# The SM9 Cryptographic Schemes

Zhaohui Cheng

Independent Consultant
zhaohui_cheng@hotmail.com

**Abstract.** SM9 is a Chinese official cryptography standard which defines a set of identity-based cryptographic schemes from pairings. This report describes the technical specification of SM9 as a reference for those practitioners who have difficult to access the Chinese version of the standard.

## 1 Introduction

In this document, the identity-based signature (IBS), the identity-based key agreement(IB-KA) and the identity-based encryption (IBE) schemes from SM9 are described. These schemes are instantiated with an efficient bilinear pairing on elliptic curves [3] such as the optimal Ate pairing [7] or the R-Ate pairing [6].

Without loss of generality, a pairing is defined as a bilinear map

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$$

where $\mathbb{G}_1, \mathbb{G}_2$ are additive groups and $\mathbb{G}_T$ is a multiplicative group. All three groups have prime order $r$.

The map $\hat{e}$ has the following properties:

1. Bilinearity. For all $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}$, $\hat{e}([a]P, [b]Q) = \hat{e}(P, Q)^{ab}$.
2. Non-degeneracy. For generator $P_1 \in \mathbb{G}_1$ and $P_2 \in \mathbb{G}_2$, $\hat{e}(P_1, P_2) \neq 1$

## 2 Notation

The following list briefly describes the notation used in the document. One may refer to ISO/IEC 18033-2 [4] for detailed definitions.

1. $BITS(m)$ the primitive to count bit length of a bit string $m$.
2. $BS2IP(m)$ the primitive to convert a bit string $m$ to an integer.
3. $EC2OSP(C)$ the primitive to convert an elliptic curve point $C$ to an octet string.
4. $FE2OSP(w)$ the primitive to convert a field element $w$ to an octet string.
5. $I2OSP(m, l)$ the primitive to convert an integer $m$ to an octet string of length $l$.

## 3 Supporting Functions

Before presenting the main schemes, two supporting functions used in the schemes are described here.

The first function is a key derivation function (KDF) which works as KDF2 in ISO/IEC 18033-2 [4].

**KDF2** $(H_v, Z, l)$. Given a hash function $H_v$ with output bit length $v$, a bit string $Z$ and a non-negative integer $l$

1. Set a 32-bit counter $ct = 0x00000001$.
2. For $i = 1$ to $\lceil l/v \rceil$.
   (a) Set $Ha_i = H_v(Z \| I2OSP(ct, 4))$.
   (b) Set $ct = ct + 1$.
3. Output the first $l$ bits of $Ha_1 \| Ha_2 \| \cdots \| Ha_{\lceil l/v \rceil}$.

The second function is a hash to range function (H2RF) which runs as follows:

$\mathbf{H2RF}_i(H_v, Z, n)$. Given a hash function $H_v$ with output bit length $v$, a bit string $Z$ and a non-negative integer $n$ and a non-negative integer index $i$

1. Set $l = 8 \times \lceil (5 \times BITS(n))/32 \rceil$.
2. Set $Ha =$**KDF2**$(H_v, I2OSP(i, 1) \| Z, l)$.
3. Set $h = BS2IP(Ha)$.
4. Output $h_i = (h \mod (n-1)) + 1$.

## 4 Identity-Based Signature

The SM9 signature scheme consists of following four operations: **Setup**, **Private-Key-Extract**, **Sign** and **Verify**.

**Setup** $\mathbb{G}_{\mathtt{ID}}(1^\kappa)$**.** On input $1^\kappa$, the operation runs as follows:

1. Generate three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order $r$ and a bilinear pairing map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Pick random generator $P_1 \in \mathbb{G}_1, P_2 \in \mathbb{G}_2$.
2. Pick a random $s \in \mathbb{Z}_r^*$ and compute $P_{pub} = [s]P_2$.
3. Set $g = \hat{e}(P_1, P_{pub})$.
4. Pick a cryptographic hash function $H_v$ and a one byte appendix $hid$.
5. Output the master public key $M_{\mathfrak{pk}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, P_1, P_2, P_{pub}, g, \mathbf{H2RF}_1(H_v, \cdot, \cdot), \mathbf{H2RF}_2(H_v, \cdot, \cdot), hid)$ and the master secret key $M_{\mathfrak{sk}} = s$. SM9 standard requires $hid = 1$.

**Private-Key-Extract** $\mathbb{X}_{\mathtt{ID}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID}_A)$**.** Given an identity string $\mathtt{ID}_A \in \{0,1\}^*$ of entity $A$, $M_{\mathfrak{pk}}$ and $M_{\mathfrak{sk}}$, the operation outputs error if

$$s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r) \mod r = 0,$$

otherwise outputs

$$D_A = [\frac{s}{s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)}]P_1.$$

**Sign**$(M_{\mathfrak{pk}}, D_A, M)$. Given the message $M$, the private key $D_A$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. Pick a random $x \in \mathbb{Z}_r^*$.
2. Set $w = g^x$.
3. Set $h = \mathbf{H2RF}_2(H_v, M \| FE2OSP(w), r)$.
4. Set $l = (x - h) \mod r$.
5. Set $S = [l]D_A$.
6. Output $\langle h, S\rangle$.

**Verify**$(M_{\mathfrak{pk}}, \mathtt{ID}_A, M, \langle h, S\rangle)$. Given the message $M$, the signer's identity string $\mathtt{ID}_A$, the signature $\langle h, S\rangle$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. If $h \notin \mathbb{Z}_r^*$ or $S \notin \mathbb{G}_1^*$, then output failure and terminate.
2. Set $h_1 = \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)$.
3. Set $Q = [h_1]P_2 + P_{pub}$.
4. Set $u = \hat{e}(S, Q)$.
5. Set $t = g^h$.
6. Set $w' = u \cdot t$.
7. Set $h_2 = \mathbf{H2RF}_2(H_v, M \| FE2OSP(w'), r)$.
8. If $h \neq h_2$, then output failure, otherwise output success.

## 5 Identity-Based Key Agreement

The SM9 key agreement is an authenticated two-pass (or three-pass) key agreement (with key confirmation). The scheme consists of following operations: **Setup**, **Private-Key-Extract**, **Message Exchange**, **Session Key Generation** and **Session Key Confirmation**.

**Setup** $\mathbb{G}_{\mathtt{ID}}(1^\kappa)$**.** On input $1^\kappa$, the operation runs as follows:

1. Generate three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order $r$ and a bilinear pairing map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Pick random generator $P_1 \in \mathbb{G}_1, P_2 \in \mathbb{G}_2$.
2. Pick a random $s \in \mathbb{Z}_r^*$ and compute $P_{pub} = [s]P_1$.
3. Set $g = \hat{e}(P_{pub}, P_2)$.
4. Pick a cryptographic hash function $H_v$ and a one byte appendix *hid*.
5. Output the master public key $M_{\mathfrak{pk}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e},\ P_1, P_2, P_{pub}, g,$ $\mathbf{H2RF}_1(H_v, \cdot, \cdot), hid)$ and the master secret key $M_{\mathfrak{sk}} = s$. SM9 standard requires $hid = 2$.

**Private-Key-Extract** $\mathbb{X}_{\mathtt{ID}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID}_A)$**.** Given an identity string $\mathtt{ID}_A \in \{0,1\}^*$ of entity $A$, $M_{\mathfrak{pk}}$ and $M_{\mathfrak{sk}}$, the operation outputs error if

$$s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r) \mod r = 0,$$

otherwise outputs

$$D_A = [\frac{s}{s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)}]P_2.$$

**Message Exchange.**

$$\begin{aligned} &A \to B : R_A = [x_A]([\mathbf{H2RF}_1(H_v, \mathtt{ID}_B \| hid, r)]P_1 + P_{pub}) \\ &B \to A : R_B = [x_B]([\mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)]P_1 + P_{pub}), S_B \\ &A \to B : S_A \end{aligned}$$

where random $x_A, x_B \in \mathbb{Z}_r^*$ are picked by $A$ and $B$ respectively and $S_B$ and $S_A$ are the optional session key confirmation parts. The method to generate such optional values is explained later.

**Session Key Generation.**

1. Entity $A$ computes intermediate values

$$g_1 = \hat{e}(R_B, D_A), g_2 = \hat{e}(P_{pub}, P_2)^{x_A} = g^{x_A}, g_3 = g_1^{x_A}.$$

2. Entity $A$ computes session key

$$\begin{aligned} SK_A = \mathbf{KDF2}(&ID_A \| ID_B \| EC2OSP(R_A) \| EC2OSP(R_B) \| \\ &FE2OSP(g_1) \| FE2OSP(g_2) \| FE2OSP(g_3), klen). \end{aligned}$$

3. Entity $B$ computes intermediate values

$$g'_1 = \hat{e}(P_{pub}, P_2)^{x_B} = g^{x_B}, g'_2 = \hat{e}(R_A, D_B), g'_3 = {g'_2}^{x_B}.$$

4. Entity $B$ computes session key

$$\begin{aligned} SK_B = \mathbf{KDF2}(&ID_A \| ID_B \| EC2OSP(R_A) \| EC2OSP(R_B) \| \\ &FE2OSP(g'_1) \| FE2OSP(g'_2) \| FE2OSP(g'_3), klen). \end{aligned}$$

**Session Key Confirmation.**

1. Entity $B$ computes its key confirmation

$$S_B = H_v(0x82 \| FE2OSP(g'_2) \|$$

$$H_v(FE2OSP(g'_1) \| FE2OSP(g'_3) \| ID_A \| ID_B \| EC2OSP(R_A) \| EC2OSP(R_B))).$$

   Entity $A$ should verify $S_B$'s correctness with $g_1, g_2, g_3$.
2. Entity $A$ computes its key confirmation

$$S_A = H_v(0x83 \| FE2OSP(g_1) \|$$

$$H_v(FE2OSP(g_2) \| FE2OSP(g_3) \| ID_A \| ID_B \| EC2OSP(R_A) \| EC2OSP(R_B))).$$

   Entity $B$ should verify $S_A$'s correctness with $g'_1, g'_2, g'_3$.

Note that entity $A(B)$ should check $R_B(R_A)$ lies in $\mathbb{G}_1^*$.

## 6 Identity-Based Encryption

The SM9 encryption is a hybrid encryption scheme built from an identity-based key encapsulation scheme (KEM) and a data encapsulation scheme (DEM). DEM can be one of those schemes standardized in ISO/IEC 18033-2 [4]. First the SM9 KEM is presented, then the hybrid encryption scheme is described. The KEM scheme consists of four operations: **Setup**, **Private-Key-Extract**, **KEM-Encap** and **KEM-Decap**. They works follows:

**Setup** $\mathbb{G}_{\mathtt{ID}}(1^\kappa)$**.** On input $1^\kappa$, the operation runs as follows:

1. Generate three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of prime order $r$ and a bilinear pairing map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Pick random generator $P_1 \in \mathbb{G}_1, P_2 \in \mathbb{G}_2$.
2. Pick a random $s \in \mathbb{Z}_r^*$ and compute $P_{pub} = [s]P_1$.
3. Set $g = \hat{e}(P_{pub}, P_2)$.
4. Pick a cryptographic hash function $H_v$ and a one byte appendix $hid$.
5. Output the master public key $M_{\mathfrak{pk}} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e},\ P_1, P_2, P_{pub}, g,$ $\mathbf{H2RF}_1(H_v, \cdot, \cdot), hid)$ and the master secret key $M_{\mathfrak{sk}} = s$. SM9 standard requires $hid = 3$.

**Private-Key-Extract** $\mathbb{X}_{\mathtt{ID}}(M_{\mathfrak{pk}}, M_{\mathfrak{sk}}, \mathtt{ID}_A)$**.** Given an identity string $\mathtt{ID}_A \in \{0, 1\}^*$ of entity $A$, $M_{\mathfrak{pk}}$ and $M_{\mathfrak{sk}}$, the operation outputs error if

$$s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r) \mod r = 0,$$

otherwise outputs

$$D_A = [\frac{s}{s + \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)}]P_2.$$

**KEM-Encap** $(M_{\mathfrak{pk}}, \mathtt{ID}_A, l)$. Given an identify string $\mathtt{ID}_A$, the DEM key length $l$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. Set $h_1 = \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)$.
2. Set $Q = [h_1]P_1 + P_{pub}$.
3. Pick a random $x \in \mathbb{Z}_r^*$.
4. Set $C_1 = [x]Q$.
5. Set $t = g^x$.
6. Set $K = \mathbf{KDF2}(H_v, EC2OSP(C_1) \| FE2OSP(t) \| ID_A, l)$.
7. Output $\langle K, C_1 \rangle$.

**KEM-Decap** $(M_{\mathfrak{pk}}, \mathtt{ID}_A, D_A, C_1, l)$. Given an identify string $\mathtt{ID}_A$, the corresponding private key $D_A$, the encapsulation part $C_1$, the DEM key length $l$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. If $C_1 \notin \mathbb{G}_1^*$, then output $\perp$ and terminate.
2. Set $t = \hat{e}(C_1, D_A)$.
3. Set $K = \mathbf{KDF2}(H_v, EC2OSP(C_1) \| FE2OSP(t) \| ID_A, l)$.
4. Output $K$.

The full SM9 encryption scheme works as follows:

**KEM-DEM-Encrypt** $(M_{\mathfrak{pk}}, \mathtt{ID}_A, m)$. Given an identify string $\mathtt{ID}_A$, the plain text $m$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. Set $h_1 = \mathbf{H2RF}_1(H_v, \mathtt{ID}_A \| hid, r)$.
2. Set $Q = [h_1]P_1 + P_{pub}$.
3. Pick a random $x \in \mathbb{Z}_r^*$.
4. Set $C_1 = [x]Q$.
5. Set $t = g^x$.
6. Set $K_1 \| K_2 = \mathbf{KDF2}(H_v, EC2OSP(C_1) \| FE2OSP(t) \| ID_A, BITS(m) + v)$.
7. Set $C_2 = K_1 \oplus m$.
8. Set $C_3 = H_v(C_2 \| K_2)$.
9. Output $\langle C_1, C_2, C_3 \rangle$.

**KEM-DEM-Decrypt** $(M_{\mathfrak{pk}}, \mathtt{ID}_A, D_A, \langle C_1, C_2, C_3 \rangle)$. Given an identify string $\mathtt{ID}_A$, the corresponding private key $D_A$, the cipher text $\langle C_1, C_2, C_3 \rangle$ and the master public key $M_{\mathfrak{pk}}$, the operation runs as follows:

1. If $C_1 \notin \mathbb{G}_1^*$, then output $\perp$ and terminate.
2. Set $t = \hat{e}(C_1, D_A)$.
3. Set $K_1 \| K_2 = \mathbf{KDF2}(H_v, EC2OSP(C_1) \| FE2OSP(t) \| ID_A, BITS(C_2) + v)$.
4. Set $C_3' = H_v(C_2 \| K_2)$.
5. If $C_3' \neq C_3$, then output $\perp$ and terminate.
6. Output $m = K_1 \oplus C_2$.

## 7 Performance Evaluation

Here we briefly compare the performance of SM9 with the identity-based signature schemes included in ISO/IEC 14888-3 [2], identity-based key agreements included in ISO/IEC 11770-3 [1] and encryption schemes in ISO/IEC 18033-5 [5]. Table 1 shows that the SM9 signature scheme is more efficient than those two IBS schemes in ISO/IEC 14888-3. Table 2 shows that the SM9 key agreement is more efficient than those two IB-KA schemes in ISO/IEC 11770-3 [1]. Table 3 shows that the SM9 KEM maintains better performance in terms of both the computation efficiency and the cipher text size than those three schemes in ISO/IEC 18033-5.

## References


1. ISO/IEC. Information technology – Secruity techniques – Key management – Part 3: Mechanisms using asymmetric techniques. *ISO11770-3*, 2015.
2. ISO/IEC. Information technology – Secruity techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms. *ISO14888-3*, 2015.

**Table 1.** Performance of IBS from Pairings

| | IBS1 [2] | IBS2 [2] | SM9-IBS |
|---|---|---|---|
| Private Key Extract | | | |
| Hash to $\mathbb{G}_1$ | 1 | 1 | |
| Mul in $\mathbb{G}_1$ | 1 | 1 | $\overline{1}$ |
| Sign | | | |
| Mul in $\mathbb{G}_1$ | $\overline{2}^{(1)}$ | $\overline{2}^{(2)}$ | $\overline{1}$ |
| Exp in $\mathbb{G}_T$ | | | $\overline{1}$ |
| Pairings | 1 | | |
| Verify | | | |
| Hash to $\mathbb{G}_1$ | 1 | 1 | |
| Mul in $\mathbb{G}_1$ | $1^{(1)}$ | 1 | |
| Mul in $\mathbb{G}_2$ | | | $\overline{1}$ |
| Exp in $\mathbb{G}_T$ | | | $\overline{1}$ |
| Pairings | 2 | 2 | 1 |
| Signature Size | $\lambda + \gamma$ | $2\gamma$ | $\lambda + \gamma$ |

1. Assume muplication in $\mathbb{G}_1$ is faster than exponentiation in $\mathbb{G}_T$.
2. Assume $Y$ is pre-computed in producing the pre-signature [2] which is reasonable for a signer.
3. Symbols $\overline{m}$ and $n$ denote $m$ fix-based multiplications or exponentiations and $n$ general operations respectively.
4. Symbols $\lambda, \gamma$ denote the length of an element in $\mathbb{Z}_r^*$ and $\mathbb{G}_1$ respectively.

**Table 2.** Performance of IB-KAs from Pairings

| | SCC [1] | FSU [1] | SM9-KA |
|---|---|---|---|
| Private Key Extract | | | |
| Hash to $\mathbb{G}_1$ | 1 | 1 | |
| Mul in $\mathbb{G}_1$ | 1 | 1 | $\overline{1}$ |
| Message Exchange | | | |
| Mul in $\mathbb{G}_1$ | $\overline{1}$ | $\overline{1}$ | $\overline{2}$ |
| Session Key Generation | | | |
| Hash to $\mathbb{G}_1$ | 1 | 1 | |
| Mul in $\mathbb{G}_1$ | $1+\overline{1}$ | $1+\overline{1}^{(1)}$ | |
| Exp in $\mathbb{G}_T$ | | | $1+\overline{1}$ |
| Pairings | 2 | 2 | 1 |
| Message Size | $\gamma$ | $\gamma$ | $\gamma$ |

1. The FSU scheme requires $\mathbb{G}_1 = \mathbb{G}_2$.

**Table 3.** Performance of IBEs from Pairings

| | BF-IBE [5] | $BB_1$-KEM [5] | SK-KEM [5] | SM9-KEM |
|---|---|---|---|---|
| Private Key Extract | | | | |
| Hash to $\mathbb{G}_2$ | 1 | | | |
| Mul in $\mathbb{G}_2$ | 1 | $\overline{2}$ | $\overline{1}$ | $\overline{1}$ |
| Encapsulate | | | | |
| Hash to $\mathbb{G}_2$ | 1 | | | |
| Mul in $\mathbb{G}_1$ | $1+\overline{1}$ | $\overline{3}$ | $\overline{2}$ | $\overline{2}$ |
| Exp in $\mathbb{G}_T$ | | $\overline{1}$ | $\overline{1}$ | $\overline{1}$ |
| Pairings | 1 | | | |
| Decapsulate | | | | |
| Mul in $\mathbb{G}_1$ | $\overline{1}$ | | $\overline{1}^{(1)}$ | |
| Mul in $\mathbb{G}_2$ | | | | |
| Pairings | 1 | 2 | 1 | 1 |
| Cipher Text Size | $\gamma+\delta+\zeta$ | $2\gamma+\eta$ | $\gamma+\delta+\eta$ | $\gamma+\eta$ |

1. Assume $Q$ is pre-computed in KEM-Decrypt [5] which is reasonable for a decryptor.
2. Symbols $\gamma, \delta, \zeta, \eta$ denote the length of an element in $\mathbb{G}_1$, a random message, a plain text and a DEM respectively.

3. ISO/IEC. Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 5: Elliptic curve generation. *ISO15946-5*, 2009.
4. ISO/IEC. Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers. *ISO18033-2*, 2006.
5. ISO/IEC. Information technology – Security techniques – Encryption algorithms – Part 5: Identity-based ciphers. *ISO18033-5*, 2015.
6. E. Lee, H. Lee and C. Park. Efficient and generalized pairing computation on abelian varieties. In *IEEE Transactions on Information Theory*, Volume 55, pp. 1793–1803, 2009.
7. F. Vercauteren. Optimal pairings. In *IEEE Transactions on Information Theory*, Volume: 56, Issue: 11, pp. 455 – 461, 2010.