Multi-party Secure Position Determination

Tord Ingolf Reistad tordr@item.ntnu.no

September 22, 2006

Abstract

We consider the problem of calculating the geographical position of nodes in a wireless network, where the privacy of location data is highly valued. A solution is presented using a multi-party computation, where the secret inputs are the position of anchor nodes and distances between nodes.

1 Introduction

Mobile ad-hoc networks (MANET) are wireless mobile devices (nodes) that cooperatively form a network without central infrastructure. Each node cooperates by being involved in routing and forwarding information between neighbors. Thus, an ad-hoc network allows devices to create a network without prior coordination or configuration.

In a non-homogeneous networks some nodes know their geographical position, and distance between nodes can be estimated. This information can be used to calculate the position of additional nodes in the network. The position data thus generated can in turn be used for a wide variety of applications, such as route tracking, location based services and many more.

On the other hand privacy of location data might be very important. This because ad-hoc networks might be implemented in future generations of mobile phones. Ad-hoc network can also be integrated into other mobile devices such as PDA's, digital cameras and laptops that are worn or carried around.

The location data can be kept private with the use of multi-party calculations. Multi-party calculations are distributed calculations done by multiple parties. Each value that should be private is split up into shares, such that an single share does not give any information about the value itself. Calculations can be performed on these shares, and the answer can be revealed by recombining shares representing the answer.

Our contribution is to show how multi-party computations can be performed on location data, to improve privacy in ad-hoc networks. This is part of ongoing research, therefore only the protocols for simple location calculations in one and two dimensions are given.

This paper was presented at the NIK-2006 conference; see http://www.nik.no/.

Related work

Multi-party computations were introduced by Yao [1] and fundamental results obtained by Ben-Or, Goldwasser, and Wigderson [2], as well as Chaum, Crepau, and Damgård [3]. An integral part of any multi-party computation are secret sharing schemes. We will use Shamir's secret sharing scheme which was introduced in [4]. Damgård et al. [5] has shown novel techniques for calculating comparison and bit extraction.

The paper is organized as follows. In the following section, we describe the model, in section 3 we fix the notation for the article. We then recall the basics of a multi-party computation in section 4, present our solution in section 5, and discuss conclusion and future work in section 6.

2 Model

Consider an ad-hoc network of communicating nodes some of which know their geographical position, henceforth called *anchor nodes*. While others would like to compute their own geographical position and will be referred to as *floater nodes*. Both sets of nodes would like their geographical position to remain secret.

We present a protocol that uses multi-party computations to provide the floater nodes with their geographical position, while ensuring the privacy of location data of all honest participants. We assume that the communicating parties can determine the distance between them. This can for example be done by examining the loss of signal strength or by calculating the time the signal takes to propagate. We also assume there exists private channels to each node e.g. each node has a public RSA key, this to ensure that shares will only be known to the correct recipients.

In the protocols we describe, we make the assumption that the positions of nodes are in a two dimensional plane. This is a good approximation for long range ad-hoc networks, while greatly simplifying the computations and reducing the communication complexity.

We also assume that there are sets of $k \geq 3$ anchor nodes that are in direct communication range with at least some floater nodes in the network, and the anchor nodes are willing to participate in the computation. When these floater nodes find their geographical position they can in turn act as anchor nodes to other floater nodes.

3 Notation

All computations on secret shares will be performed over a finite field $\mathbb{F}$. This finite field is assumed to be chosen large enough such that no overflow occurs within the field - i.e. all computation on the inputs is equivalent to computation over integers. The nodes will be denoted by capital letters A, B, C, and P, their position will be denoted by the coordinates $(x_A, y_A), (x_B, y_B), (x_C, y_C)$, and (x_P, y_P), respectively. The distance between nodes A and B will be denoted by $|AB|$.

The shares for a secret s will be denoted by $[s]_i$ where i identifies the recipient of the share.

4 Multi-party computation

Suppose m players $P_1, \ldots, P_m$ owning secret inputs $x_1, \ldots, x_m \in \mathbb{F}$, respectively, would like to compute a function $y = f(x_1, \ldots, x_m)$ without revealing more about $x_1, \ldots, x_m$ than what can be inferred from the output y.

They achieve this by first distributing *shares* of their input values to each other by using, for instance, Shamir's secret sharing scheme. Then they perform all operations given by the function f on the distributed shares and finally recombine the shares to obtain the output y.

Let a, b be secrets with shares $[a]_i$, $[b]_i$, $1 \leq i \leq m$, and let $c \in \mathbb{F}$. Since the function f can be written as a rational function in $x_1, \ldots, x_m$, only the following operations need to be carried out.

- Linear Combination

  Shamir's secret sharing scheme is linear, that is, shares for linear combinations of secrets are equal to the corresponding linear combination of shares and can be computed by the participants without any interaction. Thus, $[a+cb]_i = [a]_i + c[b]_i$.

- Multiplication

  The multiplication of two secrets a and b can be done by the following interactive protocol, described in [6].

  Each player i computes the value $h_i = [a]_i[b]_i$, splits h_i into shares $[h_i]_j$ and distributes the shares $[h_i]_j$.

  Each player can then compute a share of the product ab by using the following equation.

  $$[ab]_i = \sum_{j=1}^{m} \lambda_j [h_j]_i \tag{1}$$

  where λ_j is the first row of the inverse of the Van der Monde matrix $[i^j]_{1 \leq i \leq m, 0 \leq j \leq m-1}$.

- Multiplicative Inverse

  We use the protocol given in [7]. To compute $1/b$ the players first create a random number R as follows. Each player i distributes shares $[r_i]_j$ of a random number r_i and adds up all received shares to obtain $[R]_i = \sum_{k=1}^{m} [r_k]_i$, so that $R = \sum_{i=1}^{m} [R]_i$.

  The shares for the value bR are calculated using the multiplication protocol and then bR is revealed. The shares for the inverse of b are then calculated as

  $$\left[b^{-1}\right]_i = (bR)^{-1}[R]_i. \tag{2}$$

5 Position calculations

To simplify the exposition, we begin by solving the one dimensional analogue of our problem.

One-dimensional calculations

If a node P wants to know its position in a one-dimensional world it only needs to contact two anchor nodes A and B. In what follows, P needs to know the distances $|AP|$ and $|BP|$, while A and B only need to know their positions x_A and x_B, respectively. A, B, and P can then carry out the following protocol which allows P to learn its position x_P. All secrets will be shared using a $(2,3)$–threshold linear secret sharing scheme.

A and B distribute shares of their positions x_A and x_B, P distributes shares of $|AP|$ and $|BP|$.

Shares for the following values are then calculated by each party.

$$\begin{aligned} x_{A1} &= x_A + |AP| & \qquad x_{B1} &= x_B + |BP| \\ x_{A2} &= x_A - |AP| & \qquad x_{B2} &= x_B - |BP| \end{aligned}$$

e.g. the shares $[x_{A1}]_i$ are calculated from the sum $[x_A]_i + [|AP|]_i$, for all nodes $i \in A, B, P$

Thereafter A, B and P calculate the following function and the resulting shares are sent to P which will enable P to learn its position.

$$x_P = \frac{x_{A1}x_{A2} - x_{B1}x_{B2}}{(x_{A1} + x_{A2}) - (x_{B1} + x_{B2})} \tag{3}$$

The division can be done in the field only if x_P is known to be an integer value. Otherwise division in the field will not give the same answer as division in over integers. A solution to integer division is to use the methods in [5] to get the bits and do integer division over the bits.

The 2-dimensional case

The two dimensional case involves at least four parties, namely three anchor nodes A, B, C, and one floater node P.

The three circles described by the following equations intersect, by construction, in the point (x_P, y_P).

$$\begin{aligned} (x - x_A)^2 + (y - y_A)^2 &= |AP|^2 \\ (x - x_B)^2 + (y - y_B)^2 &= |BP|^2 \\ (x - x_C)^2 + (y - y_C)^2 &= |CP|^2 \end{aligned}$$

From each pair of circles we can construct a line passing through their intersection. The equations for these lines can be written as

$$\begin{aligned} 2(x_A - x_B)x_P + 2(y_A - y_B)y_P &= x_A^2 - x_B^2 + y_A^2 - y_B^2 + BP^2 - AP^2 \\ 2(x_A - x_C)x_P + 2(y_A - y_C)y_P &= x_A^2 - x_C^2 + y_A^2 - y_C^2 + CP^2 - AP^2 \\ 2(x_B - x_C)x_P + 2(y_B - y_C)y_P &= x_B^2 - x_C^2 + y_B^2 - y_C^2 + CP^2 - BP^2 \end{aligned}$$

If no two nodes are close by each other, it suffices to solve any pair of the above equations for (x_P, y_P). Thus the above equations can be simplified to the following linear system.

$$a_1 x_P + b_1 y_P = c_1$$
$$a_2 x_P + b_2 y_P = c_2$$

The solution of the linear system is then obtained by the following functions

$$x_P = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - b_1 a_2} \qquad y_P = -\frac{c_1 a_2 - c_2 a_1}{a_1 b_2 - b_1 a_2}$$

Again the division can generally not be done over the field, so the shares for the values $(c_1 b_2 - c_2 b_1)$, $(c_1 a_2 - c_2 a_1)$ and $(a_1 b_2 - b_1 a_2)$ are sent to P. P recombines the shares to get the values and does the divisions over integers to obtain its position (x_P, y_P).

In practice finding the geographical position of P in 2 dimensions can be done with 3 rounds of communication. In the first round all the variables are distributed. The second round the multiplication is done in parallel. The third round consists of sending the shares for the answers to P. With 4 nodes the privacy is preserved as long as 2 nodes do not cooperate.

This idea can easily be generalized to higher dimensions.

6 Conclusion and future work

In this paper we have given an introduction into multi-party computation over Shamir's secret sharing scheme. We have also shown how these methods can be used to compute geographical position using simple algorithms for calculating location in one and two dimensions. The equations are quite simple and do not perform so well when anchor points are close together or in a line, but they improve privacy.

To improve the accuracy of the calculations and to compute geographical position in ad-hoc networks which have fewer anchor nodes, one can use better algorithms. A great deal of work has been done on localization, e.g. Strang et al. [8]. Localization in wireless ad-hoc and sensor networks can be found in e.g. [9] and [10]. But multi-party computations require many rounds of communication, e.g. calculating if $a > b$ costs 114 rounds of communication as proposed by Damgård et al. [5]. Therefore further research is needed to get more round efficient computations, and find algorithms that are well suited to multi-party computations.

Homomorphic public-key systems and threshold homomorphic public-key systems could be explored as an alternative to multi-party computations. A overview of such systems is given in [11].

Also for larger networks a subset of nodes could together be used as a form of distributed trusted third party. This could reduce the computational burden for the overall system.

Acknowledgments

We would like to thank Sasa Radomirovic for many helpful comments throughout the writing process.

References

[1] A. Yao. Protocols for secure computation. In IEEE, editor, *23rd annual Symposium on Foundations of Computer Science, November 3–5, 1982, Chicago, IL*, pages 160–164, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1982. IEEE Computer Society Press. IEEE catalog no. 82CH1806-9. IEEE Computer Society order no. 440.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In ACM [12], pages 1–10. ACM order no. 508880.

[3] D. Chaum, C. Crepeau, and I. Damgård. Multiparty unconditionally secure protocols. In ACM [12], pages 11–19. ACM order no. 508880.

[4] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

[5] Ivan Damgård, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, and Tomas Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. *Proceedings of the third Theory of Cryptography Conference TCC 2006*, pages 285–304, 2006.

[6] Rosario Gennaro, Michael Rabin, and Tal Rabin. Simplified VSS and fast-track Multiparty Computations with applications to Threshold Cryptography, 1998.

[7] J. Bar-Ilan and D. Beaver. Non-cryptographic fault-tolerant computing in a constant number of rounds. In ACM, editor, *Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing: Edmonton, Alberta, Canada, August 14–16, 1989*, pages 201–209, New York, NY 10036, USA, 1989. ACM Press.

[8] Gilbert Strang and Kai Borre. *Linear algebra, geodesy, and GPS*. Wellesley-Cambridge Press, Wellesley, MA, USA, 1997.

[9] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks (Amsterdam, Netherlands: 1999)*, 43(4):499–518, November 2003.

[10] A. Savvides, H. Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *Mobile Networks and Applications*, 8(4):443 – 451, aug 2003.

[11] Kristian Gjøsten. Homomorphic public-key systems based on subgroup membership problems. *Proceedings of MyCrypt 05 volume 3715 of LNCS*, pages 314–327, 2005.

[12] ACM, editor. *Proceedings of the twentieth annual ACM Symposium on Theory of Computing, Chicago, Illinois, May 2–4, 1988*, New York, NY 10036, USA, 1988. ACM Press. ACM order no. 508880.