Protocol cheat sheet

Thomas Piellard

April 2019

1 Common setup and CRS

Parties using AZTEC protocol share the following setup:

- a prime p and $\mathbb{Z}_p$ (set of integers modulo p)
- $\mathbb{G}$,$\mathbb{G}_2$,$\mathbb{G}_T$, groups of order p
- g,g_2,g_T are the generators of $\mathbb{G}$,$\mathbb{G}_2$,$\mathbb{G}_T$
- a pairing $e : \mathbb{G} \times \mathbb{G}_2 \mapsto \mathbb{G}_T$ such that $e(g^a, g_2^b) = g_T^{ab}$ for all $a, b \in \mathbb{Z}$
- a Hash function H

Parties share a CRS, composed of:

- k_{max}, maximal value of a note
- $\mu_1, \ldots, \mu_{k_{max}}, h \in \mathbb{G}$
- $t \in \mathbb{G}_2$
- There exists $y \in \mathbb{Z}_p - \{0 \ldots k_{max}\}$ such that $\mu_k = h^{\frac{1}{y-k}}$ and $t = g_2^y$

2 Commitment to a value

The commitment to a value in $[1, k_{max}]$ is done like this:

$$Comm(k) = (\mu_k^a, \mu_k^{k.a} h^a) = (\gamma, \sigma)$$

where a is randomly chosen in $\mathbb{Z}_p$.

To check that a commitment (γ, σ) is valid one checks that equality

$$e(\gamma, t) = e(\sigma, g_2)$$

holds.

3 Proving and Verifying balance relation

Aztec deals with aztec notes. Each note is a tuple $((\gamma, \sigma), s, a)$ where a is the one used for the commitment of (γ, σ). a is the *viewing key*, s is the *spending key*.

Remark 1. k_{max} is chosen such that retrieving k from $((\gamma, \sigma), a)$ is fast enough (this is why a is called the *viewing key*), and such that retrieving k from (γ, σ) is hard.

JoinSplit can consume or create notes. A user wants to prove the following:

$$\sum_{i=1}^{m} k_i = \sum_{i=m+1}^{n} k_i + k_{public}$$

where k_i are the values of some notes, and k_{public} is a public input provided by the user. If $k_{public} > 0$ the user converts some notes in plain erc20 tokens, if $k = 0$ it is full confidential transaction, otherwise she converts plain erc20 tokens into aztec notes. Verification of balance is done with $P_{balance}$ (proof) and$V_{balance}$ (verfiying):

$P_{balance}((note_1 \dots note_n)$,$m$,$k_{public})$:

- Pick $b_{a_1}, (b_{a_2}, b_{k_2}), \dots (b_{a_n}, b_{k_n}) \in \mathbb{Z}_p$ (blinding parameters for a_i's and k_i's)
- Set $b_{k_1} = \sum_{i=m+1}^{n} b_{k_i} - \sum_{i=2}^{m} b_{k_i}$
- Compute $B_i = \gamma_i^{b_{k_i}} h^{b_{a_i}}$
- Compute a challenge $c = H((\gamma_i, \sigma_i), m, (B_i))$
- Compute $\hat{k_i} = ck_i + b_{k_i}$, $\hat{a_i} = ca_i + b_{a_i}$ (blinding of values k_i and viewing keys a_i)
- Return $\pi = (c, \hat{a_1}, (\hat{a_i}, \hat{k_i})_{i \in \{2,n\}})$

$V_{balance}((\gamma_i, \sigma_i), m, k_{public}, \pi)$

- Set $\hat{k_1} = \sum_{i=m+1}^{n} k_i - \sum_{i=2}^{m} k_i + k_{pubic}.c$
- Compute $B_i = \gamma_i^{\hat{k_i}} h^{\hat{a_i}} \sigma_i^{-c}$
- Return 1 if $c = H((\gamma_i, \sigma_i), m, (B_i))$

Remark 2. To check why it works one notices that $\gamma_i^{k_i} h^{a_i} = \sigma_i$. So

$$\gamma_i^{\hat{k_i}} h^{\hat{a_i}} = \gamma_i^{ck_i + b_{k_i}} h^{ca_i + b_{a_i}} = (\gamma_i^{k_i} h^{a_i})^c . \gamma_i^{b_{k_i}} h^{b_{a_i}} = \sigma_i^c \gamma_i^{b_{k_i}} h^{b_{a_i}}$$

Remark 3. To create a proof one must have access to the viewing key: only the note owners (next section) can create proofs.

4 Ownership

4.1 Note owner

An aztec note belongs to someone. To encode this, when creating an aztec note the following happens:

- One specifies the note owner's public key k
- The creator of the note generates an ephemeral key pair $(Pub, Priv)$
- The viewing key is the result of a Diffie Hellman $shared = DH(k, Priv)$

The note contains only Pub and $H(shared)$. Only the note owner can derive the viewing key.

4.2 Spending key

To spend a note, one must have:

- The viewing key to construct the proof therefore be the note owner
- The spending key to sign the note

For the moment, the spending key is the private key associated to the note owner's public key, but a stealth address protocol is on the way.

The spending key and owner public key are standard ethereum public/private key (on secp256k1).