Trends in Mathematics
Information Center for Mathematical Sciences
Volume 8, Number 1, June, 2005, Pages 21–28

# PRIVACY-ENHANCED PUBLIC-KEY CERTIFICATE: HOW TO EMBED AN INDIVIDUAL'S SENSITIVE INFORMATION INTO A CERTIFICATE

SEUNGJOO KIM AND DONGHO WON

Abstract. When a Certification Authority (CA) issues X.509 public-key certificate to bind a public key to a user, the user is specified through one or more subject names in the "subject" field and the "subjectAltName" extension field of a certificate. The "subject" field or the "subjectAltName" extension field may contain a hierarchically structured distinguished name, an electronic mail address, IP address, or other name forms that correspond to the subject. In this paper, we propose the methods to protect the user's privacy information contained in the "subject" field or the "subjectAltName" extension field of a public-key certificate.

## 1. Introduction

Public-key certificates, also called "digital IDs" , typically use the X.509 file format and consist of the information about the certificate's format, a unique serial number, information about the algorithm used to sign the certificate, the name of the Certification Authority (CA) that issued the certificate, the validity period of the certificate, the certificate holder's information, the certificate holder's public key, and the issuing CA's signature.

Among these, the certificate holder's information includes the name of the certificate holder and other identification information, such as nationality, email address, the holder's organization, and the department within that organization where the holder works. It could also include a picture of the holder, a codification of the holder's fingerprints, holder's passport number, and so on.

Here we should note that the current public-key certificates contain too much sensitive information related to the certificate holder's privacy, which should not be disclosed except to a few trusted parties. In terms of "information privacy" [1], this is not desirable. In this paper, we present the privacy enhanced certificate formats

*Key words and phrases.* PKI, Public-key certificate, Privacy.

This research was done, while the third author of this article was working at University of California, Irvine.

[1]Information privacy is defined as "an individual's claim to control the terms under which personal information – information identifiable to the individual – is acquired, disclosed, and used". [1]

in which the certificate holder's sensitive information is not universally verifiable but can only be checked with the certificate holder's agreement.

## 2. Privacy requirements

Need-to-know principle is one of the most fundamental security principles [2, 3, 4]. This principle means that the information is only provided to those with a legitimate need for that information – similar in concept and effect to the principle of least privilege. The practice of need-to-know limits the damage that can be done by a trusted insider who betrays our trust.

Based on the need-to-know principle, the certificate verifier should know only the information needed to perform his job. Thus, the following requirements, R1 and R2 should be considered to enhance the privacy protection of the certificate. Furthermore, in order for the certificate holder to control the abuse of his privacy information, the R3 should also be considered.

- R1 *(Invisibility)* The privacy information of a certificate holder should not be exposed to others without the certificate holder's agreement.
- R2 *(Soundness)* The certificate holder should not illegally change his/her privacy information contained in the certificate.
- R3 *(Non-transferability)* If necessary, the certificate holder can recover his privacy information from the certificate, and show it to a verifier through a confirming operation. However, a cheating verifier obtains no information from the confirming operation that allows him to convince other third party that the alleged certificate holder's information is valid or invalid.

## 3. Related works and analysis

Firstly, four relevant schemes that have been used as the (de facto) standards are analyzed. Then one scheme in literature will be discussed.

3.1. **X.509 public-key certificate.** The X.509 protocol defines the following structure for public-key certificates [5] : $X.509Cert = \langle Version, SerialNumber, Algorithm, Issuer, Validity, Subject, subjectPublicKeyInfo, Extensions, Signature\rangle$, where

- $Version$ : we use a version field to identify the certificate format.
- $SerialNumber$ : this is unique number within the CA.
- $Algorithm$ : this field identifies the algorithm used to sign the certificate.
- $Issuer$ : this field defines the name of the CA.
- $Validity$ : this field defines the date of expiration.
- $Subject$ : subject field is the information about the user to whom the certificate is issued. This field may include other information such as certificate holder's nationality, email address, organization, passport number, and so on.
- $subjectPublicKeyInfo$ : this field includes the algorithm name and the public key itself.

- *Extension* : these extensions can convey such data as additional subject identification information, key attribute information, policy information, and certification path constraints.
- *Signature* : CA's signature on the certificate.

As seen above, in X.509 public-key certificate, there is no mechanism to protect the certificate holder's privacy contained in the "subject" field or the "subjectAltName" extension field of a certificate. Thus X.509 public-key certificate violates the requirements R1 and R3.

3.2. **Hongkong post office e-Cert format.** Hongkong post office e-Cert personal certificate format [6] is based on the X.509 public-key certificate. However, in Hongkong post office e-Cert personal certificate format, the certificate holder's HKID(Hongkong Identity Card No.) number is stored in the certificate in the form of a hash value of the HKID number which has been signed by the private key of the certificate holder :

$$cert\_hkid\_hash = H(RSA_{privatekey,H}(hkid\_number)),$$

where the $H(\cdot)$ is a secure hash function and the RSA$(\cdot)$ is the signing function. The hash value *cert_hkid_hash* is put into the "subjectAltName" extension field of the certificate.

Hongkong e-Cert personal certificate format satisfies the requirements R1 and R2. However, [6] did not propose any mechanism for the certificate holder to recover HKID from *cert_hkid_hash* and prove it to a third party. This makes HKID number stored in the certificate less useful.

3.3. **KISA's Subject Identification Method.** In 2002, Korea Information Security Agency (KISA) suggested Subject Identification Method (SIM) as an IETF standard [7]. SIM is a method to securely embed a human memorizable sensitive identification information, such as social security number, driver license number, and so on, into the "subjectAltName" extension field of the certificate in the form of double-hashed value :

$$SIM = \langle r, H(H(pwd, r, sensitive\_id\_info))\rangle,$$

where the $H(\cdot)$ is a secure hash function, *pwd* is a password chosen by a certificate holder, and $r$ is a random salt generated by Registration Authority (RA).

In order to show a third party the certificate holder's sensitive identification information embedded into the certificate, we can use the following confirming process :

(1) Certificate holder $\mathcal{A}$ sends $\mathcal{B}$ *pwd*, $sensitive\_id\_info$, and her certificate in a secure manner.
(2) Verifier $\mathcal{B}$ extracts $r$ from the "subjectAltName" extension field of $\mathcal{A}$'s certificate.
(3) $\mathcal{B}$ computes the double-hashed value of (*pwd*, $r$, $sensitive\_id\_info$), and then compares the result with the double-hashed value, $H(H(pwd, r, sensitive\_id\_info))$, recorded in the $\mathcal{A}$'s certificate.

If $\mathcal{A}$ wants to prove just the possession of the sensitive identification information, $sensitive\_id\_info$, without revealing it (for example, in case the driver license number being embedded, $\mathcal{A}$ can only prove the fact that she has a driver license without divulging the license number.) she can send $\mathcal{B}$ the hash value, $H(pwd, r, sensitive\_id\_info)$ instead of $pwd$ and $sensitive\_id\_info$ in step (1).

KISA's SIM satisfies the requirements R1 and R2. However, once the verifier $\mathcal{B}$ gets $\mathcal{A}$'s $sensitive\_id\_info$ and $pwd$, he can prove other third party that the given $sensitive\_id\_info$ is $\mathcal{A}$'s sensitive identification information and is valid. Thus KISA's SIM violates the requirements R3.

3.4. **VeriSign's CZAG extension.** In 1997, VeriSign announced an optional One-Step Registration feature that included a user's country, zip code, date of birth, and gender (informally called CZAG information) in Class 1 certificates when the user do not opt-out [8]. When provided, the CZAG information is stored as encrypted form in the certificate and can be read by the participating web sites using software available from VeriSign for a licensing fee.

However, in [9], Renfro pointed out that the VeriSign CZAG feature suffered from several weaknesses : First, the system used unexpectedly weak encryption and had no revocation mechanism to revoke the participating sites whose contract lapsed or was terminated for misuse. Second, there was a clear discrepancy between users' expectations and the actual protection promised and delivered.

3.5. **Hwang's revised SET certificate.** In the original SET [10, 11, 12], sensitive credit card information including the card number, the expiry date, etc., is recorded as a hashed value rather than plain text in the cardholder's certificate. When the acquirer[2] need to verify the cardholder's signature on the Payment Instruction (PI), it first extracts the credit card information from PI, computes the hash value of credit card information, and then compares the result with the subject name recorded in the cardholder's certificate.

In this scenario, the card holder's sensitive information is exposed at the acquirer's place, and it may possibly cause unexpected loss from the cardholder's point of view. Hence, in [13], Hwang *et al.* proposed a revision of SET certificate to address the cardholder's concern.

Hwang *et al.* suggested that the cardholder's credit card information is recorded in the certificate after two times of hash computation instead of one. If it is $H(H(credit\_card\_info))$ stored in the certificate, only $H(credit\_card\_info)$ has to be shown in PI. By validating the consistency between the certificate and PI, the acquirer still can verify the validity of the card without knowing the card number. Then the acquirer sends $H(credit\_card\_info)$ received from PI to the issuer for authorization.

However, since most of the credit card number consist of only 16 digits, the $credit\_card\_info$ could be easily obtained from $H(H(credit\_card\_info))$ by the

[2]There are four roles involved in the transaction model of credit card payment. The "issuer" is a financial institution that issues a credit card to the "cardholder". The "acquirer" is a financial institution that processes payment authorizations and payments for the "merchant".

brute force attack. Thus Hwang's revised SET certificate violated the requirements R1 and R3.

Even though we can modify Hwang's scheme like KISA's SIM, this modified version still violates the requirements R3.

## 4. Our proposed privacy-enhanced public-key certificate formats

In this paper, we assume that the "subject" field of a certificate contains only the basic (non-sensitive) information on the certificate holder and the "subjectAltName" extension field includes $n$ sensitive information related to the certificate holder's privacy such as a picture of the holder, a codification of the holder's fingerprints, holder's passport number, and so on : i.e., $subjectAltName = \langle Privacy_1, Privacy_2, \cdots, Privacy_n \rangle$, where $Privacy_i$ $(1 \leq i \leq n)$ contains the certificate holder's privacy information.

4.1. **Solution 1.** First, the CA chooses a Diffie-Hellman prime $p$ (where $|p| > |Privacy_i|$ for all $1 \leq i \leq n$, $|p|$ and $|Privacy_i|$ denote the bit-length of $p$ and $Privacy_i$ respectively), a generator $g \in Z_p$ with order $p-1$, a cryptographically secure hash function $H(\cdot)$, and a secure redundancy (or padding) function $R(\cdot)$ as public system parameters [14].

In our proposed format, "subjectAltName" extension field is re-defined as follows : $NewSubjectAltName = \langle BlindedPrivacy_1, \cdots, BlindedPrivacy_n \rangle$. Here, $BlindedPrivacy_i$ $(1 \leq i \leq n)$ is defined as

$$BlindedPrivacy_i = R(Privacy_i) \bigoplus ((g^{H(i, SK_{subject})} \bmod p) \bmod 2^{|Privacy_i|})$$

with the secret key $SK_{subject}$ corresponding to the public key $PK_{subject}$ recorded in the $subjectPublicKeyInfo$ field.

From now on, the user can selectively open and show his/her privacy information to a web site as follows :

(1) The certificate holder $\mathcal{A}$ submits her certificate to the server $\mathcal{B}$.
(2) $\mathcal{B}$ tells $\mathcal{A}$ what kind of certificate holder's information is additionally required.
(3) According to the $\mathcal{B}$'s request, $\mathcal{A}$ sends $y_i = g^{H(i, SK_{\mathcal{A}})} \bmod p$, where $BlindedPrivacy_i$ contains the $\mathcal{B}$'s requested information, to $\mathcal{B}$ in a secure way.
(4) Then, $\mathcal{A}$ and $\mathcal{B}$ execute the zero-knowledge proof protocol for knowledge of the discrete logarithm of $y_i$ to the base $g$ [15] :
    (a) $\mathcal{A}$ computes $t = g^v \bmod p$ with $v \in_R Z_{p-1}$, and sends it to $\mathcal{B}$.
    (b) $\mathcal{B}$ chooses a challenge $c$ from a "small set of possible challenges", $\{1, 2, \cdots, k\}$ and sends it back to $\mathcal{A}$.
    (c) $\mathcal{A}$ computes $r = v - c \cdot H(i, SK_{\mathcal{A}}) \bmod p-1$, and sends it to $\mathcal{B}$.
    (d) $\mathcal{B}$ checks the equation, $t \stackrel{?}{=} g^r \cdot y_i^c \bmod p$.
(5) If $\mathcal{A}$ succeeds in constructing the valid proof, $\mathcal{B}$ recovers $Privacy_i$ and checks the redundancy (or padding) by computing

$$Privacy_i = R^{-1}(BlindedPrivacy_i \bigoplus (y_i \bmod 2^{|BlindedPrivacy_i|})),$$

where $R^{-1}(\cdot)$ is the inverse of $R(\cdot)$.

4.2. **Solution 2.** If $Privacy_i$ $(1 \leq i \leq n)$ is human memorizable, we can define $BlindedPrivacy_i$ $(1 \leq i \leq n)$ as $BlindedPrivacy_i$
$= R(HashedPrivacy_i) \bigoplus ((g^{H(i,SK_{subject})} \bmod p) \bmod 2^{|Privacy_i|})$,
where $HashedPrivacy_i = H(Privacy_i)$.

As in KISA's SIM, if $\mathcal{A}$ wants to prove just the possession of $Privacy_i$ without revealing it, she send $\mathcal{B}$ $y_i = g^{H(i,SK_{\mathcal{A}})} \bmod p$. After $\mathcal{A}$ and $\mathcal{B}$ execute the zero-knowledge protocol for the discrete logarithm of $y_i$ to the base $g$, $\mathcal{B}$ recovers $HashedPrivacy_i$ and checks the redundancy (or padding) of it.

If $\mathcal{B}$ additionally requests to show the privacy information, $\mathcal{A}$ can also send $\mathcal{B}$ her memorized $Privacy_i$. Then $\mathcal{B}$ checks if $H(Privacy_i) \stackrel{?}{=} HashedPrivacy_i = R^{-1}(BlindedPrivacy_i \bigoplus (y_i \bmod 2^{|BlindedPrivacy_i|}))$.

## 5. Analysis

5.1. **Invisibility.** A generator $g$ with order $p-1$ has the property that every element in $Z_p$ can be expressed as a power of the $g$, and, in the random oracle model [16], we assume that the hash function $H(\cdot)$ behaves like a random function. Thus $g^{H(i,SK_{subject})} \bmod p$ is indistinguishable from an integer randomly chosen in $Z_p$, and $R(Privacy_i) \bigoplus (g^{H(i,SK_{subject})} \bmod p)$ is also random.

Hence no one without $SK_{subject}$ can extract an information on $Privacy_i$ from a certificate.

5.2. **Soundness.** Assume that cheating certificate holder $\mathcal{A}$ tries to illegally change her information $Privacy_i$ into $\overline{Privacy_i}$. To do this, in step (3), $\mathcal{A}$ computes and sends $\overline{y_i} = BlindedPrivacy_i \bigoplus R(\overline{Privacy_i})$ to $\mathcal{B}$, and makes a zero-knowledge proof for knowledge of the discrete logarithm of $\overline{y_i}$ to the $g$. However, because of the hardness of discrete logarithm problem, cheating $\mathcal{A}$ can succeed with a probability of at most $k^{-1}$ in constructing a valid proof of knowledge of the discrete logarithm of $\overline{y_i}$ to the base $g$.

Let's consider another scenario. In this, the cheating $\mathcal{A}$ repeats to find $sk \in_R Z_{p-1}$, until $BlindedPrivacy_i \bigoplus ((g^{sk} \bmod p) \bmod 2^{|BlindedPrivacy_i|})$ has a correct format. However, because of the secure redundancy function $R(\cdot)$, the success probability is negligible.

5.3. **Non-transferability.** Furthermore, the server $\mathcal{B}$ cannot leak $\mathcal{A}$'s privacy information to others. Of course, $\mathcal{B}$ can make a transcript of the exchange between $\mathcal{A}$ and $\mathcal{B}$. But $\mathcal{B}$ cannot use this transcript to convince others, because, for any privacy information $\overline{Privacy_i}$, he can always build a simulator that fakes $\mathcal{A}$'s knowledge. To simulate the transcripts, it needs to apply the common trick for proving zero-knowledge by picking the challenge in advance :

(1) Choose the privacy information $\overline{Privacy_i}$.
(2) Set $\overline{y_i} = (pad_i \| (BlindedPrivacy_i \bigoplus R(\overline{Privacy_i}))) \in_R Z_p$, where $pad_i$ is a $(|p| - |BlindedPrivacy_i|)$-bit random padding.
(3) Pick $\overline{r} \in_R Z_{p-1}$ and $\overline{c} \in_R \{1, 2, \cdots, k\}$.

(4) Set $\overline{t} = g^{\overline{r}} \cdot \overline{y_i}^{\overline{c}} \bmod p$.
(5) Output($\overline{y_i}$, $\overline{t}$, $\overline{c}$, $\overline{r}$, $\overline{Privacy_i}$).

Note that the transcript of the simulator is indistinguishable from the transcript of a correct execution of the protocol. (In fact the distribution of the transcript of a simulator is identical to the distribution of the transcript of a honest execution of the protocol.) This is the reason why we do not blind the certificate holder's privacy information with the block cipher such as 3DES, AES, etc.: for example, $BlindedPrivacy_i = E_{H(i,SK_{\mathcal{A}})}(Privacy_i)$.

## 6. Conclusion

In this paper, we summarized the requirements for certificate holder's privacy protection, and analyzed four relevant standards : X.509 public-key certificate, Hongkong's e-Cert, KISA's SIM, VeriSign's CZAG, and a protocol proposed in recent paper. Based on the need-to-know principle, we also proposed the modified public-key certificate formats to secure certificate holder's privacy.

## References

[1] IITF principles, supra note 19, at 5.
[2] J.J. Hwang and S.C. Hsueh, "Greater protection for credit card holders : a revised SET protocol", Computer Standards and Interfaces 19, 1998, pp.1-8.
[3] M. Bellare, J.A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, G. Tsudik, E.V. Herreweghen, and M. Waidner, "Design, implementation, and deployment of the iKP secure electronic payment system", IEEE Journal on Selected Areas in Communications 18(4), April 2000, pp.611-627.
[4] Australian Transaction Report and Analysis Center, "RGEC report – research and technical advice volume 3", Dec. 1999, http://www.austrac.gov.au/text/publications/rgec/3/pdf/ch1.pdf.
[5] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and Certificate Revocation List (CRL) profile", IETF RFC 3280, April 2002.
[6] Hongkong Post, "e-Cert certification practice statement", 2001
[7] J. Park, J. Yoon, S. Kim, S. Park, J. Lee, H. Lee, and T. Polk, "Internet X.509 public key infrastructure subject identification method", draft-ietf-pkix-sim-03.txt, February 2004.
[8] VeriSign, "VeriSign enhances digital IDs to enable universal website login and one-step registration", http://www.verisign.com/press/product/isv.html, April 1997.
[9] S.G. Renfro, "VeriSign CZAG: privacy leak in X.509 certificates" Proceedings of the 11th USENIX Security Symposium, August 2002.
[10] MasterCard and VISA, "Secure Electronic Transaction (SET) specification", Book 1 : Business Description, version 1.0 (1997).
[11] MasterCard and VISA, "Secure Electronic Transaction (SET) specification", Book 2 : Programmer's Guide, version 1.0 (1997).
[12] MasterCard and VISA, "Secure Electronic Transaction (SET) specification", Book 3 : Formal Protocol Definition, version 1.0 (1997).
[13] J.J. Hwang, T.C. Yeh, and J.B. Li, "Securing on-line credit card payments without disclosing privacy information", Computer Standards and Interfaces 25, 2003, pp.119-129.
[14] W. Diffie and M.E. Hellman, "New directions in cryptography", IEEE Trans. Inform. Theory, IT-22, 6, 1976, pp.644-654.
[15] J. Camenisch and M. Stadler, "Proof systems for general statements about discrete logarithms", Technical Report TR 260, 13 pages Department of Computer Science, ETH Zurich, March 1997.

[16] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols", Proc. First Annual Conference on Computer and Communications Security, ACM, 1993.

School of Information and Communication Engineering,, Sungkyunkwan University,, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Korea

*E-mail address*: skim@ece.skku.ac.kr

*E-mail address*: dhwon@simsan.skku.ac.kr