A Novel Approach to Hide Credentials

Zhensong Liao, Chisong Li, and Ruhan He

(Corresponding author: Zhensong Liao)

School of Computer, Huazhong University of Science and Technology,

430074, PR. China, Wuhan. (Email: {zsliao, lizan, ruhanhe}@mail.hust.edu.cn)

(Received July 18, 2005; revised and accepted Aug. 15 & Aug. 21, 2005)

Abstract

Credentials are useful in many applications. For example, automated trust negotiation can be attained through the exchange of credentials. However, credentials endure all kinds of attacks during the transport while credentials bring us a lot of benefits. At the same time, how to store credentials is still a key problem to manage trust management systems. Good credential-storage design can greatly reduce much overhead, such as easy to search credentials, less storage space, high application convenience and so on. In this paper, we propose a new approach to hide credentials, which satisfies all requirements of common credentials and provides a higher security as well without affecting the credentials transferring efficiency. It aims at attaining little storage space and high security.

Keywords: Authentication, automated trust negotiation, authorization, credentials

1 Introduction

During the trust negotiation, the negotiators can establish the trust between each other via the exchange of credentials. In the trust-management systems, credential is the identity of the legitimate user. A credential shows the membership of a person in a security domain or an organization. In Globus Toolkit (GT), a user will be denied to access unless he holds a valid credential issued by CA. Here, we give an instance to illustrate the importance of credentials.

A medical organization called MO provides online service for doctors and patients. Every user, including the clerks and guests, can access the information according to his role in the organization. Assume Tom to be a doctor, and Jerry, a patient. Now, Tom is examining Jerry via real-time video. In order to know more information about Jerry's illness records, Tom contacts to MO, submits his certified credential and gets the relative data smoothly.

However, as the credentials bring us a lot of favors, they are compromised by many aspects. First, since credentials are transferred via insecure channels like Internet, they are easy to be attacked. Second, the storage of credentials is still a key problem under way. Distributed credentials' discovery poses a great overhead. Third, when the credentials are stolen, opponents can use them to access the system or perform some tasks.

In this paper we present a new method to hide credentials, which adopts the technique of digital watermarking in the image process. In total, the contributions of this paper are as follows:

- It is the first time to introduce digital watermark hiding techniques to the trust negotiation domain on protecting credentials. Since watermark is often used to protect copyrights, it can be applied to embed the credentials into a meaningful and representative image.
- We present a new model to hide credentials. In our model, we treat credentials as a series of information or copyright of a legitimate user. Meanwhile, we provide several alternative algorithms on how to insert the credentials into the mediator.
- It can make the authentication more secure. Our work aims at enhancing the security level of current trust management systems. Exchange of credentials can provide a secure trust negotiation. When the watermark is implemented, the identification of the image can be another security method.
- The model here is flexible and can be easily extended. During the design, we carefully consider the practicability and the complexity of realization so that the model is effective and efficient. In the model, we provide flexible interfaces and some algorithms can be replaced. For example, we can use DCT to substitute DFT as the insertion algorithm.

Note that this paper deals only with hiding credentials as to protect sensitive information, mainly for authenti-

cation during the trust negotiation. And the subsequence is based on such a scenario: HUST library issues image-credential and provide online services. Surely there is much related research on how to protect sensitive information, such as hidden credentials [2], policy and so on. The detailed description is in next section.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the concrete model on how to hide credentials. We describe in details the model and the work flow as well. Later, we conclude the features of the model in Section 4. Section 5 focuses on the relative issues about the model. Meanwhile, a simple example is given on how to implement the model. Then, we give a conclusion of the paper in Section 6 and the acknowledgements in Section 7.

2 Related Work

Since credential is introduced to computer system, it is always used to stand up for a user's identity. Winslett analyzed the pitfall of current paper-credentials and proposed to use digital credentials in trust negotiation systems [8].

Credentials are taken part into identity-based credentials and attribute-based ones according to their application and purpose. The former includes X.509 [9] credential, while the latter contains SPKI/SDSI [1]. The two types of credentials are still on the development, and they can be used together.

Policy [4] describes the requirements of the resource/service provider towards the accesser. During a trust negotiation, the accesser needs to show many credentials, for credentials can denote the access rights of a user.

Holt et al. introduced hidden credentials in [2]. They gave a formal description for hidden credentials, including the concept of credential indistinguishability, and showed how to build them using the Franklin/Boneh IBE. Their work also gave compelling examples of the utility of hidden credentials. In a hidden credential system, there are four steps to build such a system: (1) create CA, (2) issue, (3) encrypt, and (4) decrypt. Our work is different from hidden credentials as follows:

- Hidden credentials focus on non-showing the contents of credentials during the authentication. Our work aims at passing credentials through a carrier, which contains the credential information.
- Hidden credentials are built on identity-based encryption (IBE), attribute values are incorporated into the identity, and the credential issuer's public key is the PKG public key. Our work provides a method to hide credentials. Surely, we can build the model into a system.

Digital watermark is mainly used to protect copyrights of publications. In order to realize the protection, there are many methods to handle this, and the mainly work focuses on the operations(insert, distill, distort, divide etc) of an image. The common methods are based on mathematic, for instance, DCT makes use of the following two formulas:

$$\begin{aligned} F &= (u,v) \\ &= \frac{2}{N}c(u)c(v)\cdot\sum_{x=0}^{N-1}\sum_{y=0} \\ &\qquad f(x,y)cos\frac{(2x+1)u\pi}{2N}sin\frac{(2y+1)v\pi}{2N}, \end{aligned}$$

and

$$\begin{aligned} f(x,y) &= \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}c(u)c(v)F(u,v) \\ &\qquad cos\frac{(2x+1)u\pi}{2N}sin\frac{2y+1}{v}\pi 2N, \end{aligned}$$

where $f(x,y)$ is the value of the image in position $(x,y), x,y = 0,1,\cdots,N-1$; $F(u,v)$ is the DCT value of the image in position $(u,v), u,v = 0,1,\cdots,N-1$.

Yusuk et al. presented a framework for web based image authentication using invisible fragile watermark in [3]. Schneider and Chang [7] proposed a method to embed content-based signature using private key as a watermark. Both of the authentication schemes require distribution of public key to verify the watermarked image. In our scheme, we avoid the public key distribution so that reduce the overhead of the authentication. In the next section, we will introduce the model.

3 Hiding Credentials Model

In this section, we will describe the hiding credentials model in detail. The model takes the features of authentication into consideration, and integrates with the digital watermark techniques. And it can be implemented in trust negotiation.

3.1 Main Idea of the Model

To simplify the problem, we give an example to illustrate such a scenario. HUST library plans to provide online services for all the users who hold library credentials issued by LIB. The library stores all the pictures for every user. Now in order to meet the requirements of the online services, the library needs to store all the credentials, which will inevitably bring a great burden. To address such a dilemma, we give a feasible resolution. The resolution can be described as four steps: (1) every user submits his digital picture to library for registering; (2) the library checks the pictures one by one and generates the relative permission credentials; (3) the library embeds the credential information into the user's picture so as to generate image-credential; (4) the library distributes the image-credentials to every user. The users can download their image-credentials at the specific URL or get it in his email based on his own choice.

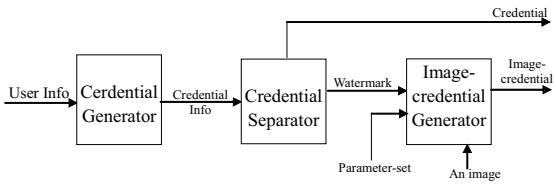

Figure 1: Process of image-credential generation

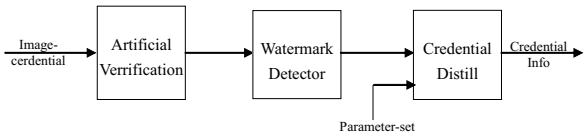

Figure 2: Process of image-credential authentication

3.2 The Detailed Model

The purpose of hiding credentials is to implement image-credentials in trust management for trust negotiation, credential storage etc. The total model consists of two parts: (1) the generation of image-credential and (2) the implementation of image-credential.

3.2.1 Image-credential's Generation

In the part, we describe how an image-credential generates. The components are shown in Figure 1. During the generation, the administrator needs to input (1) user's basic information, (2) common parameter-set for embedding algorithm and (3) a representative image, usually to be the user's picture.

The three components include as follows:

- Credential Generator (CG):
  Filter user information and prepare basic information for credential. Commonly in the manual way, after a user submits his basic individual information for registering, the administrator will have a routine to check the information. Later, an effective credential will be generated for the user if the examination passes. Since different system has various certificate form, how to generate the credential, and the required registry information are distinguishable. For instance, in GT (Globus Toolkit) systems, credentials are automated built based on the available functions such as grid-ca-sign, CA.pl -newcert, and the required information is {name, state, zip_code, city, organization, unit, email, PEM phrase}.

- Credential Separator (CS):
  Generate a credential for user and a digital watermark so as to be embedded into an image. After CG, the output (credential information) includes a user's country, city, organization, role, email, PEM pass phrase etc. The CS then produces (1) a complete credential for user to download and (2) a watermark. The credential is the common one, which can be stored as the form of ".pem". Alternatively, the watermark can be the hash of credential information so that it occupies a little storage space.

- Image-credential generator (ICG):
  Generate image-credential. There are three input parts: (1) a watermark: the information to be embedded into an image; (2) an image: the mediator, to carry watermark; (3) parameter-set: information about algorithm type (such as DCT, DFT, DWT), insertion point. Each user provides his own picture as the input image. For the embedding algorithm is variable, ICG is flexible.

3.2.2 Image-credential's Implementation

In the practical applications, an image-credential will experience three examinations, shown in Figure 2.

- Artificial Verification (AV):
  Before authentication, the other part can have an artificial verification towards the image through the comparing the image-credential to his portrait. It can ensure that only the user himself can use his image-credential. Of course, AV will have a big cost in human resource so that this examination is optional.

- Watermark Detector (WD):
  AV cannot find whether a picture is just an image or an image-credential, while WD can do so. WD will detect the watermark hidden in the image-credential. Of course, there is a relative algorithm to detect watermark according to the insertion algorithm.

- Credential Distill (CD):
  CD will separate credential information from image-credential. It needs administrator to input parameter-set about embedding information (algorithm, insertion point). Note that, in order to avoid the provision of original image, here we use the blind-watermark insertion/distillion techniques, which need not the original image when distilling the watermark.

3.3 How to Build Image-Credential System

Just like hidden credential system, we use a series of functions to realize image-credential system. The functions are listed as follows:

- Credential_create(country, organization, PEM password, email, etc.)

This function is used to create a credential. Generally, a standard credential at least contains (1) country name, (2) city name, (3) zip code, (4) organization name, (5) PEM pass phrase, (6) user information, (7) email and so on. Since the implement is various, the specific parameters are decided by the practical requirements, here we mainly give a basic framework to illustrate the issue.

- Watermark_setup(credential info)
  To produce watermark using credential information. In order to embed credential information into an image, we select some meaningful parts, such as user's role, dependency between trust participants, email. We use another hash function such as *mhash(mhash_md5*,credential info) to encode the data.

- Imagecredential_create(image, watermark, insertion algorithm, etc.)
  To insert watermark into a specified image with the selective algorithm. According to the specific application, watermark algorithms are classified as blind and non-blind. The former needs the original image to distill the watermark, while the latter does not. Usually, we use blind watermark algorithm so that will not need the original image to distill the credential information..

- Watermark_detect(image)
  The function is to detect whether an image is a picture only or an image-credential. During the process of authentication, if a user submits a useless image, it will lead the authentication to failure. Then ,we adapt a detection function to filter the meaningless images as to enhance the success rate to authentication or trust negotiation.

- Watermark_distill (image-credential, distill algorithm)
  This function is to separate the credential from image-credential. Just like the function of Im *agecredential_create()*, we need to specify algorithm to distill watermark.

4 Features of the Model

Based on the description of the model above, we can conclude the features of the model as follows:

1) More secure than common credential
   In our model, the watermarked credential provides the second security line of defense. The image is a user's personal picture, the participant can judge whether the other is the accessor himself by examining the picture. Meanwhile, the approach can prevent image-credential from being used by others. Generally, when a user logins on a trust-management system, he is required to input his id and password as well as credential. If one user lost his image-credential, others could not use his due to the consistency verification between the login id and image-credential.

2) Simple credential form easy to manage
   In the model, we use an image as the current credential. Surely, the image-credential is easy to store and manage. Meanwhile, an image can scratch the attacker's head over for no one would doubt that an image is in fact a credential. However, if the image-credential is stolen, one must submit his image again to get a new one.

3) Flexible framework easy to extend
   As mentioned above, some components are designed so flexible that they are easy to extend. For example, the component of Image-credential Generator has three input parameters. It does not specify concrete algorithm so that the administrator can select different algorithms for different applications. At the same time, it provides interfaces for new techniques.

5 Related Issues

In this section, we will discuss some much-concerned questions. Since watermark has some important features, the image-credential should meet the requirements, otherwise, anything is of no value.

5.1 How to Ensure the Quality

Generally, a successful watermark has at least four characteristics [6]: (1) invisibility: difficult to find the hidden information; (2) robustness: a watermark can resist some kinds of attacks as possible as it can; (3) non-erasure: a watermark should not be erased and (4) security: even the data is stolen by others, the contents are still unknown. The image-credential should follow the rules mentioned above. That's to say, the image-credential should be undistinguishable to the original image, resist attacks, be not easy to erase hashed credential information and can leak nothing about the credential.

5.2 How to Embed the Credentials into an Image

There are too many existing techniques to answer such a question. In the field of image process, DCT, DFT, litterwave etc are used to embed watermark into an image. Surely, we first need to turn the credential data into watermark. Then, we choose an algorithm according to the practical application and requirement. In the subsequent section, we will give a simple experiment to illustrate the total process.

Figure 3: Contrast of original image and image-credential

5.3 How Much Data Can Be Embedded into an Image

Since the storage space of an image very limited, we cannot embed too much data into an image. Pierre Moulin et al. [5] proposed a framework to evaluate the data-hiding capacity of image sources. How much data an image can be embedded is decided by (1) the space of an image, (2) the algorithm selected, (3) the insertion point and (4) the data form. So we will consider which data to be embedded into the image.

5.4 A Simple Experiment

In this part, we give a simple experiment to show how image-credential works. Assume Tom to be a professor of HUST, he submits his picture (JPEG, 2160 1440) to the online library, since we need to hide a little more data, so the image is bigger than the standard picture of lena(51251224b). The library accepts his image (if the image is valid and acceptable) and embeds some relative data into it. Suppose credential information follows the form of "subject, country, city, zip code, organization, unit,role, email", and we use *md*5_32() as the hash function. Here, we give the relative data as follows:

- Credential data: Tom, CH, wuhan, 430074, HUST, CS, professor, Tom@hust.edu.cn.
- Hashed code of credential data: 05c98a0e453094badab8684e0e1e829b.
- Image watermark: shown in Figure 3(a).
- Original image: shown in Figure 3(b), it is 50% of the real image, the form is (JPEG, 2160 1440).
- Embedding algorithm: Discrete Cosine Transfer (DCT).
- Experiment environment: we use Matlab and VC to realize the total experiment.
- Image-credential: shown in Figure 3(c), it is 50% of the real image-credential.
- Distill credential data: shown in Figure 3(d).
- Analysis on quality of image-credential: the three basic performance parameters are SNR, PSNR, NC. The relative values are described at Table 1.

6 Conclusions and Future Work

Watermark is a kind of technique in image process. We introduce it to trust management and implement it in trust negotiation, which can bring much convenience and security. In this paper, we consider both features of credential and watermark, and present a new approach to hide credential by embedding the credential data into an image. We describe the model and discuss the relative issues. At the same time, we give a simple experiment to show how it works. Our future work is to design a practical system and implement image-credential to the system.

Acknowledgements

This paper is written in the context of ChinaGrid project at CGCL lab. The project is a part of the important special project supported by National Natural Science Foundation of China under Grant No. 90412010. During the writing, members of Grid Security Team give us much useful instruction. We here thank them for their helpful comments.

References

[1] D. Clarke, J. E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Local names in SPKI/SDSI," in *Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW'2000)*, pp. 14-23, 2000.

[2] J. Holt, R. Bradshaw, K. E Seamons, and H. Orman, "Hidden crdentials," in *2nd ACM Workshop on Privacy in the Electronic Society*, pp. 1-8, 2003.

[3] Yusuk Lim, Changsheng Xu, and David Dagan Feng, "Web based image authentication using invisible fragile watermark," in *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing Conference on Visual Information Processing*, pp. 94-98, 2001.

[4] D. Marriott, and M. Sloman, "Management policy service for distributed systems," in *Proceedings of Third IEEE International Workshop Services in Distributed and Networked Environments (SDNE'96)*, pp. 2-9, Jun. 1996.

Table 1: Evaluation of image-credential

Item	Computing formula	Computing value
SNR	$S_{SNR} = 10log_{10}(\frac{\sum_{j=1}^{m}\sum_{i=1}^{n}X_{ij}}{\sum_{j=1}^{m}\sum_{i=1}^{n}(X_{i,j}-\tilde{X}_{i,j})^2})$	47.1655
PSNR	$PSNR = 10log_{10}(\frac{m\times n\times 255^2}{\sum_{j=1}^{m}\sum_{i=1}^{n}(X_{i,j}-\tilde{X}_{i,j})^2})$	49.4600
NC	$NC = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}W(i,j)W(i,j)}{\sum_{j=1}^{m}\sum_{i=1}^{n}W(i,j)^2}$	0.9796

[5] P. Moulin and J. A. O'Sullivan, "A framework for evaluating the data-hiding capacity of image sources," *IEEE Transaction on Image Processing*, vol. 11, no. 9, pp. 45-55, sept. 2002.

[6] N. Nikolaidis, and I. Pitas, "Digital image watermarking: an overview multimedia computing and systems", in *IEEE International Conference on Neural Networks for Signal Processing*, pp. 1-6, Jun. 7-11, 1999.

[7] M. Schneider and S. F. Chang , "A robust content based digital signature for image authentication," in *Proceeding IEEE International Conference on Image Process*, pp. 17-26, 1996.

[8] M. Winslett, *An Introduction to Trust Negotiation*, http://dais.cs.uiuc.edu/trustbuilder/people/Winslett.html

[9] P. Wohlmacher, "Digital certificates: a survey of revocation methods," in *Proceedings of the 2000 ACM workshops on Multimedia*, ACM Press, pp. 38-47, 2000.

Zhensong Liao male, born in 1979, a Ph.D. candidate. He studies in the department of computer science and technology, in Huazhong University of Science and Technology. He majors in Grid Security and Grid Performance Analysis. Email: zsliao@hust.edu.cn Web: http://grid.hust.edu.cn/zsliao/ Add: School of computer, Huazhong University of Science and Technology, Wuhan City Hubei Province , P.R. China Zip code: 430074

Chisong Li female, born in 1976, a Ph.D. candidate and an assistant. He studies in the department of computer science and technology, in Huazhong University of Science and Technology. He majors in Grid Security and High Performance Computing. Email: lizan@hust.edu.cn Add: School of computer, Huazhong University of Science and Technology, Wuhan City Hubei Province , P.R. China Zip code: 430074

Ruhan He male, born in 1974, a Ph.D. candidate. He studies in the department of computer science and technology, in Huazhong University of Science and Technology. He majors in Grid Security and Image Grid etc. Email: ruhanhe@hust.edu.cn Web: http://grid.hust.edu.cn/heruhan/ Add: School of computer, Huazhong University of Science and Technology, Wuhan City Hubei Province , P.R. China Zip code: 430074