Rusk: Dusk genesis circuits

Dusk Network

Victor Lopez <victor@dusk.network>

January 6, 2022

1. Constants

- G JubJub generator point
- G′ JubJub generator, $G' \neq G$
- I JubJub identity point

2. Functions

- H Hash to BLS12-381
- H' Hash to BLS12-381 truncated to 249 bits
- O Merkle opening over H

3. Gadgets

$$commitment(p, v, b, s) \rightarrow p == v \cdot \mathrm{G} + b \cdot \mathrm{G}' \wedge v < 2^s \tag{1}$$

$$schnorr(\sigma, k \cdot \mathrm{G}, m) \rightarrow \sigma = schnorrSign(k, m) \tag{2}$$

$$\sigma := doubleSchnorrSign(s, m, r) \rightarrow c := H'(r \cdot G, r \cdot G', m), u := r - c \cdot s, \sigma = (u, r \cdot G, r \cdot G') \tag{3}$$

$$doubleSchnorrVerify(\sigma, s \cdot G, m) -> c := H'(\sigma_R, \sigma_{R'}, m), \sigma_R = \sigma_u \cdot G + c \cdot \sigma_R \wedge \sigma_{R'} = \sigma_u \cdot G' + c \cdot \sigma_{R'} \tag{4}$$

$$opening(\mathbf{b}, r, l) \rightarrow O(\mathbf{b}) \wedge (\mathbf{b_0}, \mathbf{b}_{|\mathbf{b}|}) == (l, r) \tag{5}$$

$$s := selectPair(x, i, (a, b), (c, d)) \rightarrow x \in \{0, 1\} \wedge (s, i) == \begin{cases} ((a, b), (c, d)), & \text{if } x == 1. \\ ((c, d), (a, b)), & \text{if } x == 0. \end{cases} \tag{6}$$

$$s := stealthAddress(r, (a, b)) \rightarrow s = H'(r \cdot a) \cdot \mathrm{G} + b \tag{7}$$

$$\psi := encrypt(s, n, \mathbf{m}) \rightarrow \mathbf{m} == decrypt(s, n, \psi) \tag{8}$$

4. Execute

4.1. Structures

- $I = (t, v, b, c, n, s, r, p, \psi, h, o, \sigma)$ Input note
  - t Note type
  - v Value
  - b Blinder
  - c Value commitment
  - k Stealth address
  - k' Stealth address alternate
  - σ Double Schnorr proof
  - r Public entropy
  - p Position in the notes tree
  - n Encryption nonce
  - ψ Encryption cipher
  - h Note hash
  - o Merkle tree path
- $O = (v, b)$ Output note
  - v Value
  - b Blinder
- $C = (v, b, c)$ Crossover
  - v Value
  - b Blinder
  - c Value commitment

4.2. Private Inputs

- (C_v, C_b)
- $\mathbb{I}$ Set of input notes I
- $\mathbb{O}$ Set of input notes O

4.3. Public Inputs

- C_c
- A Notes tree Merkle anchor
- F Fee value
- $\mathbb{N}$ Set of nullifiers of $\mathbb{I}$
- $\mathbb{V}$ Set of value commitments of $\mathbb{O}$
- T Transaction hash

4.4. Circuit

1. $\forall(i, n) \in \mathbb{I} \times \mathbb{N} \mid \mathbb{I} \mapsto \mathbb{N}$
   (a) $\mathit{opening}(i_o, A, i_h)$
   (b) $i_h == H(i_t, i_c, i_n, i_k, i_r, i_p, i_\psi)$
   (c) $\mathit{doubleSchnorrVerify}(i_\sigma, i_k, T)$
   (d) $n == H(i_{k'}, i_p)$
   (e) $\mathit{commitment}(i_c, i_v, i_b, 64)$
2. $\mathit{commitment}(C_c, c_v, c_b, 64)$
3. $\forall(o, v) \in \mathbb{O} \times \mathbb{V} \mid \mathbb{O} \mapsto \mathbb{V}$
   (a) $\mathit{commitment}(v, o_v, o_b, 64)$
4. $\sum(i_v \in \mathbb{I}) - \sum(o_v \in \mathbb{O}) - C_v - F = 0$

5. Send to contract transparent

5.1. Structures

- $C = (v, b, c, n, \psi)$ Crossover
  - v Value
  - b Blinder
  - c Value commitment
  - n Encryption nonce
  - ψ Encryption cipher

5.2. Private Inputs

- (C_b, C_n, C_ψ)
- σ Schnorr signature
- A Contract address

5.3. Public Inputs

- (C_c, C_v)
- F_a Fee stealth address
- S Signed message

5.4. Circuit

1. $commitment(C_c, C_v, C_b, 64)$
2. $S == H(C_c, C_n, C_\psi, C_v, A)$
3. $schnorr(\sigma, F_a, S)$

6. Send to contract obfuscated

6.1. Structures

- $C = (v, b, c, n, \psi)$ Crossover
  - v Value
  - b Blinder
  - c Value commitment
  - n Encryption nonce
  - ψ Encryption cipher
- $M = (r, v, b, c, x, p, s, a, n, \psi)$ Message
  - r Entropy
  - v Value
  - b Blinder
  - c Value commitment
  - x Flag to use public derive key
  - p Public derive key pair
  - s Secret derive key pair
  - a Stealth address
  - n Encryption nonce
  - ψ Encryption cipher

6.2. Private Inputs

- v value
- $(C_b, M_r, M_b, M_x, M_s)$
- σ Schnorr signature

6.3. Public Inputs

- $(C_c, C_n, C_\psi, M_c, M_p, M_a, M_n, M_\psi)$
- A Contract address
- S Signed message
- F_a Fee stealth address

6.4. Circuit

1. $commitment(C_c, v, C_b, 64)$
2. $commitment(M_c, v, M_b, 64)$
3. $(p_a, p_b) := selectPair(M_x, \mathrm{I}, M_p, M_s)$
4. $M_a == stealthAddress(M_r, (p_a, p_b))$
5. $M_\psi == encrypt(M_r \cdot p_a, M_n, [v, M_b])$
6. $S == H(C_c, C_n, C_\psi, M_c, M_n, M_\psi, A)$
7. $schnorr(\sigma, F_a, S)$

7. Withdraw from transparent

7.1. Structures

- $N = (v, b, c)$ Phoenix note
  - v Value
  - b Blinder
  - c Value commitment

7.2. Private Inputs

- N_b

7.3. Public Inputs

- (N_v, N_c)

7.4. Circuit

1. $commitment(N_c, N_v, N_b, 64)$

8. Withdraw from obfuscated

8.1. Structures

- $I = (v, b, c)$ Input
  - v Value
  - b Blinder
  - c Value commitment
- $C = (r, v, b, c, x, p, s, a, n, \psi)$ Message change
  - r Entropy
  - v Value
  - b Blinder
  - c Value commitment
  - x Flag to use public derive key
  - p Public derive key pair
  - s Secret derive key pair
  - a Stealth address
  - n Encryption nonce
  - ψ Encryption cipher
- $O = (v, b, c)$ Output Phoenix note
  - v Value
  - b Blinder
  - c Value commitment

8.2. Private Inputs

- $(I_v, C_v, O_v, I_b, C_b, O_b, C_r, C_x, C_s)$

8.3. Public Inputs

- $(I_c, C_c, O_c, C_p, C_a, C_n, C_\psi)$

8.4. Circuit

1. $commitment(I_c, I_v, I_b, 64)$
2. $commitment(C_c, C_v, C_b, 64)$
3. $commitment(O_c, O_v, O_b, 64)$
4. $(p_a, p_b) := selectPair(C_x, \mathrm{I}, C_p, C_s)$
5. $C_a == stealthAddress(C_r, (p_a, p_b))$
6. $C_\psi == encrypt(C_r \cdot p_a, C_n, [C_v, C_b])$
7. $I_v - C_v - O_v == 0$