Zero Knowledge Proof of Location Platin Yellow Paper Draft

Lionel Wolberger, Vadym Fedyukovych October 25, 2018

Abstract

Many location based services authorize a user by assessing whether or not the user is within a given range of the service. To assess this range, systems request the user's geospatial coordinates, and often store them for later analysis. We describe a system where the service authorization is based on a zero knowledge verification of a commitment. The commitment has no geographical coordinate data, yet can be reliably verified to prove that the user is within range of the service eligibility. The service has the assurance required to deliver the service while having zero knowledge of the user's geographical coordinates.

1 Zero Knowledge Proof of Location

We present two sets of equations with a graphical illustration followed by notes and decisions.

The illustration describes a location based service with a radius that defines the range of that service. Two users are shown colored green and red, the green user inside the radius and the red user, outside. The diagram illustrates parameters that are significant to the zero knowledge proof.

The equations are presented in two sets and specify the steps required to perform the full protocol, from commitment to verification. This protocol allows a verifier to test whether position committed is inside or outside the radius of the service area. Both interactive and non-interactive protocols are defined, each presented as a series of equations.

The notes and decisions section records issues related to the protocol and illustration. Each note discusses a mathematical decision that we have made. Some of these decisions are not yet reflected in the equations and protocols shared in this paper.

A git repository is associated with this paper. C++ reference code can be found there enabling testing and efficiency metrics.

This zero knowledge proof of location protocol is sufficient for many use cases, efficient, supports large scale analytics, and preserves users' privacy.

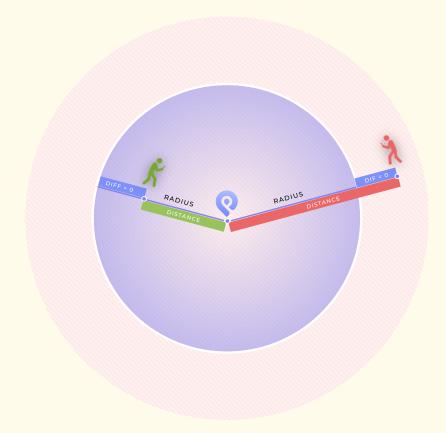


Figure 1: Two users with a predefined radius. green node and a red node are positioned in relation to a circle with a predefined radius. The Pan airdrop is described by circular geometry with a center and radius. A node has three significant parameters: a distance, radius and difference.

Common input of Prover and Verifier is commitment s_U to node location (1), airdrop location (x_l, y_l, z_l) and threshold d^2 (integers), parameters $(N, g, g_x, g_y, g_z, g_r, \{h_j\})$.

$$s_U = g_x^{x_n} g_y^{y_n} g_z^{z_n} g^r \pmod{N} \tag{1}$$

Private input of Prover is node location (x_n, y_n, z_n) (integers) and location commitment randomness r, four numbers $\{a_j\}$ calculated according to (2). Statement being proved is

$$d^{2} - ((x_{n} - x_{l})^{2} + (y_{n} - y_{l})^{2}) + (z_{n} - z_{l})^{2}) = \sum_{j=1}^{4} a_{j}^{2}$$
 (2)

Protocol runs as follows:

1. Prover picks random $\{\alpha_j\}$, η , γ , β_x , β_y , β_z , β_r , ρ_0 , ρ_1 , produces f_0 , f_1 , sends b_0 , b_1 , t_a , s_a , t_n :

$$f_0 = -(\beta_x^2 + \beta_y^2 + \beta_z^2) - \sum_{j=1}^4 \alpha_j^2$$
 (3)

$$f_1 = -((x_n - x_l)\beta_x + (y_n - y_l)\beta_y + (z_n - z_l)\beta_z) - \sum_{i=1}^4 a_i \alpha_i$$
 (4)

$$t_n = g_x^{\beta_x} g_y^{\beta_y} g_z^{\beta_z} g^{\beta_r}, \ s_a = g^{\gamma} \prod_{i=1}^4 h_j^{a_j}, \ t_a = g^{\eta} \prod_{i=1}^4 h_j^{\alpha_j}$$
 (5)

$$b_0 = g^{f_0} g_r^{\rho_0}, \ b_1 = g^{2f_1} g_r^{\rho_1} \pmod{N}$$
 (6)

- 2. Verifier chooses and sends his challenge c
- 3. Prover produces and sends responses

$$X_n = cx_n + \beta_x, \ Y_n = cy_n + \beta_y, \ Z_n = cz_n + \beta_z, \ R = cr + \beta_r$$
 (7)
 $A_j = ca_j + \alpha_j, \ R_a = c\gamma + \eta, \ R_d = c\rho_1 + \rho_0$ (8)

4. Verifier accepts if

$$g_x^{X_n} g_y^{Y_n} g_z^{Z_n} g^R s_U^{-c} = t_n, \quad g^{R_a} \left(\prod_{j=1}^4 h_j^{A_j} \right) s_a^{-c} = t_a$$
 (9)

$$g^{c^2d^2 - ((X_n - cx_l)^2 + (Y_n - cy_l)^2 + (Z_n - cz_l)^2) - (A_1^2 + A_2^2 + A_3^2 + A_4^2)}g_r^{R_d} = b_1^c b_0 \pmod{N}$$
(10)

Figure 2: Private location verification protocol, interactive version

Input of Prover is location commitment s_U (1), location (x_n, y_n, z_n) and random r to open this commitment, airdrop location (x_l, y_l, z_l) , threshold d^2 , parameres $(N, g, g_x, g_y, g_z, g_r, h_j)$ and public information pubp.

Non-interactive proof is produced as follows:

1. Prover calculates $a_1
ldots a_4$ from locations and threshold, picks random $\{\alpha_j\}, \eta, \gamma, \beta_x, \beta_y, \beta_z, \beta_r, \rho_0, \rho_1, \text{ produces } t_n, s_a, t_a, b_0, b_1$:

$$t_n = g_x^{\beta_x} g_y^{\beta_y} g_z^{\beta_z} g^{\beta_r}, \ s_a = g^{\gamma} (\prod_{j=1}^4 h_j^{a_j}), \ t_a = g^{\eta} (\prod_{j=1}^4 h_j^{\alpha_j}) \pmod{N}$$
 (11)

$$\tilde{f}_0 = \beta_x^2 + \beta_y^2 + \beta_z^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 \tag{12}$$

$$\tilde{f}_1 = (x_n - x_l)\beta_x + (y_n - y_l)\beta_y + (z_n - z_l)\beta_z + a_1\alpha_1 + a_2\alpha_2 + a_3\alpha_3 + a_4\alpha_4$$
(13)

$$b_0 = g^{\tilde{f}_0} g_r^{\rho_0}, \quad b_1 = g^{2\tilde{f}_1} g_r^{\rho_1} \pmod{N}$$
 (14)

2. Prover produces his challenge with a hash function from text representation of commitments generated at previous step and public information:

$$c = H(t_n||s_a||t_a||b_1||b_0||s_U||pubp)$$
(15)

3. Prover produces responses:

$$X_n = -cx_n + \beta_x, \ Y_n = -cy_n + \beta_y, \ Z_n = -cz_n + \beta_z, \ R = -cr + \beta_r$$

 $A_j = -ca_j + \alpha_j, \ R_a = -c\gamma + \eta, \ R_d = -c\rho_1 + \rho_0$ (16)

Non-interactive proof is $(c, X_n, Y_n, Z_n, R, \{A_j\}, R_a, R_d, s_a, b_1)$. Proof verification:

$$F_{d} = ((X_{n} + cx_{l})^{2} + (Y_{n} + cy_{l})^{2} + (Z_{n} + cz_{l})^{2}) + (A_{1}^{2} + A_{2}^{2} + A_{3}^{2} + A_{4}^{2}) - c^{2}d^{2}$$

$$H(g_{x}^{X_{n}}g_{y}^{Y_{n}}g_{z}^{Z_{n}}g^{R}s_{U}^{c}||s_{a}||g^{R_{a}}(\prod_{j=1}^{4}h_{j}^{A_{j}})s_{a}^{c}||b_{1}||g^{F_{d}}g_{r}^{R_{d}}b_{1}^{c}||s_{U}||pubp) = c \quad (17)$$

Figure 3: Location proof generation and verification, non-interactive version

2 Notes and Decisions

Notes and decisions capture issues related to zero knowledge proofs of location and are presented in no particular order. Some of the topics discussed relate to our general approach to zero knowledge proofs, and do not directly reflect the equations or protocols above. The contents of notes and decisions may be integrated with the main body of the paper

in future.

2.1 Some 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):

$$d^{2} - ((x_{n} - x_{l})^{2} + (y_{n} - y_{l})^{2} + (z_{n} - z_{l})^{2}) = a_{1}^{2} + a_{2}^{2} + a_{3}^{2} + a_{4}^{2}$$
 (18)

We rely on 4-squares Lagrange theorem to prove equality statement [2]. Proofs for integer relations are possible in hidden group order setup.

2.2 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 [1].

2.3 Representation-Based Commitment

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

$$s_U = g_x^{x_n} g_y^{y_n} g_z^{z_n} g^r (19)$$

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.

2.4 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.

2.5 A 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 that proving "distance is smaller than" (18), complementary "is larger" proof is given. In the following, we only show changes required to the main protocol.

$$((x_n - x_l)^2 + (y_n - y_l)^2 + (z_n - z_l)^2) - d^2 = a_1^2 + a_2^2 + a_3^2 + a_4^2$$
 (20)

$$g^{((X_n - cx_l)^2 + (Y_n - cy_l)^2 + (Z_n - cz_l)^2) - c^2 d^2 - (A_1^2 + A_2^2 + A_3^2 + A_4^2)} h^{R_a} = b_1^c b_0$$
 (21)

$$f_{V}(v) = f_{2}v^{2} + f_{1}v + f_{0} =$$

$$(((vx_{n} + \beta_{x}) - vx_{l})^{2} + ((vy_{n} + \beta_{y}) - vy_{l})^{2} + ((vz_{n} + \beta_{z}) - vz_{l})^{2}) - v^{2}d^{2}$$

$$- ((va_{1} + \alpha_{1})^{2} + (va_{2} + \alpha_{2})^{2} + (va_{3} + \alpha_{3})^{2} + (va_{4} + \alpha_{4})^{2})$$
(22)

2.6 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 it's 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 $k \neq p$.

$$\prod_{k=1} ((d_k^2 - ((x_n - x_k)^2 + (y_n - y_k)^2 + (z_n - z_k)^2) - (a_{1,k}^2 + a_{2,k}^2 + a_{3,k}^2 + a_{4,k}^2)) = 0 \quad (23)$$

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

$$f_{KV}(v) = \sum_{j=0}^{2K} f_j v^j = \prod_{k=1}^{K} (v^2 d_k^2 - (((vx_n + \beta_x) - vx_k)^2 + ((vy_n + \beta_y) - vy_k)^2 + ((vz_n + \beta_z) - vz_k)^2) - ((va_{1,k} + \alpha_1)^2 + (va_{2,k} + \alpha_2)^2 + (va_{3,k} + \alpha_3)^2 + (va_{4,k} + \alpha_4)^2))$$
 (24)

References

- [1] Jan Camenisch. Specification of the identity mixer cryptographic library RZ 3730 version 2.3.0, 2010.
- [2] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In ASIACRYPT, volume 2894 of Lecture Notes in Computer Science, pages 398–415. Springer, 2003.