GM/T 0003.2

SM2 Public Key Cryptographic Algorithms Based on Elliptic Curves

Part 2: Digital Signature Algorithm

Cryptography Standardization
Technical Committee of China

Issued on 2012-03-21 Translated on 2024-10-30

Contents

Foreword

GM/T 0003 “SM2 public key cryptographic algorithm based on elliptic curves” consists of 5 parts:

- Part 1: General
- Part 2: Digital signature algorithm
- Part 3: Key exchange protocol
- Part 4: Public key encryption algorithm
- Part 5: Parameter definition

This section is the second part of GM/T 0003.

1 Scope

This part of GM/T 0003 specifies the digital signature algorithm of SM2 the public key cryptographic algorithms based on elliptic curves, including the digital signature generation and verification algorithms, and demonstrates examples of digital signature and verification and the corresponding processes.

This part is applicable to the generation and verification of digital signatures in commercial cryptographic applications. It meets the security requirements of identity verification and data integrity and authenticity in various types of cryptographic applications. Besides, this part also provides standardization guidance and reference to products and technology for manufacturers of security products, improving the credibility and maneuverability of security products.

2 Normative references

The following parts are necessary for the application of this document. For all the references with specified dates, only the version with those dates is suitable for this document. For the references without dates, the latest version (including all the modified lists) is suitable for this part.

GM/T 0003.1, SM2 Public key cryptographic algorithm based on elliptic curves — Part 1: General

3 Terms and definitions

The following terms and definitions are applicable to this document.

3.1 message

any bit string of finite length.

3.2 signed message

group of data elements that consists of a message and its digital signature.

3.3 signature key

private key of the signer; the secret data element used by the signer in the digital signature generating process.

3.4 signature process

the process of inputting a message, a signature private key, and system parameters of an elliptic curve and outputting a digital signature.

3.5 distinguishable identifier

the information that can unambiguously identify the identity of an entity.

4 Symbols

The following symbols are applicable to this part.

A, B: two users who use the public key cryptography system.
a, b: elements in F_q which define an elliptic curve E over F_q.
d_A: the private key of user A.
$E(F_q)$: the set of rational points on elliptic curves E over F_q (including the infinity point O).
e: the output of a cryptographic hash function on input M.
e': the output of a cryptographic hash function on input M'.
F_q: the finite field with q elements.
G: a base point of an elliptic curve with prime order.
$H_v()$: a cryptographic hash function with v bits message digest.
ID_A: the distinguishable identifier of user A.
M: the message to be signed.
M' : the message to be verified.
mod n: the operation of modulo n, for example, $23 \bmod 7 = 2$.
n: the order of a base point G where n is a prime factor of $\#E(F_q)$.
O: a special point on an elliptic curve called the infinity point or zero point. It is the identity of the additive group of an elliptic curve.

P_A: the public key of user A.

q: the number of elements of finite field F_q.

$x||y$: the concatenation of x and y, where x and y are bit strings or byte strings.

Z_A: the hash value of the distinguishable identifier of user A, part of the system parameters of elliptic curves and the public key of user A.

(r, s): the sent signature.

(r', s'): the received signature.

$[k]P$: a point which is k times of point P on elliptic curves, i.e., $[k]P = \underbrace{P + P + \cdots + P}_{Add\ k\ times}$, where k is a positive integer.

$[x, y]$: the set of integers which are greater than or equal to x and less than or equal to y.

$\lceil x \rceil$: ceiling function which maps x to the smallest integer greater than or equal to x. For example, $\lceil 7 \rceil = 7$, $\lceil 8.3 \rceil = 9$.

$\lfloor x \rfloor$: floor function which maps x to the largest integer less than or equal to x. For example, $\lfloor 7 \rfloor = 7$, $\lfloor 8.3 \rfloor = 8$.

$\#E(F_q)$: the number of points on $E(F_q)$, called the order of the elliptic curve $E(F_q)$.

5 Digital signature algorithm

5.1 General

The digital signature algorithm generates a digital signature of data by a signer and verifies the validity of the signature by a verifier. Every signer has a private key and the corresponding public key, where the private key is used to generate a signature, and the public key is used by the verifier to verify the signature. Before the signature generation process, message $\bar{M}$ (consisting of Z_A and M) should be compressed using a cryptographic hash function. Before the verification process, $\bar{M}'$ (consisting of Z_A and M') should be compressed using a cryptographic hash function.

5.2 System parameters of elliptic curves

The system parameters of elliptic curves include the size q of finite field F_q (additionally including identifiers of representation of elements and reduced polynomial when $q = 2^m$), two elements $a, b \in F_q$ which define an equation of the elliptic curves

$E(F_q)$, a base point $G = (x_G, y_G)(G \neq O)$ over $E(F_q)$ where x_G and y_G are two elements of F_q, the order n of G and other optional parameters (e.g., the cofactor h of n).

The system parameters of elliptic curves and their verification should be in line with the regulations in Clause 5 of GM/T 0003.1.

5.3 User key pair

The key pair of user A includes the private key d_A and the public key $P_A = [d_A]G = (x_A, y_A)$.

The generation algorithm of user's key pair and the verification of public key should be in line with the regulations in Clause 6 of GM/T 0003.1.

5.4 Auxiliary functions

5.4.1 Overview

Two kinds of auxiliary functions are involved in the digital signature algorithm based on elliptic curves specified in this part: cryptographic hash functions and random number generators.

5.4.2 Cryptographic hash functions

This part should adopt secure cryptographic hash functions, approved by the State Cryptography Administration, such as the GM/T 0004 SM3, Cryptographic Hash Algorithm.

5.4.3 Random number generators

This part should adopt random number generators, approved by the State Cryptography Administration, such as GM/T 0105, software-based random number generators.

5.5 Other information of user

User A, as the signer, has a distinguishable identifier ID_A of length $entlen_A$ bits. Notation $ENTL_A$ is the two bytes converted from integer $entlen_A$. In the digital signature algorithm based on elliptic curves specified in this part, both the signer and the verifier should use cryptographic hash functions to obtain user A's hash value Z_A. Convert the data types of the elliptic curve equation parameters a, b, the coordinates (x_G, y_G) of G, and P_A's coordinates (x_A, y_A) to bit string as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1. Then $Z_A = H_{256}(ENTL_A||ID_A||a||b||x_G||y_G||x_A||y_A)$.

6 Digital signature generation algorithm and its process

6.1 Digital signature generation algorithm

Let M be the message to be signed. To obtain a signature (r, s) of the message M, user A as a signer should do the following:

A1: Set $\bar{M} = Z_A \parallel M$;

A2: Compute $e = H_v(\bar{M})$, and convert the type of data e to be an integer as specified in Clauses 4.2.4 and 4.2.3 of GM/T 0003.1;

A3: Generate a random number $k \in [1, n-1]$ using a random number generator;

A4: Compute $(x_1, y_1) = [k]G$, and convert the type of data x_1 to be an integer as specified in Clause 4.2.8 of GM/T 0003.1;

A5: Compute $\mathrm{r} = (e + x_1) \bmod n$. If $r = 0$ or $r + k = n$, then go to A3;

A6: Compute $s = \left((1 + d_A)^{-1} \cdot (k - r \cdot d_A)\right) \bmod n$. If $s = 0$, then go to A3;

A7: Convert the type of data r, s to be bit strings according to the details in Clause 4.2.2 of GM/T 0003.1. Then, the signature of message M is (r, s).

Note: for examples of digital signature generation process, see Annex A.

6.2 Process of digital signature generation algorithm

The process of digital signature generation algorithm is depicted in Figure 1.

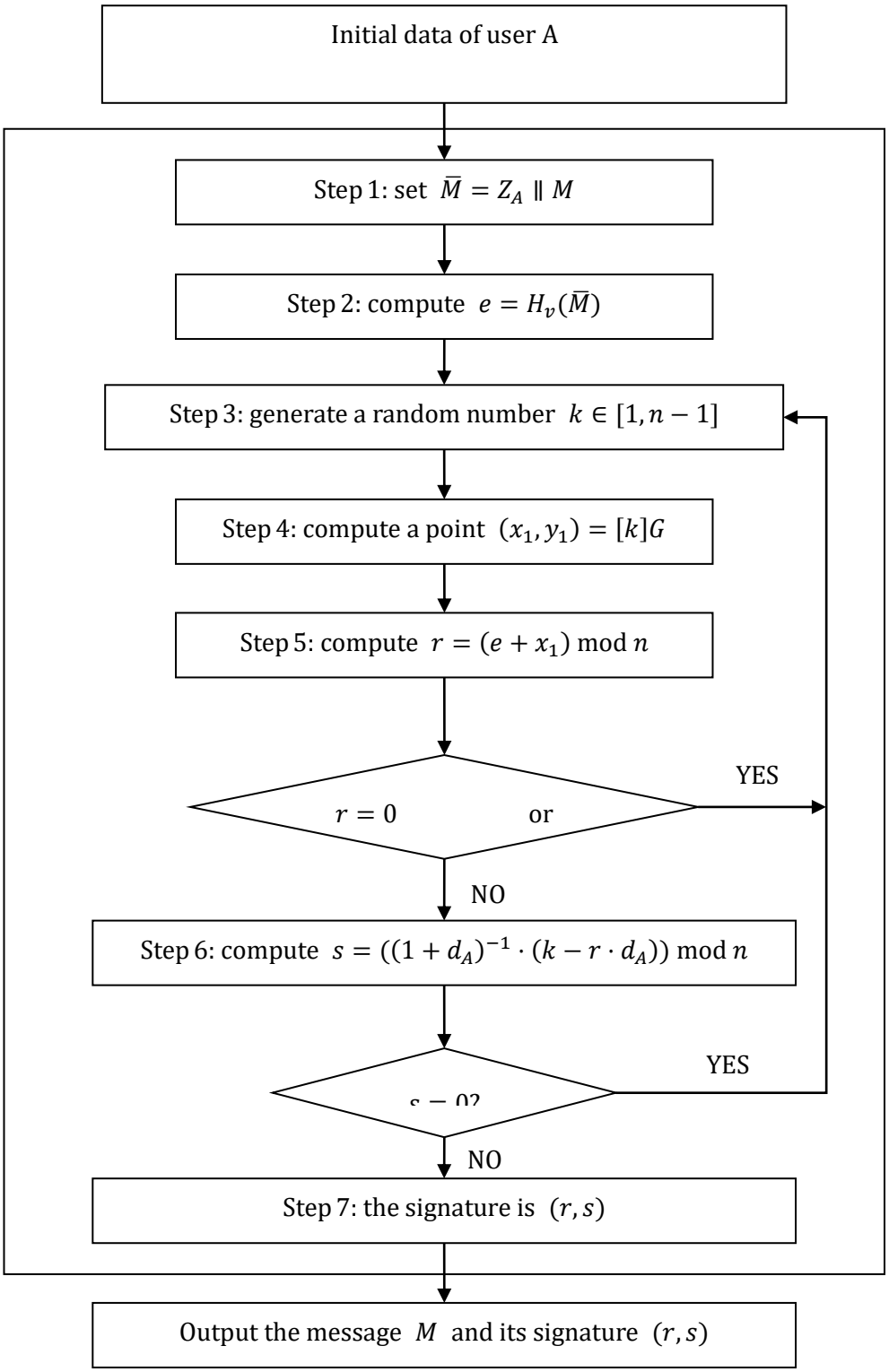

Figure 1: Digital signature generation algorithm process

7 Digital signature verification algorithm and its process

7.1 Digital signature verification algorithm

To verify the received message M' and its digital signature (r', s'), as a verifier, user B should implement the following operations:

B1: Verify whether $r' \in [1, n-1]$ holds; If not, the verifier outputs reject.

B2: Verify whether $s' \in [1, n-1]$ holds; If not, the verifier outputs reject.

B3: Set $\bar{M}' = Z_A \| M'$.

B4: Compute $e' = H_v(\bar{M}')$ and convert the type of data e' to be integer as specified in Clauses 4.2.4 and 4.2.3 of GM/T 0003.1.

B5: Convert the type of data r', s' to be integers as specified in Clause 4.2.3 of GM/T 0003.1 and compute $t = (r' + s') \bmod n$. If $t = 0$, then the verifier outputs reject.

B6: Compute the point $(x_1', y_1') = [s']G + [t]P_A$.

B7: Convert the type of data x_1' to integer with the way in Clause 4.2.8 of GM/T 0003.1. Compute $R = (e' + x_1') \bmod n$ and verify whether $R = r'$ holds. If so, the verifier outputs accept; otherwise outputs reject.

Note: If Z_A is not the hash function value of user A, the verifier obviously outputs reject". For examples of the digital signature verification process, see Annex A.

7.2 Process of digital signature verification algorithm

The process of digital signature verification algorithm is shown in Figure 2.

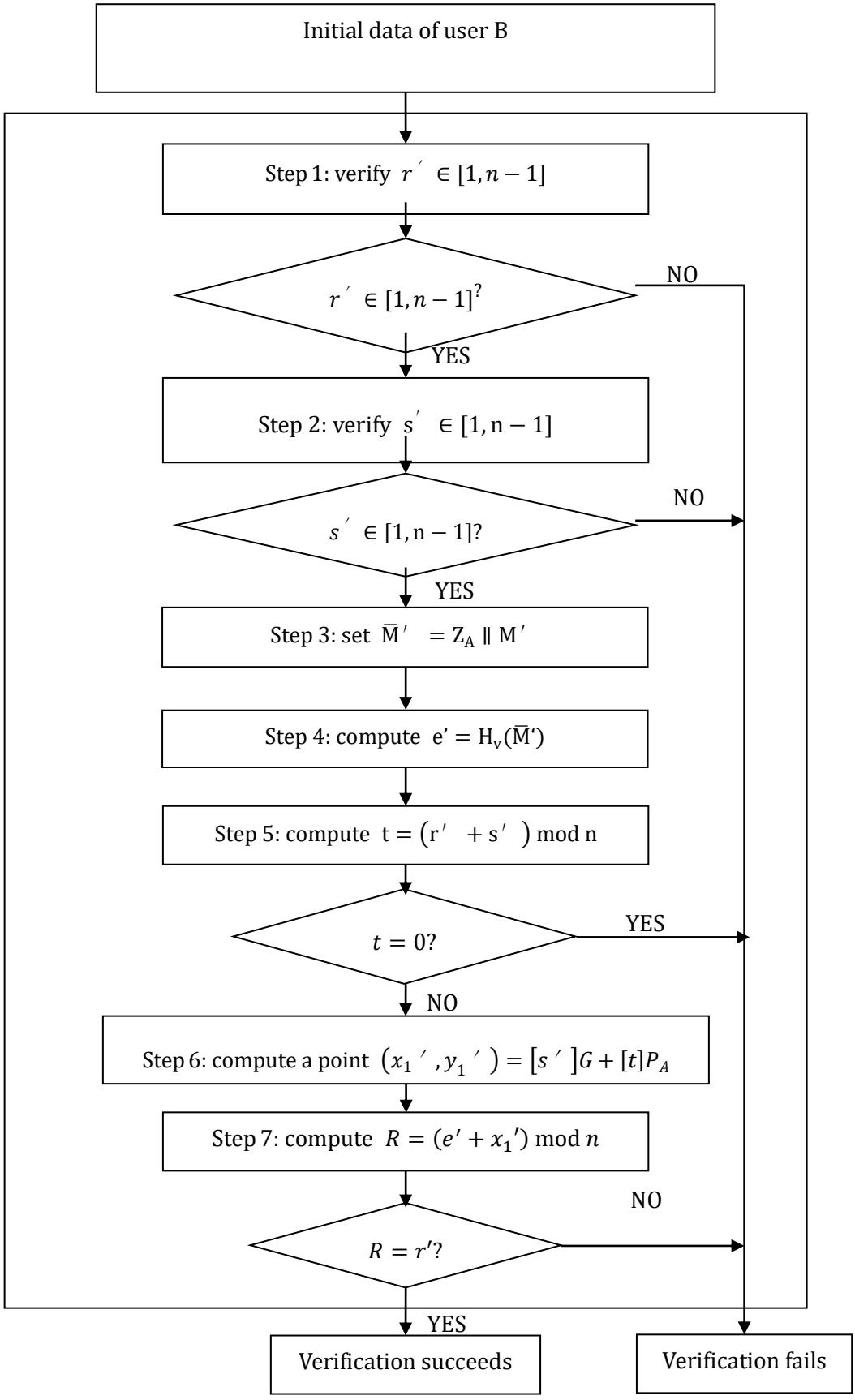

Figure 2: Digital signature verification algorithm process

Annex A
(informative)
Examples of digital signature and verification

A.1 General requirements

This annex adopts the cryptographic hash function, which is specified in the GM/T 0004 Cryptographic Hash Algorithm SM3, whose input is bit strings of length less than 2^{64}, and output is a hash value of length 256 bits, denoted by $H_{256}(\ )$.

In this annex, for all values represented in the hexadecimal form, the left is the most significant side, and the right is the least significant side.

In this annex, all messages are denoted as ASCII encoding.

Suppose user A's identity is ALICE123@YAHOO.COM. Its ASCII encoding ID_A is 414C 49434531 32334059 41484F4F 2E434F4D. $ENTL_A = 0090$.

A.2 Digital signature of elliptic curves over F_p

The elliptic curve equation is: $y^2 = x^3 + ax + b$

Example 1: $F_p - 256$

prime p: 8542D69E 4C044F18 E8B92435 BF6FF7DE 45728391 5C45517D 722EDB8B 08F1DFC3

coefficient a: 787968B4 FA32C3FD 2417842E 73BBFEFF 2F3C848B 6831D7E0 EC65228B 3937E498

coefficient b: 63E4C6D3 B23B0C84 9CF84241 484BFE48 F61D59A5 B16BA06E 6E12D1DA 27C5249A

base point $G = (x_G, y_G)$ whose order is n

coordinate x_G: 421DEBD6 1B62EAB6 746434EB C3CC315E 32220B3B ADD50BDC 4C4E6C14 7FEDD43D

coordinate y_G: 0680512B CBB42C07 D47349D2 153B70C4 E5D7FDFC BFA36EA1 A85841B9 E46E09A2

order n: 8542D69E 4C044F18 E8B92435 BF6FF7DD 29772063 0485628D 5AE74EE7 C32E79B7

message to be signed M: message digest

private key d_A:128B2FA8 BD433C6C 068C8D80 3DFF7979 2A519A55 171B1B65 0C23661D 15897263

public key $P_A = (x_A, y_A)$:

coordinate x_A: 0AE4C779 8AA0F119 471BEE11 825BE462 02BB79E2 A5844495 E97C04FF 4DF2548A

coordinate y_A: 7C0240F8 8F1CD4E1 6352A73C 17B7F16F 07353E53 A176D684 A9FE0C6B B798E857

hash value $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$

Z_A: F4A38489 E32B45B6 F876E3AC 2168CA39 2362DC8F 23459C1D 1146FC3D BFB7BC9A

Intermediate values in the steps of generating signature:

$\bar{M} = Z_A \parallel M$:

F4A38489 E32B45B6 F876E3AC 2168CA39 2362DC8F 23459C1D 1146FC3D BFB7BC9A 6D657373 61676520 64696765 7374

cryptographic hash value $e = H_{256}(\bar{M})$: B524F552 CD82B8B0 28476E00 5C377FB1 9A87E6FC 682D48BB 5D42E3D9 B9EFFE76

generate random number k: 6CB28D99 385C175C 94F94E93 4817663F C176D925 DD72B727 260DBAAE 1FB2F96F

compute point: $(x_1, y_1) = [k]G$ of the elliptic curve:

coordinate x_1: 110FCDA5 7615705D 5E7B9324 AC4B856D 23E6D918 8B2AE477 59514657 CE25D112

coordinate y_1: 1C65D68A 4A08601D F24B431E 0CAB4EBE 084772B3 817E8581 1A8510B2 DF7ECA1A

compute $r = (e + x_1) \bmod n$: 40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

$(1 + d_A)^{-1}$: 79BFCF30 52C80DA7 B939E0C6 914A18CB B2D96D85 55256E83 122743A7 D4F5F956

compute $s = (1 + d_A)^{-1} \cdot (k - r \cdot d_A) \bmod n$: 6FC6DAC3 2C5D5CF1 0C77DFB2 0F7C2EB6 67A45787 2FB09EC5 6327A67E C7DEEBE7

The signature of message m is (r, s):

Value r: 40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

Value s: 6FC6DAC3 2C5D5CF1 0C77DFB2 0F7C2EB6 67A45787 2FB09EC5 6327A67E C7DEEBE7

Verify the related values:

cryptographic hash value $e' = H_{256}(\bar{M}')$: B524F552 CD82B8B0 28476E00 5C377FB1 9A87E6FC 682D48BB 5D42E3D9 B9EFFE76

compute $t = (r' + s') \bmod n$: 2B75F07E D7ECE7CC C1C8986B 991F441A D324D6D6 19FE06DD 63ED32E0 C997C801

compute point $(x'_0, y'_0) = [s']G$ of the elliptic curve:

coordinate x'_0: 7DEACE5F D121BC38 5A3C6317 249F413D 28C17291 A60DFD83 B835A453 92D22B0A

coordinate y'_0: 2E49D5E5 279E5FA9 1E71FD8F 693A64A3 C4A94611 15A4FC9D 79F34EDC 8BDDEBD0

compute point $(x'_{00}, y'_{00}) = [t]P_A$ of the elliptic curve:

coordinate x'_{00}: 1657FA75 BF2ADCDC 3C1F6CF0 5AB7B45E 04D3ACBE 8E4085CF A669CB25 64F17A9F

coordinate y'_{00}: 19F0115F 21E16D2F 5C3A485F 8575A128 BBCDDF80 296A62F6 AC2EB842 DD058E50

compute point $(x'_1, y'_1) = [s']G + [t]P_A$ of the elliptic curve

coordinate x'_1: 110FCDA5 7615705D 5E7B9324 AC4B856D 23E6D918 8B2AE477 59514657 CE25D112

coordinate y_1': 1C65D68A 4A08601D F24B431E 0CAB4EBE 084772B3 817E8581 1A8510B2 DF7ECA1A

compute $R = (e' + x_1') \bmod n$: 40F1EC59 F793D9F4 9E09DCEF 49130D41 94F79FB1 EED2CAA5 5BACDB49 C4E755D1

A.3 Digital signature of elliptic curves over F_{2^m}

The elliptic curve equation is: $y^2 + xy = x^3 + ax^2 + b$

Example 2: $F_{2^m} - 257$

generator polynomial of the base field: $x^{257} + x^{12} + 1$

coefficient a : 0

coefficient b : 00 E78BCD09 746C2023 78A7E72B 12BCE002 66B9627E CB0B5A25 367AD1AD 4CC6242B

base point $G = (x_G, y_G)$ with order n

coordinate x_G: 00 CDB9CA7F 1E6B0441 F658343F 4B10297C 0EF9B649 1082400A 62E7A748 5735FADD

coordinate y_G: 01 3DE74DA6 5951C4D7 6DC89220 D5F7777A 611B1C38 BAE260B1 75951DC8 060C2B3E

order n: 7FFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF BC972CF7 E6B6F900 945B3C6A 0CF6161D

message to be signed M: message digest

private key d_A: 771EF3DB FF5F1CDC 32B9C572 93047619 1998B2BF 7CB981D7 F5B39202 645F0931

public key $P_A = (x_A, y_A)$:

coordinate x_A: 01 65961645 281A8626 607B917F 657D7E93 82F1EA5C D931F40F 6627F357 542653B2

coordinate y_A: 01 68652213 0D590FB8 DE635D8F CA715CC6 BF3D05BE F3F75DA5 D5434544 48166612

hash value $Z_A = H_{256}(ENTL_A \parallel ID_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$

Z_A: 26352AF8 2EC19F20 7BBC6F94 74E11E90 CE0F7DDA CE03B27F 801817E8 97A81FD5

Intermediate values in the steps of signature generation:

$\overline{M} = Z_A \parallel M$:

26352AF8 2EC19F20 7BBC6F94 74E11E90 CE0F7DDA CE03B27F 801817E8 97A81FD5

6D657373 61676520 64696765 7374

cryptographic hash value $e = H_{256}(\overline{M})$: AD673CBD A3114171 29A9EAA5 F9AB1AA1 633AD477 18A84DFD 46C17C6F A0AA3B12

generate random number k: 36CD79FC 8E24B735 7A8A7B4A 46D454C3 97703D64 98158C60 5399B341 ADA186D6

compute point: $(x_1, y_1) = [k]G$ of the elliptic curve:

coordinate x_1: 00 3FD87D69 47A15F94 25B32EDD 39381ADF D5E71CD4 BB357E3C 6A6E0397 EEA7CD66

coordinate y_1: 00 80771114 6D73951E 9EB373A6 58214054 B7B56D1D 50B4CD6E B32ED387 A65AA6A2

compute $r = (e + x_1) \bmod n$: 6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B

$(1 + d_A)^{-1}$: 73AF2954 F951A9DF F5B4C8F7 119DAA1C 230C9BAD E60568D0 5BC3F432 1E1F4260

Compute $s = (1 + d_A)^{-1} \cdot (k - r \cdot d_A) \, mod \, n$: 3124C568 8D95F0A1 0252A9BE D033BEC8 4439DA38 4621B6D6 FAD77F94 B74A9556

Signature (r, s) of message M:

Value r: 6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B

Value s: 3124C568 8D95F0A1 0252A9BE D033BEC8 4439DA38 4621B6D6 FAD77F94 B74A9556

Verify the related values:

cryptographic hash value $e' = H_{256}\left(\bar{M}'\right)$: AD673CBD A3114171 29A9EAA5 F9AB1AA1 633AD477 18A84DFD 46C17C6F A0AA3B12

compute $t = (r' + s') mod \, n$: 1E647F8F 784891A6 51AFC342 0316F44A 042D7194 4C91910F 835086C8 2CB07194

compute the point on elliptic curve $(x_0', y_0') = \left[s'\right] G$:

coordinate x_0': 00 252CF6B6 3A044FCE 553EAA77 3E1E9264 44E0DAA1 0E4B8873 89D11552 EA6418F7

coordinate y_0': 00 776F3C5D B3A0D312 9EAE44E0 21C28667 92E4264B E1BEEBCA 3B8159DC A382653A

compute point $(x_{00}', y_{00}') = [t] P_A$ of the elliptic curve

coordinate x_{00}': 00 07DA3F04 0EFB9C28 1BE107EC C389F56F E76A680B B5FDEE1D D554DC11 EB477C88

coordinate y_{00}': 01 7BA2845D C65945C3 D48926C7 0C953A1A F29CE2E1 9A7EEE6B E0269FB4 803CA68B

compute point $\left(x_1', y_1'\right) = \left[s'\right] G + [t] P_A$ of the elliptic curve

coordinate x_1': 00 3FD87D69 47A15F94 25B32EDD 39381ADF D5E71CD4 BB357E3C 6A6E0397 EEA7CD66

coordinate y_1': 00 80771114 6D73951E 9EB373A6 58214054 B7B56D1D 50B4CD6E B32ED387 A65AA6A2

compute $R = \left(e' + x_1'\right) \, mod \, n$: 6D3FBA26 EAB2A105 4F5D1983 32E33581 7C8AC453 ED26D339 1CD4439D 825BF25B