Wild Encryption Scheme

James Bond

November 2025

Abstract

Introduce a non-interactive, aggregateable encryption scheme where users possess a single master key pair, allowing child public keys to be derived publicly, and child secret keys to be disclosed without leaking the master secret key. The proposed approach can be used in Private Multisig [1] as a solution to keys derivation and rotation problem.

1 Protocol description

1.1 Prerequisites

The proposed scheme is based on IBE [2], and for it to function properly, a pairing-friendly curve such as BLS12-381 [3] is required. Let G_1 and G_2 be the curve source groups and G_t be the target group. Let G be the generator point in G_1 and Q be the generator point in G_2. Let pairing be a function $e, e : G_1 \times G_2 \to G_t$.

Note that the paper does not explicitly define the boundaries for $\sum$ and $\prod$ for brevity. The implicit convention is that these operations are performed on the entire set of the provided values.

1.2 Encryption key construction

Let sk_i and $pk_i = sk_i \times G$ be a user's master key pair and let d be equal to the publicly known derivation challenge. Then the decryption and encryption key shares are computed as follows:

$$H = hashToCurve(d) \in G_2$$

$$x_i = sk_i \times H$$

$$p_i = e(pk_i, H)$$

where x_i — child secret key (decryption key share),
p_i — child public key (encryption key share).

Correspondingly, the aggregated decryption and encryption keys are derived in the following way:

$$x = \sum x_i = \sum sk_i \times H$$

$$P = \prod p_i = \prod e(pk_i, H) = \prod e(sk_i \times G, H) = \prod e(G, H)^{sk_i} = e(G, H)^{\sum sk_i}$$

1.3 Message mapping

Let m be the message to be encrypted. In the context of Private Multisig, let m_i be equal to zero if a participant votes "against" the proposal and equal to one if the vote is "in favor". Then choose a generator $K = e(G, Q)$ and calculate $M_i = K^{m_i}$.

1.4 Message encryption

Let r_i be a random encryption nonce. The encryption scheme is then defined as follows:

$$R_i = r_i \times G$$
$$C_i = M_i \cdot P^{r_i}$$

1.5 Encryption aggregation

In order to aggregate the individually encrypted messages, the following scheme is used:

$$R = \sum R_i = \sum r_i \times G$$
$$C = \prod C_i = (\prod M_i) \cdot P^{\sum r_i} = (\prod M_i) \cdot e(G, H)^{\sum sk_i \sum r_i}$$

1.6 Message decryption

To decrypt the aggregated message, perform the following steps:

$$K^{\sum m_i} = C \cdot e(R, -x) = (\prod M_i) \cdot e(G, H)^{\sum sk_i \sum r_i} \cdot e(\sum r_i \times G, -\sum sk_i \times H) =$$
$$= (\prod M_i) \cdot e(G, H)^{\sum sk_i \sum r_i} \cdot e(G, H)^{-\sum sk_i \sum r_i} = \prod M_i$$

Loop incrementally through z, until $K^z = C \cdot e(R, -x)$. As soon as the equality is found, z will be equal to the number of voters "in favor" of the given proposal.

References

[1] Artem Chystiakov, Oleh Komendant, and Mariia Zhvanko. *Private Multisig*. 2025. URL: https://ethresear.ch/t/private-multisig-v0-1/23244.

[2] Dan Boneh and Matthew Franklin. *Identity Based Encryption From the Weil Pairing*. Cryptology ePrint Archive, Paper 2001/090. 2001. URL: https://eprint.iacr.org/2001/090.

[3] Ben Edgington. *BLS12-381 For The Rest Of Us*. 2025. URL: https://hackmd.io/@benjaminion/bls12-381.