DOI: 10.19729/j.cnki.1673-5188.2018.04.005
http://kns.cnki.net/kcms/detail/34.1294.TN.20181112.1001.002.html, published online November 12, 2018

A New Direct Anonymous Attestation Scheme for Trusted NFV System

CHEN Liquan[1], ZHU Zheng[1], WANG Yansong[2], LU Hua[2], and CHEN Yang[1]
(1. School of Information Science and Engineering, Southeast University, Nanjing 210096, China;
2. Nanjing R&D Center, ZTE Corporation, Nanjing 210021, China)

Abstract

How to build a secure architecture for network function virtualization (NFV) is an important issue. Trusted computing has the ability to provide security for NFV and it is called trusted NFV system. In this paper, we propose a new NFV direct anonymous attestation (NFV-DAA) scheme based on trusted NFV architecture. It is based on the Elliptic curve cryptography and transfers the computation of variable D from the trusted platform module (TPM) to the issuer. With the mutual authentication mechanism that those existing DAA schemes do not have and an efficient batch proof and verification scheme, the performance of trusted NFV system is optimized. The proposed NFV-DAA scheme was proved to have a higher security level and higher efficiency than those existing DAA schemes. We have reduced the computation load in Join protocol from $3G_1$ to $2G_1$ exponential operation, while the time of NFV-DAA scheme's Sign protocol is reduced up to 49%.

Keywords

NFV; trusted computation; DAA; bilinear pairings

1 Introduction

Network function virtualization (NFV) is a new network architecture based on standard virtualization technology. It can realize the network entities such as servers, switches and storages on industrial standard hardware platforms, achieving various network functions by running different software on the virtualized platform. Network entities loaded onto the virtualized platform can achieve dynamic resource allocation, and network flexibility and scalability enhancement. Moreover, replacing the existing special hardware devices with industrial standardized servers could decrease the operators' network cost. Low cost and high flexibility are the great features of NFV technology [1].

Since NFV technology is being widely used in the foreseen future, the security issues in NFV system need resolving. In an NFV system, the virtualized network function (VNF) is the key model of achieving network functionality and it possibly becomes the first target to be attacked. Furthermore, compared to other modules in NFV, VNF needs to interact with outside environments (e.g. another VNF) frequently, which makes it to be another breakthrough for the malicious counterpart. Therefore, the VNF security is the most important part for the entire NFV system. To guarantee the security of interaction among different VNF modules, a two-way authentication protocol for mutual authentication is necessary. At the same time, a security channel needs to be established for this authentication protocol, which prevents the conversation between VNF parties from eavesdropping. Based on direct anonymous attestation (DAA), we propose a new NFV-DAA scheme that is applied to the authentication between VNF modules. The proposed scheme also provides VNF with identification and mutual authentication, and establishes secure communication channel between the VNF parties.

The DAA scheme was first developed by Brickell, Camenisch, and Chen [2] for remote authentication of a trusted computing platform while preserving the privacy of the platform. It has been adopted by trusted computing group (TCG) in the trusted platform module (TPM) specification version 1.2 [3]. DAA is a new group signature scheme without the capability to open signature but with a mechanism to detect rogue members. It draws on the techniques that have been developed for group signatures, identity escrow and credential systems. In the DAA scheme, a suitable signature scheme is employed to issue certificate on a membership public key generated by a TPM. This certificate can help one platform to be authenticated as a group

This work was supported by the national Natural Science Foundation of China (NSFC) under grant No. 61372103 and the ZTE Industry-Academia-Research Cooperation Funds.

member. A valid TPM proves to the verifier that it possesses a certificate. Each TPM has a secret key, which is used to sign a credential and detect rogue TPMs by the verifier. Many researchers have proposed different DAA schemes to meet the requirements in different applications and environments [4]–[6].

Generally, the DAA schemes have the characteristics of efficiency, anonymity, and privacy.

The Join protocol, by which the issuer receives the TPM's application for joining and sends back the credential to TPM, runs once a new platform with TPM begins to join this trusted system. When this platform receives the DAA credential from the issuer, it can use this credential to conduct the following sign/verify processes many times. Compared to the privacy certificate authority (CA) scheme, the issuer in DAA has no need to conduct in each of the following sign/verify processes [2]. Therefore, DAA is more efficient than the Privacy CA scheme that have a bottleneck because the Privacy CA server has to be included in every processing section.

Since the DAA scheme uses zero-knowledge proof theory to prove the trust of a new platform which possesses legitimate credential, it prevents any adversary from seeking the identity of the real communicating TPM. Meanwhile, many DAA schemes use the credential randomization technique to mask the real transmitted credential [7], [8]. It is difficult for the adversary to track the identity of the target TPM even when the verifier can collude with the credential issuer.

The trusted credential issuer has endorsement key (EK) lists to check the legitimation of the applying TPM, and the verifier employs the Camenisch-Lysyanskaya (C-L) signature scheme [9] and the discrete logarithms based proofs to prove the possession of a certificate, while the unforgeability, privacy and anonymity are guaranteed under the decisional Diffie-Hellman (DDH) assumption.

The DAA scheme in [2] is based on the strong RSA assumption and is named as RSA-DAA. Theory analysis results have shown that the protocols and algorithms in RSA-DAA are complicated and inefficient. In recent years, researchers have worked on how to create new DAA schemes with elliptic curves cryptography (ECC) and bilinear pairings [10], [11]. We call these DAA schemes as ECC-DAA for short. Generally speaking, ECC-DAA is more efficient in both computation and communication than RSA-DAA. The operation of TPM is much simpler and the key and signature length is shorter in ECC-DAA than that in RSA-DAA.

However, there are no existing DAA schemes proposed to meet the requirements of NFV system by now. According to the security requirements of mutual authentication between the signer and verifier, bundling rogue check of TPM and host in NFV system, an enhanced DAA scheme (hereinafter as NFV-DAA scheme) with mutual authentication which can meet all the above requirements is proposed. A remote anonymous authentication architecture for NFV system is constructed. The proposed NFV-DAA scheme has the following advantages with efficiency and security.

1) We put off J, K variables and those computations in the sign/verify stage in [10], and use a new variable $c_2 = H_2(f \parallel bsn)$ instead. In order to realize rogue list checking and user-controlled-linkability, the verifier can check the received c_2 with the RogueList to find out those rogue TPMs. Meanwhile, with the same bsn (base name) value, we can control the verifier to find out what messages are coming from the same TPM by getting out the same c_2 value. This scheme can reduce one scalar multiplication induced by the J, K pair computation.

2) Considering the low computing ability of the TPM, the computation of variable D is transferred from the TPM to issuer.

3) An efficient batch proof and verification scheme is used to reduce the computation of both the TPM and Host. In our NFV-DAA scheme, the TPM needs only to perform one exponentiation in the sign stage. However, this operation requires at least three exponentiations in the existing DAA schemes that provide the same functionality.

4) Elliptic curve is used in the Join, Sign and Verify protocols. It is shown in theoretical analysis that the protocols and algorithms used in RSA-DAA are complex and inefficient compared to those in ECC-DAA. Generally, TPM has lower computation load and higher communication efficiency in ECC-DAA, while the length of key and signature is also shorter.

5) The identity of TPM and Host is tied up and checked by the issuer and verifier. This technology prevents the attack of plugging a valid TPM into a malicious host. This security problem has not been considered in the existing other DAA schemes.

6) Traditional DAA schemes do not take mutual authentication into account. Despite that the issuer has a thorough mechanism to check the identity of TPM, TPM does not have any method to check the authenticity of the issuer and host. Therefore, the TPM and host would receive a forged certificate from a fake issuer, or the TPM be deceived by a fake host. In NFV-DAA scheme, a thorough mutual authentication is proposed, which ensures the legitimacy of the identity of all the protocol parties.

The rest of the paper is organized as following. Section 2 presents the NFV-DAA scheme for trusted NFV system. In Section 3, the security and performance analysis of the proposed scheme is presented. Finally, we conclude the paper in Section 4.

2 NFV-DAA Scheme for Trusted NFV System

2.1 Preliminary knowledge

1) Bilinear mapping

G_1, G_2 and G_T are cyclic groups with order of prime q,

$G_1=\langle P_1\rangle$, and $G_2=\langle P_2\rangle$, where P_1, P_2 is a generator of G_1, G_2 respectively. And calculating discrete logarithm on groups G_1, G_2, and G_T is difficult. Here, we also use the G_1, G_2 and G_T to represent the computation costs of the group G_1, G_2 and G_T.

If the mapping $\hat{t}:G_1\times G_2\rightarrow G_T$ satisfies the following conditions:

- $P_1\in G_1$, $P_2\in G_2$, $1\in G_T$, and $\hat{t}(P_1,P_2)\neq 1$
- $x\in G_1$, $y\in G_2$, then $\hat{t}(x,y)$ can be computed in polynomial time
- for $x\in G_1$, $y\in G_2$, and $a,b\in Z_P$, $\hat{t}(x,y)^{ab}=\hat{t}(x^a,y^b)$

Then $\hat{t}:G_1\times G_2\rightarrow G_T$ can be called as bilinear mapping.

2) The CDH problem

The computational Diffie-Hellman (CDH assumption) is the assumption that a certain computational problem within a cyclic group is hard. Consider a cyclic group $G_1=\langle P_1\rangle$. The CDH assumption states that, given aP_1, bP_1, and $a,b\in Z_q$ it is computationally difficult to get the value of abP_1.

Moreover, each party in the NFV-DAA scheme is presented as follows.

- Trusted Center (TC) and Issuer: They are the entity who issues the certificate. In this paper, we make no distinction between these two terms.
- TPM: It is the trusted platform module.
- Host: the host for the TPM. It is also the physical platform in trusted NFV system which providing resources for different VNF.
- Verifier: It verifies the signature. In this paper, the verifying operation is carried out by the counterpart VNF' s Host, therefore, the counterpart Host plays the role of Verifier in the NFV-DAA scheme. However, in the following discussion, we still put Host and Verifier as different logical entity.

The overall NFV-DAA scheme similarly includes Setup protocol which establishes system parameters, Join protocol which obtain certificate and the Sign/Verify protocol which do the Sign and authentication process. Specific description of each protocol is shown as follows.

2.1.1 Setup Protocol

Assuming that the bilinear pairs are $\hat{t}:G_1\times G_2\rightarrow G_T$ and $H_1:\{0,1\}^*\rightarrow Z_q$, we define the common parameter set $par_c=(G_1,G_2,G_T,\hat{t},P_1,P_2,q,H_1)$. Given that the public key and private key parameters of the issuer are ipk and isk respectively. Here, isk is $x,y\leftarrow Z_q$, ipk is (X,Y), while $X=xP_2\in G_2$, $Y=yP_2\in G_2$. In addition, the issuer will generate a pair of key (PK_I,SK_I) for mutual authentication. Finally, the issuer will provide a unique value H_2 to generate secret value f. Then we get the issuer parameter set $par_I=(ipk,K_I,PK_I)$.

Suppose that the parameter par_R of TPM is H_2, $H_2:\{0,1\}^*\leftarrow Z_q$. The public key of TPM is par_T, and the public and private EK key is (PK_T,SK_T), $par_T=PK_T$. The public and private key pair of the host is (PK_H,SK_H), $par_H=PK_H$. The parameters of sign/verify are par_s: $H_3:\{0,1\}^*\rightarrow Z_q$, $H_4:\{0,1\}^*\rightarrow Z_q$. After the protocol setup, the system public parameter set par is defined as $(par_C,par_I,par_T,par_H,par_S)$.

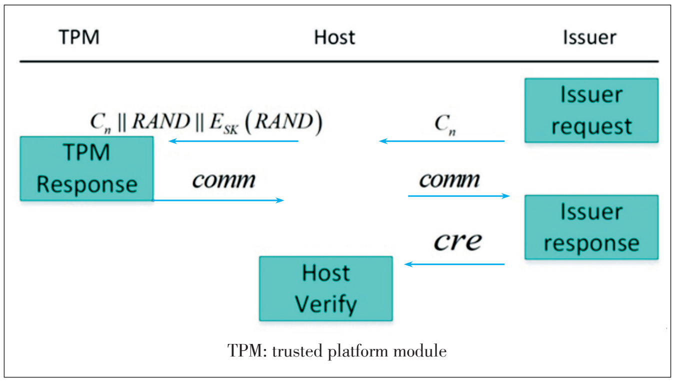
2.1.2 Join Protocol

The Join protocol is realized based on the request/response interaction between the issuer and TPM/host. We can divide the Join protocol into four parts in chronological order: the issuer request, TPM response, issuer response, and host verification. The overall process of the Join protocol is shown in **Fig. 1**.

The operation process of issuer request is shown in **Fig. 2**.

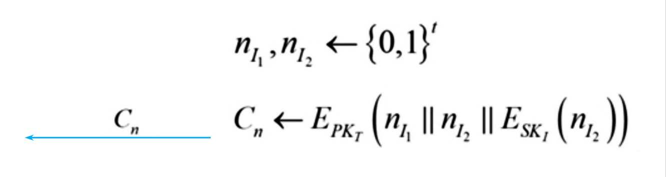
The issuer needs to confirm firstly that it is a legitimate trusted platform who has issued the DAA certificate, while the TPM also needs to check the legitimacy of the issuer. That is to say, an authentication channel should be established between the issuer and TPM in advance. The establishment is completed with the random numbers n_{I_1} and n_{I_2} chosen by the issuer, while n_{I_2} is encrypted with SK_I and then $n_{I_1}\|n_{I_2}\|E_{SK_I}(n_{I_2})$ is encrypted with PK_T to the host. Similar to the issuer, the host also generates a random number $RAND$, encrypts $RAND$ with pre - shared private key SK_H, and sends $C_n\|RAND\|E_{SK_H}(RAND)$ to TPM.

The TPM decrypts $E_{SK_H}(RAND)$ with the pre-shared public key PK_H to get $RAND'$. The result of comparison between $RAND'$ and $RAND$ indicates whether the host is legitimate. Only when $RAND'$ is equal to $RAND$ will the TPM continue the protocol. Next, if the TPM can successfully decrypt C_n with SK_T, obtain the value of n'_{I_1} and return the hash value of



TPM: trusted platform module

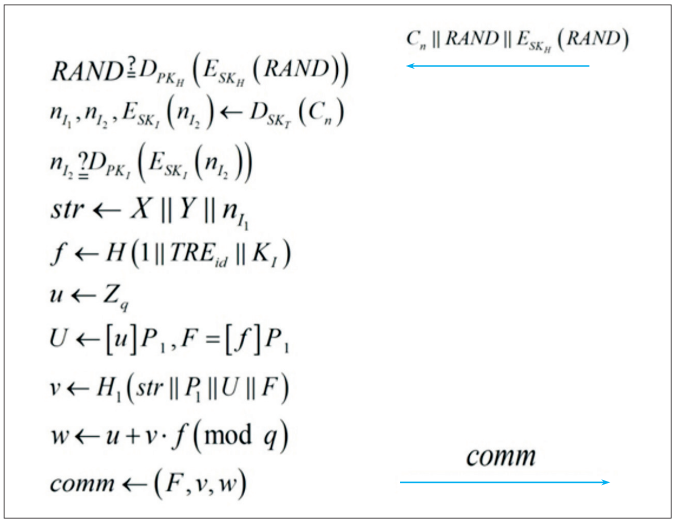
▲Figure 1. Overall process of the Join protocol.


$n_{I_1}, n_{I_2} \leftarrow \{0,1\}^t$
C_n
$C_n \leftarrow E_{PK_T}\left(n_{I_1} \| n_{I_2} \| E_{SK_I}\left(n_{I_2}\right)\right)$

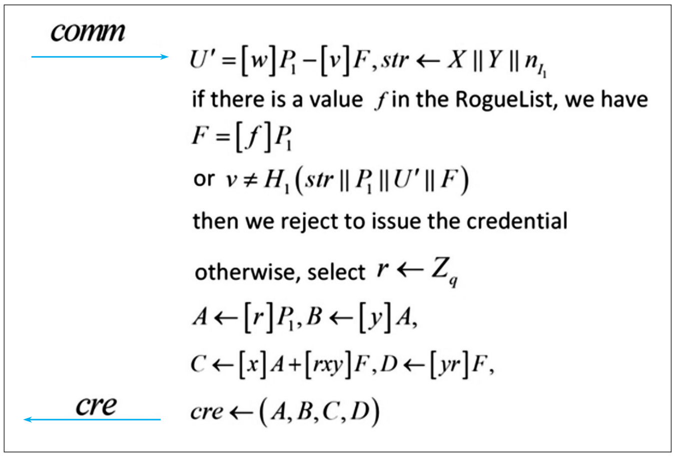
▲Figure 2. Flowchart of the issuer request.

n'_{I_1} to the issuer, it indicates that the TPM owns its legitimate EK private key. Besides, the TPM decrypts $E_{SK_I}(n_{I_2})$ with the legitimate EK public key of the issuer and compares the decrypted result n'_{I_2} with n_{I_2}. If they are equal, it indicates the legitimacy of the issuer. In this way, mutual authentication between the issuer and TPM is completed. **Fig. 3** shows the operating process of TPM response.

The TPM generates secret value f with K_I and TRE_{id} (a stable security parameter stored in TPM) and $f=H(1\|TRE_{id}\|K_I)$. It also generates the comm value from f, and delivers it to the certificate issuer. **Fig. 4** shows the process of the issuer response.


$C_n \| RAND \| E_{SK_H}(RAND)$
$RAND \overset{?}{=} D_{PK_H}(E_{SK_H}(RAND))$
$n_{I_1}, n_{I_2}, E_{SK_I}(n_{I_2}) \leftarrow D_{SK_T}(C_n)$
$n_{I_2} \overset{?}{=} D_{PK_I}(E_{SK_I}(n_{I_2}))$
$str \leftarrow X \| Y \| n_{I_1}$
$f \leftarrow H(1 \| TRE_{id} \| K_I)$
$u \leftarrow Z_q$
$U \leftarrow [u]P_1, F=[f]P_1$
$v \leftarrow H_1(str \| P_1 \| U \| F)$
$w \leftarrow u + v \cdot f (\bmod q)$
$comm \leftarrow (F, v, w)$
comm

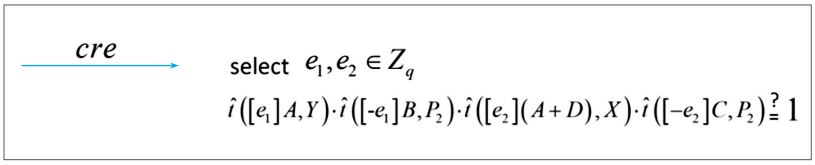
▲Figure 3. Response flowchart of the trusted platform module.


comm
$U' = [w]P_1 - [v]F, str \leftarrow X \| Y \| n_{I_1}$
if there is a value f in the RogueList, we have
$F=[f]P_1$
or $v \neq H_1(str \| P_1 \| U' \| F)$
then we reject to issue the credential
otherwise, select $r \leftarrow Z_q$
$A \leftarrow [r]P_1, B \leftarrow [y]A,$
$C \leftarrow [x]A + [rxy]F, D \leftarrow [yr]F,$
cre
$cre \leftarrow (A, B, C, D)$

▲Figure 4. Flowchart of the issuer response.

The issuer will authenticate the *comm* value, that is to say, it will judge the zero-knowledge proof process the TPM performed to discrete logarithm f, and check whether the TPM owns the legitimate f or not. If the authentication is successful, the issuer will generate DAA certificate with F in comm. It is important to note that, the computation of D in certificate *cre* uses $D=[yr]F$, which uses F provided by TPM rather than f to compute $D=[f]B$ [12]. This is mainly due to the fact that F itself is generated from value f. For the generation of DAA certification (A,B,C,D), according to the B-bLRSW principle of blind-bilinear assumption, the computation amount of TPM Join in NFV-DAA is reduced from $3G_1$ to $2G_1$. This computation amount is the lowest among the existing DAA schemes which are based on the LRSW or DDH difficulty assumption.

Furthermore, the process of host authentication is shown in **Fig. 5**. After receiving the certification *cre*, the host verifies correctness of *cre*. Based on the batch authentication technology, the host finds out whether the certificate is correct or not by using a P^4 computation. The P^4 computation will cost less than four independent bilinear pair computations (4*P*) [13]. The platform here does not have to perform very strict authentication of certificate and simple authentication is enough. The reason is that it does not affect the security of the entire DAA even if we cannot completely guarantee the dependability of certificate at this time. In subsequent Sign/Verify, there are other operations to verify the certificate.


cre
select $e_1, e_2 \in Z_q$
$\hat{\imath}([e_1]A, Y) \cdot \hat{\imath}([-e_1]B, P_2) \cdot \hat{\imath}([e_2](A+D), X) \cdot \hat{\imath}([-e_2]C, P_2) \overset{?}{=} 1$

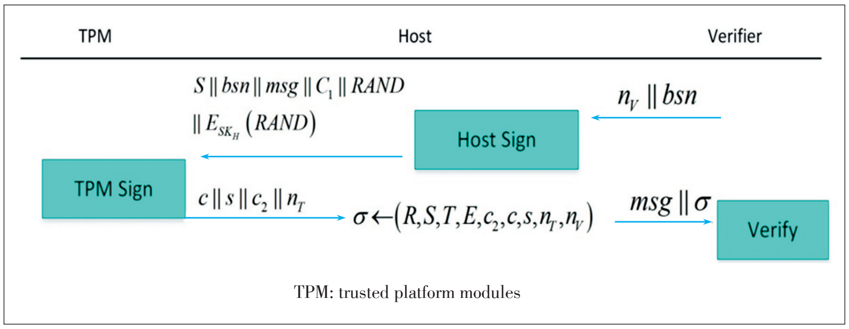
▲Figure 5. Flowchart of host verify.

2.1.3 Sign/Verify Protocol

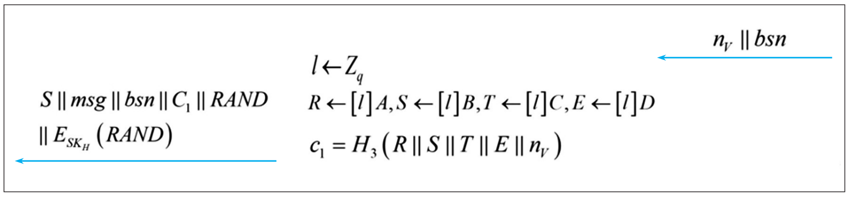
The Sign/Verify protocol refers to the process in which the TPM, together with the host, performs the knowledge sign of message *msg* and generates DAA signature σ, and then sends σ to the verifier. The operations in chronological order in the Sign/Verify protocol can be divided into three parts: host sign, TPM sign, and verify. The total framework of the Sign/Verify protocol is shown in **Fig. 6**.

The operation of host sign is shown in **Fig. 7**. The host performs blind computation of DAA certification value (A,B,C,D) after receiving n_V and base name $b$sn. Then the host generates blind certificate (R,S,T,E). Meanwhile, the host generates a random number *RAND* and encrypts it with the pre-shared private key SK_H, which is similar to the procedure in the Join protocol. Next, the host sends $S\|bsn\|msg\|c_1\|RAND\|E_{SK_H}(RAND)$ to TPM.

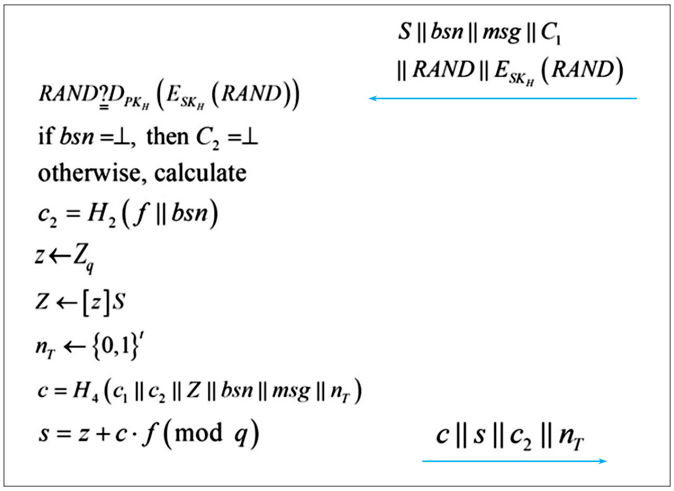
The operation of TPM sign is shown in **Fig. 8**. After receiving from the host, the TPM verifies the legitimacy of the verifier and host using the same method as the Join protocol. The TPM computes $RAND'$ in the way of decrypting the $E_{SK_H}(RAND)$ with the pre-shared key PK_H, and compares


TPM
Host
Verifier
$S \| bsn \| msg \| C_1 \| RAND$
$\| E_{SK_H}(RAND)$
Host Sign
$n_V \| bsn$
TPM Sign
$c \| s \| c_2 \| n_T$
$\sigma \leftarrow (R,S,T,E,c_2,c,s,n_T,n_V)$
$msg \| \sigma$
Verify
TPM: trusted platform modules

▲Figure 6. Overall framework of the Sign/Verify protocol.


$n_V \| bsn$
$l \leftarrow Z_q$
$S \| msg \| bsn \| C_1 \| RAND$
$\| E_{SK_H}(RAND)$
$R \leftarrow [l]A, S \leftarrow [l]B, T \leftarrow [l]C, E \leftarrow [l]D$
$c_1 = H_3(R \| S \| T \| E \| n_V)$

▲Figure 7. Flowchart of the host sign operation.


$S \| bsn \| msg \| C_1$
$\| RAND \| E_{SK_H}(RAND)$
$RAND \overset{?}{=} D_{PK_H}(E_{SK_H}(RAND))$
if $bsn = \perp$, then $C_2 = \perp$
otherwise, calculate
$c_2 = H_2(f \| bsn)$
$z \leftarrow Z_q$
$Z \leftarrow [z]S$
$n_T \leftarrow \{0,1\}^t$
$c = H_4(c_1 \| c_2 \| Z \| bsn \| msg \| n_T)$
$s = z + c \cdot f \pmod q$
$c \| s \| c_2 \| n_T$

▲Figure 8. The sign operation of the trusted platform module.

$RAND'$ with $RAND$. Only when the values are equal is the host proved to be legitimate.

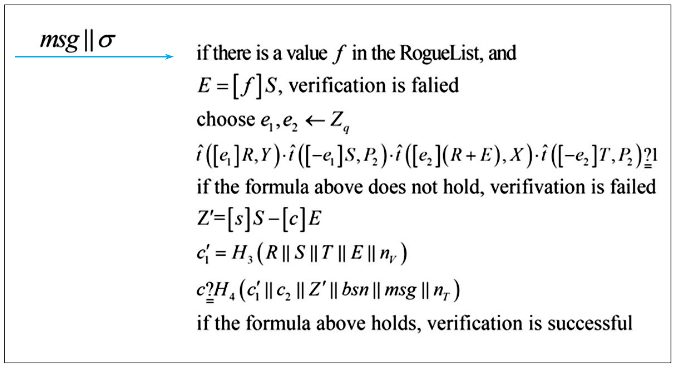
The TPM continues to finish the rest computation of the signature value after checking the legitimacy of the host. It generates independent value c_2 for relevance detection as $c_2 = H_2(f \| bsn)$. Then it considers the c_2 as the public signature member value, and performs the zero-knowledge proof of possessing a legitimate DAA certification. It also generates c and s values, while c_2 is add into the Hash computation of c. The TPM sends c, s and the random number n_T all together to the host, the final signature σ generated by the host is constructed as $(R,S,T,E,c,s,n_2,n_T,n_V)$. Then, the operation of Verify is shown in **Fig. 9**.

The verifier performs the verify operation after receiving the signature σ. It firstly substitutes the blind certificate value $E = [f]S$ for the counterfeit f on RogueList to check if the f value used in the signature has been disclosed, then verifies whether the blind signature value (R,S,T,E) is correct or not, and judges whether the zero-knowledge proof of the legitimate DAA certification in signature is correct. If all these are correct, it indicates that the signer owns a legitimate secret value f and the legitimate DAA certificate based on the same f. If the verifier has been provided with a specific bsn in advance, the relevance detection of signature is also needed. Relevance detection can be performed by using the signature member value c_2 generated from the secret value f and the base name bsn of the verifier. The entire verify process is completed only if all these verification steps are completed.

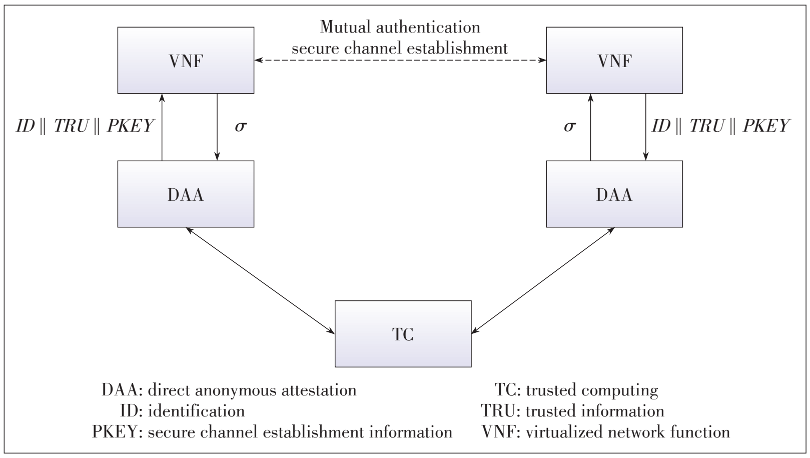
2.2 VNF Mutual Authentication and Secure Channel Establishment

In trusted NFV architecture, the VNF modules are able to mutually authenticate in a security and effective way as well as to establish secure a communication channel by the support of the NFV-DAA scheme. Mutual authentication refers to two parties authenticating each other. Under trusted NFV architecture, any two VNF should authenticating each other before communication in order to verify the identity and establish a security channel, i.e. exchanging the session key. The mutual authentication and security channel establishment is shown in **Fig. 10**.

As stated in Section 2.1, DAA's Sign protocol can generate signatures for any *msg* (message) accordingly. Therefore, during the mutual authentication, VNF sends 3 parameters (identification ID, trusted information TRU and secure channel establishment information PKEY) as the msg to NFV-DAA for the signature. The generated signature will be $\sigma_{ID\|TRU\|PKEY} = Sign(ID\|TRU\|PKEY)$. It will be sent to the counterpart VNF module in the other side. Then the recipient side will send the received signature to its own NFV-DAA scheme


$msg \| \sigma$
if there is a value f in the RogueList, and
$E = [f]S$, verification is falied
choose $e_1, e_2 \leftarrow Z_q$
$\hat{t}([e_1]R, Y) \cdot \hat{t}([-e_1]S, P_2) \cdot \hat{t}([e_2](R+E), X) \cdot \hat{t}([-e_2]T, P_2) \overset{?}{=} 1$
if the formula above does not hold, verifivation is failed
$Z' = [s]S - [c]E$
$c_1' = H_3(R \| S \| T \| E \| n_V)$
$c \overset{?}{=} H_4(c_1' \| c_2 \| Z' \| bsn \| msg \| n_T)$
if the formula above holds, verification is successful

▲Figure 9. Flowchart of the verify operation.


VNF
Mutual authentication
secure channel establishment
VNF
ID ‖ TRU ‖ PKEY
σ
σ
ID ‖ TRU ‖ PKEY
DAA
DAA
TC
DAA: direct anonymous attestation
ID: identification
PKEY: secure channel establishment information
TC: trusted computing
TRU: trusted information
VNF: virtualized network function

▲Figure 10. NFV-DAA scheme application in trusted NFV system.

to verify its legality. After the verification, TRU is verified by the credibility verification service provided by NFV infrastructure. It finishes the entire process by establishing a secure communication channel under the exchange of keys through PKEY. ID and TRU are all security parameters coming from the two parties of mutual authentication, i.e. two VNFs.

3 Security and Performance Analysis of NFV-DAA Scheme

3.1 Security

The correctness and security of overall NFV-DAA protocol are analyzed in this section.

The CDH problem can be solved if the NFV-DAA scheme can be a breakthrough when the security parameters saved in TPM are not leaked.

At the time of verifying, (1) guarantees the correctness of NFV-DAA process.

$$\hat{t}\left(\left[e_1\right]R,Y\right)\cdot\hat{t}\left(\left[-e_1\right]S,P_2\right)\cdot\hat{t}\left(\left[e_2\right]\left(R+E\right),X\right)\cdot\hat{t}\left(\left[-e_2\right]T,P_2\right)=1. \qquad (1)$$

To prove (1) true, $\hat{t}\left(R,Y\right)=\hat{t}\left(R,yP_2\right)=\hat{t}\left(yR,P_2\right)=\hat{t}\left(S,P_2\right)$ and $\hat{t}\left(R+fS,X\right)=\hat{t}\left(R+fS,xP_2\right)=\hat{t}\left(x\left(R+fS\right),P_2\right)=\hat{t}\left(T,P_2\right)$ are needed to be true. It indicates that the DAA certificate of the signature is generated in a correct way. The security of NFV-DAA is mainly reflected as follows: as long as the secret value of TPM f and DAA certification have not been disclosed, an attacker is unable to carry out any attacks, which ensure the security of NFV-DAA scheme.

The proof is as follows: in the context where no the TPM secret value or DAA certification is disclosed, without the verifier, an attacker should provide $\left(R,S,T,E\right)$ alone to make (1) true, which expects $S=\left[y\right]R$ and $T=\left[x\right]\left(R+E\right)$ to be true. Assume that attacker A selects $R=\left[\alpha\right]P_1$, $S=\left[\beta\right]P_1$, $T=\left[\gamma\right]P_1$ and $E=\left[\delta\right]P_1$, and the public key of the issuer is known as $X=\left[x\right]P_2$, $Y=\left[y\right]P_1$ and $S=\left[y\cdot\alpha\right]P_1$ which is needed to make $S=\left[y\right]R$ true. It means that given $\left[\alpha\right]P_1$ and $\left[y\right]P_1$, the attacker must be able to calculate $\left[y\cdot\alpha\right]P_1$, so as to solve the CDH problem in group G_1. However, it is obviously impossible for the attacker to solve the CDH problem. The non-symmetric bilinear pair is generally considered to be difficult.

In addition, different from the existing DAA schemes, NFV-DAA has the feature of mutual authentication. We assume that prior to any system setup, each issuer has its private endorsement private key SK_I and each TPM has the corresponding public key PK_I. The issuer generates a random number n_{I_2} and encrypts it with SK_I. The TPM admits the legitimacy of the issuer if the decrypting result is equal to the received n_{I_2}. In other words, the issuer sends its signature to the TPM for checking. Considering the issuer has checked the TPM by the endorsement key pair SK_T/PK_T in the Join protocol, mutual authentication is realized.

In order to prevent a corrupted host from taking advantages of an honest TPM to sign on an illegal message, it is necessary to bind the TPM and host when manufacturing the devices. A pair of pre-shared public/private key PK_H/SK_H is embedded into the TPM and host respectively. For the Join protocol and Sign protocol, the host needs to generate a random number $RAND$ and sends $RAND\|E_{SK_H}\left(RAND\right)$ to the TPM. The TPM checks the consistency of the decrypting result $RAND'$ and $RAND$ to verify the legitimacy of the host. Here, we assume that the embedded key cannot be extracted from the TPM and host due to hardware protection.

The existing DAA schemes prevent the change of *bsn* by the host to make signatures linkable, but they cannot prevent that a malicious message is delivered to TPM by the host to generate a legal signature. Since a valid TPM may be used in an illegal way in existing DAA schemes, verifying the identity of the host is necessary. In the NFV-DAA scheme, mutual authentication ensures the legitimacy of the host, hence the host will not deliver illegal messages to the TPM or disclose the identity of the TPM directly.

Based on all of the above results, it is easy to find out that the proposed NFV-DAA scheme is secure enough on the premise that the TPM is secure and credible. Compared with those DAA schemes based on LRSW and DDH assumptions, NFV-DAA has no security weakness and can improve overall protocol efficiency. It has the highest running efficiency among all the existing DAA schemes based on LRSW and DDH assumption at present. Both the Join and Sign protocols of the NFV-DAA scheme have been improved, the computation amount of TPM Join is reduced to $2G_1$, and that of TPM Sign is as low as $1G_1$. The detailed analysis of NFV-DAA efficiency will be presented in next section. The benefits of secured security, low cost, high efficiency, and being easy to implement help the NFV-DAA scheme meet the dual requirements for security and economic benefits of the trusted NFV system.

3.2 Performance

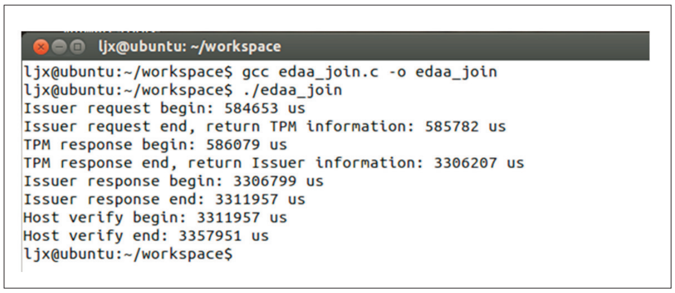
In this paper, we compare the performance of the scheme in [12] with that of the NFV-DAA scheme. In other words, we compare the time overhead of the Join protocol and Sign/verify protocol in both schemes. Here, the alternative simulation is used to make the experiments. It means that without considering the communication time between the TPM and the remote issuer, and between the TPM and the remote verifier, we just focus on the time overhead on the protocol operations of each protocol entity.

Based on the above consideration, we set the host, issuer and verifier to the same PC host. The software simulation scheme [14] is used to internally install the TPM to the same PC host, which communicates with the TPM via the hardware interface. Statistics on the time overhead of each protocol are provided in the stand-stone simulation environment. Besides, the protocol parameters are chosen the same as those in [15].

3.2.1 Join

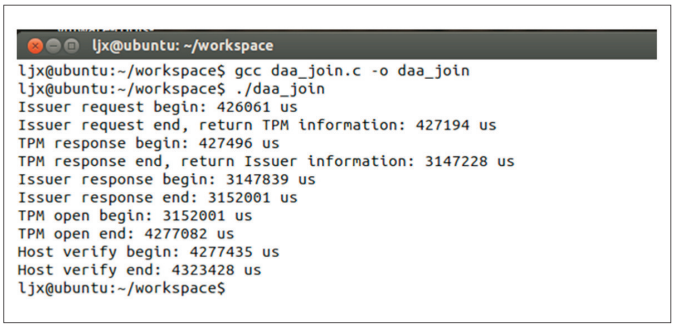
In accordance with the NFV-DAA Join protocol process, with the cryptographic algorithm library package in OpenSSL software, we used C++ to write the client program edaa_join.c in ubuntu 9.10. Main parts of the edaa_join.c include the issuer requirement, issuer response and host verification. The TPM response is fulfilled by the TPM software. The program computed the time overhead of each protocol in microseconds and ran by calling the timing function gettimeofday( ) of the system. The experimental results after running edaa_join.c are shown in **Fig. 11**.

Furthermore, we did experiments to verify the scheme [12] and the experimental results are shown in **Fig. 12**.



```
ljx@ubuntu: ~/workspace
ljx@ubuntu:~/workspace$ gcc edaa_join.c -o edaa_join
ljx@ubuntu:~/workspace$ ./edaa_join
Issuer request begin: 584653 us
Issuer request end, return TPM information: 585782 us
TPM response begin: 586079 us
TPM response end, return Issuer information: 3306207 us
Issuer response begin: 3306799 us
Issuer response end: 3311957 us
Host verify begin: 3311957 us
Host verify end: 3357951 us
ljx@ubuntu:~/workspace$
```

▲Figure 11. Time overhead of the Join protocol in NFV-DAA scheme.



```
ljx@ubuntu: ~/workspace
ljx@ubuntu:~/workspace$ gcc daa_join.c -o daa_join
ljx@ubuntu:~/workspace$ ./daa_join
Issuer request begin: 426061 us
Issuer request end, return TPM information: 427194 us
TPM response begin: 427496 us
TPM response end, return Issuer information: 3147228 us
Issuer response begin: 3147839 us
Issuer response end: 3152001 us
TPM open begin: 3152001 us
TPM open end: 4277082 us
Host verify begin: 4277435 us
Host verify end: 4323428 us
ljx@ubuntu:~/workspace$
```

▲Figure 12. Time overhead of the Join protocol in the scheme in [12].

According to the results in Figs. 11 and 12, it is easy to compute the time overhead in the two schemes (**Table 1**).

In Table 1, the main differences between NFV-DAA and the scheme in [12] lie in two aspects. On one hand, the NFV-DAA scheme does not have the TPM Open process. On the other hand, the time overhead of the issuer response in the NFV-DAA scheme is 996 us larger than that in the scheme in [12]. The reason is that in NFV-DAA, the TPM does not have TPM Open operation, while the issuer makes the operation on behalf of the TPM. In this way, the issuer fulfills a group G_1 exponential operation originally conducted by the TPM. With the usage of Batch technology, the issuer response costs just 996 us more than the scheme in [12], which is much smaller than the TPM Open cost of 1,125,081 us. It is also found that the Join's total time of NFV-DAA is 2,770,411 us. Compared with the time overhead 3,896,101us in the scheme [12], the performance of NFV-DAA Join is improved up to about 29%.

3.2.2 Sign/Verify

Two more software programs edaa_sign.c and edaa_verify.c are written to test the performance of Sign/Verify protocol. **Table 2** provides the final statistics of comparing Sign/Verify time overhead in NFV-DAA and the scheme in [12].

From Table 2, the time of the scheme in [12] is different for the signature with correlation ($bsn = \perp$) and without correlation

▼Table 1. Comparison of NFV-DAA and the scheme in [12] on the time overhead of Join

Scheme	Join protocol					
	Issuer request (us)	TPM response (us)	Issuer response (us)	TPM open (us)	Host verifies (us)	Total time (us)
NFV-DAA	1129	2,720,128	5158	-	45,994	2,770,411
The scheme in [12]	1133	2,719,732	4162	1,125,081	45,993	3,896,101

DAA: direct anonymous attestation NFV: network function virtualization TPM: trusted platform module

▼Table 2. Comparison of NFV-DAA and the scheme in [12] on the time of Sign/Verify

Scheme		Sign/Verify procedure			
		Host Sign (us)	TPM Sign (us)	Verify (us)	Total time (us)
NFV-DAA		4088	1,149,213	47,301	1,200,602
The scheme in [12]	$bsn = \perp$	4146	1,148,901	47,232	1,200,279
	$bsn \neq \perp$	6312	2,292,876	48,310	2,347,498

DAA: direct anonymous attestation NFV: network function virtualization TPM: trusted platform module

($bsn \neq \perp$). The improvement of the NFV-DAA scheme is that no matter whether the signature has correlation, the computation time of each entity is the same as that of the scheme without correlation in the scheme in [12]. It is found that the NFV-DAA scheme costs 1,200,602 us that is similar to the time overhead in the scheme in [12] when $bsn = \perp$. However, compared to the time in the scheme in [12] when $bsn \neq \perp$, it is obvious that the performance of the NFV-DAA scheme's Sign/Verify is improved up to 49%. The reason is that the NFV-DAA scheme uses $c_2 = H_2\left(f \| bsn\right)$ instead of J and K in [12] for signature correlation detection so that TPM Sign gets less group G_1 exponential operation than in [12].

4 Conclusions

A secure and high efficient NFV-DAA scheme is proposed in this paper. The scheme is designed based on the architecture of trusted NFV system, taking advantages of existing security TPM in the architecture. Therefore, the scheme can be integrated into the architecture seamlessly. With a mutual authentication mechanism that the existing DAA schemes do not have and an efficient batch proof and verification scheme, the trusted NFV system has optimized performance. From the experiment results, we can find out that the proposed NFV-DAA scheme has higher security level and efficiency than those existing DAA schemes. The computation load in Join protocol is reduced from $3G_1$ to $2G_1$ exponential operation, while the time of NFV-DAA scheme's Sign/Verify protocol is improved up to 49%.

References

[1] ETSI. (2016 Jul.). *Network functions virtualization (NFV)* [Online]. Available: http://portal.etsi.org/NFV

[2] E. Brickel, J. Camenisch, and L. Q. Chen, "Direct anonymous attestation," in *Proc. 11th ACM Conference on Computer and Communications Security*, Washington DC, USA, 2004, pp. 132–145. doi:10.1145/1030083.1030103.

[3] *Information Technology Security Techniques-Trusted Platform Module*, ISO/IEC 11889, 2009.

[4] E. Brickell and J. Li, "A pairing-based DAA scheme further reducing TPM resource," in *International Conference on Trust and Trustworthy Computing*, Heidelberg, Germany, 2010, pp. 902–915. doi: 10.1007/978-3-642-13869-0_12.

[5] X. M. Wang, H. Y. Heyou, and R. H. Zhang, "One kind of cross-domain DAA scheme from bilinear mapping," in *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Beijing, China, 2014, pp. 237–243. doi:10.1109/TrustCom.2014.62.

[6] B. Zhu, H. H. Cui, L. Chen, and C. Tang, "Improvement of the DAA protocol based on TPM," in *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Chengdu, China, 2010, pp. 401–404. doi: 10.1109/ICCSIT.2010.5564832.

[7] L. Q. Chen, P. Morrissey, and N. P. Smart, "Pairings in trusted computing," in *International Conference on Pairing-Based Cryptography*, Heidelberg, Germany, 2008, pp. 1–17. doi: 10.1007/978-3-540-85538-5_1.

[8] L. Q. Chen, P. Morrissey, and N. P. Smart. (2016 Jul.). Fixing the pairing based protocols. Cryptology ePrint Archive Report 2009/198 [Online]. Available: http://eprint.iacr.org/2009/198

[9] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology-CRYPTO*, Berlin, Germany, 2004, pp. 56–72. doi: 10.1007/978-3-540-28628-8_4.

[10] L. Tan and M. T. Zhou, "A new process and framework for direct anonymous attestation based on asymmetric bilinear maps," *Chinese Journal of Electronics*, vol. 22, no. 4, pp. 695–701, 2013.

[11] L. Yang, J. F. Ma, and W. Wang, "Multi-domain direct anonymous attestation scheme from pairings," in *International Conference on Network and System Security*, Xi'an, China, 2014, pp. 566–573. doi: 10.1007/978-3-319-11698-3_47.

[12] L. Q. Chen, "A DAA scheme using batch proof and verification," in *International Conference on Trust and Trustworthy Computing*, Heidelberg, Germany, 2010, pp. 166–180. doi: 10.1007/978-3-642-13869-0_11.

[13] R. Granger, N. P. Smart. (2016 Jul.). On computing products of pairings. Cryptology ePrint Archive Report 2006/172 [Online]. Available: http://eprint.iacr.org/2006/172

[14] M. Strasser and H. Stamer, "A software-based trusted platform module emulator," in *International Conference on Trusted Computing*, Heidelberg, Germany, 2008, pp. 33–47. doi: 10.1007/978-3-540-68979-9_3.

[15] L. Q. Chen, D. Page, and N. P. Smart, "On the design and implementation of an efficient DAA scheme," in *International Conference on Smart Card Research and Advanced Applications*, Heidelberg, Germany, 2008, pp. 223–238. doi: 10.1007/978-3-642-12510-2_16.

Manuscript received: 2017-12-05

Biographies

CHEN Liquan (Lqchen@seu.edu.cn) received his B.Sc. degree in electronic engineering from Nanjing University, China in 1998, M.Sc. degree in radio engineering from the Purple Mountain Observatory, Chinese Academic of Sciences in 2001, and Ph.D. degree in signal processing from Southeast University, China in 2005. He is presently engaged in information processing and communication network research as a professor at Southeast University, China.

ZHU Zheng (zhuzheng@seu.edu.cn) is currently a master student at Southeast University, China. His research interests include information security and computer networks.

WANG Yansong (wang.yansong@zte.com.cn) is a principle product manager of ZTE Corporation. His research interests include communications technologies and computer networks.

LU Hua (lu.hua@zte.com.cn) is an engineer of ZTE Corporation. His research interests include 5G technologies, computer networks.

CHEN Yang (chenyang90@seu.edu.cn) is currently a master student at Southeast University, China. His research interests include information security and digital communications.