1 ECC ElGamal

1.1 Elliptic Curve Over $\mathbb{F}_p$

The implementation of our scheme is based on elliptic curve groups for efficiency. Let $\sigma := (p, a, b, g, q, \zeta)$ be the elliptic curve domain parameters over $\mathbb{F}_p$, consisting of a prime p specifying the finite field $\mathbb{F}_p$, two elements $a, b \in \mathbb{F}_p$ specifying an elliptic curve $E(\mathbb{F}_p)$ defined by $E : y^2 \equiv x^3+ax+b \pmod p$, a base point $g = (x_g, y_g)$ on $E(\mathbb{F}_p)$, a prime q which is the order of g, and an integer ζ which is the cofactor $\zeta = \#E(\mathbb{F}_p)/q$. We denote the cyclic group generated by g by $\mathbb{G}$, and it is assumed that the DDH assumption holds over $\mathbb{G}$, that is for all p.p.t. adversary $\mathcal{A}$:

$$\mathsf{Adv}_{\mathbb{G}}^{\mathsf{DDH}}(\mathcal{A}) = \left| \Pr\left[\begin{array}{l} x, y \leftarrow \mathbb{Z}_q; b \leftarrow \{0,1\}\,; h_0 = g^{xy}; \\ h_1 \leftarrow \mathbb{G} : \mathcal{A}(g, g^x, g^y, h_b) = b \end{array} \right] - \frac{1}{2} \right| \leq \epsilon(\lambda) \ ,$$

where $\epsilon(\cdot)$ is a negligible function.

1.2 Lifted (Threshold) ElGamal

We employ lifted ElGamal encryption scheme as the candidate of the additively homomorphic public key cryptosystem in our protocol construction. It consists of the following 4 PPT algorithms:

- $\mathsf{Gen}_{\mathsf{gp}}(1^\lambda)$: take input as security parameter λ, and output $\sigma := (p, a, b, g, q, \zeta)$.
- $\mathsf{EC.Gen}(\sigma)$: pick $\mathsf{sk} \leftarrow \mathbb{Z}_q^*$ and set $\mathsf{pk} := h = g^{sk}$, and output $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{EC.Enc}_{\mathsf{pk}}(m; r)$: output $e := (e_1, e_2) = (g^r, g^m h^r)$.
- $\mathsf{EC.Dec}_{\mathsf{sk}}(e)$: output $\mathsf{Dlog}(e_2 \cdot e_1^{-\mathsf{sk}})$, where $\mathsf{Dlog}(x)$ is the discrete logarithm of x. (Note that since $\mathsf{Dlog}(\cdot)$ is not efficient, the message space should be a small set, say $\{0,1\}^\xi$, for $\xi \leq 30$. In practice, we can use lookup tables.)

It is well known that lifted ElGamal encryption scheme is IND-CPA secure under the DDH assumption. It has additively homomorphic property:

$$\mathsf{EC.Enc}_{\mathsf{pk}}(m_1; r_1) \cdot \mathsf{EC.Enc}_{\mathsf{pk}}(m_2; r_2) = \mathsf{EC.Enc}_{\mathsf{pk}}(m_1 + m_2; r_1 + r_2) \ .$$

Remark: The key generation and decryption algorithm of the lifted ElGamal encryption can be efficiently distributed. (cf. below)

1.3 A Hybrid Encryption

When we need to encrypt longer strings, we will use the following hybrid encryption scheme, which consists of the following 2 PPT algorithms in addition to the algorithms described above:

- $\mathsf{EC.Enc}_{\mathsf{pk}}(m; (r, s))$, output $e := (e_1, e_2, e_3) = (g^r, g^s h^r,)$.
- $\mathsf{EC.Dec}_{\mathsf{sk}}(e)$: output $\mathsf{Dlog}(e_2 \cdot e_1^{-\mathsf{sk}})$, where $\mathsf{Dlog}(x)$ is the discrete logarithm of x. (Note that since $\mathsf{Dlog}(\cdot)$ is not efficient, the message space should be a small set, say $\{0,1\}^\xi$, for $\xi \leq 30$. In practice, we can use lookup tables.)

2 NIZK for lifted Elgamal Encryption of 0

$$\mathsf{NIZK}\{(\mathsf{pk}, C), (m, r) : C = \mathsf{EC.Enc}_{\mathsf{pk}}(m; r) \wedge m = 0\}$$

Statement: Public key, $\mathsf{pk} := h \in \mathbb{G}$, and ciphertext $C := (C_1, C_2) = (g^r, g^m h^r)$
Witness: $m = 0$ and $r \in \mathbb{Z}_p$

Prover:

- Pick random $w \leftarrow \mathbb{Z}_p$; Compute $A_1 := g^w$ and $A_2 := h^w$
- Compute $e_1 = \mathsf{hash}(h, C, A_1, A_2)$ and $z = r * e_1 + w$
- Return $\pi_1 := (A_1, A_2, z_1)$

Verifier:

- Compute $e_1 = \mathsf{hash}(h, C, A_1, A_2)$ and return valid if and only if
  - $g^{z_1} = {C_1}^{e_1} \cdot A_1$
  - $h^{z_1} = {C_2}^{e_1} \cdot A_2$

Figure 1: Non-Interactive Zero Knowledge proof for Lifted-Elgamal Encryption of 0

3 NIZK for lifted Elgamal Encryption of 1

$$\mathsf{NIZK}\{(\mathsf{pk}, C), (m, r) : C = \mathsf{EC.Enc}_{\mathsf{pk}}(m; r) \wedge m = 1\}$$

Statement: Public key, $\mathsf{pk} := h \in \mathbb{G}$, and ciphertext $C := (C_1, C_2) = (g^r, g^m h^r)$
Witness: $m = 1$ and $r \in \mathbb{Z}_p$

Prover:

- Pick random $v \leftarrow \mathbb{Z}_p$; Compute $B_1 := g^v$ and $B_2 := h^v$
- Compute $e_2 = \mathsf{hash}(h, C, B_1, B_2)$ and $z_2 = r * e_2 + v$
- Return $\pi_2 := (B_1, B_2, z_2)$

Verifier:

- Compute $e_2 = \mathsf{hash}(h, C, B_1, B_2)$ and return valid if and only if
  - $g^{z_2} = {C_1}^{e_2} \cdot B_1$
  - $h^{z_2} = (C_2 \cdot g^{-1})^{e_2} \cdot B_2$

Figure 2: Non-Interactive Zero Knowledge proof for Lifted-Elgamal encryption of 1

4 NIZK for lifted Elgamal Encryption of 0 or 1

$$\mathsf{NIZK}\{(\mathsf{pk}, C), (m, r) : C = \mathsf{EC.Enc}_{\mathsf{pk}}(m; r) \wedge m \in \{0, 1\}\}$$

Statement: Public key, $\mathsf{pk} := h \in \mathbb{G}$, and ciphertext $C := (C_1, C_2) = (g^r, g^m h^r)$
Witness: $m \in \{0, 1\}$ and $r \in \mathbb{Z}_p$

Prover:

- if m=0:
  - Pick random $w \leftarrow \mathbb{Z}_p$; Compute $A_1 := g^w$ and $A_2 := h^w$
  - Pick random $z_2 \leftarrow \mathbb{Z}_p, e_2 \leftarrow \{0, 1\}^{256}$; Compute $B_1 := \frac{g^{z_2}}{C_1{}^{e_2}}$ and $B_2 := \frac{h^{z_2}}{(\frac{C_2}{g})^{e_2}}$
  - Compute $e := \mathsf{hash}(C, h, A_1, A_2, B_1, B_2)$ and $e_1 := e \oplus e_2$
  - Compute $z_1 := r \cdot e_1 + w$
  - Return: $\pi_3 := (A_1, A_2, B_1, B_2, e_1, e_2, z_1, z_2)$
- else m=1:
  - Pick random $v \leftarrow \mathbb{Z}_p$; Compute $B_1 := g^v$ and $B_2 := h^v$
  - Pick random $z_1 \leftarrow \mathbb{Z}_p, e_1 \leftarrow \{0, 1\}^{256}$; Compute $A_1 := g^{z_1} \cdot C_1{}^{-e_1}$ and $A_2 := \frac{h^{z_1}}{C_2{}^{e_1}}$
  - Compute $e := \mathsf{hash}(C, h, A_1, A_2, B_1, B_2)$ and $e_2 := e \oplus e_1$
  - Compute $z_2 := r \cdot e_2 + v$
  - Return: $\pi_3 := (A_1, A_2, B_1, B_2, e_1, e_2, z_1, z_2)$

Verifier:

- Computer $e = \mathsf{hash}(C, h, A_1, A_2, B_1, B_2)$ and return valid if and only if
  - $e = e_1 \oplus e_2$
  - $g^{z_1} = C_1{}^{e_1} \cdot A_1$
  - $h^{z_1} = C_2{}^{e_1} \cdot A_2$
  - $g^{z_2} = C_1{}^{e_2} \cdot B_1$
  - $h^{z_2} = (\frac{C_2}{g})^{e_2} \cdot B_2$

Figure 3: Non-Interactive Zero Knowledge proof for Lifted-Elgamal encryption of 0 or 1

5 NIZK for correct lifted Elgamal decryption

$$\mathsf{NIZK}\{(\mathsf{pk}, C, D), (\mathsf{sk}) : D = C_1^{\mathsf{sk}} \wedge \mathsf{pk} = g^{\mathsf{sk}}\}$$

Statement: Public key, $\mathsf{pk} := h \in \mathbb{G}$, ciphertext $C := (C_1, C_2)$, and the decryption share D
Witness: $\mathsf{sk} \in \mathbb{Z}_p$

Prover:

- Pick random $w \leftarrow \mathbb{Z}_p$; Compute $A_1 := g^w$ and $A_2 := C_1^w$
- Compute $e = \mathsf{hash}(C, D, A_1, A_2)$ and $z = \mathsf{sk} * e + w$
- Return $\pi_4 := (A_1, A_2, D, z)$

Verifier:

- Compute $e = \mathsf{hash}(C, D, A_1, A_2)$; and $D = \frac{C_2}{g^m}$, and return valid if and only if
  - $g^z = h^e \cdot A_1$
  - $C_1^z = D^e \cdot A_2$

Figure 4: Non-Interactive Zero Knowledge proof for secret key sk

6 Alternative NIZK for lifted Elgamal Encryption of $0/1$

$$\mathsf{NIZK}\{(\mathsf{pk}, c), (m, r) : C = \mathsf{EC.Enc}_{\mathsf{pk}}(m; r) \wedge m \in \{0, 1\}\}$$

Statement: Public key, $\mathsf{pk} := h \in \mathbb{G}$, and ciphertext $C := (C_1, C_2) = (g^r, g^m h^r)$
Witness: $m \in \{0, 1\}$ and $r \in \mathbb{Z}_p$

Prover:

- Pick random $\beta, \gamma, \delta \leftarrow \mathbb{Z}_p$; Compute $B = \mathsf{EC.Enc}_{\mathsf{pk}}(\beta; \gamma)$ and $A = \mathsf{EC.Enc}_{\mathsf{pk}}(m \cdot \beta; \delta)$
- Compute $e = \mathsf{hash}(C, pk, B, A)$;
- Compute $f = m \cdot e + \beta$ and $w = r \cdot e + \gamma$; and $v = r \cdot (e - f) + \delta$
- Return $\pi_5 := (B, A, f, w, v)$

Verifier:

- Compute $e = \mathsf{hash}(C, pk, B, A)$ and return valid if and only if
  - $C^e \cdot B = \mathsf{EC.Enc}_{\mathsf{pk}}(f; w)$
  - $C^{e-f} \cdot A = \mathsf{EC.Enc}_{\mathsf{pk}}(0; v)$

Figure 5: Non-Interactive Zero Knowledge proof for lifted Elgamal Encryption of 0/1