Private Location Verification

Lionel Wolberger, Vadym Fedyukovych

October 2, 2018

Abstract

Location based services (LBS) rely on users' disclosing their location. A zero knowledge procedure is proposed where a user provides a commitment that can be verified to show that the user is within a perimeter.

The simplest perimeter is a circle with radius R. The simplest polygon is a triangle. The procedure is then shown to work for an arbitrary perimeter consisting of n points.

This proof is sufficient for many LBS, and preserves the users privacy.

1 Introduction

Location based services (LBS) are a multimillion dollar market. An LBS generally relies on a user's smartphone to disclose its location data, as people carry these devices with them almost everywhere, use them over four hours each day on average, and touch them over two thousand times a day [**?**]

While LBS are increasingly popular and expanding into many areas of our daily lives, there is a parallel and counter trend towards increasing privacy. Privacy preserving tactics such as Data Minimization, Selective Disclosure and Progressive Trust are increasingly expected to be applied to our digital interactions. [**?**]

For example, a user may attempt to use an online service that asks him to share his location in order to prove his geolocation. The user may hesitate, since he knows that the service may share this information with other parties without meaningful consent on his part.

Some LBS may not need a precise location, but only an assertion that the user is within a certain area.

This paper details a zero knowledge procedure to enable such assertions. The user would share a cryptographic commitment rather than his exact location. The commitment is validated, and a proof is given that Diego is within a perimeter, without revealing the secret of his actual location within this perimeter.

This zero knowledge procedure is described below.

1.1 Our contribution

2 Protocol

2.1 Definitions

2.2 Notations

A location based service publishes a location of interest. The location is bounded within a perimeter, or boundary.

A simple case is that of a circle with a central point and a radius.

The center of the circle is shared in the clear, (x_l, y_l, z_l). The radius of interest about this point is shared as a maximum distance d from (x_l, y_l, z_l).

The user's smartphone device or node is at location (x_n, y_n, z_n). The node keeps these coordinates a secret. It shares a cryptographic commitment that can prove its location is within the geometry in question. The procedure's setup requires a group description and parameters, selection of group elements for making commitments, an initial message, and for the interactive version a challenge and response of a Schnorr-like protocol.

We then describe a non-interactive version. In this version we replace the challenge/response by using Fiat Shamir.

2.3 Interactive proof

This section presents the interactive version of the zero knowledge proof.

The location geometry is considered as a range, and we have only to prove that the node is within the range. We rely on Lagrange's classic result that every non-negative integer is a sum of four squares. Lipmaa extended this into a non-interactive range proof.

A mathematical group is constructed that can support an interactive query with integers. The order of the group is kept secret and remains unknown to the Prover.

Schnorr proof was extended into a proof systems for polynomial relations with polynomials of higher degree in challenge for a number of applications [2, 1].

A comparable proof system for integers was introduced at [3].

2.4 Proof setup

We restrict ourselves to the use of integers, following the practice of Idemix [**?**].

We setup a group based on quadratic residues, a property of numbers raised to the second power (squared) as they behave in modular arithmetic. The modulus is generated based on a safe RSA key pair. This constitutes a multiplicative group of residue classes, with an RSA-like modulus.

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

2.5 Non-Interactive proof

2.6 Security properties

3 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 [4] 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.

[**?**]

4 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] 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.

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

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

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