Elliptic Curve Abe-Okamoto-Suzuki Message Recovery Signature Scheme (ECAOS) Working Draft 20080825

August 25, 2008

1 Introduction

This section gives an overview of this standard, its use, its aims, and its development.

1.1 Overview

This document specifies public-key cryptographic schemes based on elliptic curve cryptography (ECC). In particular, it specifies signature scheme.

It also describes cryptographic primitives which are used to construct the schemes, and ASN.1 syntax for identifying the schemes.

The schemes are intended for general application within computer and communications systems.

1.2 Aim

The aim of this document is threefold:

- Firstly, to facilitate deployment of ECC by completely specifying efficient, well-established, and well-understood public-key cryptographic schemes based on ECC.
- Secondly, to encourage deployment of interoperable implementations of ECC by profiling standards such as ANSI X9.62 [X9.62a], WAP WTLS [WTLS], ANSI X9.63 [X9.63] and IEEE 1363 [1363], and recommendation NIST SP 800-56 [800-56A], but restricting the options allowed in these standards to increase the likelihood of interoperability and to ensure conformance with as many standards as possible.
- Thirdly, to help ensure ongoing detailed analysis of ECC by cryptographers by clearly, completely, and publicly specifying baseline techniques.

1.3 Compliance

Implementations may claim compliance with the cryptographic schemes specified in this document provide the external interface (input and output) to the schemes is equivalent to the interface specified here. Internal computations may be performed as specified here, or may be performed via an equivalent sequence of operations.

Note that this compliance definition implies that conformant implementations must perform all the cryptographic checks included in the scheme specifications in this document. This is important because the checks are essential to the prevention of subtle attacks.

It is intended to make a validation system available so that implementors can check compliance with this document — see the SECG website, `http://www.secg.org`, for further information.

1.4 Document Evolution

This document will be reviewed every five years to ensure it remains up to date with cryptographic advances. The next scheduled review will therefore take place in November, 2013.

This current draft, version 0.9.0, is a work in progress and is subject to change without notice. Its purpose is review at the annual SECG meeting in November, 2008.

Additional intermittent reviews may also be performed occasionally, as deemed necessary by the Standards for Efficiency Cryptography Group.

1.5 Intellectual Property

The reader's attention is called to the possibility that compliance with this document may require use of an invention covered by patent rights. By publication of this document, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder(s) may have filed with the SECG a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Additional details may be obtained from the patent holder and from the SECG website, `http://www.secg.org`.

1.6 Organization

This document is organized as follows.

The main body of the document focuses on the specification of signature scheme based on ECC. Section 2 describes the Elliptic Curve Abe-Okamoto-Suzuki Message Recovery Signature Scheme (ECAOS).

The appendices to the document provide additional relevant material. Appendix A provides recommendation for mask generation function. Appendix B provides requirement for domain parameters. Appendix C provides recommendation for domain parameters. Appendix D provides test vectors.

2 ECAOS Message Recovery Signature Scheme

The Elliptic Curve Abe-Okamoto-Suzuki Message Recovery Signature Scheme (ECAOS) [2] is a message recovery signature scheme based on ECC. It is designed to be existentially unforgeable, even in the presence of an adversary capable of launching chosen-message attacks.

The setup procedure for ECAOS is specified in Section 2.1, the key deployment procedure is specified in Section 2.2, the signing operation is specified in Section 2.3, and the verifying operation is specified in Section 2.4.

2.1 Scheme Setup

The domain parameters of ECAOS consist of the elliptic curve domain parameters and the octet length domain parameters described below.

The elliptic curve domain parameters T of ECAOS are described below:

- $T = (p, a, b, G, n, h)$ or $(m, f(x), a, b, G, n, h)$ or $(p, m, f(x), a, b, G, n, h)$, where $\langle G \rangle$ is an additive cyclic group of prime order n, with cofactor h, with a base point G, in an elliptic curve E with coefficients a, b, over an finite field F explicitly given by characteristic p, extension degree m, and defining polynomial $f(x)$.

These elliptic curve domain parameters are specified in section 3.1 of SEC1[3] and section ?? of SEC6[4].

The octet length domain parameters of ECAOS are described below:

- L_n: octet length of prime integer n,
- L_F: octet length of an element of finite field F,
- K: octet length of the extended part of a mask string,
- L_{red}: octet length of added redundancy, and
- L_{rec}^{min}: octet length of minimum recovable message.

These length domain parameters are required to satisfy the following conditions w.r.t. security parameter $\kappa > 0$ to ensure 8κ-bit security: $L_n \geq 2\kappa$, $L_F \geq 2\kappa$, $K \geq \kappa$, $L_{red} \geq \kappa$, and $L_{red} + L_{rec}^{min} \geq 2\kappa$.

L_{red} and L_{rec}^{min} are required to satisfy the following conditions because of the maximum output octet length L_{MGF}^{out} of mask generation function MGF specified in appendix A: $L_{red} \leq L_{\mathrm{MGF}}^{out}$ and $L_{rec}^{min} \leq L_{\mathrm{MGF}}^{out}$.

ECAOS uses mask generation function MGF specified in appendix A.

ECAOS uses conversion functions OS2IP and I2OSP specified in section 2.3 of SEC1[3], EC2OSP specified in section 2.3 of SEC1[3], and EC2OSP for OEF specified in section ?? of SEC6[4].

We denote by $L(s)$ octet length of octet string s.

2.2 Key Deployment

ECAOS key generation algorithm $KeyGen$ is defined as follows:
Input:

- elliptic curve domain parameters T.

Output:

- private key $x \in [1, n-1]$ and
- public key $Y \in \langle G \rangle$.

Actions: ECAOS key generation algorithm $KeyGen$ creates keys as follows:

1. select random integer $x \in [1, n-1]$,
2. compute $Y = xG \in \langle G \rangle$, and
3. output private key $x \in [1, n-1]$ and public key $Y \in \langle G \rangle$.

2.3 Signing Operation

ECAOS signing algorithm Sig takes as input octet strings M_{rec} of octet length $0 \leq L(M_{rec}) \leq min(L^{in}_{\text{MGF}} - (L_F + 11), L^{out}_{\text{MGF}} - 1)$ and M_{clr} of octet length $0 \leq L(M_{clr}) \leq L^{in}_{\text{MGF}} - (L_{red} + \widetilde{L}_{rec} + 1)$, and is defined as follows:
Input:

- domain parameters,
- private key $x \in [1, n-1]$,
- recoverable message $M_{rec} \in \{0,1\}^{8*}$, and
- non-recoverable message $M_{clr} \in \{0,1\}^{8*}$.

Output:

- signature $(r, s) \in \{0,1\}^{8(L_{red}+\widetilde{L}_{rec})} \times [1, n-1]$.

Actions: ECAOS signing algorithm Sig creates signature as follows:

1. compute $L_{rec} = L(M_{rec})$ and $L_{clr} = L(M_{clr})$,
2. compute $\widetilde{L}_{rec} = \max(L^{min}_{rec}, L_{rec}+1)$ and $\widetilde{M}_{rec} = \text{I2OSP}(1, \widetilde{L}_{rec} - L_{rec}) || M_{rec} \in \{0,1\}^{8\widetilde{L}_{rec}}$,
3. compute $L'_{rec} = \text{I2OSP}(L_{rec}, 8) \in \{0,1\}^{64}$,
4. select random integer $k \in [1, n-1]$,
5. compute $R = kG \in \langle G \rangle$,

6. compute $R' = \mathrm{EC2OSP}_E(R, compressed) \in \{0,1\}^{8(L_F+1)}$,
7. compute $h_0 = \mathrm{MGF}(\widetilde{M}_{rec}||L'_{rec}||R'||\mathrm{I2OSP}(0,1), L_{red}) \in \{0,1\}^{8L_{red}}$,
8. compute $h_1 = \mathrm{MGF}(h_0||R'||\mathrm{I2OSP}(1,1), \widetilde{L}_{rec}) \in \{0,1\}^{8\widetilde{L}_{rec}}$,
9. compute $r = h_0||(\widetilde{M}_{rec} \oplus h_1) \in \{0,1\}^{8(L_{red}+\widetilde{L}_{rec})}$,
10. compute $u = \mathrm{MGF}(M_{clr}||r||\mathrm{I2OSP}(2,1), L_n + K) \in \{0,1\}^{8(L_n+K)}$,
11. compute $t = \mathrm{OS2IP}(u) \mod n \in [0, n-1]$,
12. if $t = 0$, goto step 4,
13. compute random integer $s = k - xt \mod n \in [0, n-1]$,
14. if $s = 0$, goto step 4, and
15. output signature $(r, s) \in \{0,1\}^{8(L_{red}+\widetilde{L}_{rec})} \times [1, n-1]$.

2.4 Verifying Operation

ECAOS verification algorithm Ver takes as input octet string r and integer s and octet string M_{clr}, and is defined as follows:
Input:

- domain parameters,
- public key $Y \in \langle G \rangle$,
- signature $(r, s) \in \{0,1\}^{8*} \times \mathbb{Z}$, and
- non-recoverable message $M_{clr} \in \{0,1\}^{8*}$.

Output:

- recoverable message $M_{rec} \in \{0,1\}^{8L_{rec}}$ or 'invalid'.

Actions: ECAOS verification algorithm Ver checks signature as follows:

1. compute $L_r = L(r)$ and $L_{clr} = L(M_{clr})$,
2. check $L_{red} + L_{rec}^{min} \le L_r \le L_{red} + min(L_{\mathrm{MGF}}^{in} - (L_F + 10), L_{\mathrm{MGF}}^{out})$, $1 \le s \le n-1$, and $0 \le L_{clr} \le L_{\mathrm{MGF}}^{in} - (L_{red} + \widetilde{L}_{rec} + 1)$,
3. if it does not hold, output 'invalid' and stop,
4. compute $u = \mathrm{MGF}(M_{clr}||r||\mathrm{I2OSP}(2,1), L_n + K) \in \{0,1\}^{8(L_n+K)}$,
5. compute $t = \mathrm{OS2IP}(u) \mod n \in [0, n-1]$,
6. if $t = 0$, output 'invalid' and stop,

7. compute $R = sG + tY \in \langle G \rangle$,
8. if R is the point at infinity, output 'invalid' and stop,
9. compute $R' = \mathrm{EC2OSP}_E(R, \mathit{compressed}) \in \{0,1\}^{8(L_F+1)}$,
10. compute $\widetilde{L}_{rec} = L_r - L_{red}$,
11. decompose $r = r_0||r_1 \in \{0,1\}^{8(L_{red}+\widetilde{L}_{rec})}$ s.t. $r_0 \in \{0,1\}^{8L_{red}}$ and $r_1 \in \{0,1\}^{8\widetilde{L}_{rec}}$,
12. compute $h_1 = \mathrm{MGF}(r_0||R'||\mathrm{I2OSP}(1,1), \widetilde{L}_{rec}) \in \{0,1\}^{8\widetilde{L}_{rec}}$,
13. compute $\widetilde{M}_{rec} = r_1 \oplus h_1 \in \{0,1\}^{8\widetilde{L}_{rec}}$,
14. decompose $\widetilde{M}_{rec} = \mathrm{I2OSP}(1,i)||M_{rec}$ s.t. $1 \leq i \leq L_{rec}^{min}$, if there exists such unique i,
15. if there exists no such i, output 'invalid' and stop,
16. compute $L_{rec} = \widetilde{L}_{rec} - i$,
17. compute $L'_{rec} = \mathrm{I2OSP}(L_{rec}, 8) \in \{0,1\}^{64}$,
18. compute $h_0 = \mathrm{MGF}(\widetilde{M}_{rec}||L'_{rec}||R'||\mathrm{I2OSP}(0,1), L_{red}) \in \{0,1\}^{8L_{red}}$,
19. if $r_0 \neq h_0$, output 'invalid' and stop, and
20. output recovered message $M_{rec} \in \{0,1\}^{8L_{rec}}$.

References

[1] Iso/iec 9796-3. In *ISO/IEC*, 2006.

[2] Masayuki Abe, Tatsuaki Okamoto, and Koutarou Suzuki. Message recovery signature schemes from sigma-protocols. In *NTT Technical Review, January 2008 Vol. 6 No. 1*, 2008.

[3] Certicom Corp. Sec 1. In *SECG specification*, 2000.

[4] NTT Laboratories. Sec 6. In *SECG specification*, 2008.

A Mask Generation Function

ECAOS uses mask generation function MGF that takes as input octet string x and octet length l and output octet string $\mathrm{MGF}(x,l)$ of length l. We denote by L^{in}_{MGF} maximal octet length of input x, and by L^{out}_{MGF} maximal octet length of output $\mathrm{MGF}(x,l)$.

The following implementation of mask generation function MGF is recommended. It uses mask generation function MGF1 specified in ISO9796-3[1], and with $L^{in}_{\text{MGF}} = L^{in}_{\text{Hash}} - 8$ and $L^{out}_{\text{MGF}} = 2^{64} L^{out}_{\text{Hash}}$. Here, L^{in}_{Hash} and L^{out}_{Hash} are maximal octet length of input and output of the hash function used in MGF1, respectively.

$$\text{MGF}(x, l) = \text{MGF1}(x||\text{I2OSP}(0, 4), 2^{32} L^{out}_{\text{Hash}})||\text{MGF1}(x||\text{I2OSP}(1, 4), 2^{32} L^{out}_{\text{Hash}})|| \\ \ldots ||\text{MGF1}(x||\text{I2OSP}(b - 1, 4), l - 2^{32} L^{out}_{\text{Hash}}(b - 1)) \in \{0, 1\}^{8l},$$

where $b = \lceil l/(2^{32} L^{out}_{\text{Hash}}) \rceil$. If $l \leq 2^{32} L^{out}_{\text{Hash}}$, it can be written as follows.

$$\text{MGF}(x, l) = \text{MGF1}(x||\text{I2OSP}(0, 4), l) \in \{0, 1\}^{8l}.$$

B Requirements for Parameters

Length parameters L_n, L_F, K, L_{red}, and L^{min}_{rec} are required to satisfy the following conditions.

- $L_n \geq 20$,
- $L_F \geq 20$,
- $K \geq 10$,
- $L_{red} \geq 10$,
- $L^{min}_{rec} \geq 0$ and $L_{red} + L^{min}_{rec} \geq 20$.

Hash function in MGF1 is required to be one of the following hash functions.

- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512.

C Recommendations for Parameters

Length parameters L_n, L_F, K, L_{red}, and L^{min}_{rec} are recommended to be the following value, and hash function in MGF1 is recommended to be the following hash function.

- $L_n = 32$, $L_F = 32$, $K = 16$, $L_{red} = 16$, $L^{min}_{rec} = 16$.
- SHA-256.

In general, length parameters L_n, L_F, K, L_{red}, and L^{min}_{rec} are recommended to be the following value, and hash function in MGF1 is recommended to be the following hash function.

- $\kappa \geq 16$, $L_n = 2\kappa$, $L_F = 2\kappa$, $K = \kappa$, $L_{red} = \kappa$, $L^{min}_{rec} = \kappa$.
- Hash function with κ-bit security from the list in requirements.