order to link the attribute certificate issued to the customer with any public key certificate issued to that customer by the PDF viewer manufacturer's CA, the customer should be identified by their DN in the attribute certificate.

When the public key certificate of an attribute authority expires, however, all attributes issued by that authority can no longer be verified. Thus, attribute authorities must have keys that are very long lived (e.g. 20 years).

6.3 Are Global Root Keys Required?

The description in this paper assumed for simplicity that there would be one global root key for the PMI that would be used for all types of electronic content, one global root key for the PKI that would identify each customer of electronic content, and one master symmetric key for decrypting content. This is not strictly necessary. The PDF viewer manufacturer, for example, could establish its own roots for the PKI and PMI and master symmetric key that would be embedded within each PDF viewer. In some situations, this configuration may be more desirable.

7 A Comparison with Other Schemes

This section will describe other possible solutions for distributing electronic content and compare them with the solution proposed in this paper.

7.1 Encrypting the Content Just for the Customer

Another method of allowing the secure downloading of electronic content so that it is only accessible by the legitimate customer is to simply encrypt the content for the customer. The customer simply generates a (symmetric or asymmetric) key and sends it to the e-content distributor who encrypts the content for the user. This solution is conceptually simple and also allows the user to gain access to the content on different devices. However, it is now very easy for malicious customers to decrypt the content and distribute the plaintext. While it is also possible with the scheme described in this paper for malicious customers to gain access to plaintext by gaining access to the master symmetric key, it is much more difficult than simply performing a decryption using a key known to the customer.

Similarly, it is possible for a malicious customer to sell his/her PSE and password, thus allowing others to obtain access to all content he/she has purchased. Customers will be deterred from doing this for two reasons. First, any one with access to the customer's PSE would also be able to impersonate the customer, thus potentially incurring a large amount of costs for the customer. Secondly, if unauthorized redistribution of electronic content occurs on a large scale, the presence of the customer's PSE among a large number of people allows authorities to trace the source back to the malicious customer.

Instead of encrypting the content directly for the customer, an alternative solution is to encrypt it for the customer's computer. A (symmetric or asymmetric) key could be generated on the customer's computer and stored in such a way that it is only accessible on that computer. For example, it could be encrypted by a key generated from unique data on the host computer. This makes it difficult for malicious customers to gain access to plaintext, but does not allow customers to view/play the content on different computers or devices.

In addition, neither of these solutions allows secure delegation of access.

7.2 Encrypting the Content Just for the Viewer

It may also be tempting to encrypt the content using just a key that is embedded in the viewer. This solution allows anyone with a copy of the viewer to have access to the content, however. This may make sense in situations where sales of the viewer are projected to be more important than sales of the content, but that business model is seldom the one envisioned in current and projected e-content distribution ventures.

This solution also suffers from the problem that if someone is able to find the decryption key in the viewer and distribute it, unlimited access to all content for everyone may be available.

While the solution described in this paper also relies upon a key embedded in the viewer, loss of this key does not immediately provide unlimited access to all content. Only a legitimate customer can obtain access. Thus, this solution provides additional security over simply encrypting content for the viewer, and also allows a more realistic business model.

8 Conclusion

This paper described a method for enforcing copyright protection on a per-customer basis. The described solution allows both online and off-line use, provides customers access to content on any device that has the appropriate viewer, allows further delegation of access, and is secure except against very determined malicious legitimate customers.

References

1. C. Adams, "The Simple Public-Key GSS-API Mechanism (SPKM)", RFC 2025, October 1996.
2. C. Adams and S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC 2510, March 1999.
3. C. Adams and S. Lloyd, *Understanding Public-Key Infrastructure; Concepts, Standards, Deployment Considerations*, Macmillan Technical Publishing, 1999.
4. T. Dierks and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
5. Housley, R., "Cryptographic Message Syntax", RFC 2630, June 1999.
6. Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509 Public Key Infrastructure: Certificate and CRL Profile", RFC 2459, January 1999.
7. *"Information Technology – Open Systems Interconnections – The Directory: Authentication Framework"*, ISO/IEC International Standard 9594-8 — ITU-T Recommendation X.509 (1997).
8. *"Information Technology – Security Techniques – Entity authentication - Part 3: Mechanisms using asymmetric signature techniques"*, ISO/IEC International Standard 9798-3: 1998. (2nd edition)
9. ITU-T Recommendation X.509 (1997) – ISO/IEC 9594-1:1997, Information Technology – Open Systems Interconnections – The Directory: Authentication Framework — Draft Amendment (DAM) 1: Draft Amendment on Certificate Extensions.
10. Kaliski, B., "PKCS #7: Cryptographic Message Syntax, Version 1.5.", RFC 2315, March 1998.