HAL
archives-ouvertes.fr

A Generalization of ISO/IEC 24761 to Enhance Remote Authentication with Trusted Product at Claimant

Asahiko Yamada

▶ **To cite this version:**

Asahiko Yamada. A Generalization of ISO/IEC 24761 to Enhance Remote Authentication with Trusted Product at Claimant. Hannes Federrath; Dieter Gollmann. 30th IFIP International Information Security Conference (SEC), May 2015, Hamburg, Germany. IFIP Advances in Information and Communication Technology, AICT-455, pp.145-158, 2015, ICT Systems Security and Privacy Protection. <10.1007/978-3-319-18467-8_10>. <hal-01345102>

HAL Id: hal-01345102
https://hal.inria.fr/hal-01345102

Submitted on 13 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A Generalization of ISO/IEC 24761 to Enhance Remote Authentication with Trusted Product at Claimant

Asahiko Yamada

National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, Japan
yamada.asahiko@aist.go.jp

Abstract. In this paper, a data structure to enhance remote authentication is proposed generalizing the concept of ISO/IEC 24761. Current technologies do not provide sufficient information on products which are used in the authentication process at the Claimant to the Verifier. As a result, the Verifier cannot sufficiently distinguish the authentication result executed with a trusted product from that without a trusted product. The difference is made clear if an evidence data of the execution of authentication process at the Claimant is generated by the trusted product and used for verification by the Verifier. Data structure for such a data is proposed in this paper as client Authentication Context (cAC) instance. Relation to other works and extension of the proposal are also described for further improvement of remote authentication. For this proposal to realize, standardization activities are to be taken as the next steps.

Keywords: Biometric authentication· Cryptographic Message Syntax (CMS) · digital signature· IC card· initial authentication· Public Key Infrastructure (PKI) · remote authentication· tamper-resistant device· trusted device.

1 Introduction

In networked IT environments, remote authentication is essential. Remote authentication is one of the most important elements of the security of innumerous applications of governmental, commercial, academic systems and so forth and it is applied to them. Although policy-based authorization makes the Relying Party (RP) possible to change the service level reflecting the level of assurance of the identity, the level of trust of the environment of the Claimant where the authentication protocol is executed is not taken into account appropriately and sufficiently. This paper proposes a mechanism with which the Verifier can know the level of trust in the authentication process executed at the Claimant of remote authentication under the condition that a trusted product with the digital signature generation function such as a tamper-resistant IC card is used for authentication at the Claimant. There are two cases for the activation of the private key, with passphrase or biometrics. Both cases are discussed in this paper, extending the former to the latter.

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Current technologies

Progresses in authentication technologies have been significant in the last decade. Single Sign-On (SSO) technologies such as Security Assertion Markup Language (SAML) and OpenID have made general service providers free from authentication itself and only consume the assertion generated by the Verifier, which is called Identity Provider (IdP) in SAML and OpenID Provider (OP) in OpendID. While the technologies in subsequent authentication between the Verifier and the RP, the consumer of the assertion, have been progressed, the technologies in initial authentication between the Claimant and the Verifier have been stable. In Web systems, Transport Layer Security (TLS) protocol including its predecessor Secure Sockets Layer (SSL) has been dominant for about twenty years and is still the most major and standard technology. The variation of tokens has not changed, something you know, something you have, and something you are.

In SAML [1], authentication context is optionally used in assertions to give additional information for the RP in determining the authenticity and confidence of assertions. Authentication context contains information how the user is authenticated at the Claimant. Although the IdP generates an authentication context at the initial authentication, the IdP does not always have sufficiently trustable information about the authentication process at the Claimant in order to generate an authentication context, considering that the execution environment of the Claimant is not always so sufficiently trustable to the IdP as that of the IdP to the RP. For example, the IdP does not have sufficient information to judge whether a tamper-resistant IC card with digital signature function is used at the Claimant or not. It is true that a private key stored in a tamper-resistant IC card can be distinguished with the qualified certificate specified in RFC 3739[16] with the qualified certificate statement 5.2.4 in ETSI TS 101 862 [3] which is for secure signature-creation devices with the conditions in Annex III of Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures [4]. But the purpose of X.509 certificate itself is to describe the attributes of the user and his/her public key. So the use of the extension of X.509 certificate in such ways does not match its original purpose.

Guidelines and requirements on authentication have been also studied well. One of the most important results in this area is NIST SP 800-63-2 Electronic Authentication Guideline [13]. It assigns requirements on tokens, token and credential management, authentication process, and assertions to each Level of Assurance (LoA) from Level 1 to Level 4 each of which was introduced in OMB M-04-04 [14]. Although SP 800-63-2 requires Level 4 to use Multi-Factor (MF) hardware cryptographic token such as tamper-resistant cryptographic IC card, any of the current authentication protocols does not show sufficient evidence that such a token is used at the Claimant. At Level 4, such a protocol may be unnecessary because only in-person registration is allowed at Level 4 and it can be assured that such a token is issued and used in authentication process at the Claimant. But in Level 2 and Level 3 to which remote registration is allowed, it is not evident for the Registration Authority (RA) or the Credential Service Provider (CSP) whether a public key pair is generated and stored in a tamper-resistant IC card in registration process or not, and it is not evident either to the Verifier whether such a

product is used in authentication process or not, for example. In Level 2 and Level 3, it would be desirable for the RP to know more information about the trust level of the authentication process executed at the Claimant. Then the RP can provide its services according to the level of trust.

In the area of biometric authentication, a similar motivation and the solution can be found in ISO/IEC 24761 Authentication Context for Biometrics (ACBio) [10]. The work in this paper is a generalization of the idea in ISO/IEC 24761.

In the following, terms and definitions in SP 800-63-2 are basically applied unless otherwise specified.

3 ISO/IEC 24761, a related work in biometric authentication

ISO/IEC 24761 referred as ACBio is an enhancement using evidence data generated by execution environment for biometric authentication while this proposal is that for PKI based authentication.

ACBio is a solution to the technological issues of biometric authentication used in the Internet environment. The issues are listed in the threat analysis done in [15] and they are categorized into three. The first is that subprocesses may be replaced with malware. Here a subprocesses is an execution component in biometric authentication, namely data capture to sense human body to output raw biometric sample, intermediate signal processing to process raw biometric sample to intermediate processed biometric sample, final signal processing to process intermediate biometric sample to processed biometric sample, storage to store and retrieve enroled biometric reference template , biometric comparison to compare and calculate the score of similarity of processed biometric sample to biometric reference template, or decision to decide match or non-match from the score. The second is that the enroled biometric reference template may be replaced with that of another person such as an attacker. The last is that the data transmitted between subprocesses may be replaced with another data.

ACBio has solved these issues by generation and verification of evidence data of the executed biometric processing under the assumption that trusted biometric products are used. Authentication using the specification of ACBio is called ACBio authentication. A trusted biometric product is called a Biometric Processing Unit (BPU) in ACBio.

In production process, the BPU manufacturer has to generate the BPU report to BPU product in ACBio authentication. In the BPU report which is a data of type `SignedData` digitally signed by the BPU manufacture, information about the BPU such as the modality which the BPU processes, the subprocesses implemented and the data flow in the BPU are contained. In ACBio authentication, a key pair for the BPU is generated and the X.509 certificate for the public key of the BPU is issued. The data generated at production process are all stored in the BPU.

At registration process, Biometric Reference Template (BRT) certificate is issued to BRT by BRT certificate authority in ACBio authentication. The BRT certificate is a digitally signed data by a BRT certificate authority and links a BRT to a user. For privacy reasons, the BRT certificate does not contain the BRT itself but contains the hash value of the BRT. There is an evidence data named ACBio instance for enrolment,

which is digitally signed with the private key of the BPU, to show the generation and storage of the BRT is securely done in the BPU. In ACBio authentication, each BPU used in the enrolment generates its ACBio instance for enrolment. The ACBio instances for enrolment show the BPUs used in the enrolment and the integrity of the data transmitted between the BPUs if the enrolment is done with multiple BPUs. The ACBio instances for enrolment are optionally set in the BRT certificate. From the ACBio instances for enrolment, the BPU where the BRT is stored is also identified. ACBio instances for enrolment may be used to check whether the enrolment satisfies the security requirement or not by the BRT certificate authority to issue the BRT certificate, and also by the Verifier later in authentication process, depending on the security policies of the BRT certificate authority and the Verifier respectively.

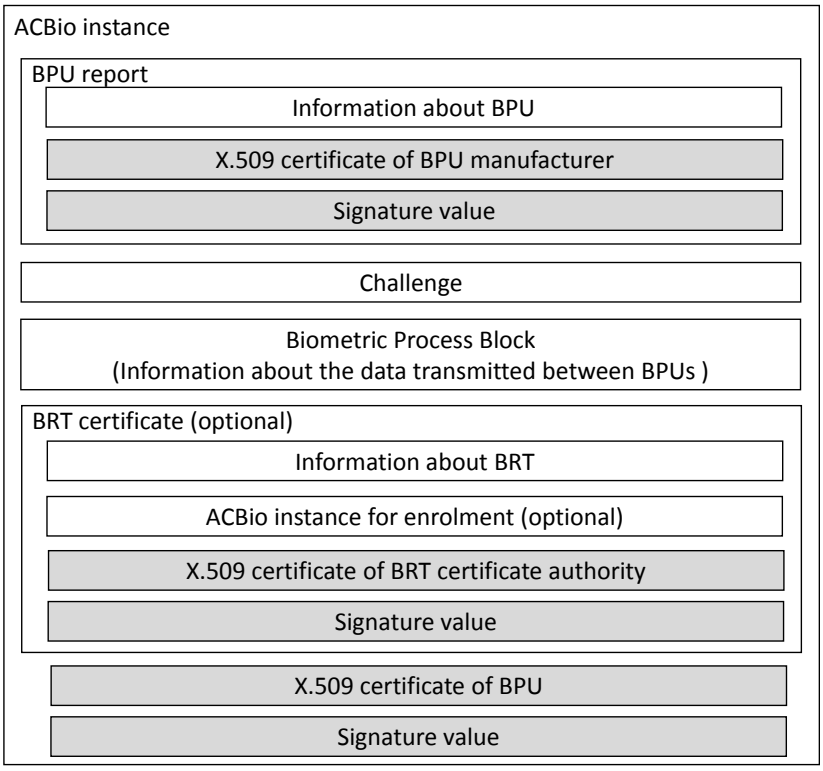

Fig. 1. Simplified data structure of ACBio instance

At authentication process, ACBio authentication assumes challenge response mechanism. An ACBio instance is generated in each BPU which takes part in biometric authentication process. Fig. 1 overviews the data structure of ACBio instance.

An ACBio instance contains the BPU report. This gives information to the Verifier about the product which executes authentication protocol at the Claimant.

The triple of the challenge which is called Control Value in ACBio, the Biometric Process Block, and the BRT certificate, which is contained only if the BPU stores the

BRT, is contained in an ACBio instance. This shows that the authentication process at the Claimant is successfully executed.

The ACBio instance contains all the data mentioned above and the digital signature of those with the private key of the BPU. This gives the evidence of the successful execution of the authentication protocol done at the Claimant.

Toshiba Solutions Corporation in Japan has already implemented ACBio authentication into a product. Using this product, its customer company has built an in-house system.

The idea of ACBio enhances biometric authentication used in the Internet but the name ACBio (Authentication Context for Biometrics) is inappropriate. As written in 2, authentication context in SAML is information in assertions, i.e., information sent from the Verifier to the RP while ACBio instance is not but is sent form the Claimant to the Verifier. In this context, the name cAC (client Authentication Context) is used in this paper.

4 Problem definition

In an environment where a trusted product is not used at the Claimant for PKI based authentication protocol, there may be possibilities that the private key is compromised, i.e., an attacker may get and misuse it for spoofing. When a trusted product is used, it will be assured that the private key is not stolen under certain conditions, as assumptions listed in 5. There should be an authentication protocol for the Verifier to distinguish the above two cases.

5 Assumptions

In this paper, suppose that the trusted products considered have the following assumptions.

(A) The trusted product has digital signing function.

(B) The trusted product has generation function of public key pairs.

(C) The private key embedded in production process or generated in the trusted product cannot be exported.

(D) The trusted product has a function to manage the triples of private key, public key, and X.509 certificate of the public key.

(E) The trusted product can digitally sign only with a private key embedded in production process or one generated in the trusted product.

(F) The trusted product has functions proposed in this paper for authentication process.

In addition to the above assumption to the trusted products, assume that the whole production process of trusted products is trusted. Therefore the private key embedded to the trusted product is never leaked in the production process.

The assumptions (A) and (D) are necessary to generate data such as SignedData in a product. If a trusted product can digitally sign with an imported private key, then the private key may have been already compromised before it is imported. Therefore the

assumption (E) is necessary to assure that the digital signature is generated by the trusted product. To assume (E), the private key has to be generated in production process or it has to be generated in the trusted product after production process. Therefore the assumption (B) is necessary. Without (C) the private key may be misused.

These assumptions are appropriate since tamper-resistant PKI cards conformant to ISO/IEC 7816-4 [5] and 8 [6] satisfy (A) to (E). The implementation of (F) is not difficult as is to be seen later.

In the following, the detailed communication protocol including negotiation is not discussed.

6 Proposal

In this paper, a data structure named client Authentication Context (cAC) is proposed to enhance the PKI based authentication protocol under the condition that a trusted product with assumptions from (A) to (F) is used at the Claimant. Hereafter a trusted product with the assumptions is called a cAC product and authentication using cAC is called cAC authentication. The cAC authentication enables the Verifier to judge whether a cAC product is used for the authentication process or not. In short, this is done with a combination of product authentication and user authentication techniques, PKI based user authentication assured by PKI based product authentication. Authentication protocol for cAC authentication is also discussed. The problem cannot be solved only with the authentication protocol but with a series of processes beginning from the production process as in ACBio. This proposal tries to give a solution to the problem as universal as possible.

6.1 Production process

In the production process of cAC products, the cAC product manufacturer needs several procedures for cAC authentication afterwards.

The cAC product manufacturer has to generate its public key pair and have the X.509 certificate issued in advance. The private key is used to digitally sign cAC product report which gives information about the cAC product. Digitally signed by the cAC product manufacturer, cAC product report becomes a trusted data if there is an assumption that the Verifier trusts the cAC product manufacturer. Hereafter `certificateMnf` denotes the X.509 certificate of the cAC product manufacturer.

In the following, type means ASN.1 type.

For generation of cAC product report, a type `SignedData`, specified in RFC 3852 [17] /RFC 5911 [18] Cryptographic Message Syntax (CMS), is applied. In `SignedData`, the signed object is the field `encapContentInfo` of type `EncapsulatedContentInfo` which consists of two fields. The first is a field to indicate the data type of the data which is DER encoded in the second field. To indicate the data type, `OBJECT IDENTIFIER` type is used. The second is the content itself carried as an octet string whose data type is identified with the first field.

There are some categories of cAC products. For example, in a category, a cAC products activates the private key with a passphrase, in another it may activate the private key with biometric authentication. Here only the former category is discussed. The latter will be discussed later.

There is another categorization of products into a category of software and one of hardware.

The type identifier for the content of cAC product report is defined as `id-content-cPR-passphrase` of type `OBJECT IDENTIFIER`. The corresponding content type `ContentCPRPassphrase` identified by `id-content-cPR-passphrase`, is define to have four fields. The first field `productType` gives information that the product is a software or hardware product. The second field `levelCMVP` is to show the level of Cryptographic Module Validation Program specified in FIPS 140-2 [12] and ISO/IEC 19790 [9] if the cryptographic module in the cAC product is certified. The third `reqLengthPassPhrase` and fourth `minLength` are a field to show whether there is a requirement for the length of passphrase, and a field for the required minimal length of passphrase if there is. With the above information, the Verifier knows the extent to which it can trust the cAC product. In ASN.1 notation, `ContentCPRPassphrase` is specified as follows:

```
ContentCPRPassphrase :: = SEQUENCE {
     productType         ProductType,
     levelCMVP           LevelCMVP,
     reqLengthPassPhrase BOOLEAN,
     minLength           INTEGER OPTIONAL}
ProductType ::= ENUMERATED {
     software  (0),
     hardware  (1) }
LevelCMVP ::= ENUMERATED {
     none    (0),
     level1  (1),
     level2  (2),
     level3  (3),
     level4  (4) }
```

Let `SIGNEDDATA(eCTypeID, ContentType)` denote a type which is derived from `SignedData` where the fields `eContentType` in `encapContentInfo` is specified to take `eCTypeID` and `eContent` in `encapContentInfo` is `OCTET STRING` of the DER encoding of a data of type `ContentType`.

Then a type `CACProductReport` for cAC product report is defined as `SIGENDDATA(id-content-cPR-passphrase, ContentCPRPassphrase)`. A data of this type shall be digitally signed with the private key of a cAC product manufacturer. Therefore `certificateMnf` is set in one of certificates in the cAC product report.

At the last of production process of cAC product, a public key pair shall be generated and the X.509 certificate for the public key, which is denoted by `certificatePrd`

hereafter, shall be issued. In the X.509 certificate, the field subject of type `Name` in the field `tbsCertificate` of type `TBSCertificate` shall contain the name of the cAC product and that of the cAC product manufacturer. The name of the cAC product manufacturer in the field `subject` shall be the same as that in the field `subject` in the X.509 certificate of the cAC product manufacturer in the cAC product report. The public key pair and the X.509 certificate shall be stored in the cAC product together with the already generated cAC product report.

6.2 Registration process

To become a Claimant in PKI based authentication process, a user has to generate the public key pair and get the X.509 certificate. It is also the same in cAC authentication, but the Claimant has to generate the key pair in the cAC product. Otherwise, if the public key pair is generated outside the cAC product, the imported key pair cannot generate digital signature because of assumption (E).

There is no corresponding data in cAC authentication to the ACBio instance for enrolment. There seems to have to be "key generating context" in cAC authentication. But it is redundant because the private key used in authentication process is assured to have been generated in the same cAC product in registration process by assumptions (B) and (E). Furthermore it is assured that the digital signature is generated in the cAC product by assumption (C) and (E).

6.3 Authentication process

In the cAC product, the pair of the private key and X.509 certificate for the cAC product, the pair of the private key and X.509 certificate for the user, and the cAC product report are stored before the authentication process starts. With these data, a cAC instance, an evidence data of the cAC authentication process at the Claimant, is defined. In this paper, challenge response mechanism is assumed to be applied in the authentication protocol in order to prevent replay attacks. This assumption is appropriate since most of the protocols used in remote authentication apply challenge response mechanism. But before defining the authentication protocol, the data structure is defined.

A type `ChallengeSignedByUser` is defined as `SIGNEDDATA(id-data, OCTET STRING)`. When the Claimant receives a challenge from the Verifier, a data of type `ChallengeSignedByUser` is generated at the Claimant setting the challenge of type `OCTET STRING` into `eContent` and digitally signing with the user's private key which is activated with a passphrase input by the user. Hereafter a data of type `ChallengeSignedByUser` is called a CSBU. A type `ContentClientAC` identified by the type identifier `id-contentClientAC` is defined as:

```
ContentClientAC :: = SEQUENCE {
   cACProductReport      CACProductReport,
   challengeSignedByUser ChallengeSignedByUser }
```

Then a type `ClientACInstance` is defined as: `SIGNEDDATA(id-contentClientAC, ContentClientAC)`. To generate a cAC instance of type `ClientACInstance`, the cAC product report and the data of `ChallengeSignedByUser` generated as in the above are used. For digital signing, the private key of the cAC product is used. Therefore the X.509 certificate set in certificates in the cAC instance is `certificatePrd`. Fig. 2 shows a simplified data structure of cAC instance where shaded boxes indicate data specified in RFC 3852.

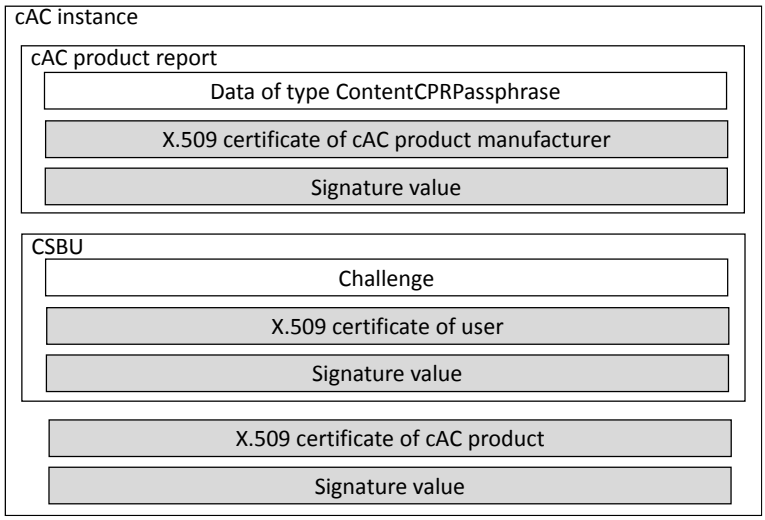

Fig. 2. Simplified data structure of cAC instance

At the Verifier, a cAC instance is verified as follows:

(1) The Verifier checks the cAC product report. This consists of signature verification, checking of the product type, the level of cryptographic function, and the passphrase policy implemented on the cAC product. By checking the cAC product report, the Verifier can know if the cAC product satisfies the authentication policy of the RP for example, and that the cAC product is manufactured by the cAC product manufacturer with the X.509 certificate in the cAC product report.

(2) The Verifier checks the CSBU. The Verifier can know whether there was a replay attack or not by checking the challenge in the CSBU, and whether the Claimant generated the digital signature or not. The digital signature of the challenge is verified with the public key in the X.509 certificate in the CSBU.

(3) The Verifier verifies the digital signature of the cAC instance. With this verification, the Verifier can conclude that the Claimant has done the authentication process in the cAC product because the digital signature has been calculated with the private key of the cAC product which has been stored in the cAC product since key generation because of assumptions from (A) to (E). This solves the problem stated in 4.

Fig. 3 summarizes all the operations in all the processes that are proposed in this paper. In the region of processing in cAC product in Fig.3, surrounded by the dotted

line, the relations of "contained" and "digitally sign" are assured by the assumption from (A) to (F). These make the evidence of the execution of authentication process in cAC product trusted.

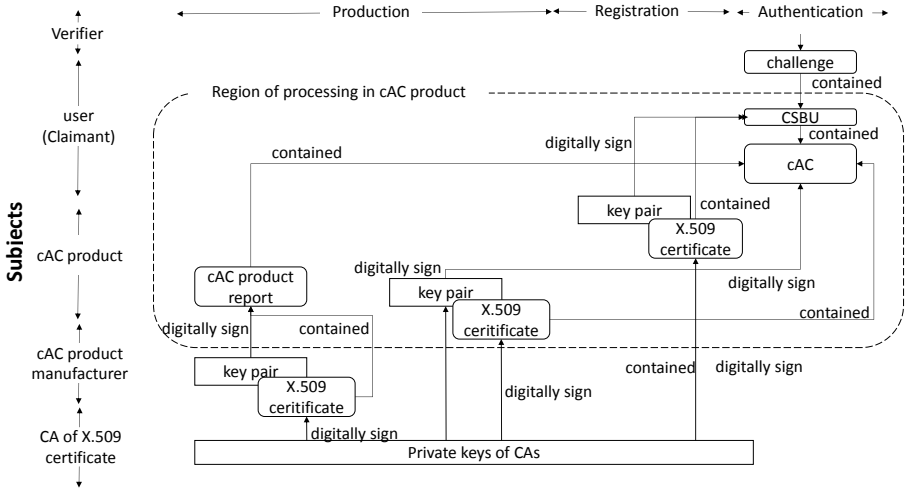

Fig. 3. Trust relation in cAC authentication

7 Considerations

7.1 Comparison with the qualified certificate model

The qualified certificate can be used to show that the private key paired to the public key to which the certificate is issued is stored and used in a trusted product. If a vulnerability of the trusted product concerning the storage of the private key is found, then all the qualified certificates to the users who use the trusted product to digitally sign with have to be revoked while only one cAC product report of the trusted product has to be invalidated in the proposed model. There is a big difference in efficiency of the validation process at the Verifier. For the CA to revoke the qualified certificates, the information about the trusted product has to be stored to each of the qualified certificates issued.

The qualified certificate does not show which trusted product the private key is stored and used in. As a result, the Verifier can know nothing about the trusted product used in the authentication process and can give less information about the authentication process to the RP than in the proposed model. This information to the RP is important for policy-based authorization. Therefore the proposed model is more appropriate for the policy-based authorization than the qualified certificate model.

7.2 **Application of the proposal to ITU-T X.1085 | ISO/IEC 17922 BHSM**

In ITU-T SG 17 and ISO/IEC JTC 1/SC 27, a project to make a common text on Biometric Hardware Security Module (BHSM) is going on. It is at Committee Draft stage in SC 27 at the time of writing this paper. A typical example of BHSM is a PKI card in which the private key is activated by biometric authentication. To show that a BHSM is used in the authentication process, cAC can be applied with modification where the modification depends on the security policy on authentication. If only the modality used has to be known by the Verifier, then replacement of `ContentCPRPassphrase` in `CACProductReport` with `ContentCPRBiometicsSimple`, which is defined as follows, suffices to apply cAC to BHSM:

```
ContentCPRBiometicsSimple ::= SEQUENCE {
    productType      ProductType,
    levelCMVP        LevelCMVP,
    biometricType    BiometricType,
    biometricSubype  BiometricSubtype OPTIONAL }
```

Here `BiometricType` and `BiometricSubtype` are types for modalities defined in ISO/IEC 19785-3 [8]. This is the simplest case of the application of cAC to BHSM.

If the Verifier needs to validate the biometric authentication executed in the BHSM through the authentication protocol, the combination of cAC and ACBio will be required. This is the most complex case of the application of cAC to BHSM. To deal with this issue, `ContentCPRBiometicsFull` shall be defined as follows to replace `ContentCPRBiometicsSimple`:

```
ContentCPRBiometicsFull ::= SEQUENCE {
    productType        ProductType,
    levelCMVP          LevelCMVP,
    bpuFunctionReport  BPUFunctionReport,
    bpuSecurityReport  BPUSecurityReport}
```

Here `BPUFunctionReport` and `BPUSecurityReport` are types defined in ACBio to show the specification of function and security of BPU. In this case, a BHSM is also considered as a BPU from the view point of ACBio. A cAC product report with `ContentCPRBiometicsFull` is regarded as an extension of BPU report with two fields, `productType` and `levelCMVP`, added to the data structure of `BPUReportContentInformation` in a BPU report.

In registration process, X.509 certificate and BRT certificate shall be issued. The issuance of these two types of certificate will be done at different TTPs. In ACBio, harmonization with PKI authentication has been considered. When both PKI and biometrics are used, the X.509 certificate shall be issued before the BRT certificate is issued. From a BRT certificate, the corresponding X.509 can be referenced with the field `pkiCertificateInformation` of type `PKICertificateInformation` in the BRT certificate. This correlates PKI authentication and biometric authentication.

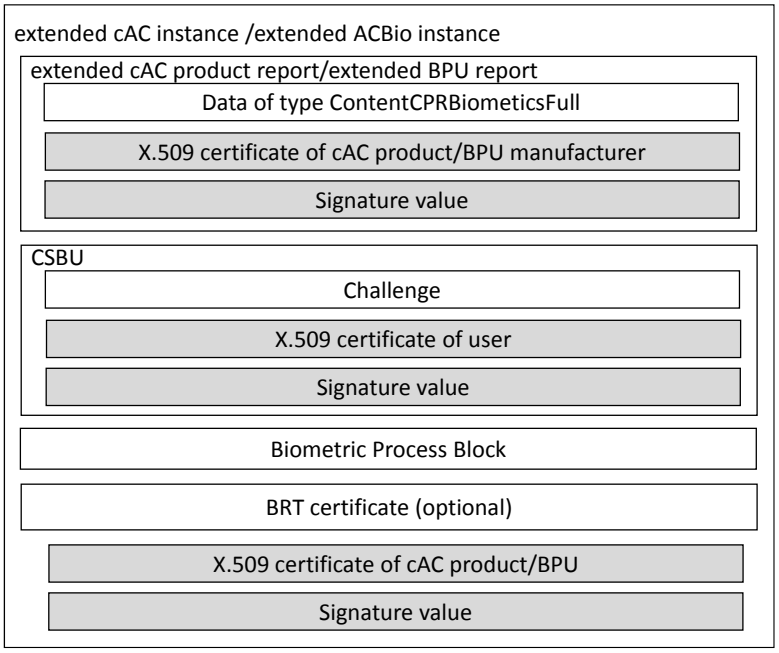

Fig. 4. Simplified data structure of extended cAC instance

In authentication process, extended cAC instance whose data structure is depicted in Fig.4 shall be used. The extended cAC instance can be also regarded as extended ACBio instance. If it is regarded as extended ACBio instance, the BPU report and the control value shall be replaced with the above defined cAC product report and CSBU respectively. With this extended cAC instance (or extended ACBio instance), cAC authentication and ACBio authentication are unified.

7.3 Future works

In this paper, there is an assumption that the Verifier trusts the cAC product manufacturer. This is a strong assumption because it is difficult for the Verifier to know all the cAC product manufacturers that are trusted beforehand. It is desirable to weaken this assumption.

For security evaluation and certification of products, there is a scheme Common Criteria (CC) [2] which is also internationally standardized as ISO/IEC 15408 [7]. In the CC world, there is a movement to specify collaborative Protection Profile (cPP) to share security requirements for certain categories of security related products. At the time of writing this paper, Full Disk Encryption cPPs are posted for comments. For informing security features of a CC certified product, it is appropriate to show the cPPs which the product conforms to because cPPs are security requirements.

Let `CPPsConformantTo` be a type defined as follows:

```
CPPsConformantTo ::= SEQUENCE OF IdentifierCPP
IdentifierCPP ::= OBJECT IDENTIFIER
```

Here `IdentifierCPP` is used to assign an object identifier to a cPP. Then the type `CPPsConformantTo` can mean a set of cPPs which a CC certified product conforms to. Let `ContentCCCertificate` be a type defined as follows:

```
ContentCCCertificate::= SEQUENCE {
    nameProduct      Name,
    cPPsConformantTo CPPsConformantTo }
```

and let `id-content-CCCertificate` be the object identifier for the type `ContentCCCertificate`. Then the type `CCCertificate` defined as `SIGNEDDATA(id-cPPs-ConformantTo,CPPsConformantTo)` is used to express a CC certificate of a product if the private key of a CC certificate authority is used to digitally sign in generating a data of this type. The operation of the verification of this digital CC certificate will be easy to deal with for the Verifier since it needs to prepare only seventeen X.509 certificates in advance as there are only seventeen CC certification authorities worldwide (See http://www.commoncriteriaportal.org/ccra/members/). If signed CC certificate is standardized, the Verifier only needs to trust seventeen CC certification authorities. This will weaken the assumption stated at the beginning of this subsection. When `CCCertificate` becomes commonly used, the redefinition of type `ContentCPRPassphrase` adding a new field of type `CCCertificate` will make the cAC product report a more trustable data to the Verifier.

As is written at the end of 5, the communication protocol including negotiation is not discussed and to be specified in the next step. Adding new authentication contexts corresponding to cAC authentications to the OASIS standard related to authentication context is also necessary.

8 Conclusion

A new data cAC instance is proposed to improve the authentication process between the Claimant and the Verifier in remote authentication by giving the evidence data of execution of authentication process at the Claimant. To realize this proposal, standardization activities on the specification of cAC instance, the authentication protocol applying cAC authentication are necessary as the next steps.

Acknowledgement. The author appreciates Mr. Tatsuro Ikeda of Toshiba Solutions Corporation for a lot of discussions related to this work. Without these discussions, the author could not have reached the basic concept of this proposal.

References

1. Advancing open standards for the information society (OASIS). Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard (2005)
2. Common Criteria Recognition Arrangement. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, September 2012, Version 3.1 Revision 4, CCMB-2012-09-001 (2012)
3. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures (2000)
4. European Telecommunications Standards Institute (ETSI). ETSI TS 101 862 V1.3.1 Qualified Certificate profile (2004)
5. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 7816-4:2013, Identification cards -- Integrated circuit cards — Part 4: Organization, security and commands for interchange (2013)
6. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 7816-8:2004, Identification cards -- Integrated circuit card — Part 8: Commands for security operations (2004)
7. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 15408-1:2009, Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model (2009)
8. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 19785-3:2007, Information technology — Common Biometric Exchange Formats Framework — Part 3: Patron format specifications (2007)
9. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC).ISO/IEC 19790:2012, Information technology — Security techniques — Security requirements for cryptographic modules (2012).
10. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 24761:2009, Information technology — Security techniques — Authentication context for biometrics (2009)
11. International Organization for Standardization (ISO), International Electrotechnical Committee (IEC). ISO/IEC 24761:2009/Cor 1:2013 (2013)
12. National Institute of Standards and Technology (NIST). Federal Information Processing Standardization (FIPS) 140-2 (2001)
13. National Institute of Standards and Technology (NIST). NIST Special Publication (SP) 800-63-2 Electronic Authentication Guideline (2013)
14. Office of Management and Budget (OMB). E-Authentication Guidance for Federal Agencies, M-04-04 (2003)
15. Ratha, N. K., Connell, J. H., and Bolle R. M. A biometrics-based secure authentication system, Proc. of IEEE Workshop on Automatic Identification Advanced Technologies (AutoId 99), Summit, NJ, pp. 70-73 (1999)
16. Santesson, S., Nystrom, M., Polk, T.: Requests for Comments (RFC) 3739, Internet X.509 Public Key Infrastructure: Qualified Certificates Profile, The Internet Engineering Task Force (IETF) (2004)
17. Housley, R.: Requests for Comments (RFC) 3852, Cryptographic Message Syntax (CMS) , The Internet Engineering Task Force (IETF) (2004)
18. Hoffman, P., Schaad, J.: Requests for Comments (RFC) 5911, New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S/MIME, The Internet Engineering Task Force (IETF) (2010)