Zero Knowledge Proof of Location

Platin Yellow Paper
Draft

Lionel Wolberger, Vadym Fedyukovych

October 24, 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 geographical 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

The zero knowledge location We present a series of equations illustrated with a diagram and followed by notes.

The diagram This introduction describes a circle with geometric radius R. A node positioned inside the circle can calculate a mathematical commitment and share it. In the "green" cases the difference is greater than zero, in the "red" cases it is less than zero.

The equations outline all steps required to perform the full procedure, from commitment to verification. A protocol is specified where a verifier can process that commitment and determine the difference. A non-interactive and interactive protocol is defined. This proof is sufficient for many use cases, efficient, supports large scale analytics, and preserves users' privacy.

The notes can be read separately from the equations. Each note discusses a mathematical decision that we have made. Some of these decisions are not yet reflected in the equations shared in this paper.

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

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

$$d^2 - ((x_n - x_l)^2 + (y_n - y_l)^2) + (z_n - z_l)^2) = \sum_{j=1}^{4} a_j^2 \tag{2}$$

Protocol runs as follows:

1. Prover picks random $\{\alpha_j\}, \eta, \gamma, \beta_x, \beta_y, \beta_z, \beta_r, \rho_0, \rho_1$, produces f_0, f_1, sends b_0, b_1, t_a, s_a, t_n:

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

$$f_1 = -((x_n - x_l)\beta_x + (y_n - y_l)\beta_y + (z_n - z_l)\beta_z) - \sum_{j=1}^{4} a_j \alpha_j \tag{4}$$

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

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

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

$$X_n = cx_n + \beta_x,\ Y_n = cy_n + \beta_y,\ Z_n = cz_n + \beta_z,\ R = cr + \beta_r \tag{7}$$

$$A_j = ca_j + \alpha_j,\ R_a = c\gamma + \eta,\ R_d = c\rho_1 + \rho_0 \tag{8}$$

4. Verifier accepts if

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

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

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

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

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

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

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

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

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

3. Prover produces responses:

$$X_n = -cx_n + \beta_x,\ Y_n = -cy_n + \beta_y,\ Z_n = -cz_n + \beta_z,\ R = -cr + \beta_r$$
$$A_j = -ca_j + \alpha_j,\ R_a = -c\gamma + \eta,\ R_d = -c\rho_1 + \rho_0 \quad (16)$$

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 2: Location proof generation and verification, non-interactive version

1.1 Diagram

1.2 Non-Interactive proof

1.3 Security properties

2 Implementation

We have this protocol implemented on top of Crypto++ library [1] serving as a bignumbers backend.

[1] `https://cryptopp.com/`

Producing four-squares witness [6] is a work in progress, and is not a part of the protocol reported. To facilitate larger proof-of-concept application, a temporary approximate solution was introduced producing four squares.

[1]

3 Discussion and Conclusion

We have shown how a node can keep its location secret, yet allow a verifier to prove that the secret location is within a perimeter. We offer two methods to do so: interactive and non-interactive.

References

[1] Jan Camenisch. Specification of the identity mixer cryptographic library RZ 3730 version 2.3.0, 2010.

[2] Giovanni Di Crescenzo and Vadym Fedyukovych. Zero-knowledge proofs via polynomial representations. In *Proceedings of the 37th International Conference on Mathematical Foundations of Computer Science*, pages 335–347, 2012.

[3] DScout. Mobile touches. Accessed Online October 2018, 2018.

[4] Vadym Fedyukovych. An argument for Hamiltonicity. Cryptology ePrint Archive, Report 2008/363, 2008.

[5] Vadym Fedyukovych. Proving outcome of private statistical signal testing. In *Statistical Methods of Signal and Data Processing*, pages 172–175, 2010.

[6] Paul Pollack and Enrique Trevino. Finding four squares in Lagrange's theorem.

[7] Lionel et al. Wolberger. Engineering privacy for verified credentials: In which we describe data minimization, selective disclosure, and progressive trust. Credentials Community Group Report 2018, 2018.