Threshold Public-Release Encryption via BLS DKG and Signature-Based Witness Encryption

1 Overview

This document specifies a pairing-based threshold encryption scheme with *public release* semantics. The construction combines:

- a robust distributed key generation (DKG) protocol using Pedersen verifiable secret sharing (VSS) for BLS signatures over BLS12-381, and
- a pairing-based key encapsulation mechanism (KEM) in which possession of a valid BLS signature on a designated tag serves as a witness for decryption.

No participant ever learns the master secret key. A plaintext is released publicly once a threshold ($t = 2f + 1$) of participants cooperate to produce a valid BLS signature on a specified tag.

2 Cryptographic Setting

Let (G_1, G_2, G_T) be cyclic groups of prime order r equipped with a bilinear pairing

$$e : G_1 \times G_2 \to G_T$$

defined over the BLS12-381 curve.

Let:

- $g_1 \in G_1$, $g_2 \in G_2$ be fixed generators,
- $h_2 \in G_2$ be a generator such that $\log_{g_2}(h_2)$ is unknown,
- $H_{\mathsf{BLS\_sig}} : \{0,1\}^* \to G_1$ be a hash-to-curve function instantiated as `BLS12381G1_XMD:SHA-256_SSWU_RO_` with DST `DST_sig`,

- KDF be HKDF-SHA256,
- AEAD be an authenticated encryption scheme (e.g. AES-256-GCM or ChaCha20-Poly1305).

All group elements are serialized using the standard compressed encodings for BLS12-381. The security parameter is $\lambda = 256$. The HKDF salt is the fixed protocol constant `salt_kem`=`MEMP-ENC-KEM-V1`. The domain-separation constants are `DST_sig`=`MEMP-ENC-SIG-V1` and `DST_kdf`=`MEMP-ENC-KDF-V1`. All protocol constants are interpreted as ASCII byte strings (identical to UTF-8 for these values).

We consider a committee of $n = 3f+1$ parties with threshold $t = 2f+1$. The adversary may corrupt up to $f < n/3$ parties. Each party has a fixed, non-zero index $i \in \{1, \ldots, n\}$ interpreted as a distinct element of $\mathbb{F}_r$ via the natural integer embedding.

3 Distributed Key Generation with Pedersen VSS

The goal of the DKG protocol is to jointly generate a BLS public key

$$PK = g_2^x,$$

where the secret exponent $x \in \mathbb{F}_r$ is never reconstructed or revealed to any party.

3.1 Local Polynomial Sharing

Each party P_j independently samples:

- a secret polynomial

$$f_j(z) = a_{j,0} + a_{j,1}z + \cdots + a_{j,t-1}z^{t-1},$$

- an independent random blinding polynomial

$$r_j(z) = b_{j,0} + b_{j,1}z + \cdots + b_{j,t-1}z^{t-1},$$

  both over $\mathbb{F}_r$.

Party P_j privately sends to each party P_i the pair

$$(f_j(i),\, r_j(i))\,.$$

3.2 Commitment Broadcast

For each coefficient index $k = 0, \ldots, t-1$, party P_j broadcasts a Pedersen commitment

$$C_{j,k} = g_2^{a_{j,k}} h_2^{b_{j,k}} \in G_2.$$

These commitments are computationally binding and information-theoretically hiding.

3.3 Encoding Conventions

We use the standard compressed encodings for BLS12-381 group elements. For any byte string x, define $\mathsf{enc}(x) = \mathsf{len}(x) \parallel x$ where $\mathsf{len}(x)$ is a fixed-width 4-byte big-endian length. For a tuple $(x_1, \ldots, x_k)$, define

$$\mathsf{enc}(x_1, \ldots, x_k) = \mathsf{enc}(x_1) \parallel \cdots \parallel \mathsf{enc}(x_k).$$

3.4 Share Verification and Complaints

Upon receiving $(f_j(i), r_j(i))$ from P_j, party P_i verifies consistency by checking:

$$g_2^{f_j(i)} h_2^{r_j(i)} \stackrel{?}{=} \prod_{k=0}^{t-1} C_{j,k}^{i^k}.$$

If the check fails or a share is missing, P_i issues a complaint against P_j. A standard complaint-resolution and disqualification phase is executed; parties whose shares cannot be validated are excluded from the qualified set QUAL. This can be instantiated with the synchronous DKG of Gennaro et al. (1999), Sec. 4 (Pedersen VSS), which provides complaint resolution, dealer disqualification, and a well-defined QUAL set.

3.5 Share Aggregation and Public Keys

Each qualified party P_i computes its final secret share as

$$x_i = \sum_{j \in \mathsf{QUAL}} f_j(i).$$

The implicit master secret is

$$x = \sum_{j \in \mathsf{QUAL}} f_j(0),$$

which remains unknown to all parties.

The group public key is derived from commitments as

$$PK = \prod_{j \in \mathsf{QUAL}} g_2^{a_{j,0}} = g_2^x.$$

Each party also derives a public verification key

$$PK_i = g_2^{x_i},$$

which can be checked for consistency using the published commitments.

4 Threshold BLS Signatures

For a tag $\mathsf{tg} \in \{0,1\}^*$:

Each party P_i computes a partial signature

$$\sigma_i = H_{\mathsf{BLS\_sig}}(\mathsf{tg})^{x_i} \in G_1.$$

Partial signatures are verified using:

$$e(\sigma_i, g_2) \stackrel{?}{=} e(H_{\mathsf{BLS\_sig}}(\mathsf{tg}), PK_i).$$

Given any subset $S \subseteq \mathsf{QUAL}$ with $|S| \geq t$, an aggregator computes Lagrange coefficients $\{\lambda_i\}_{i \in S}$ and forms the full signature

$$\sigma = \prod_{i \in S} \sigma_i^{\lambda_i} = H_{\mathsf{BLS\_sig}}(\mathsf{tg})^x.$$

5 Witness Encryption / Key Encapsulation

We define a pairing-based KEM in which possession of a valid BLS signature on tg acts as a witness for decapsulation. Accordingly, the KEM uses $H_{\mathsf{BLS\_sig}}$ on tg so that the signature and encapsulation bind to the same group element.

5.1 Encryption (Encapsulation)

To encrypt a message M under public key PK and tag tg:

1. Sample a random symmetric key $K \in \{0,1\}^{256}$.
2. Sample $r \leftarrow \mathbb{F}_r$ and compute $U = g_2^r \in G_2$.

3. Compute
$$W = e(H_{\mathsf{BLS\_sig}}(\mathsf{tg}), PK)^r \in G_T.$$

4. Derive a masking key

$$\mathrm{PRK} = \mathsf{HKDF\text{-}Extract}(\mathtt{salt\_kem}, \mathsf{enc}(W)), \quad K' = \mathsf{HKDF\text{-}Expand}(\mathrm{PRK}, \mathtt{DST}_{\mathsf{kdf}} \parallel \mathsf{enc}(\mathsf{tg}, U, PK),$$

5. Compute $C_K = K \oplus K'$.

6. Sample a fresh nonce $N \in \{0,1\}^{96}$ uniformly at random.

7. Encrypt the payload using associated data:
$$C_M = \mathsf{AEAD.Enc}(K, N, M;\ \mathsf{AD}),$$
where $\mathsf{AD} = \mathsf{enc}(\mathsf{tg}, U, PK)$.

The ciphertext is
$$C = (\mathsf{tg}, U, C_K, N, C_M).$$

5.2 Decryption (Decapsulation)

Given a valid BLS signature $\sigma = H_{\mathsf{BLS\_sig}}(\mathsf{tg})^x$:

1. Compute
$$W' = e(\sigma, U).$$

2. Derive

$$\mathrm{PRK} = \mathsf{HKDF\text{-}Extract}(\mathtt{salt\_kem}, \mathsf{enc}(W')), \quad K' = \mathsf{HKDF\text{-}Expand}(\mathrm{PRK}, \mathtt{DST}_{\mathsf{kdf}} \parallel \mathsf{enc}(\mathsf{tg}, U, PK),$$

3. Recover $K = C_K \oplus K'$.

4. Output $M = \mathsf{AEAD.Dec}(K, N, C_M;\ \mathsf{AD})$ with the same $\mathsf{AD} = \mathsf{enc}(\mathsf{tg}, U, PK)$.

Correctness follows from bilinearity:
$$e(\sigma, U) = e(H_{\mathsf{BLS\_sig}}(\mathsf{tg})^x, g_2^r) = e(H_{\mathsf{BLS\_sig}}(\mathsf{tg}), PK)^r.$$

6 Threshold Public Release

To release the plaintext publicly:

1. The ciphertext C is published.
2. Once the release condition is met, at least $t = 2f + 1$ parties produce partial BLS signatures on tg.
3. The partial signatures are combined into a full signature σ and published.
4. Anyone can decrypt C using σ.

7 Security Assumptions and Guarantees

- **Confidentiality:** The KEM is IND-CPA secure under the decisional bilinear Diffie–Hellman (DBDH) assumption in (G_1, G_2, G_T), modeled in the random oracle model for $H_{\mathsf{BLS\_sig}}$ and HKDF.
- **Threshold enforcement:** Fewer than t parties cannot produce a valid signature under the EUF-CMA security of BLS signatures.
- **Key secrecy:** The master secret x is never reconstructed.
- **DKG robustness:** Under $f < n/3$ corruptions, Pedersen VSS prevents adaptive biasing and ensures correctness of derived public keys.
- **Context binding:** Associated data in AEAD binds $\mathsf{enc}(\mathsf{tg}, U, PK)$ to the ciphertext.

8 Remarks

This construction is suitable for time-lock encryption, event-triggered disclosure, and governance-controlled release. It intentionally provides public-release semantics; once the threshold signature is published, decryption is non-interactive and available to all parties.