IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Changes to Hash-to-Curve | | | | |
| Date: 2019-10-31 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Dan Harkins | HPE | 3333 Scott boulevard  Santa Clara, California,  United States of America | +1 408 555 1212 |  |

Abstract

REVmd uses an algorithm from an Internet-Draft to convert an arbitrary string into a point on an elliptic cufve. One of the parameters in this algorithm has changed and criteria for selecting it has also changed. This submission updates REVmd to use the new selection criteria and to indicate the parameter values for popular elliptic curves.

**Discussion**: We can’t make reference to an Internet-Draft (beyond “a work in progress”) because they change so we had to copy the relevant information out of it into REVmd. Well that Internet-Draft changed. The criteria to select the curve-specific constant z has changed and therefore the constant for each curve has also changed. We need to update REVmd to reflect this change and be consistent with the Internet-Draft.

**Proposal**: Include the new z selection criteria and update the table that lists z for popular curves.

*Instruct the editor to modify section 12.4.4.2.3 as indicated:*

**12.4.4.2.3 Hash-to-curve generation of the password element with ECC groups**

The SSWU method takes a curve-specific parameter, z, which is determined by finding the number that satisfies the following rules, given p, a, and b, from the curve’s domain parameter set:

1. z = n if
   1. n is not a quadratic residue modulo p
   2. n is not -1
   3. the polynomial x3 + a \* x + b – n is irreducible
   4. (b/(n\*a)3 + a \* (b/(n\*a)) + b is a quadratic residue modulo p
2. z = -n if
   1. -n is not a quadratic residue modulo p
   2. -n is not -1
   3. the polynomial x3 + a \* x + b + n is irreducible
   4. (b/(-n\*a)3 + a \* (b/(-n\*a)) + b is a quadratic residue modulo p
3. there is no other number n’ that satisfies the above two criteria such that the absolute value of n’ is less than the absolute value of n; and
4. if both n and -n satisfy the above criteria then z is n, the positive value

Values for some defined groups based on their IANA-assigned values are listed in Table 12-2:

**Table 12-2—Unique curve parameter**

|  |  |  |
| --- | --- | --- |
| **Curve name** | **IANA value** | **z** |
| NIST p256 | 19 | -10 |
| NIST p384 | 20 | -12 |
| NIST p521 | 21 | -4 |
| NIST p192 | 25 | -5 |
| NIST p224 | 26 | 31 |
| Brainpool p256 | 28 | -2 |
| Brainpool p384 | 29 | -5 |
| Brainpool p512 | 30 | 7 |

*Instruct the editor to replace the hash-to-curve portion of J.10 with the following:*

Hash to Curve technique of PT/PWE Generation

group 19

SSID: byteme

identifier: psk4internet

password: mekmitasdigoat

MAC address 1: 00:09:5b:66:ec:1e

MAC address 2: 00:0b:6b:d9:02:46

-----------------------------------------------------------------------

calculating PT for group 19

salt:

62797465 6d65

ikm:

6d656b6d 69746173 6469676f 61747073 6b34696e 7465726e 6574

prk:

3bd53fe9 223dc028 0fbfce17 d7a35640 64e20f48 c6ec7224 6ce367b5 569a22af

info:

53414520 48617368 20746f20 456c656d 656e7420 75312050 31

okm:

a5044469 ab16f25b 6abf1e0e 37a36b56 f50be733 69053df8 db87989a 6b66fd1a

491f1cda cbd07931 620f8300 8ffc0ecc

SSWU(u1):

---------

u1:

dc941bc3 c6a2b494 8b6c61d5 5590ecb1 f0c51c4b 1bebaff6 77e59369 8d5a53c6

m = z^2 \* u^4 + z \* u^2:

7f655b40 2e73946f 02101f64 a56dc14e 691d8808 689c6cc5 4a3347c7 ad082e2d

t = 1/m:

f07ea718 98e90f41 1064aa04 0469609d 2246579b 1b986ffc 0958f26a ecdb2dcf

(-b/a) \* (1 + t):

a07c2607 64a13445 ff8cd97c 5acc644e 7119bde5 1bad4258 3eed6f41 09639e6b

b/(z \* a):

a528bd86 96bdaf99 6c65b982 d94959d3 146fe6a0 20693090 bdba1313 2375f224

m != 0, so x1 = (-b/a)\*(1+t)

x1:

a07c2607 64a13445 ff8cd97c 5acc644e 7119bde5 1bad4258 3eed6f41 09639e6b

gx1:

b21f71a7 645b281a c384ae40 9d4d56dc 919134fb c11ac0ac a3c4c0d6 8607d8b0

x2:

bd07b38a 812ed2cc 8472cdca 4beee58a bd55f40b 17216097 bc2b547d a12a692e

gx2:

8d8ba5d5 6b6fea0b 7087af73 0bb1fca5 092ba140 d0d7777a f9f07c26 815794fb

gx1 is a quadratic residue

gx2 is not a quadratic residue

point P1...

P1.x:

a07c2607 64a13445 ff8cd97c 5acc644e 7119bde5 1bad4258 3eed6f41 09639e6b

P1.y:

3bdc8df0 d3233793 6c74df60 4933a454 142251c5 3c576c03 51b28dea f9428d7e

-----------------------------------

salt:

62797465 6d65

ikm:

6d656b6d 69746173 6469676f 61747073 6b34696e 7465726e 6574

prk:

3bd53fe9 223dc028 0fbfce17 d7a35640 64e20f48 c6ec7224 6ce367b5 569a22af

info:

53414520 48617368 20746f20 456c656d 656e7420 75322050 32

okm:

9b4e0d5b 1879f253 c5319615 099b05ae c5b06fa5 e788bcfd 1e9ea60d 33436927

190814c3 22a62585 c93c577b baa3d307

SSWU(u2):

---------

u2:

1b8375a5 18bc2139 6ad6a65e 5597e0bf 80d793b6 d66e2534 a6e7dfe3 ee22616f

m = z^2 \* u^4 + z \* u^2:

5b17c6b0 531ae0ab 1a346116 7f2c309e 731fe072 aa5043e0 307d71dd fc0983be

t = 1/m:

8c7200be e794cc45 f7d1c0a1 4843b641 40366f7e 42e6fbd3 49b92320 4ebe4615

(-b/a) \* (1 + t):

72cd2a96 7a837fea 5051f013 3db46227 775ba09f 7b6dfb99 ae7a8ef2 2c7d34a0

b/(z \* a):

a528bd86 96bdaf99 6c65b982 d94959d3 146fe6a0 20693090 bdba1313 2375f224

m != 0, so x1 = (-b/a)\*(1+t)

x1:

72cd2a96 7a837fea 5051f013 3db46227 775ba09f 7b6dfb99 ae7a8ef2 2c7d34a0

gx1:

ad50cf4c 878e2a40 a3f9bc5a 85137fef 7ff5887a 2d46a23e 3a534f99 543fa53d

x2:

e7c2d3f3 580f1c04 5c7e5f06 8fcfc03d 2a7dcd32 bb42166f 95ff5c04 04109782

gx2:

7578d68b d9c60233 508ab8a1 90313ae4 d0215f41 d1a3646f be639434 a8710d55

gx1 is a quadratic residue

gx2 is not a quadratic residue

point P2...

P2.x:

72cd2a96 7a837fea 5051f013 3db46227 775ba09f 7b6dfb99 ae7a8ef2 2c7d34a0

P2.y:

864390d7 97d352b3 68d311af 515bde11 6fe54459 fec867ee 18a8a161 9ca3ff59

-----------------------------------

PT = P1 + P2:

x:

b6e38c98 750c684b 5d17c3d8 c9a4100b 39931279 187ca6cc ed5f37ef 46ddfa97

y:

5687e972 e50f73e3 898861e7 edad21be a7d5f622 df88243b b804920a e8e647fa

-----------------------------------

val:

bb7f9cac 5aa8b72c 02b5daac c2771abe 74e72604 612295ec a2ce1836 3ae9a927

PWE = val \* PT:

x:

c93049b9 e64000f8 48201649 e999f2b5 c22dea69 b5632c9d f4d633b8 aa1f6c1e

y:

73634e94 b53d82e7 383a8d25 8199d9dc 1a5ee826 9d060382 ccbf33e6 14ff59a0

**References:**

Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R., Wood, C., “Hashing to Elliptic Curves”, draft-irtf-cfrg-hash-to-curve, A work in progress, November 2019