GM/T 0003.3

SM2 Public Key Cryptographic Algorithms Based on Elliptic Curves

Part 3: Key Exchange Protocol

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 third part of GM/T 0003.

Copyright Notice

1 Scope

This part of GM/T 0003 specifies the key exchange protocol of SM2 public key cryptographic algorithms based on elliptic curves, and gives examples of key exchange and verification and the corresponding processes.

This part is applicable to key exchanges in commercial cryptographic applications. It meets the requirements that two parties establish their shared private key (session key) by computation via two or three optional exchanges of information. Besides, this part also provides standardization guidance and standardization references to products and technology for manufacturers of security products, improving the credibility and maneuverability of security products.

2 Normative references

The following documents are necessary for the application of this document. For the references with noted dates, only the version on the specific date applies to this part. For the references without dates, the newest version (including all the modified lists) applies to this part.

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

3 Terms and definitions

The following terms and definitions apply to this part.

3.1 key confirmation from A to B

Confirmation from A to B that A holds a specific private key.

3.2 key derivation function

The function that generates one or more shared secret keys on input shared secrets and other parameters known to the two parties.

3.3 initiator

The user who sends the first round exchanging message in the execution of a protocol.

3.4 responder

The user who does not send the first round exchanging message in the execution of a protocol.

3.5 distinguishable identifier

The information that can identify the identity of an entity without ambiguity.

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.

d_B: the private key of user B.

$E(F_q)$: the set of rational points on elliptic curves E over F_q (including the infinity point O).

F_q: the finite field with q elements.

G: a base point of an elliptic curve with prime order.

$Hash(\,)$: a cryptographic hash function.

$H_v(\,)$: a cryptographic hash function with v bits message digest.

h: the cofactor is defined as $h = \#E(F_q)/n$, where n is the order of the base point G.

ID_A, ID_B: the distinguishing identifiers of user A and user B, respectively.

K, K_A, K_B: the shared private keys generated in the key exchange protocol.

$KDF()$: key derivation function.

$mod\ n$: the operation of modulo n, for example, $23\ mod\ 7 = 2$.

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, which is the identity of the additive group of an elliptic curve.

P_A: the public key of user A.

P_B: the public key of user B.

q: the number of elements of finite field F_q.

r_A: ephemeral key generated by user A in key exchange.

r_B: ephemeral key generated by user B in key exchange.

$x \parallel y$: 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.

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

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

$[k]P$: 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, $\lfloor 7 \rfloor = 7$, $\lfloor 8.3 \rfloor = 8$.

&: the bit-wise AND operation of two integers.

5 Algorithm parameters and auxiliary functions

5.1 General

Key exchange protocol is the process of two users A and B use their respective private key and the other's public key to share a secret key, by interactive information exchanges. The shared secret key is generally used in some symmetric cryptographic algorithms. The key exchange protocol can be used in key management and key agreements.

5.2 System parameters of elliptic curves

The system parameters of elliptic curves include the size q of finite field F_q (when $q = 2^m$, identifiers of representation of elements and reduced polynomial are also involved), two elements $a, b \in F_q$, which define the equation of the elliptic curve $E(F_q)$, 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, etc.).

The system parameters of elliptic curves and their validation shall 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 key pair of user B includes the private key d_B and the public key $P_B = [d_B]G = (x_B, y_B)$.

The generation algorithm of the user key pair and the verification of the 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

Three kinds of auxiliary functions are involved in the key exchange protocol based on elliptic curves specified in this part: cryptographic hash functions, key derivation functions and random number generators. These three types of auxiliary functions have direct impact on security of the key exchange protocol.

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 Key derivation function

The functionality of key derivation functions is to derive key data from a shared secret bit string. In the process of key agreement, on input of the shared secret bit string obtained by key exchange protocol, the key derivation function outputs a required session key or a key data required by further encryption.

Key derivation function needs to invoke the cryptographic hash function.

Let $H_v(\,)$ be a cryptographic hash function which outputs a hash value of length v bits.

Key derivation function $KDF(Z, klen)$ is defined as follows:

Input: a bit string Z, an integer $klen$ which represents the bit length of the resulting secret key data and is required to be smaller than $(2^{32}-1)v$.

Output: the secret key bit string K of length $klen$ bits.

a) Initialize a 32-bit counter $ct = 0\text{x}00000001$;

b) For i from 1 to $\left\lceil \frac{klen}{v} \right\rceil$:

b.1) compute $Ha_i = H_v(Z \parallel ct)$;

b.2) $ct++$;

c) If $\frac{klen}{v}$ is an integer, let $Ha!_{\left\lceil \frac{klen}{v} \right\rceil} = Ha_{\left\lceil \frac{klen}{v} \right\rceil}$; Otherwise let $Ha!_{\left\lceil \frac{klen}{v} \right\rceil}$ be the leftmost $\left(klen - \left(v \times \left\lfloor \frac{klen}{v} \right\rfloor\right)\right)$ bits of $Ha_{\left\lceil \frac{klen}{v} \right\rceil}$;

d) Let $K = Ha_1 \parallel Ha_2 \parallel \cdots \parallel Ha_{\left\lceil \frac{klen}{v} \right\rceil - 1} \parallel Ha!_{\left\lceil \frac{klen}{v} \right\rceil}$.

5.4.4 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 has a distinguishable identifier ID_A of length $entlen_A$ bits. Notation $ENTL_A$ is the two bytes converted from integer $entlen_A$. User B has a distinguishable identifier ID_B of length $entlen_B$ bits. Notation $ENTL_B$ is the two bytes converted from integer $entlen_B$. In the key exchange protocol based on elliptic curves specified in this part, both parties A and B participating in the key agreement use the cryptographic hash function to get their respective hash values Z_A and Z_B. Convert the data types of the elliptic curve equation parameters a, b, coordinates (x_G, y_G) of G, coordinates (x_A, y_A) of P_A, and coordinates (x_B, y_B) of P_B to bit strings as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1, then $Z_A = H_{256}(ENTL_A||ID_A||a||b||x_G||y_G||x_A||y_A)$ and $Z_B = H_{256}(ENTL_B||ID_B||a||b||x_G||y_G||x_B||y_B)$.

6 Key exchange protocol and the process

6.1 Key exchange protocol

Suppose the length of the secret key established by A and B is $klen$. A is the initiator and B is the responder. In order to get the same secret key, A and B should perform the following operations:

Let $w = \lceil(\lceil\log_2(n)\rceil/2)\rceil - 1$.

User A:

A1: Generate a random number $r_A \in [1, n-1]$ with random number generators;

A2: Compute the point $R_A = [r_A]G = (x_1, y_1)$ on elliptic curve;

A3: Send R_A to B;

User B:

B1: Generate a random number $r_B \in [1, n-1]$ with random number generators;

B2: Compute the point $R_B = [r_B]G = (x_2, y_2)$ of the elliptic curve;

B3: Take the element x_2 from R_B, convert its data type to an integer as specified in Clause 4.2.8 of GM/T 0003.1, and compute $\bar{x}_2 = 2^w + \left(x_2 \& (2^w - 1)\right)$;

B4: Compute $t_B = (d_B + \bar{x}_2 \cdot r_B) \bmod n$;

B5: Verify whether R_A satisfies the elliptic curve equation. If not, the agreement is failed. Otherwise, take the element x_1 from R_A, compute $\bar{x}_1 = 2^w + \left(x_1 \& (2^w - 1)\right)$, and convert its data type to an integer as specified in Clause 4.2.8 of GM/T 0003.1;

B6: Compute the point $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$ of the elliptic curve. If V is the infinity point, the agreement is failed; Otherwise, convert the types of data x_V, y_V to bit strings as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1;

B7: Compute $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$;

B8: (Optional) Convert the data type of R_A's coordinates x_1, y_1 and R_B's coordinates x_2, y_2 to bit strings as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1, and compute $S_B = Hash(0\text{x}02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$;

B9: Send R_B and S_B(optional) to A;

User A:

A4: Take the element x_1 from R_A, and compute $\bar{x}_1 = 2^w + \left(x_1 \& (2^w - 1)\right)$ by converting its data type to integer as specified in Clause 4.2.8 of GM/T 0003.1;

A5: Compute $t_A = (d_A + \bar{x}_1 \cdot r_A) \bmod n$;

A6: Verify whether R_B satisfies the elliptic curve equation. If not, the agreement is failed. Otherwise, take the element x_2 from R_B, and compute $\bar{x}_2 = 2^w + \left(x_2 \& (2^w - 1)\right)$ by converting its data type to integer as specified in Clause 4.2.8 of GM/T 0003.1;

A7: Compute the point $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$ of the elliptic curve. If U is the infinity point, the agreement is failed; otherwise, convert the data type of x_U, y_U to integers as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1;

A8: Compute $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B, klen)$;

A9: (Optional) Convert the data type of R_A's coordinates x_1, y_1, and R_B's coordinates x_2, y_2 to bit strings as specified in Clauses 4.2.6 and 4.2.5 of GM/T 0003.1, compute $S_1 = Hash(0\text{x}02 \parallel y_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$, and verify whether $S_1 = S_B$ holds; If not, key confirmation from B to A is failed;

A10: (Optional) Compute $S_A = Hash(0\text{x}03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, and send S_A to B.

User B:

B10: (Optional) Compute $S_2 = Hash(0\text{x}03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, and verify whether $S_2 = S_A$ holds. If not, key confirmation from A to B is failed.

Note: If Z_A, Z_B are not the corresponding hash values of user A and B respectively, they obviously cannot achieve a consistent shared secret key. The examples of the process of key exchange protocol are described in Annex A.

6.2 Process of key exchange protocol

The process of the key exchange protocol is depicted in Figure 1.

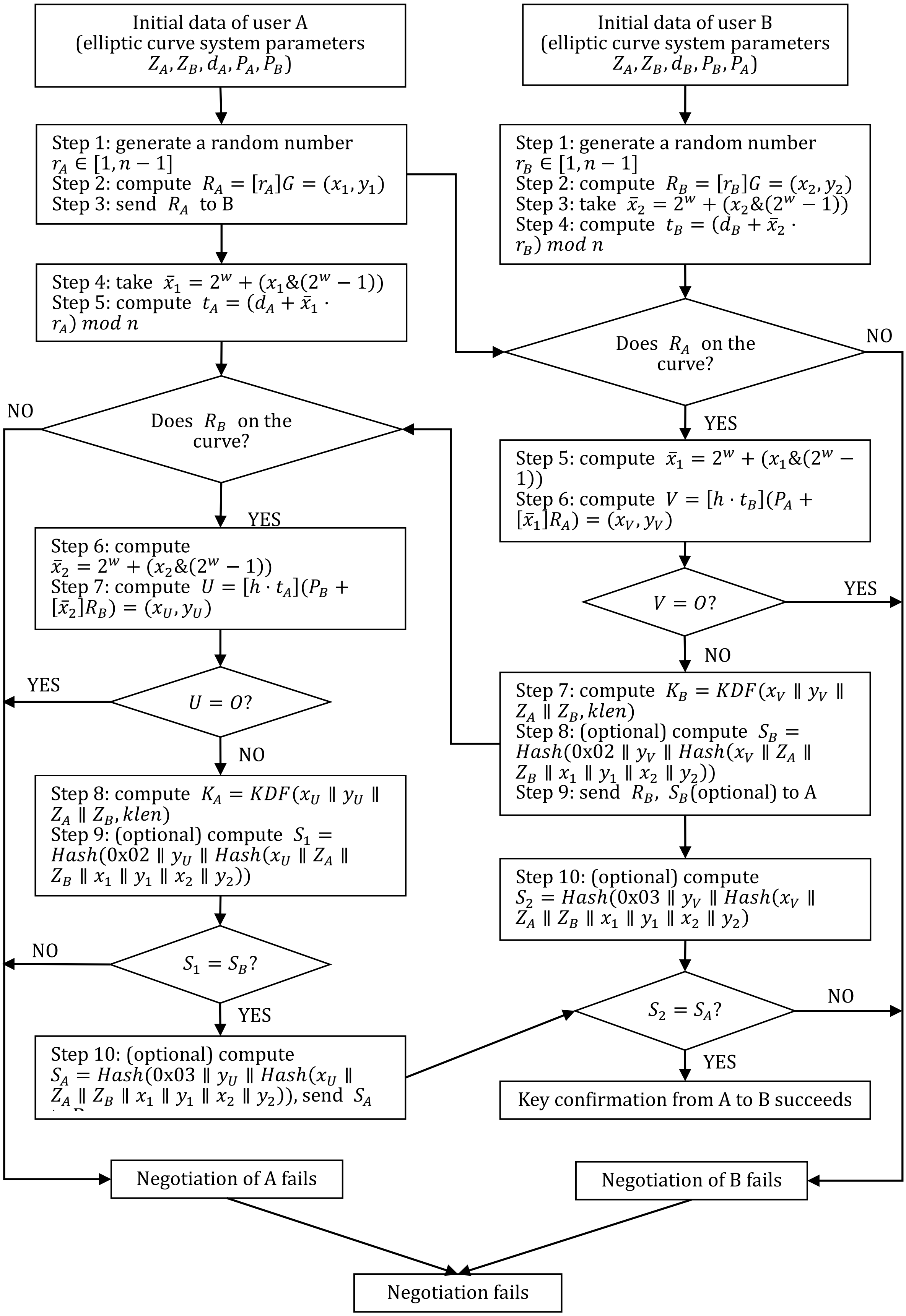
Initial data of user A (elliptic curve system parameters Z_A, Z_B, d_A, P_A, P_B)
Initial data of user B (elliptic curve system parameters Z_A, Z_B, d_B, P_B, P_A)
Step 1: generate a random number $r_A \in [1, n-1]$
Step 2: compute $R_A = [r_A]G = (x_1, y_1)$
Step 3: send R_A to B
Step 1: generate a random number $r_B \in [1, n-1]$
Step 2: compute $R_B = [r_B]G = (x_2, y_2)$
Step 3: take $\bar{x}_2 = 2^w + (x_2\&(2^w - 1))$
Step 4: compute $t_B = (d_B + \bar{x}_2 \cdot r_B) \, mod \, n$
Step 4: take $\bar{x}_1 = 2^w + (x_1\&(2^w - 1))$
Step 5: compute $t_A = (d_A + \bar{x}_1 \cdot r_A) \, mod \, n$
Does R_A on the curve?
NO
YES
Does R_B on the curve?
NO
YES
Step 5: compute $\bar{x}_1 = 2^w + (x_1\&(2^w - 1))$
Step 6: compute $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$
Step 6: compute $\bar{x}_2 = 2^w + (x_2\&(2^w - 1))$
Step 7: compute $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$
$V = O?$
YES
NO
$U = O?$
YES
NO
Step 7: compute $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$
Step 8: (optional) compute $S_B = Hash(0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$
Step 9: send R_B, S_B(optional) to A
Step 8: compute $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B, klen)$
Step 9: (optional) compute $S_1 = Hash(0x02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$
Step 10: (optional) compute $S_2 = Hash(0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$
$S_1 = S_B?$
NO
YES
$S_2 = S_A?$
NO
YES
Step 10: (optional) compute $S_A = Hash(0x03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$, send S_A
Key confirmation from A to B succeeds
Negotiation of A fails
Negotiation of B fails
Negotiation fails

Figure 1: Key exchange protocol process

Annex A
(informative)
Examples of key exchange and verification

A.1 General requirements

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

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

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

Suppose user B's identity is BILL456@YAHOO.COM. Its ASCII encoding ID_B is: 42 494C4C34 35364059 41484F4F 2E434F4D. $\mathrm{ENTL_B} = 0088$.

A.2 Key exchange protocol on elliptic curves over F_p

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

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

cofactor h: 1

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

user A's private key d_A: 6FCBA2EF 9AE0AB90 2BC3BDE3 FF915D44 BA4CC78F 88E2F8E7 F8996D3B 8CCEEDEE

user A's public key $P_A = (x_A, y_A)$:

coordinate x_A: 3099093B F3C137D8 FCBBCDF4 A2AE50F3 B0F216C3 122D7942 5FE03A45 DBFE1655

coordinate y_A: 3DF79E8D AC1CF0EC BAA2F2B4 9D51A4B3 87F2EFAF 48233908 6A27A8E0 5BAED98B

user B's private key d_B: 5E35D7D3 F3C54DBA C72E6181 9E730B01 9A84208C A3A35E4C 2E353DFC CB2A3B53

user B's public key $P_B = (x_B, y_B)$:

coordinate x_B: 245493D4 46C38D8C C0F11837 4690E7DF 633A8A4B FB3329B5 ECE604B2 B4F37F43

coordinate y_B: 53C0869F 4B9E1777 3DE68FEC 45E14904 E0DEA45B F6CECF99 18C85EA0 47C60A4C

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: E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31

hash value $Z_B = H_{256}(ENTL_B \parallel ID_B \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_B \parallel y_B)$

Z_B: 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

Related values in steps A1-A3 in the key exchange protocol:

generate random number r_A:

83A2C9C8 B96E5AF7 0BD480B4 72409A9A 327257F1 EBB73F5B 073354B2 48668563

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

coordinate x_1: 6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0

coordinate y_1: 0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A

Related values in steps B1-B9 in the key exchange protocol:

generate random number r_B:

33FE2194 0342161C 55619C4A 0C060293 D543C80A F19748CE 176D8347 7DE71C80

compute point $R_B = [r_B]G = (x_2, y_2)$ of the elliptic curve:

coordinate x_2: 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5

coordinate y_2: 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

take $\bar{x}_2 = 2^{127} + \left(x_2 \& \left(2^{127} - 1\right)\right)$: B8F2B533 7B3DCF45 14E8BBC1 9D900EE5

compute $t_B = (d_B + \bar{x}_2 \cdot r_B) \, mod \, n$:

2B2E11CB F03641FC 3D939262 FC0B652A 70ACAA25 B5369AD3 8B375C02 65490C9F

take $\bar{x}_1 = 2^{127} + \left(x_1 \& \left(2^{127} - 1\right)\right)$: E856C095 05324A6D 23150C40 8F162BF0

Compute the point $[\bar{x}_1]R_A = (x_{A0}, y_{A0})$ of the elliptic curve:

coordinate x_{A0}: 2079015F 1A2A3C13 2B67CA90 75BB2803 1D6F2239 8DD8331E 72529555 204B495B

coordinate y_{A0}: 6B3FE6FB 0F5D5664 DCA16128 B5E7FCFD AFA5456C 1E5A914D 1300DB61 F37888ED

compute point $P_A + [\bar{x}_1]R_A = (x_{A1}, y_{A1})$ of the elliptic curve:

coordinate x_{A1}: 1C006A3B FF97C651 B7F70D0D E0FC09D2 3AA2BE7A 8E9FF7DA F32673B4 16349B92

coordinate y_{A1}: 5DC74F8A CC114FC6 F1A75CB2 86864F34 7F9B2CF2 9326A270 79B7D37A FC1C145B

compute $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$:

coordinate x_V: 47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905

coordinate y_V: 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295

compute $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$:

$x_V \parallel y_V \parallel Z_A \parallel Z_B$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 2AF86EFE 732CF12A D0E09A1F

2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66

8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

$klen = 128$

shared secret key K_B: 55B0AC62 A6B927BA 23703832 C853DED4

compute optional term $S_B = Hash(0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 E4D1D0C3 CA4C7F11 BC8FF8CB

3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC

336E6656 119ABD67 6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0 0D6FCF62

F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A 1799B2A2 C7782953 00D9A232 5C686129

B8F2B533 7B3DCF45 14E8BBC1 9D900EE5 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91

D0DB3CF4

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0x02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 FF49D95B D45FCE99

ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

optional term S_B: 284C8F19 8F141B50 2E81250F 1581C7E9 EEB4CA69 90F9E02D F388B454 71F5BC5C

Related values in steps A4-A10 in the key exchange protocol:

take $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: E856C095 05324A6D 23150C40 8F162BF0

compute $t_A = (d_A + \bar{x}_1 \cdot r_A)\, mod\, n$:

236CF0C7 A177C65C 7D55E12D 361F7A6C 174A7869 8AC099C0 874AD065 8A4743DC

take $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: B8F2B533 7B3DCF45 14E8BBC1 9D900EE5

compute point $[\bar{x}_2]R_B = (x_{B0}, y_{B0})$ of the elliptic curve:

coordinate x_{B0}: 66864274 6BFC066A 1E731ECF FF51131B DC81CF60 9701CB8C 657B25BF 55B7015D

coordinate y_{B0}: 1988A7C6 81CE1B50 9AC69F49 D72AE60E 8B71DB6C E087AF84 99FEEF4C CD523064

compute point $P_B + [\bar{x}_2]R_B = (x_{B1}, y_{B1})$ of the elliptic cuve:

coordinate x_{B1}: 7D2B4435 10886AD7 CA3911CF 2019EC07 078AFF11 6E0FC409 A9F75A39 01F306CD

coordinate y_{B1}: 331F0C6C 0FE08D40 5FFEDB30 7BC255D6 8198653B DCA68B9C BA100E73 197E5D24

compute $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$:

coordinate x_U: 47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905

coordinate y_U: 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295

compute $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B, klen)$:

$x_U \parallel y_U \parallel Z_A \parallel Z_B$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67

$klen = 128$

shared secret key K_A: 55B0AC62 A6B927BA 23703832 C853DED4

compute optional term $S_1 = Hash(0\text{x}02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67 6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0 0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0\text{x}02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

optional term S_1: 284C8F19 8F141B50 2E81250F 1581C7E9 EEB4CA69 90F9E02D F388B454 71F5BC5C

compute optional term $S_A = Hash(0\text{x}03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 E4D1D0C3 CA4C7F11 BC8FF8CB 3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC 336E6656 119ABD67 6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0 0D6FCF62 F1036C0A 1B6DACCF 57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45 14E8BBC1 9D900EE5 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0\text{x}03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

03 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

optional term S_A: 23444DAF 8ED75343 66CB901C 84B3BDBB 63504F40 65C1116C 91A4C006 97E6CF7A

Related values in step B10 in the key exchange protocol:

compute optional term $S_2 = Hash(0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

47C82653 4DC2F6F1 FBF28728 DD658F21 E174F481 79ACEF29 00F8B7F5 66E40905 E4D1D0C3 CA4C7F11 BC8FF8CB

3F4C02A7 8F108FA0 98E51A66 8487240F 75E20F31 6B4B6D0E 276691BD 4A11BF72 F4FB501A E309FDAC B72FA6CC

6CB56338 16F4DD56 0B1DEC45 8310CBCC 6856C095 05324A6D 23150C40 8F162BF0 0D6FCF62 F1036C0A 1B6DACCF

57399223 A65F7D7B F2D9637E 5BBBEB85 7961BF1A 1799B2A2 C7782953 00D9A232 5C686129 B8F2B533 7B3DCF45

14E8BBC1 9D900EE5 54C9288C 82733EFD F7808AE7 F27D0E73 2F7C73A7 D9AC98B7 D8740A91 D0DB3CF4

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

FF49D95B D45FCE99 ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

$0x03 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

03 2AF86EFE 732CF12A D0E09A1F 2556CC65 0D9CCCE3 E249866B BB5C6846 A4C4A295 FF49D95B D45FCE99

ED54A8AD 7A709110 9F513944 42916BD1 54D1DE43 79D97647

optional term S_2: 23444DAF 8ED75343 66CB901C 84B3BDBB 63504F40 65C1116C 91A4C006 97E6CF7A

A.3 Key exchange protocol on elliptic curves over F_{2^m}

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

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

cofactor h: 4

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

user A's private key d_A: 4813903D 254F2C20 A94BC570 42384969 54BB5279 F861952E F2C5298E 84D2CEAA

user A's public key $P_A = (x_A, y_A)$:

coordinate x_A: 00 8E3BDB2E 11F91933 88F1F901 CCC857BF 49CFC065 FB38B906 9CAAE6D5 AFC3592F

coordinate y_A: 00 4555122A AC0075F4 2E0A8BBD 2C0665C7 89120DF1 9D77B4E3 EE4712F5 98040415

user B's private key d_B: 08F41BAE 0922F47C 212803FE 681AD52B 9BF28A35 E1CD0EC2 73A2CF81 3E8FD1DC

user B's public key $P_B = (x_B, y_B)$:

coordinate x_B: 00 34297DD8 3AB14D5B 393B6712 F32B2F2E 938D4690 B095424B 89DA880C 52D4A7D9

coordinate y_B: 01 99BBF11A C95A0EA3 4BBD00CA 50B93EC2 4ACB6833 5D20BA5D CFE3B33B DBD2B62D 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: ECF00802 15977B2E 5D6D61B9 8A99442F 03E8803D C39E349F 8DCA5621 A9ACDF2B

hash value $Z_B = H_{256}(ENTL_B \parallel ID_B \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_B \parallel y_B)$:

Z_B: 557BAD30 E183559A EEC3B225 6E1C7C11 F870D22B 165D015A CF9465B0 9B87B527

Related values in steps A1-A3 in the key exchange protocol:

generate random number r_A:

54A3D667 3FF3A6BD 6B02EBB1 64C2A3AF 6D4A4906 229D9BFC E68CC366 A2E64BA4

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

coordinate x_1: 01 81076543 ED19058C 38B313D7 39921D46 B80094D9 61A13673 D4A5CF8C 7159E304

coordinate y_1: 01 D8CFFF7C A27A01A2 E88C1867 3748FDE9 A74C1F9B 45646ECA 0997293C 15C34DD8

Related values in steps B1-B9 in the key exchange protocol:

generate random number r_B:

1F219333 87BEF781 D0A8F7FD 708C5AE0 A56EE3F4 23DBC2FE 5BDF6F06 8C53F7AD

compute point $R_B = [r_B]G = (x_2, y_2)$ of the elliptic curve:

coordinate x_2: 00 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A

coordinate y_2: 00 16599BB5 2ED9EAFA D01CFA45 3CF3052E D60184D2 EECFD42B 52DB7411 0B984C23

take $\bar{x}_2 = 2^{127} + (x_2 \& (2^{127} - 1))$: E8CC43FF A5F2623B 9BD04E46 8D322A2A

compute $t_B = (d_B + \bar{x}_2 \cdot r_B) \, mod \, n$:

3D51D331 14A453A0 5791DB63 5B45F8DB C54686D7 E2212D49 E4A717C6 B10DEDB0

compute $h \cdot t_B \, mod \, n$: 75474CC4 52914E81 5E476D8D 6D17E36F 5882EE67 A1CDBC26 FE4122B0 B741A0A3

take $\bar{x}_1 = 2^{127} + (x_1 \& (2^{127} - 1))$: B80094D9 61A13673 D4A5CF8C 7159E304

compute point $[\bar{x}_1]R_A = (x_{A0}; y_{A0})$ of the elliptic curve:

coordinate x_{A0}: 01 98AB5F14 349B6A46 F77FBFCB DDBFCD34 320DC1F4 C546D13C 3A9F0E83 0C39B579

coordinate y_{A0}: 00 BFB49224 ACCE2E51 04CD4519 C0CBE3AD 0C19BF11 805BE108 59069AA6 9317A2B7

compute point $P_A + [\bar{x}_1]R_A = (x_{A1}, y_{A1})$ of the elliptic curve:

coordinate x_{A1}: 00 24A92F64 66A37C5C 12A2C68D 58BFB0F0 32F2B976 60957CB0 5E63F961 F160FE57

coordinate y_{A1}: 00 F74A4F17 DC560A55 FDE0F1AB 168BCBF7 6502E240 BA2D6BD6 BE6E5D79 16B288FC

compute $V = [h \cdot t_B](P_A + [\bar{x}_1]R_A) = (x_V, y_V)$:

coordinate x_V: 00 DADD0874 06221D65 7BC3FA79 FF329BB0 22E9CB7D DFCFCCFE 277BE8CD 4AE9B954

coordinate y_V: 01 F0464B1E 81684E5E D6EF281B 55624EF4 6CAA3B2D 37484372 D91610B6 98252CC9

compute $K_B = KDF(x_V \parallel y_V \parallel Z_A \parallel Z_B, klen)$:

$x_V \parallel y_V \parallel Z_A \parallel Z_B$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 5401F046 4B1E8168 4E5ED6EF 281B5562 4EF46CAA 3B2D3748 4372D916 10B69825 2CC9ECF0 08021597 7B2E5D6D 61B98A99 442F03E8 803DC39E 349F8DCA 5621A9AC DF2B557B AD30E183 559AEEC3 B2256E1C 7C11F870 D22B165D 015ACF94 65B09B87 B527

$klen = 128$

shared secret key K_B: 4E587E5C 66634F22 D973A7D9 8BF8BE23

compute optional term $S_B = Hash(0\text{x}02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 54ECF008 0215977B 2E5D6D61 B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF 2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01 5ACF9465 B09B87B5 27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159 E30401D8 CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15 C34DD800 2A4832B4 DCD399BA AB3FFFE7 DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4 2B52DB74 110B984C 23

$Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0\text{x}02 \parallel y_V \parallel Hash(x_V \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C C9E05FE2 87B73B0C E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

optional term S_B: 4EB47D28 AD3906D6 244D01E0 F6AEC73B 0B51DE15 74C13798 184E4833 DBAE295A

Related values in steps A4-A10 in the key exchange protocol:

take $\bar{x}_1 = 2^{127} + (x_1\&(2^{127} - 1))$: B80094D9 61A13673 D4A5CF8C 7159E304

compute $t_A = (d_A + \bar{x}_1 \cdot r_A)\ mod\ n$:

18A1C649 B94044DF 16DC8634 993F1A4A EE3F6426 DFE14AC1 3644306A A5A94187

compute $h \cdot t_A\ mod\ n$: 62871926 E501137C 5B7218D2 64FC692B B8FD909B 7F852B04 D910C1AA 96A5061C

take $\bar{x}_2 = 2^{127} + (x_2\&(2^{127} - 1))$: E8CC43FF A5F2623B 9BD04E46 8D322A2A

compute point $[\bar{x}_2]R_B = (x_{B0}, y_{B0})$ of the elliptic curve

coordinate x_{B0}: 01 0AA3BAC9 7786B629 22F93414 57AC64F7 2552AA15 D9321677 A10C7021 33B16735

coordinate y_{B0}: 00 C10837F4 8F53C46B 714BCFBF AA1AD627 11FCB03C 0C25B366 BF176A2D C7B8E62E

compute point $P_B + [\bar{x}_2]R_B = (x_{B1}, y_{B1})$ of the elliptic curve:

coordinate x_{B1}: 00 C7A446E1 98DB4278 60C3BB50 ED2197DE B8161973 9141CA61 03745035 9FAD9A99

coordinate y_{B1}: 00 602E5A42 17427EAB C5E3917D E81BFFA1 D806591A F949DD7C 97EF90FD 4CF0A42D

compute $U = [h \cdot t_A](P_B + [\bar{x}_2]R_B) = (x_U, y_U)$:

coordinate x_U: 00 DADD0874 06221D65 7BC3FA79 FF329BB0 22E9CB7D DFCFCCFE 277BE8CD 4AE9B954 coordinate y_U: 01 F0464B1E 81684E5E D6EF281B 55624EF4 6CAA3B2D 37484372 D91610B6 98252CC9

compute $K_A = KDF(x_U \parallel y_U \parallel Z_A \parallel Z_B; klen)$:

$x_U \parallel y_U \parallel Z_A \parallel Z_B$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 5401F046 4B1E8168 4E5ED6EF

281B5562 4EF46CAA 3B2D3748 4372D916 10B69825 2CC9ECF0 08021597 7B2E5D6D 61B98A99 442F03E8 803DC39E

349F8DCA 5621A9AC DF2B557B AD30E183 559AEEC3 B2256E1C 7C11F870 D22B165D 015ACF94 65B09B87 B527

$klen = 128$

shared secret key K_A: 4E587E5C 66634F22 D973A7D9 8BF8BE23

compute optional term $S_1 = Hash(0\text{x}02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 54ECF008 0215977B 2E5D6D61

B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF 2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01

5ACF9465 B09B87B5 27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159 E30401D8

CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15 C34DD800 2A4832B4 DCD399BA AB3FFFE7

DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4

2B52DB74 110B984C 23

$Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0\text{x}02 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

02 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C C9E05FE2 87B73B0C

E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

optional term S_1: 4EB47D28 AD3906D6 244D01E0 F6AEC73B 0B51DE15 74C13798 184E4833 DBAE295A

compute optional term $S_A = Hash(0\text{x}03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2))$:

$x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 54ECF008 0215977B 2E5D6D61

B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF 2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01

5ACF9465 B09B87B5 27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159 E30401D8

CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15 C34DD800 2A4832B4 DCD399BA AB3FFFE7

DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4

2B52DB74 110B984C 23

$Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0\text{x}03 \parallel y_U \parallel Hash(x_U \parallel Z_A \parallel Z_B \parallel x_1 \parallel y_1 \parallel x_2 \parallel y_2)$:

03 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C C9E05FE2 87B73B0C

E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

optional term S_A: 588AA670 64F24DC2 7CCAA1FA B7E27DFF 811D500A D7EF2FB8 F69DDF48 CC0FECB7

Related values in step B10 in the key exchange protocol:

compute optional term $S_2 = Hash(0x03 \| y_V \| Hash(x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2))$:

$x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2$:

00DADD08 7406221D 657BC3FA 79FF329B B022E9CB 7DDFCFCC FE277BE8 CD4AE9B9 54ECF008 0215977B 2E5D6D61

B98A9944 2F03E880 3DC39E34 9F8DCA56 21A9ACDF 2B557BAD 30E18355 9AEEC3B2 256E1C7C 11F870D2 2B165D01

5ACF9465 B09B87B5 27018107 6543ED19 058C38B3 13D73992 1D46B800 94D961A1 3673D4A5 CF8C7159 E30401D8

CFFF7CA2 7A01A2E8 8C186737 48FDE9A7 4C1F9B45 646ECA09 97293C15 C34DD800 2A4832B4 DCD399BA AB3FFFE7

DD6CE6ED 68CC43FF A5F2623B 9BD04E46 8D322A2A 0016599B B52ED9EA FAD01CFA 453CF305 2ED60184 D2EECFD4

2B52DB74 110B984C 23

$Hash(x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2)$:

E05FE287 B73B0CE6 639524CD 86694311 562914F4 F6A34241 01D885F8 8B05369C

$0x03 \| y_V \| Hash(x_V \| Z_A \| Z_B \| x_1 \| y_1 \| x_2 \| y_2)$:

03 01F0464B 1E81684E 5ED6EF28 1B55624E F46CAA3B 2D374843 72D91610 B698252C C9E05FE2 87B73B0C

E6639524 CD866943 11562914 F4F6A342 4101D885 F88B0536 9C

optional term S_2: 588AA670 64F24DC2 7CCAA1FA B7E27DFF 811D500A D7EF2FB8 F69DDF48 CC0FECB7