

# Platin Privacy Protocol

## Private location verification with incentivization

in-progress

Vadym Fedyukovych

<https://platin.io/>

June 12, 2018

### Abstract

We design an interactive proof system for “location is close enough” statement, in the form of specification for further implementation.

## 1 Definitions

Node proves the statement “distance is within a threshold” (less or equal) for node coordinates  $(x_n, y_n, z_n)$ , given location  $(x_l, y_l, z_l)$ , and some threshold  $d$  (all integers):

$$d^2 - ((x_n - x_l)^2 + (y_n - y_l)^2 + (z_n - z_l)^2) = a_1^2 + a_2^2 + a_3^2 + a_4^2 \quad (1)$$

We rely on 4-squares Lagrange theorem to prove equality statement (Lipmaa). Proofs for integer relations are possible in hidden group order setup (Camenisch-Stadler).

## 2 Proof setup

Let  $g$  be a generator of a proper group of a hidden order, and  $h$  be a group element (Pedersen commitment scheme). We use multiplicative group of invertible residue classes modulo a composite  $n$  such that  $n = pq$ ,  $p = 2p' + 1$ ,  $q = 2q' + 1$  and  $p, q, p', q'$  primes (Idemix).

### 3 Signals harvesting

Node picks random  $(r_x, r_y, r_z)$ , creates commitment  $(s_x, s_y, s_z)$  to it's coordinates

$$s_x = g^{x_n} h^{r_x}, \quad s_y = g^{y_n} h^{r_y}, \quad s_z = g^{z_n} h^{r_z} \quad (2)$$

and keeps coordinates-randoms pairs  $(x_n, y_n, z_n), (r_x, r_y, r_z)$  private.

#### 3.1 Representation-based commitment

Following U-Prove, consider commitment scheme resulting in a single group element:

$$s_U = g_x^{x_n} g_y^{y_n} g_z^{z_n} h^r \quad (3)$$

where  $g_x, g_y, g_z$  are group elements, and  $r$  is a random. This scheme admits a proof of knowledge with the same responses required form threshold location verification. This scheme can be extended with additional components.

#### 3.2 Two-level commitment

To achieve expected properties of Merkle-tree based scheme while keeping an option to run location proof protocols, representation-based commitments could be leaves of Merkle tree.

## 4 Proof

Sigma-protocol with 3 messages. Public information is node location commitment, given location, threshold, proof parameters. Private information is node location and randomness to commitment.

1. Prover (node) picks random  $\alpha_j, \beta_x, \beta_y, \beta_z, \rho_0, \rho_1, \eta_x, \eta_y, \eta_z$  and computes initial commitments  $b_0, b_1, t_x, t_y, t_z$  ( $f_0$  and  $f_1$  explained at Background section)

$$b_0 = g^{f_0} h^{\rho_0}, \quad b_1 = g^{f_1} h^{\rho_1} \quad (4)$$

$$t_x = g^{\beta_x} h^{\eta_x}, \quad t_y = g^{\beta_y} h^{\eta_y}, \quad t_z = g^{\beta_z} h^{\eta_z} \quad (5)$$

2. Challenge  $c$

### 3. Prover computes responses

$$X_n = cx_n + \beta_x, \quad Y_n = cy_n + \beta_y, \quad Z_n = cz_n + \beta_z \quad (6)$$

$$R_x = cr_x + \eta_x, \quad R_y = cr_y + \eta_y, \quad R_z = cr_z + \eta_z \quad (7)$$

$$A_j = ca_j + \alpha_j, \quad j = 1..4 \quad (8)$$

$$R_a = c\rho_1 + \rho_0 \quad (9)$$

### 4. Proof verification

$$g^{c^2d - ((X_n - cx_l)^2 + (Y_n - cy_l)^2 + (Z_n - cz_l)^2) - (A_1^2 + A_2^2 + A_3^2 + A_4^2)} h^{R_a} = b_1^c b_0 \quad (10)$$

$$g^{X_n} h^{R_x} s_x^{-c} = t_x, \quad g^{Y_n} h^{R_y} s_y^{-c} = t_y, \quad g^{Z_n} h^{R_z} s_z^{-c} = t_z \quad (11)$$

## 5 Background

Consider quadratic (degree 2 in  $v$ ) polynomial

$$\begin{aligned} f_V(v) = f_2 v^2 + f_1 v + f_0 = \\ v^2 d - (((vx_n + \beta_x) - vx_l)^2 + ((vy_n + \beta_y) - vy_l)^2 + ((vz_n + \beta_z) - vz_l)^2) \\ - ((va_1 + \alpha_1)^2 + (va_2 + \alpha_2)^2 + (va_3 + \alpha_3)^2 + (va_4 + \alpha_4)^2) \end{aligned} \quad (12)$$

This polynomial is actually linear ( $f_2 = 0$ , degree-one in  $v$ ) if, and only if statement about distance (1) holds for node coordinates that are hidden from verifier. We evaluate this polynomial at a random point chosen as the challenge of verifier. It follows, distance verification equation (10) only needs constant  $b_0$  and degree-one  $b_1^c$  components.

$$f_0 = -\beta_x^2 - \beta_y^2 - \beta_z^2 - \alpha_1^2 - \alpha_2^2 - \alpha_3^2 - \alpha_4^2 \quad (13)$$

$$\begin{aligned} f_1 = -2(x_n - x_l)\beta_x - 2(y_n - y_l)\beta_y - 2(z_n - z_l)\beta_z \\ - 2a_1\alpha_1 - 2a_2\alpha_2 - 2a_3\alpha_3 - 2a_4\alpha_4 \end{aligned} \quad (14)$$

Prover calculates  $b_0, b_1$  from  $f_0, f_1$ .

## 6 Not-at-location proof

Proving a negative location statement is a valid usecase, that could be demonstrated with “not at the grocery store” scenario. Rather than proving “distance is smaller than” (1), complementary “is larger”

proof is given. In the following, we only show changes required to the main protocol.

$$((x_n - x_l)^2 + (y_n - y_l)^2 + (z_n - z_l)^2) - d^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2 \quad (15)$$

$$g^{((X_n - cx_l)^2 + (Y_n - cy_l)^2 + (Z_n - cz_l)^2) - c^2 d - (A_1^2 + A_2^2 + A_3^2 + A_4^2)} h^{R_a} = b_1^c b_0 \quad (16)$$

$$\begin{aligned} f_V(v) &= f_2 v^2 + f_1 v + f_0 = \\ &(((vx_n + \beta_x) - vx_l)^2 + ((vy_n + \beta_y) - vy_l)^2 + ((vz_n + \beta_z) - vz_l)^2) - v^2 d \\ &\quad - ((va_1 + \alpha_1)^2 + (va_2 + \alpha_2)^2 + (va_3 + \alpha_3)^2 + (va_4 + \alpha_4)^2) \end{aligned} \quad (17)$$

## 7 Logical-OR threshold location

Consider a franchise operating multiple stores, and a usecase of proving location is “at Starbucks” without telling which one of  $K$  known. We define each such store with its center  $(x_k, y_k, z_k)$  and radius (size)  $d_k$ ,  $k \in [1..K]$ . We elaborate basic threshold proof such that prover can produce 4-squares representation for center-size of some store  $k = p$ , and pick arbitrary 4-tuples for all other stores  $k \neq p$ .

$$\begin{aligned} \prod_{k=1}^K ((d_k^2 - ((x_n - x_k)^2 + (y_n - y_k)^2 + (z_n - z_k)^2) \\ - (a_{1,k}^2 + a_{2,k}^2 + a_{3,k}^2 + a_{4,k}^2)) = 0 \end{aligned} \quad (18)$$

Verifier is testing that polynomial  $f_{KV}(v)$  is of degree  $2K - 1$ , not  $2K$ .

$$\begin{aligned} f_{KV}(v) &= \sum_{j=0}^{2K} f_j v^j = \\ \prod_{k=1}^K (v^2 d_k - (((vx_n + \beta_x) - vx_k)^2 + ((vy_n + \beta_y) - vy_k)^2 + ((vz_n + \beta_z) - vz_k)^2) \\ &\quad - ((va_{1,k} + \alpha_1)^2 + (va_{2,k} + \alpha_2)^2 + (va_{3,k} + \alpha_3)^2 + (va_{4,k} + \alpha_4)^2)) \end{aligned} \quad (19)$$

## 8 Private incentivization

To engage users, verifier (map service) is giving tokens in exchange for verifying location proofs. To keep privacy of users, proof verification and token issuance are separated with intermediate tokens. Intermediate tokens should unlink locations that users were confirming.

Intermediate token is a Schnorr proof instance, non-interactive variant (Fiat-Shamir), in another group of a known prime order  $q$

generated by  $g_i$ . Issuing (private) key  $x_i$  and public key  $X_i$  of map service.

A new blinded intermediate token is sent to user on each successful location verification. Users unblind their intermediate tokens and periodically exchange them for Platin tokens.

1. Issuer (map service) chooses random  $\gamma$  and sends  $\tilde{w}$  to Recipient

$$\tilde{w} = g_i^\gamma \quad (20)$$

2. Recipient (node) chooses random blinding  $(\delta, \mu)$ , produces blinded challenge  $\tilde{c}_i$  and sends it to Issuer

$$w = \tilde{w} g_i^{-\delta} X_i^\mu \quad (21)$$

$$\tilde{c}_i = H(w) + \mu \pmod{q} \quad (22)$$

3. Issuer produces blinded response  $\tilde{r}_i$  and sends it to Recipient

$$\tilde{r}_i = \tilde{c}_i x_i + \gamma \pmod{q} \quad (23)$$

4. Recipient verifies validity of response received and unblinds challenge and response

$$\tilde{c}_i = H(g_i^{\tilde{r}_i} X_i^{-\tilde{c}_i}) \quad (24)$$

$$c_i = \tilde{c}_i - \mu \pmod{q} \quad (25)$$

$$r_i = \tilde{r}_i - \delta \pmod{q} \quad (26)$$

Intermediate token is  $(c_i, r_i)$ .

5. Intermediate token verification (while exchange for Platin tokens)

$$c_i = H(g_i^{r_i} X_i^{-c_i}) \quad (27)$$

## 9 Intermediate token properties

Both components of intermediate token are statistically (unconditionally) independent from blinded intermediate token.

Consider an Adversary trying to distinguish two pairs of unblinded intermediate tokens  $(c_1, r_1)$  and  $(c_2, r_2)$ , matching them to issuing session identified with  $\tilde{w}$ . View of Issuer is  $(\tilde{c}_i, \tilde{r}_i, c_j, r_j)$  for  $j = 1, 2$ .

$$g_i^{r_j} X_i^{-c_i} \stackrel{?}{=} \tilde{w} g_i^{-\delta_j} X_i^{\mu_j} = \tilde{w} g_i^{r_j - \tilde{r}_i} X_i^{\tilde{c}_i - c_j} = (\tilde{w} g_i^{-\tilde{r}_i} X_i^{\tilde{c}_i}) (g_i^{r_j} X_i^{-c_j}) \quad (28)$$

It could be seen that the only possible relation does not depend on issuing session identifier  $\tilde{w}$ , resulting in unconditional privacy for intermediate tokens.

## 10 Platin Encoding Standard 01

With the scenario of user device (node) being close enough to a cell tower from sections 1 and 4, we define encoding of coordinates, commitment, witness and proof of location.

### 10.1 Non-interactive proof

Fiat-Shamir technique could be applied to produce a challenge  $c$  with proper hash function from concatenation of group elements  $t_x, t_y, t_z, b_1, b_0$  from first step of interactive protocol. Non-interactive proof verification would be re-creating most of group elements and re-producing the challenge:

$$c = H(g^{X_n} h^{R_x} s_x^{-c} || g^{Y_n} h^{R_y} s_y^{-c} || g^{Z_n} h^{R_z} s_z^{-c} || b_1 || g^{c^2 d - ((X_n - cx_l)^2 + (Y_n - cy_l)^2 + (Z_n - cz_l)^2) - (A_1^2 + A_2^2 + A_3^2 + A_4^2)} h^{R_a} b_1^{-c}) \quad (29)$$

Non-interactive proof is

$$(c, (X_n, Y_n, Z_n, A_1, A_2, A_3, A_4), b_1) \quad (30)$$

with group element  $b_1$  and 8 integers. With this proof, multiplicative group is used, so group elements are represented as integers.

### 10.2 Plaintext encoding of non-interactive proof

Popular bignumber libraries have plaintext hex representation for bignumbers, designed to inter-operate with each other. Non-interactive proof is encoded as 9-lines, with a large number on each line, in hex encoding, in the order given by (30).

### 10.3 Plaintext encoding of location commitment

U-Prove style commitment (3) is a single group element, represented as a single-line number in hex encoding.

### 10.4 Plaintext encoding of location witness

Witness is a common term for private data held by prover that allows to verify validity of a statement, open a commitment instance (equation (3)), or run a proof protocol (section 4).

Witness is represented as four lines plaintext integer numbers; 3 lines with  $x_n, y_n, z_n$  in decimal, and a big random number  $r$  in hex encoding.

Additional locations that are input to the protocol like coordinates of cell tower  $x_l, y_l, z_l$  are encoded the same as witness: 3 lines, integer numbers, decimal notation.

## 10.5 Plaintext encoding of group elements

Group is specified with two bignumbers, composite modulo  $n$  and generator  $g$ , as two lines plaintext, with hex encoding. Additional group elements required to produce commitment (3) are encoded on following lines the same way.

## 11 Platin Encoding Standard 02

Geocoordinates are generally represented as latitude-longitude-elevation triple, that could be further converted into some local x-y-z coordinate system. This choice of coordinates apply globally, while PES 01 better fit local (like city-scale) statements. To commit to lat-lon-el triple, we amend two-level commitment approach by converting GPS data into integers and commit with the U-Prove-style scheme (3). We commit to a fixed set of locations with Merkle tree, with U-Prove commitments considered leaves of the tree.

This two-level scheme requires storage of

1. Original raw lat-lon-el data and scales (multipliers) used to convert them to integers;
2. Random integer used to produce location commitment;
3. Commitments (Merkle leaves) and intermediate hashes of the tree;
4. Merkle root as commitment to the set of location.

Further design decisions would be taken to account for project ideas other than location proofs, and for implementation-level needs. Location security policy apply for original data and commitment random input.