Zero Knowledge Proof of Location

Platin ZK Yellow Paper
Draft

L. Wolberger[*], V. Fedyukovych[*]

November 5, 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 geospatial 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 geospatial 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 associated with this paper will be released. C++ reference code will 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.

[*] Platin.io, with contributions from A. Mason, M. Tiutin and Y. Semelyak

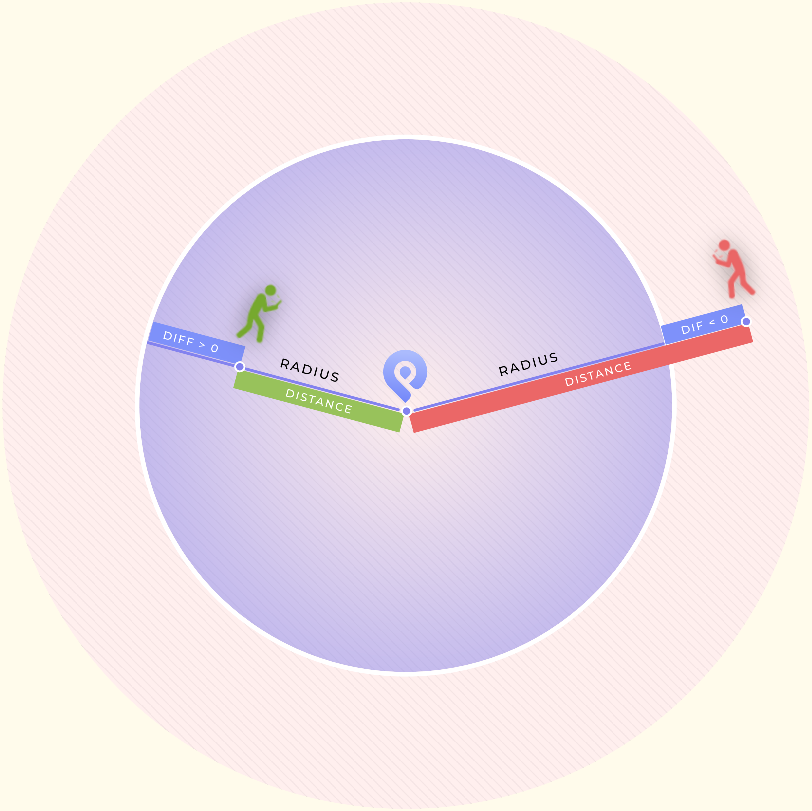

Figure 1: A location based service is shown. The center of the circle and its preset radius define the range within which users are to be authorized for the service. Two users request authorization, colored green and red. With data known only to itself, each user calculates a mathematical commitment based on radius, distance and difference. For green the difference is greater than zero. For red the difference is less than zero. The mathematical commitment is verified by applying the zero knowledge protocol below, without revealing the user's geospatial coordinates.

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 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 $\{\alpha_j\}, \eta, \gamma, \beta_x, \beta_y, \beta_z, \beta_r, \rho_0, \rho_1$, produces t_n, s_a, t_a, b_0, b_1:

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

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

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

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

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

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

3. Prover produces responses:

$$\tilde{X}_n = -cx_n + \beta_x,\ \tilde{Y}_n = -cy_n + \beta_y,\ \tilde{Z}_n = -cz_n + \beta_z,\ \tilde{R} = -cr + \beta_r$$
$$\tilde{A}_j = -ca_j + \alpha_j,\ \tilde{R}_a = -c\gamma + \eta,\ \tilde{R}_d = -c\rho_1 + \rho_0 \tag{16}$$

Non-interactive proof is $(c, \tilde{X}_n, \tilde{Y}_n, \tilde{Z}_n, \tilde{R}, \{\tilde{A}_j\}, \tilde{R}_a, \tilde{R}_d, s_a, b_1)$.
Proof verification:
Verifier produces F_d and then re-produces the challenge as follows

$$F_d = ((\tilde{X}_n + cx_l)^2 + (\tilde{Y}_n + cy_l)^2 + (\tilde{Z}_n + cz_l)^2) + (\tilde{A}_1^2 + \tilde{A}_2^2 + \tilde{A}_3^2 + \tilde{A}_4^2) - c^2 d^2$$
$$H(g_x^{\tilde{X}_n} g_y^{\tilde{Y}_n} g_z^{\tilde{Z}_n} g^{\tilde{R}} s_U^c || s_a || g^{\tilde{R}_a} (\prod_{j=1}^{4} h_j^{\tilde{A}_j}) s_a^c || b_1 || g^{F_d} g_r^{\tilde{R}_d} b_1^c || s_U || pubp) = c \tag{17}$$

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

1. Statement of the problem. Let EMBED Equation.DSMT4 , EMBED Equation.DSMT4 be two points on the plane or in the space. One of these points, EMBED Equation.DSMT4 , is some reference point: well-known landmark, or monument, or market, or other place of interest. It's location is available to all. Some person, named Bob, has the task to

visit some point EMBED Equation.DSMT4 , which is somewhere near the point EMBED Equation.DSMT4 . He may choose the point EMBED Equation.DSMT4 himself with only one restriction: the distance from the point EMBED Equation.DSMT4 to the point EMBED Equation.DSMT4 must be smaller than some fixed value (see Fig.1). If Bob do the task, he will get some reward from the other person named Alice. Here will be Figure 1. From the one hand, Bob, for some reasons, doesn't want to open to Alice his exact location at the point EMBED Equation.DSMT4 . From the other hand, he should persuade Alice that he did visit some place which is close to the point EMBED Equation.DSMT4 . The zero-knowledge protocol from this article is just the instrument designed to solve this contradictionary problem. In what follows we should make some additional assumptions. We assume that Bob has some device, "black box", which realizes the next function: when activated, it connects to GPS, define coordinates of its real location and send Alice some values, calculated as the values of some function from the coordinate (in 2.5 we will explain how can it be). It is important that this function is one-way: given the value of the function, it is impossible to evaluate the correspondent argument of function, i.e. Bob's real coordinates. Also we assume that Alice and Bob both trust this device: Alice trusts that it's information is true and Bob trusts that information it send doesn't allow Alice to reveal his location. We also assume that only Bob can activate the device (for example, the device checks fingerprint or so) and that Bob can't interfere into its work (in particular, can't enter some false data about his location).

2. Definitions and Auxiliary Statements. In this chapter we remind some definitions and statements that are used in zk-location proof.

2.1. Distance between two points. To set a point on the plane we use coordinate plane, or two perpendicular number lines. The horizontal line is called x-axis, the vertical one is called y-axis. These lines intersect at their zero point, which is called origin (see Fig.2). A point in a coordinate plane is named by its ordered pair of the form EMBED Equation.DSMT4 . The first number corresponds to its x-coordinate, the second corresponds to its y-coordinate. Here will be Figure 2. When given two points, EMBED Equation.DSMT4 and EMBED Equation.DSMT4 on the coordinate plane, the distance formula is derived from the Pythagorean theorem, see Fig.3. Here will be Figure 3. The distance EMBED Equation.DSMT4 between EMBED Equation.DSMT4 and EMBED Equation.DSMT4 on Fig. 3 is the length of the hypotenuse of the right triangle EMBED Equation.DSMT4 . The lengths of its cathets are: EMBED Equation.DSMT4 ; EMBED Equation.DSMT4 . Then, according to the Pythagorean theorem, EMBED Equation.DSMT4 , or EMBED Equation.DSMT4 . Note that EMBED Equation.DSMT4 , so formulas are symmetrical with respect to EMBED Equation.DSMT4 and EMBED Equation.DSMT4 . In the 3-dimentional space each point has 3 coordinates. Then distance between two points, EMBED Equation.DSMT4 and EMBED Equation.DSMT4 , is evaluated using Pythagorean theorem twice: EMBED Equation.DSMT4 , (1) or EMBED Equation.DSMT4 . Example. Distance between two points, EMBED Equation.DSMT4 and EMBED Equation.DSMT4 is EMBED Equation.DSMT4 .

2.2. Circle. Points inside circle, points outside circle. On the 2-

dimentional plane, the circle with the centre O is the set of all points of the plane equidistant from the point O. The distance EMBED Equation.DSMT4 from the point O to any point on the circle is called radius (see Fig.4). Here will be Figure 4. Each circle divides the plane into 3 parts. One is the set of all points in the interior region of the circle, or the set of points inside the circle. The distance from the point O to any point EMBED Equation.DSMT4 inside the circle is strictly less then the circle's radius EMBED Equation.DSMT4 . Second one is the set of all points in the exterior region of the circle, or the set of points outside the circle. The distance from the point O to any point EMBED Equation.DSMT4 outside the circle is strictly greater then the circle's radius EMBED Equation.DSMT4 . And the third one is the set of points on the circle. The distance from the point O to any point EMBED Equation.DSMT4 on the circle is just equal to EMBED Equation.DSMT4 . Let the point O has coordinates EMBED Equation.DSMT4 . Then, according to the definitions of circle and formula (1), the circle consists of all points with such coordinates EMBED Equation.DSMT4 that EMBED Equation.DSMT4 . For any point EMBED Equation.DSMT4 inside the circle the inequality EMBED Equation.DSMT4 holds, or EMBED Equation.DSMT4 . Analogically, for any point EMBED Equation.DSMT4 outside the circle the inequality EMBED Equation.DSMT4 holds, or EMBED Equation.DSMT4 . So, the next three conditions are equivalent: 1) the distance EMBED Equation.DSMT4 from the point O to the point EMBED Equation.DSMT4 is less than EMBED Equation.DSMT4 ; 2) the point EMBED Equation.DSMT4 lays inside the circle with the center O and radius EMBED Equation.DSMT4 ; 3) the inequality holds: EMBED Equation.DSMT4 . The same equivalent conditions may be formulated for the exterior rejoin of the circle: 1) the distance EMBED Equation.DSMT4 from the point O to the point EMBED Equation.DSMT4 is greater than EMBED Equation.DSMT4 ; 2) the point EMBED Equation.DSMT4 lays outside the circle with the center O and radius EMBED Equation.DSMT4 ; 3) the inequality holds: EMBED Equation.DSMT4 .

The 3-dimentional analog of circle is called sphere. Like a circle in 2-dimentional space, a sphere is defined as the set of all points that are equidistant from some point EMBED Equation.DSMT4 , which is called the center of the sphere. The distance from the point O to any point EMBED Equation.DSMT4 on the sphere is called the radius of the sphere. Each sphere also divides the 3-dimentional space into 3 parts: interior region of the sphere, exterior region of the sphere, and the set of points on the sphere. The properties of these three parts are very similar to the properties of the correspondent parts in the case of plane. In particular, the next three statements are equivalent: 1) the distance EMBED Equation.DSMT4 from the point O to the point EMBED Equation.DSMT4 is less than EMBED Equation.DSMT4 ; 2) the point EMBED Equation.DSMT4 lays inside the sphere with the center O and radius EMBED Equation.DSMT4 ; 3) the inequality holds: EMBED Equation.DSMT4 . The same equivalent conditions may be formulated for the exterior rejoin of the sphere: 1) the distance EMBED Equation.DSMT4 from the point O to the point EMBED Equation.DSMT4 is greater than EMBED Equation.DSMT4 ; 2) the point EMBED Equation.DSMT4 lays outside the sphere with the center O and radius EMBED Equation.DSMT4 ; 3) the inequality holds: EMBED Equation.DSMT4 . Note that in the article we will use the term “3-dimentional circle” or just “circle” instead of the term “sphere”.

2.3. Connections between Bob’s task and a circle. According to the Bob’s task formulated in Chapter 1, Bob should prove to Alice that the distance between his location point EMBED Equation.DSMT4 and some reference point EMBED Equation.DSMT4 is less than some given value EMBED Equation.DSMT4 : EMBED Equation.DSMT4 . This is the same as his location point EMBED Equation.DSMT4 is inside the circle with the center O and radius EMBED Equation.DSMT4 . Another words, he should prove that the difference EMBED Equation.DSMT4 is positive, i.e. EMBED Equation.DSMT4 . (2)

2.4. Using Lagrange theorem to prove that difference is positive. As we noted in Chapter 1, Bob doesn’t want to open for Alice his true location. So Alice doesn’t know the coordinates EMBED Equation.DSMT4 and, consequently, doesn’t know the value EMBED Equation.DSMT4 . Then, how can Bob prove that the unknown to Alice value is positive? He may use Lagrange theorem, which states that any positive number may be represented as the sum of four squares. Then to prove (2) Bob should prove that there exists such squares EMBED Equation.DSMT4 , EMBED Equation.DSMT4 , EMBED Equation.DSMT4 . and EMBED Equation.DSMT4 that EMBED Equation.DSMT4 . (3)

2.5. Using discrete logarithm to hide the secret information. Let EMBED Equation.DSMT4 and EMBED Equation.DSMT4 be prime, EMBED Equation.DSMT4 . Then for some (properly chosen) positive EMBED Equation.DSMT4 the function EMBED Equation.DSMT4 (4) is one-way function, if EMBED Equation.DSMT4 and EMBED Equation.DSMT4 are unknown. It means that if Alice knows EMBED Equation.DSMT4 and EMBED Equation.DSMT4 , but doesn’t know EMBED Equation.DSMT4 and EMBED Equation.DSMT4 , she can’t reveal the value EMBED Equation.DSMT4 from the value EMBED Equation.DSMT4 . We may use such one-way function, for example, in device, which Bob uses to report Alice information about his location. The simplest way to do it is to report Alice values EMBED Equation.DSMT4 , EMBED Equation.DSMT4 , and EMBED Equation.DSMT4 instead of coordinates EMBED Equation.DSMT4 , EMBED Equation.DSMT4 , and EMBED Equation.DSMT4 . Next, we can use such function (4) to rewrite the equality (3) in the form EMBED Equation.DSMT4 , (5) which doesn’t allow Alice to reveal secret information, and prove her (5) instead of (4). Note that zk-proof of location in the article uses more complicated approach, but the main idea is the same.

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):

$$d^2 - ((x_n - x_l)^2 + (y_n - y_l)^2 + (z_n - z_l)^2) = a_1^2 + a_2^2 + a_3^2 + a_4^2 \quad (18)$$

We rely on 4-squares Lagrange theorem to prove equality statement [6]. Proofs for integer relations are possible in hidden group order setup. Our proof system is an extension of Schnorr protocol [8] with verification relation quadratic in challenge of Verifier. Our interactive proof is shown on Figure 2 and non-interactive variant on Figure 3. With our engineering community in mind, we do not focus on definitions like proof of knowledge [1] and zero knowledge [5, 4], and just state extractor and simulator algorithms exist for the interactive version.

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 $n = pq$, $p = 2k_p p' + 1$, $q = 2k_q q' + 1$ and p, q, p', q' primes [3].

2.3 Representation-Based Commitment

We commit to node location with a Pedersen-like scheme [7, 2]

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

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.

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

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

$$\begin{split} f_V(v) = f_2 v^2 + f_1 v + f_0 = \\ (((vx_n + \beta_x) - vx_l)^2 + ((vy_n + \beta_y) - vy_l)^2 + ((vz_n + \beta_z) - vz_l)^2) - v^2 d^2 \\ - ((va_1 + \alpha_1)^2 + (va_2 + \alpha_2)^2 + (va_3 + \alpha_3)^2 + (va_4 + \alpha_4)^2) \end{split} \tag{22}$$

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 $k \neq p$.

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

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

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

References

[1] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, 1993.

[2] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, Cambridge, MA, USA, 2000.

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

[4] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, July 1991.

[5] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC, pages 291–304, New York, NY, USA, 1985. ACM.

[6] 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.

[7] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[8] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4:161–174, 1991.