GM

CRYPTOGRAPHIC INDUSTRY STANDARD

OF THE PEOPLE'S REPUBLIC OF CHINA

ICS 35.040

CCS L 80

GM/T 0079-2020

Direct anonymous attestation specification for trusted computing platform

可信计算平台直接匿名证明规范

Issued on: December 28, 2020 **Implemented on: July 01, 2021**

Issued by: National Cryptography Administration

Table of Contents

Direct anonymous attestation specification for trusted computing platform

1 Scope

This document specifies the functions, interfaces, data structure of the direct anonymous attestation protocol of the trusted computing platform.

This document is applicable to the development of the direct anonymous certification protocol applications, anonymous certification services, anonymous certification systems of the trusted computing platform.

2 Normative references

The provisions in following documents become the provisions of this Standard through reference in this Standard. For the dated references, the subsequent amendments (excluding corrections) or revisions do not apply to this Standard; however, parties who reach an agreement based on this Standard are encouraged to study if the latest versions of these documents are applicable. For undated references, the latest edition of the referenced document applies.

GB/T 32918-2016 (all parts) Information security techniques - Elliptic curve public-key cryptography

GM/T 0012 Trusted computing - Interface specification of trusted cryptography module

GM/Z 4001 Cryptographic terms

3 Terms and definitions

The terms as defined in GM/Z 4001, as well as the following terms, apply to this document.

3.1

Trusted cryptography module; TCM

A basic hardware module, which construct the trusted computing platform. It provides cryptographic computing functions for the trusted computing platform; has protected storage space.

for the trusted cryptographic module.

3.9

Verifier

In direct anonymous attestation, the participant who verifies the identity of the remote trusted cryptographic module.

4 Symbols and abbreviations

4.1 Symbols

The cryptographic symbols which are defined in GB/T 32918-2016 (all parts), as well as the following cryptographic symbols, apply to this document.

0: Integer 0, bit 0, or finite field addition identity element.

1: Integer 1, bit 1, or finite field multiplication identity element.

α, b: Elements in F_q, which define the elliptic curve E on F_q.

e: G_1 X G2→G_T: Bilinear mapping, which maps elements in (G_1, G_2) to elements in G_T.

exp(l, m): The m^{th} power of the finite field element l, which is also recorded as l^m.

E: An elliptic curve, which is defined by α and b on a finite field.

E(F_q): The set of all points in E whose coordinates belong to F_q (including the point at infinity O).

F_q: The q-order finite prime field.

F_{qk}: The q^k-order finite field, an extension of q-order finite field.

G_n: A base point of the elliptic curve, whose order is a prime number; the subscript n is an integer, which is used to distinguish different base points.

G_T: A base point of a finite field, the order of which is a prime number.

l + m: Field addition operation result of finite field elements l and m.

l x m: The result of the field multiplication of the finite field element l and m, which is also recorded as lm, if it does not cause ambiguity.

P: P = (x_p, y_p) is a point on the elliptic curve excluding the zero point O, the application of TCM anonymous credentials AND the attestation of TCM anonymous identity. The prover platform drives TCM, to request anonymous identity credentials, from the credential issuer, by executing the TCM_ECDAA_Join command and related host calculations. The prover platform executes the TCM_ECDAA_Sign command and related host calculations, to prove the TCM's digital identity anonymously, to the verifier platform.

The verifier platform mainly verifies the attestation data, which is provided by the prover platform, to certify the TCM identity of the prover platform; ensure that the prover platform does indeed use the security chip TCM as the identity of the platform. While verifying the anonymous identity of the TCM, it is necessary to request the issuer to verify whether the digital identity of the TCM has been revoked.

In the ECDAA system, the anonymous identity private key f of the TCM security chip is only allowed to be stored inside the TCM chip, AND is not allowed to be exported. There can be multiple anonymous identity private keys and anonymous certification credentials for TCM; however, it is recommended to use only one anonymous identity private key and credentials. The TCM anonymous certification process (including certification and verification) can only be performed by the TCM owner; meanwhile only the TCM owner can clear the insecure anonymous private key. TCM anonymous identity credentials can be stored in a host platform, which is outside the chip, OR in other storage devices.

The core computing functions of the prover platform are completed by the TCM_ECDAA_Join and TCM_ECDAA_Sign commands of TCM. Only higher authority can execute these ECDAA commands. The ECDAA command is a command, that consumes TCM and host computing resources very much. It requires a large amount of internal resources of the TCM chip, to complete a series of computing operations. When the TCM security chip executes the ECDAA command, it is necessary to prohibit the execution of other TCM command operations.

6.2.2 Basic process

The main communication process between the various participants of the ECDAA system includes the following steps:

a) System initialization: Set the public parameters of the ECDAA system; generate a public-private key pair, which is used by the issuer to issue anonymous certificates.

b) Certificate issuance: The prover applies for and obtains an anonymous certificate from the issuer.

H_3, H_4, T_1, T_2, T_3, T_w), the signature of issuer on the public parameter cre = $sign_{k_n^{-1}}$ (issuerSettings), the confidential information of the issuer isk = r.

c) Algorithm flow:

1) Prove the system parameters (q, α, b, g_1, g_2, p) directly and anonymously. Among them, α, b and F_q jointly define the elliptic curve $E(F_q)$; g_1, g_2 are the base points of $E(F_q)$ respectively; their order is a prime number p.

2) Select the bilinear mapping operation e: $G_1 \times G_2 \rightarrow G_T$. Among them, G_1, G_2 are the cyclic group, with g_1, g_2 as generators; the order is prime p. G_T is the p-order cyclic group, with g_T as the generator, on the extended field F_{qk}; k is the embedding degree of the elliptic curve. Operation e shall satisfy the following properties:

- For all P ∈ G_1, Q ∈ G_2, all l, m ∈ Z_n, it satisfies: e(lP, mQ) = $e(P,Q)^{lm}$;

- There is P ∈ G_1, Q ∈ G_2, so that e(lP, mQ)≠1_{GT};

- There is an effective algorithm to calculate e(P, Q).

3) Choose $r \in_R Z_p^*$ and $h_1 \in_R G_1$, $h_2 \in_R G_1$; calculate $w = g_2^r$.

4) Select the hash function H_1: $\{0, 1\}^* \rightarrow \{0, 1\}^{2l}$, H_2: $\{0, 1\}^{6\lambda} \rightarrow Z_p$, H_3: $\{0, 1\}^* \rightarrow G_2$, H_4: $\{0, 1\}^* \rightarrow Z_p$.

5) Calculate the bilinear mapping $T_1 = e(g_1, g_2)$, $T_2 = e(h_1, g_2)$, $T_3 = e(h_2, g_2)$, $T_w = e(h_2, w)$.

6) Calculate the cryptographic hash value H_p = HASH (p), H_{h1} = HASH (h_1) and H_{k0} = HASH (k_0). Generate the issuerSettings = (H_p, H_{h1}, H_{k0}) of the TCM_ECDAA_ISSUER data structure (defined in Appendix A), based on H_p, H_{h1}, H_{k0}. Use k_n^{-1} to generate cre = $sign_{kn}^{-1}$ (issuerSettings) for its signature.

7) Output system public parameters gpk = (q, α, b, p, g_1, g_2, e, h_1, h_2, w, H_1, H_2, H_3, H_4, T_1, T_2, T_3, T_w), for the signature cre = $sign_{kn}^{-1}$ (issuerSettings) and the confidential information isk = r, of public parameters.

6.3.2 System initialization 2

This algorithm is used by the prover host and TCM to set the public parameters the prover host;

4) The host calculates $s_r' = r_2 + cr'(\mathrm{mod}\ p)$; outputs aux = F and comm = $(C, c, s_f, s_r', n_T, n_I)$; meanwhile sends comm to the issuer.

6.3.4 Credential issuance algorithm 2

This algorithm is used by the issuer to generate anonymous credentials for the prover. Its input, output and algorithm flow are as follows:

a) Input: the credential application comm = $(C, c, s_f, s_r', n_T, n_I)$ generated by the prover, the confidential information of the issuer isk = r, the public parameter gpk.

b) Output: (part of) anonymous credentials (A, x, r'').

c) Algorithm flow: The issuer first verifies whether the value of n_I is generated by itself and is not replayed; then the issuer verifies whether the comm is valid. The verification method is: calculate $R' = h_1^{sf} h_2^{sr'} C^{-c}$ and $c'_h = H_1(gpk, C, R')$, verify $c \overset{?}{=} H_2(c'_h, n_I, n_T)$. Then randomly select $r'', x \in_R Z_p$, calculate $A = (g_1 C h_2^{r''})^{1/(x+r)}$, send (A, x, r'') to the prover host.

6.3.5 Certificate issuance algorithm 3

This algorithm is used by the prover to store anonymous credentials; its input, output and algorithm flow are as follows:

a) Input: (part of) anonymous credential cre = (A, x, r''), information required for verification aux, public parameter gpk, which are generated by the issuer for the prover.

b) Output: If the anonymous credential is valid, output "valid"; otherwise, output "invalid".

c) Algorithm flow: The prover calculates $r = r' + r''(\mathrm{mod}\ p)$; verifies whether the $e(A, wg_2^x) = e(g_1 F h_2^r, g_2)$ is established. If it is established, store cre = (A, x, r) in trust, meanwhile output "valid"; otherwise output "invalid".

6.3.6 Attestation algorithm

This algorithm is used by the prover to perform anonymous attestation AND generate the information required for the anonymous attestation. Its input,

------ The End ------