IEEE P1363.2 Submission / D2001-06-21 (draft)

Standard Specifications for Public Key Cryptography: Password-based Techniques

Abstract. This document contains possible additions to IEEE P1363.2/D2001-05-14 (rough draft), namely, inclusion of an elliptic curve group based SRP protocol.

Contents

4.6 Table Summary

This section gives a summary of all the schemes and protocols in this submission, together with the primitives and additional methods that are invoked within a scheme or protocol.

Scheme or Protocol	Primitives	Additional Methods
password-based key agreement		
ATPKAS-ECSRP-A	ECEDP and ECPEPKGP-SRP-A and ECPESVDP-SRP-A and ECPVDGP-SRP	
ATPKAS-ECSRP-B	ECEDP and ECPEPKGP-SRP-B and ECPESVDP-SRP-B and ECPVDGP-SRP	

7. Primitives Based on the Elliptic Curve Discrete Logarithm Problem

7.2 Primitives

SUBMISSION NOTE—(Here should be the generic introduction for all EC-schemes).

In this section we use *E* to denote the elliptic curve over the field *GF*(*q*) defined by *a* and *b*.

7.2.1 ECEDP

ECEDP is Elliptic Curve Element Derivation Primitive. The primitive uses a seed input value to derive an element of the elliptic curve group to be used in a discrete logarithm password-based authenticated key agreement scheme. It is based on the work of [1]. It can be invoked to construct the values in the primitives ECPESVDP-A and ECPEPKGP-B.

Input:
— the EC domain parameters *q*, *a*, *b*, *r*, *k* and *G*
— the seed octet string *t*.

Assumptions: EC domain parameters *q*, *a*, *b*, *r*, *k* and *G* are valid

Output:
— the group element *Q*, which is a point on *E* of order *r*.

Operation. The group element *Q* is computed by the following or an equivalent sequence of steps:
1. Compute a field element x=OS2FEP(t) of $GF(q)$.
2. Derive from x an elliptic curve point $P=(x_P,y_P)$ by the following or an equivalent sequence of steps:
   2.1 Set $x_P=x$.
   2.2 If $q=p$ is an odd prime, compute the field element $\alpha=x_P^3+a.x_P+b$ (mod p), and compute a square root β of α modulo p. Go to Step 2.4 if there are no square roots of α modulo p. Otherwise set $y_P=\beta$ if $\beta\equiv x_P \pmod 2$, and set $y_P=p-\beta$ if $\beta\equiv x_P+1 \pmod 2$. Go to Step 2.5.
   2.3 If $q=2^m$, then go to Step 2.1 if $x_P=0$. Otherwise compute the field element $\beta=x_P+a+b.x_P^{-2}$ in $GF(q)$, and find a field element $z=z_{m-1}x^{m-1}+\ldots+z_1x+z_0$ such that $z^2+z=\beta$ in $GF(q)$. Go to Step 2.4 if no such z exists. Otherwise set $y_P=x_P.z$ in $GF(q)$ if $z_0\equiv x_0$, and set $y_P=x_P.(z+1)$ in $GF(q)$ if $z_0\equiv 1-x_0$. Go to Step 2.5.
   2.4 Set $x_P=x_P+1$ in $GF(q)$ and go to Step 2.2.
   2.5 Set $Q=kP$. If $Q=O$, then go to Step 2.4, otherwise go to Step 3.
3. Output Q.

Conformance region recommendation. A conformance region should include:
— at least one valid set of EC domain parameters *q*, *a*, *b*, *r*, *k* and *G*.

7.2.4 ECPVDGP-SRP

ECPVDGP-SRP is Elliptic Curve Password Validation Data Generation Primitive, SRP version. This primitive derives a password validation data from a party identification string and password, using EC domain parameters.

Input:
- the party's own password-derived octet string π
- the party's salt string *salt*.
- the party's identification string ID.
- the EC domain parameters *q*, *a*, *b*, *r*, *k* and *G*.
- the hash function *hash*.

Assumptions: Salt string *salt*, password value π, and identification string ID is bound to the user; EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* are valid

Output: The derived password verification data γ, which is a point on *E* of order *r*.

Operation: The password verification data γ shall be computed by the following or an equivalent sequence of steps:

1. Compute an integer i=OS2IP(*hash*(*salt*||*hash*(ID||:||π)).
2. Compute the elliptic curve point $\gamma=iG$.
3. Output γ.

7.2.5 ECPEPKGP-SRP-A

ECPEPKGP-SRP-A is Elliptic Curve Password-Entangled Public Key Generation Primitive, SRP version for Alice. This primitive derives a password-entangled public key from the party's ephemeral private key, using EC domain parameters.

This primitive may be invoked by a scheme to generate an ephemeral public key for Alice; specifically, it may be used with the scheme ATPKAS-ECSRP-A. It assumes that the input keys are valid.

Input:
- the party's private key *s*.
- the EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* associated with the key *s*.

Assumptions: Private key *s*, EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* are valid

Output: The derived password-entangled public key value *w*, which is a point on *E* of order *r*.

Operation: The password-entangled public key value *w* shall be computed by the following or an equivalent sequence of steps:

1. Compute $w=sG$.
2. If $w=O$, output "invalid private key".
3. Output *w*.

7.2.6 ECPESVDP-SRP-A

ECPESVDP-SRP-A is Elliptic Curve Password-Entangled Secret Value Derivation Primitive, SRP version for Alice. It is based on the work of [1]. This primitive derives a shared secret value from the party's password, ephemeral private key, and the other party's ephemeral public key, where the keys have the same EC domain parameters. If two parties correctly execute this primitive with corresponding keys as inputs, they will produce the same output. This primitive may be invoked by a scheme to derive a shared secret key; specifically, it may be used with the scheme ECPEKAS-SRP-A. It assumes that the input keys are valid.

Input:
- the party's identification string ID, salt string *salt*, private key *s* and password π
- the other party's EC public key *w*'
- the EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* associated with the keys *s* and *w*'
- the hash function *hash*
- the elliptic curve element derivation primitive ECEDP.

Assumptions: Identification string ID, salt string *salt*, password π, private key *s*, public key *w'*, EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* are valid.

Output: The derived shared secret value, which is a point on *E* of order *r*.

Operation: The shared secret value *z* shall be computed by the following or an equivalent sequence of steps:

1. If *w'* = *O* output "invalid public key".
2. Compute an integer *i*=OS2IP(*hash*(*salt*||*hash*(ID||:||*π*)).
3. Compute the password validation data γ=ECPVDGP-SRP(*salt*, ID, *π*).
4. Let x_γ be the *x*-coordinate of γ.
5. Compute w_1=*w'*-ECEDP(*hash*(FE2OSP(x_γ))).
6. Let *ustring* be the octet string consisting of the first four octets of *hash*(FE2OSP(x_w)), from left to right.
7. Compute an integer *j*=OS2IP(*ustring*).
8. Let *z*=(*s*+*i.j*)w_1.
9. Output *z*.

7.2.7 ECPEPKGP-SRP-B

ECPEPKGP-SRP-B is Elliptic Curve Password-Entangled Public Key Generation Primitive, SRP version for Bob. It is based on the work of [1]. This primitive derives a password-entangled public key from the party's ephemeral private key and the other party's password validation data, using EC domain parameters.

This primitive may be invoked by a scheme to derive an ephemeral public key for Bob; specifically, it may be used with the scheme ATPKAS-ECSRP-B. It assumes that the input keys are valid.

Input:
- the other party's password validation data γ.
- the party's own private key *s*.
- the EC domain parameters, *q*, *a*, *b*, *r*, *k*, and *G* associated with the keys *s* and γ.
- the hash function *hash*.
- the elliptic curve element derivation primitive ECEDP.

Assumptions: Private key *s*, password validation data γ, EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* are valid

Output: The derived password-entangled public key value *w*, which is a point on *E* of order *r*.

Operation: The password-entangled public key value *w* shall be computed by the following or an equivalent sequence of steps:

1. Let x_γ be the *x*-coordinate of γ.
2. Compute *w*=*sG*+ECEDP(*hash*(FE2OSP(x_γ))).
3. Output *w*.

7.2.8 ECPESVDP-SRP-B

ECPESVDP-SRP-B is Elliptic Curve Password-Entangled Secret Value Derivation Primitive, SRP version for Bob. It is based on the work of [1]. This primitive derives a shared secret value from the party's private key, the other party's public key and password validation data, where the keys have the same EC domain parameters. If two parties correctly execute this primitive with corresponding keys as inputs, they

will produce the same output. This primitive may be invoked by a scheme to derive a shared secret key; specifically, it may be used with the scheme ECPEKAS-SRP-B. It assumes that the input keys are valid.

Input:
- the party's own private key *s*.
- the other party's EC public key *w*' and password validation data γ.
- the EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* associated with the keys *s*, γ, and *w*'.

Assumptions: Private key *s*, EC domain parameters *q*, *a*, *b*, *r*, *k*, and *G* are valid, *w*' is a point on *E* of order *r*, and γ is valid

Output: The derived shared value, which is a point on *E* of order *r*.

Operation: The shared secret value *z* shall be computed by the following or an equivalent sequence of steps:

1. If $w' = O$ output "invalid public key".
2. Compute w=ECPEPKGP-SRP-B(s,γ), and let x_w be the x-coordinate of w.
3. Let *ustring* be the octet string consisting of the first four octets of *hash*(FE2OSP(x_w)), from left to right.
4. Compute an integer j=OS2IP(*ustring*).
5. Compute $z=s(w'+j\gamma)$.
6. Output *z*.

9. Password-Authenticated Key Agreement Schemes

This section will be the same as the original SRP protocol.

9.4 ATPKAS-ECSRP-A

9.5 ATPKAS-ECSRP-B

10. Reference

1. Yongge Wang. Password-based protocols, Certicom Technical Report, 2000.

11. The difference between ECSRP and SRP

The only difference between ECSRP and SRP is the Step 2 in the Operation of ECPEPKGP-SRP-B, and the corresponding Step 5 in the Operation of ECPESVDP-SRP-A. All other places should be kept the same as the original SRP protocol.

The notations in this submission are subject to change for compliance with the whole standard.

12. Intellectual property issues

See Certicom Submission to IEEE P1363.