Preserving Location Privacy for Mobile Phones with Homomorphic Encryption: The False Position Protocol

Mike H. M. Teodorescu
Information School, University of Washington, Seattle, WA 98105,
miketeod@uw.edu

Sam Ransbotham
Carroll School of Management, Boston College, Chestnut Hill, MA 02467,
sam.ransbotham@bc.edu

Abstract

Sharing sensitive information, such as location data, or health data, is a complex problem. While users may desire the benefits of application that use sensitive information, adoption may be limited by user reluctance to share sensitive data with untrusted third parties. We propose the False Position Protocol, a decentralized algorithm that allows users to reveal information such as location to trusted partners through a homomorphic encryption identification process. The algorithm offers reduced computational complexity while maintaining resilience despite potential malicious actors. Potential applications of the proposed two-party sharing protocol include connecting in social networks, exchanging health information, geotagging content, as well as proximity testing for media content delivery.

Keywords: Decentralized, mHealth, Dummy Location, Homomorphic Encryption, Privacy

1. Introduction

Information sharing is tricky in practice (Jiang, Li, Zhao, Zeng, Xiao, and Iyengar 2021). Job applicants must provide information to potential employers but cannot be sure who accesses (e.g., fake job postings, Vidros, Kolias, and Kambourakis 2016) or how potential employers use (e.g., hiring discrimination, Acquisti and Fong 2020) their data. Taxpayers must file returns to government agencies, but both parties struggle to ensure that they are sharing their confidential information with the correct party (e.g., return fraud, Brody, Haynes, and Mejia 2014). Information security requires details to reinforce protection but grapples with the consequences of widespread disclosure (e.g., software vulnerabilities, Mitra and Ransbotham 2015). In health, if users sacrifice some privacy (perhaps temporarily), society could better contain the spread of deadly pathogens (Sharma and Bashir 2020).

It is fundamentally difficult to share information with a trusted party — and only that trusted party. While individuals benefit from sharing information with specific trusted parties, attackers may masquerade as trusted parties to access that information. Several solutions exist, but none are ideal (Papageorgiou, Strigkos, Politou, Alepis, Solanas, and Patsakis 2018).

In the context of health, applications commonly require users to share sensitive information (e.g., location and health) with a centralized authority (such as a public health department or app developer). Location security and privacy are essential in Mobile health (mHealth) for mobile medical applications, such as telemedicine and remote patient monitoring. Unfortunately, tracking gives a central authority the ability to use data beyond the intended scope — for instance, to track a dissident, to scoop up potential business opportunities, "to identify people's travel paths and their entire social networks" (Sharma and Bashir 2020), etc. The majority of contract tracing apps use blanket permissions to collect or access inappropriate amounts of data that have no bearing on infection control, such as photos, contacts, call information, age, and other personal attributes; some do not even guarantee encryption or anonymity of data reporting (a review of 50 contact tracing apps, Sharma and Bashir 2020). Additionally, once given away, it is hard to "regain lost liberties," and that can lead to a "chilling" future (Sharma and Bashir 2020). Therefore, users can be understandably reticent to divulge sensitive information to a third party. As a result, the need for a verified, trusted third party is a barrier to health benefits. Furthermore, reliance on a central authority can add

URI: https://hdl.handle.net/10125/106836
978-0-9981331-7-1

a bottleneck, slowing time-critical processes (Shubina, Holcer, Gould, and Lohan 2020).

Instead, peer-to-peer approaches avoid centralized intermediaries. Homomorphic encryption algorithms such as Paillier (Paillier 1999) transmit information in decentralized protocols. For example, the information could be an encrypted message containing infection status and location. However, current homomorphic encryption schemes tax the computational capabilities of mobile devices and may not perform well with malicious actors. In our context of the proposed False Position Protocol, the localization is more efficient through the use of homomorphic encryption. An efficient protocol is especially relevant in mobile applications where battery consumption may be a factor in choosing encryption approaches.

The location privacy is also essential in networks of communication devices where the devices themselves are sensitive and under attack of location privacy, whose location is sensitive information, or whose location privacy is threatened (Li, Huang, Chang, Weng, Chen, and Li 2023). Such networks of devices may be Sensor Networks (Zhang and Chow 2014), vehicles (Guo, Ma, and Gao 2018), generally networks of mobile users (Cheng, Ma, Liu, Wu, Wei, and Dong 2022), and networks of "participatory sensing" applications (Christin 2016), where the nodes of the networks collect and exchange information including location information (Vu, Zheng, and Gao 2012). No algorithms consider the spatial density probability of finding the location privacy-protected devices or persons, except a qualitative discussion in (Sun, Chang, Ramachandran, Sun, Li, Yu, and Liao 2017) of a variant of the Dummy-Location selection (DLS) algorithm. We quantitatively integrate the probabilistic knowledge on the location in a specified area, considering the map of the area that is processed using general knowledge for deriving the probabilities of location. We extend the k-anonymity method for including the location, providing multiple dummy locations in the process of conveying information about location. We automatically generate dummy locations, based on the probabilities on the map.

We propose a False Position Protocol (FPP) that mixes grid localization and a new distance measure that simplifies computation while also performing well even with misbehaving peers. FPP falls broadly within the category of nearby-friend tracking. In infection contact tracing, for example, peer-level coordination through the decentralized protocol only requires trust at the user-pair level rather than communication with a centralized authority. FPP demonstrates that 'carte blanche' ongoing geographical location tracking by a trusted central authority is unnecessary.

2. Background

Information sharing based on location is beneficial in a variety of contexts – (e.g., supply chain, Li 2002), (e.g., online communities, Ma and Agarwal 2007), (e.g., retail, Lee, So, and Tang 2000), delivery of advertising content (Yang, Vijayakumar, Shen, and Gupta 2022), contact tracing (An, Lee, Jung, Park, Song, and Ko 2021), and more. The "ubiquity of cellphones" has made many application areas possible (Zhong, Goldberg, and Hengartner 2007).

But in practice, information sharing is fraught with difficulty. For example, in healthcare, individuals want close control of their private information yet simultaneously both need and want to share that same information (Adjerid, Acquisti, Telang, Padman, and Adler-Milstein 2016). Legislation such as HIPAA in the US has been enacted to protect patient data. Yet, at the same time, patients likely want healthcare professionals to have complete information about their treatment even if they cannot actively consent in an emergency. Privacy is much less important when people are bleeding in the emergency room than when they are not. Furthermore, health conditions such as contagious infection can have externalities to other individuals — sharing of such sensitive information requires users to trust the platform.

Transparency and control are essential for trust. Individual adoption of healthcare technologies increases when applications have "Privacy by Design," giving users transparency into what they share and tools to control sharing (Sharma, Dyer, and Bashir 2021). Users often do not wish to share geolocation data (Sharma et al. 2021) in particular. Thus a protocol that identifies pairs of users in close proximity *without* sharing actual geolocation data with a central authority would reduce friction to adoption — a "nearby-friend problem" (Chatterjee, Karabina, and Menezes 2009). A nearby friend problem is "an instance of secure multiparty computation problem, where multiple parties jointly compute the output of a function without learning each other's inputs" (Zhong et al. 2007). The "nearby-friend problem" is that locations should only be shared if one can reasonably ensure that the inquirer of the location is indeed a friend and not some other party. This problem has been an active research area for over fifteen years, yet many problems still remain.

2.1. Difficulties with Finding Nearby Friends

Nearby-friend identification is difficult for multiple reasons.

Location-based services often inherently compromise location privacy. As part of their protocol, some location protocols must release information about their entities' locations regardless of whether the entities authorize the disclosure. For example, a protocol may reveal the location of an entity to another entity even if not physically near, a known failing of many nearby-friend protocols. In the most simplistic approach, if an entity publicly broadcasts its location, everyone knows the entity's private information (the location). An early potential solution to this problem was location cloaking (Cheng, Zhang, Bertino, and Prabhakar 2006), but an issue of early solutions was that they require an intermediary: either a trusted third party in the communication or the transmission channel (service provider) would need to know the locations. A two-party solution to the "nearby-friend problem" is possible but requires additive homomorphic encryption, such as the Pierre protocol (Zhong et al. 2007). Additive homomorphic encryption algorithms enable the addition of the encrypted messages to be equal to the encryption of the added original messages (e.g., if the messages are m_1 and m_2 and the encryption algorithm is ϵ, then the property is $\epsilon(m1) + \epsilon(m_2) = epsilon(m_1 + m_2)$ (Zhong et al. 2007). Examples of homomorphic additive encryption algorithms include the Paillier algorithm (Paillier 1999) and the CGS97 algorithm (Cramer, Gennaro, and Schoenmakers 1997).

Some protocols reveal locations to a known group of entities. By restricting disclosure, private information is no longer publicly disclosed. However, the problem then becomes an authentication problem. Fingerprinting attacks falsify a user's identity and masquerade as a friend or falsify position to induce the other entity to reveal its location even if that location is not near the second entity (Dong, Dave, Qiu, and Zhang 2011). Many protocols can release information to an attacker masquerading as a friend (Narayanan, Thiagarajan, Lakhani, Hamburg, Boneh et al. 2011), enabling an adversary to compromise an identity. These compromises can occur when using shared geotagged data (Vicente, Freni, Bettini, and Jensen 2011) such as photos, videos, or other mobile posts, an attack in a phenomenon known as cybercasing (Friedland and Sommer 2010). To be effective, protocols must handle malicious behavior such as masquerading.

Alternatively, other protocols require a trusted third-party (Šikšnys, Thomsen, Šaltenis, Yiu, and Andersen 2009) such as a verification server. Protocols that involve third parties must also authenticate that third party. Attackers may compromise those third parties, a known weakness of three-party protocols (Narayanan et al. 2011).

Additionally, protocols face computational constraints. Verification solutions are expensive for mobile applications, in both computation time and power consumption (particularly if they involve third-parties, Dong et al. 2011). Apart from security, desired characteristics for a two-party nearby friend protocol include efficiency, specifically making the identification in the fewest number of connections, and with the lowest computation and bandwidth costs (Narayanan et al. 2011).

2.2. Existing Protocols

Several protocols exist to address the nearby-friend problem. The protocols vary in their definitions of neighborhood (distance), their use of third-parties, their computational demands, and their resilience to malicious behavior.

2.2.1. Louis Protocol The Louis protocol is a three-party protocol with three participants, Alice, Bob, and Trent (Zhong et al. 2007) . In this protocol, Alice and Bob are the two "nearby friends" attempting to exchange location information (and validate who they are), whereas Trent is a trustworthy third party who is used for validation. The "nearby-friend protocol" (Zhong et al. 2007) has the following features:

Neighborhood definition: The entities announce their own neighborhood defined by a radius r centered around the entity.

Third party: A third additional entity, $Trent$, makes this algorithm additionally vulnerable to attacks compromising the trusted third-party. $Trent$ acts as a trusted third party but does not know either of the locations ex ante.

Algorithm (defined by (Zhong et al. 2007): "$\varepsilon_A(\cdot)$ is the Paillier additive homomorphic encryption function using $Alice$'s public key, $\varepsilon_T(\cdot)$ is a (non-homomorphic) public-key encryption function using $Trent$'s public key." (from p. 66 of (Zhong et al. 2007)). The Paillier encryption algorithm is from (Paillier 1999).

$Alice$ first communicates her location coordinates, x and y, in ε_A along with a random salt, S_A, encrypted using the cryptographic hash function.

Bob then computes the value of the following using his own location coordinates, u and v (from (Zhong et al. 2007) page 66):

$$\frac{\varepsilon_A(x^2 + y^2).\varepsilon_A(u^2 + v^2).\varepsilon_A(k)}{(\varepsilon_A(2x))^u.(\varepsilon_A(2y))^v.(\varepsilon_A(r^2))}$$

The multiplication involved in the exponential term in the denominator is especially computationally heavy.

(Additional computational steps involve the communication channel between $Alice$ and $Trent$, but these steps are not as computationally heavy as the expression above.) The challenge of this algorithm is that Trent is a trusted third party, so if either one of the "friends" Alice or Bob cooperates with Trent, they can reveal the other's location. If there are multiple options for the third party "Trent", and either Alice or Bob can choose for each iteration potentially a different "Trent" trusted third-party, educing the potential for cheating behaviors (Zhong et al. 2007). The downside of having a third party at all is that there is another set of communication steps between the two potential nearby friends and the third party, so the Louis protocol takes two to four extra communication steps compared to the other two protocols proposed in Zhong et al. (2007) which do not require a third party. Furthermore, in the Louis protocol, both Alice and Bob learn each other's exact locations ("Louis (both phases)" in Table 3, Zhong et al. 2007), which can be disadvantageous from a privacy perspective.

2.3. Lester Protocol

A second "nearby-friend" protocol (Zhong et al. 2007) can remove the need for a trusted third party and thus is a two-party protocol. This protocol allows one party to obfuscate their actual location and provide only a distance between the parties. This presumably can reduce the risk of any one of the two parties cheating, since they do not get actual locations but merely distances between them. If we assume a 2-dimensional grid coordinate system and a distance r between Alice and Bob (the two communicating parties remaining), Alice and Bob can find if they are within a circle of radius r of their position and nothing more. The issue with this approach is that if one party uses incorrect starting coordinates as their position in the protocol, they could guess the other's location "by simply entering that guess as [their] own location and seeing if the protocol finds [the other party] nearby" (Zhong et al. 2007). Another risk is that the cheating party knows the other party's frequent locations (e.g., their home and work routine) and thus can provide a location near those spots, thus fooling the other party into thinking their "friend" is nearby. This protocol does not handle this particular scenario well.

The Lester protocol makes a trade-off to avoid the use of a third party like Trent to validate that the communicating parties Alice and Bob are indeed nearby — neither learns precise location (rather just that they're within a certain distance of each other or not) — and the communication party that initiates the protocol learns more information about the other party. The latter point implies that in order for both parties to learn whether the other is nearby, the parties need to repeat the protocol with the roles reversed.

The encryption method used for Lester is CGS97 (Cramer et al. 1997) cryptosystem, which falls under homomorphic encryption. The Lester protocol allows for the inquired party (let's say Bob) to vary the amount of computational work the party receiving the information (let's say Alice, like in the example in the paper introducing the protocol (Zhong et al. 2007)) must do in order to learn the distance between the parties based on a workfactor baked into the decryption effort the receiver must do. The work factor can be set by Bob to be very high which can deter guessing attempts and is exponential in time, while the computation effort of Bob is constant regardless of the workfactor sent. Of course, the workfactor used should be reasonable based on the device doing the computation; for example, it would be meaningless to send a workfactor to a mobile phone that would take it years to decrypt the location. Realistic ranges of workfactors appear to be narrow for the algorithm to be feasible on mobile phones (Zhong et al. 2007). Readers may simulate this protocol on new hardware and come up with different recommendations based on the hardware and type of device used. The Lester protocol derives its workfactor property from the CGS97 method. Variants of the protocol use different cryptosystems and other modifications (Zhong et al. 2007).

The major concern of the Lester protocol is that the communicating parties could cheat and provide false locations in terms of the grid positions, which the other party has no way of detecting. If, for example, the inquiring party sends a guess of the other party's location as their own (false) location, the inquired party would have no way of knowing that and would respond that indeed, they are within the radius r regardless of the value of r as long as it is nonzero. Thus the inquiring party always has an advantage in this type of cheating behavior. The False Position Protocol we propose addresses this concern, but for illustration purposes let us discuss a scenario on location privacy that showcases how a weakness in a nearby friend protocol like the one described here can translate into consequences beyond user privacy.

Let's assume a scenario that is unfortunately all too common in today's interconnected world through social media: burglars mining social media for locations of homes (Rose et al. 2011). Mobile phones encode a substantial amount of metadata with every photo, including (unless disabled, exact location), which opens serious privacy concerns (Li and Chen 2010). It's conceivable that not all users know how to disable location sharing in photos, and furthermore it is common for social media users to volunteer location information through social media platforms that automatically add geolocation tags. Social media applications are at risk of hacking (Soomro and Hussain 2019) and are valuable sources of information, such as friend lists. Therefore, a plausible scenario for a protocol like Lester to fail would be for an attacker to find a user's location patterns from social media, comb through friends' lists and stolen password lists (Missaoui, Bachouch, Abdelkader, and Trabelsi 2018) to locate profiles of friends of their target they could masquerade as, and then use a nearby-friend application to inquire if their target is at home, at work, on vacation, etc. An attacker could then act in the physical space and burglarize the target's home, an unfortunately common scenario (Soomro and Hussain 2019).

2.3.1. Pierre Protocol The Pierre Protocol is a two-party protocol which improves on Lester and has the following features:

Neighborhood definition: The entities each decide on a resolution distance, r, and then use this distance to share their coordinates in integer multiples of r without revealing r itself. This allows adjusting the raster size of the coordinate grid (based on r).

Algorithm: The protocol uses the same ε_A as the Louis Protocol. An exponential calculation in the encrypt domain is computationally heavy. Bob must calculate the following (formula from Zhong et al. 2007, p. 72):

$$D_r = (x_r - u_r)^2 + (y_r - v_r)^2$$

.

The benefit of Pierre over Lester is that the inquiring party can only learn if the other party is near the shared location. If the grid locations used by say Alice are not the real location positions and the choice of r is not reasonable, the protocol will not provide any useful information about Bob's location, and vice-versa. The protocol assumes that the communication parties do not know each other's frequent locations (or, even worse for privacy: frequent locations by time of day) and cannot pretend to be near those locations. If that information is available to an attacker, like in the social media-location mining example, then this protocol can fail as well. This protocol requires the homomorphic encryption and thus significant computational resources.

3. The False Position Protocol

Each of the three protocols (Louis, Lester, Pierre) solves many aspects of the nearby-friend problem. However, each requires significant computational resources. Given the nearby-friend protocols are predominantly beneficial in mobile applications with reduced communication and computation capabilities (Zhong et al. 2007), a protocol with reduced computational demands would be desirable.

Therefore, our FPP algorithm defines a neighborhood to minimize computational demand while preserving the privacy of the communicating entities in a way that is more robust to malicious actors. By proposing an alternative definition for the distance function, the exponentiation with linear expressions reduces computation load. Our proposed approach also enables the user to control the information that the app shares yet still share such information securely because of the homomorphic encryption.

3.1. Algorithm

3.1.1. The max-Manhattan Distance Function. We use a distance function similar to a taxicab (Manhattan) distance yet less precise in determining the location and thus better at preserving privacy. A max-Manhattan-like distance, which is rounded Chebyshev (L_∞) norm (Mohibullah, Hossain, and Hasan 2015), reduces the computational cost. We naturally combine this distance measure with the idea of walking on a graph (rectangular grid), as in the Pierre protocol. The use of the specific distance with a Pierre-like protocol improves privacy protection compared to Pierre alone. Taxicab distance in a plane (2D space) is $d_{taxicab}(A, B) = |x_A - x_B| + |y_A - y_B|$.

The alternative distance measure, d_{max_M} (named max-Manhattan, equivalent rounded L_∞ norm, with the rounding done according to the grid unit, as usually done for Manhattan distance) is:

$$d_{max-M}(A, B) = \lfloor max(|x_A - x_B|, |y_A - y_B|) \rfloor.$$

Consequently, knowing that $Alice$ and Bob are at a distance (in the max-Manhattan definition) less than r is less informative than knowing that their distance in the maxi-taxicab sense is less than r, which improves privacy. The measure d_{max-M} satisfies the axioms of a distance. The proposed protocol uses this distance to determine the closeness. Yet, the parties do not get complete knowledge on the closeness in the sense of the taxicab distance, but only in the d_{max-M} (max-Manhattan) sense, which is weaker.

Next, we define a grid over the plane, with a resolution distance r. By assumption, the grid has the axes in the North-South (NS) and East-West (EW) directions. (Grid orientation is arbitrary; we select NS and EW to illustrate.) Communicating partners must agree on the value of r, or the protocol must predetermine it. As in the Pierre protocol, the true coordinates of a point (x, y) in the plane transform to grid coordinates $x_r = \lfloor x/r \rfloor$, $y_r = \lfloor y/r \rfloor$.

Let's denote the participants as $Alice$ and Bob as follows: use the subscript A for $Alice$ and B for Bob. In the first phase, the partners choose a random point, $P = (\lambda, \mu)$ in grid coordinates; $Alice$ only knows λ and Bob picks the second *random* integer number, μ. The partners give their position with respect to this point — giving rise to the name, the False Position Protocol (FPP), as neither communication partner at this point has revealed any actual location information and this shared location is not actual for either of them.

The "nearby-friend" protocols are, in fact, mathematical games. Indeed, each protocol follows the steps of a mathematical game, which can have cheating strategies that result in a non-zero-sum outcome. As such, a condition of a successful protocol could be that in most cheating situations, a cheater cannot win by getting information while not revealing any information about themselves. In this sense, losing means disclosing one's own position while the communication partner does not reveal its position. Ideally, the chance of deception is equal for both partners or even possibly null. (Total security is not achievable by any current nearby-friend algorithm as far as we know, including the FPP we propose). However, mitigation strategies discourage cheating, and therefore, we evaluate several possible scenarios (playing strategies) for players $Alice$ and Bob to check for the game's fairness.

3.2. Strategy 'Fair players'

We assume that both players are fair and act rationally. Therefore, while they do not want the other player to gain much knowledge on their position, the goal remains that when they are close enough, they meet (Figure 1). Consequently, knowing that providing a point P that is too far from their actual position will result in a response that is too far off from themselves and possibly from the partner (Figure 2), they must choose a point not too far from their position if they wish the protocol to succeed. Non-cheating players will want the protocol to complete expediently.

We assume encrypted communication according to general encryption protocols, for example Paillier protocol (Paillier 1999, Zhong et al. 2007). We also assume that the partners know to use the max-M distance. Then, the communication game works as follows:

$Alice$: I am E-W direction. You will be the N-S (i.e., y direction). My closeness limit is n_A.

Bob: Agree or abort, depending on n_A. If agree, communicate: my closeness limit is n_B (possibly equal to n_A.)

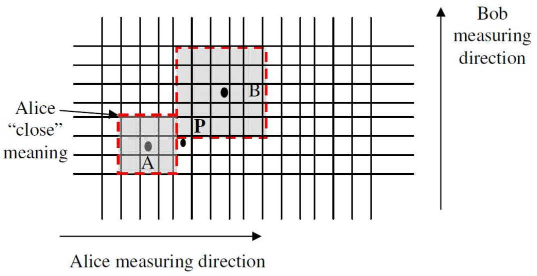

Figure 1. Favorable (reasonable, fair) choice of P, when partners are close.

Bob chooses μ such that $d_y(B, P) < n_B$.

$Alice$: My distance is $d_x(A, P)$.

Bob: My distance is $d_y(B, P)$.

Based on the answers, both compute $d_1(A, B) = max(d_x(A, P); d_y(P, B))$. If $d_1(A, B) < min(n_A, n_B)$, then they are satisfied with the first test and proceed:

$Alice$: Are you at West or East of $\lambda + k \cdot n_A$ (not revealing λ and k; k might be reasonable, e.g. 2). If the answer guarantees that Bob may be not too far, she continues.

Bob: Are you South or North of $\mu + h \cdot n_B$? If the answer guarantees that $Alice$ may be not too far, he continues.

$Alice$: Are you at West or East of $\lambda - k^+ \cdot n_A$ (not revealing lambda and k^+; k^+ might be reasonable, e.g. $k^+ = 3$). If the answer guarantees that Bob may be not too far, she continues.

Bob: Are you South or North of $\mu - h^+ \cdot n_B$? If the answer guarantees that A may be not too far, he continues.

At this point, they know either that they are too far, or that they might be close.

They may continue choosing smaller k, k^+, h, h^+ until they are satisfied, or go to the next step, revealing their grid coordinates. If the grid coordinates satisfy them, they can give the exact locations. Notice that the required encryption is minimal.

In the above case (of fair players), the game is fair because both players keep secret their position until they are satisfied with the approximate closeness value. They preserve the privacy degree they wish, based on the parameters h, k which allow customizing the number of steps in the exchanges.

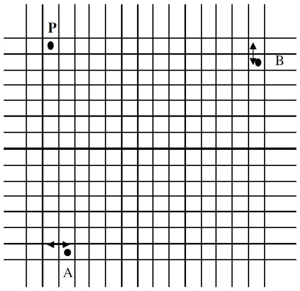

Figure 2. Unfavorable case of choice of P, because while the partners are fair, they are far apart.

4. Strategy 'Cheating players'

One or both players wish to cheat. The simplest cheating strategy is to choose a distant point from the real position and to lie about the closeness to it. For example, if Bob suspects $Alice$ might be at some place, he can lie he is close to a point P close to where he believes $Alice$ is (Figure 3). However, this strategy assumes extra knowledge of Bob about $Alice$. $Alice$ may also know about Bob that she might suspect he is in some other place and distrust him and end the communication there.

The same cheating strategy applies to guessing some position of the other partner. However, the partners still have equal chances to cheat and may deploy various defenses. Moreover, they can refuse to provide the precise (non-grid) coordinates, or may give a new point of meeting.

The protocol is a non-zero-sum game, meaning that strategies exist when both players gain information they desire, losing no privacy when they do not want to lose it. This result is typical of the class of nearby-friend algorithms.

5. Analysis: Protocol Variants and Computational Cost

Given the possibility of cheating, we could alter the protocol to have two versions:

Version 1: Trusting partners. $Alice$ and Bob agree to tell the other what *close* means for them. Denote these numbers by n_A and n_B. Then, they determine $max(n_A, n_B)$ and if that number satisfies both of them, they can declare the values they picked for (λ, μ). At this stage, they still do not know their precise position. One of them may chose to declare the exact position then.

Version 2: Suspicious partners. $Alice$ and Bob still do not trust each other and remain suspicious. While they do not tell what they mean by *close*, they agree to provide hints, at least up to some limit. So, they ask questions to limit the uncertainty. They may ask:

$Alice$: Is *close* less than m_A? $Alice$ may stop the communication if the answer is *no* and $m_A > n_A$. If the answer is *yes*, she may ask if *close* $< m_A/2$ etc., until $Alice$ is satisfied that *close* is close enough for her.

Bob may choose to ask the same questions, before providing further answers.

The advantages of the above protocol relative to existing protocols (Louis, Lester, Pierre), are:

- Communication load is lower, as the first phase only requires the values for r, λ or μ, and binary answers, and the second phase only requires values of m_A. (Of course, data encryption requires communicating signatures).
- It is easier to compute, as only subtractions and comparisons are made.
- It preserves privacy better because only partial information circulates.
- The protocol works even for non-rectangular grids.

5.1. Computational Cost

In order to compare the computational time efficiency of the FPP protocol to the other three protocols (Louis, Lester, and Pierre), we compare the time complexity of the most computationally heavy protocol (Pierre) from the three with the FPP protocol. If we assume it takes time t to perform an u-bit by v-bit encrypted key multiplication, then the time complexity of Pierre is $max(O(t^u), O(t^v))$. Due to the difference in the distance computation, with max-M being much lower cost, the time complexity for the equivalent step within FPP is just $O(t)$. Of course, both protocols are prone to certain cheating behaviors, discussed below. In particular, knowing information outside of the protocol (like, for example, frequently visited locations as from geotagged data) enables an attacker to defeat both protocols with ease. This vulnerability is indicative of the need for users to safeguard geotagged information and not distribute content with geotagged embedded information in public where an attacker could build a profile of the user.

6. Scenarios of Misbehavior

The protocol is not secure when partners cheat. Consider the case Bob is cheating. He can cheat in several manners. In the first place, he can choose a very distant P by forcing the value of μ such that $|\mu - y_B|$ is very large, while deceiving $Alice$ about his distance, saying he is close to P. A strategy for the partners to find out if the other partner is close to a specified point, Q, while not revealing her/his position, is:

1. Choose a point P close to the specified point Q (with one coordinate close to P);
2. Say you are close to Q, while it may be false;
3. Determine the position of the other partner.

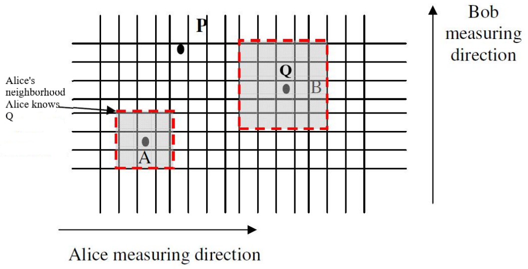

Figure 3. Example where cheating may take place: partners far apart but A guesses B's location.

As an example, assume Bob uses to frequently visit a specific pub Q. At some moment of time, $Alice$ wants to know if Bob is at that pub. Then, $Alice$ can falsely send a location close to Q and induce Bob to reveal his position. Even if the position is only approximately known, if $Alice$ has some extra knowledge ex ante of running the near-friend protocol, namely that Bob has no other business close to Q but could only be at Q, $Alice$ obtains the piece of information she desires. Other examples include home locations, office locations, frequent vacation rental locations, etc. With information from vacation photos, for example, an attacker can assemble a map of likely locations, and then defeat a nearby-friend protocol. Notice that virtually all protocols are prone to this type of attack.

However, Bob could do the following: never to provide information on his position when he is close to a frequently used location such as Q.

An extra check could alleviate the risk of such attacks. If $Alice$ and Bob say that they are close to some Q, the partners can agree to a very large radius around Q and they can agree to ask the communication provider to certify the truth of the statement 'the other is at a distance less than R from Q'. This additional check, however, adds a third-party, like $Trent$ in the Louis protocol, with all the risks involved adding an additional communication partner.

The max-Manhattan distance used in this protocol may provide an indication of gross deceit by one of the partners, while effectively concealing the true location, thus providing some added protection compared to the Louis, Lester, Pierre family of algorithms.

Attackers can use prior knowledge to defeat any protocol: the knowledge a partner may have on the most probable locations for the other partner is an advantage in the game of finding the location of the other partner while concealing one's own location.

No real-world space has the same probability that a communication party is there at some moment of time at all locations. The non-uniform spatial distribution of probability is due to regions of the space where someone cannot be — for examples spaces occupied by walls — and to regions where the party is known to have visited before. Knowing the map of where the communication partners may be allows search optimization and may reduce privacy. Denote by $p_A(t, u, v, r)$ the probability that A is located within a radius r from (u, v), (that is, that the distance from A to (u, v) is lower than r), $d((u, v), (x_A, y_A)) < r$. An attacker may be able to find the location of a target – before the target is probable to move – and with minimal energy (computational load, energy consumption) based on knowledge of $p_A(t, u, v, r)$ and by optimizing the search. For instance, a building with high population density like a skyscraper in the search radius would have a different probability than a patch of sea even if both are within the search radius r. Of course, knowledge of the target's prior locations within the search space (for instance, from prior cybersecurity breaches) would reduce the search cost even further. To mitigate this, one may apply k-anonymity (for a review of k-anonimity and its limits, see (Aslan, Matschak, Greve, Trang, and Kolbe 2023)).

In all protocols, the players can declare false locations (coordinates), regardless of a trusted third party. Prior knowledge allows a user, whatever the protocol, to lure the other user to declare if close to a specific previously known location. No protocol offers a perfect defense against this attack. On the other hand, the attacked partner may also have prior knowledge, about the fact that the attacker is aware about these preferred locations.

7. Future Work

The False Position Protocol and variants outlined here have a number of adversarial user scenarios which were briefly touched upon in this short conference paper. We are exploring the potential of expanding the False Position Protocol from enhancing location privacy to additional types of information such as, for example, vaccination status or infection status, or basic demographics, which could be transmitted as encrypted text at the same time as the position data. Such a protocol may be called, for instance, the False Information Protocol, and could be particularly useful for mobile health applications.

The COVID pandemic recently made this information-sharing tension particularly salient in the healthcare context. Containing the spread of infection requires knowing who infected people have closely contacted – individual location and health information. Fortunately, health professionals and governments now have information technology tools not available during previous pandemics. The world is now full of smart devices capable of recording identity, geolocation, proximity (e.g., Bluetooth, De Carli, Franco, Gassmann, Killer, Rodrigues, Scheid, Schoenbaechler, and Stiller 2020), and health data. As a result, organizations and governments developed a wide variety of smartphone-based contact tracing applications (Shubina et al. 2020). For example, Singapore's government released a Bluetooth-based application that required users to share data with the Ministry of Health (Cho, Ippolito, and Yu 2020). The core premise of such applications is that if users sacrifice some privacy (perhaps temporarily), society could better contain the spread of a deadly pathogen (Sharma and Bashir 2020).

For future mobile health applications in pandemics, using a decentralized contact tracing solution based on the protocol shared here could be useful to increase user trust. In such an application, for instance, only the information of when two application IDs (different from device IDs) were in proximity to one another and their infection status at the time, together with a duration of contact. This information would not require actual position data — simply knowing the self-reported infection status and that contact existed should be sufficient, as each pair of users who satisfy the conditions of the FPP would have been in contact. Of course, determining whether user trust could be enhanced with this protocol in contact tracing applications is not currently an answered question and would be worth exploring for instance through surveys before implementing it.

8. Conclusion

The proposed False Position Protocol uses a mixture of grid localization and a new distance that simplifies computations. The reduced computational load will substantially benefit the algorithm's applicability to mobile devices used in social applications. FPP also avoids requiring users to trust and coordinate with a central third party. Yet, FPP remains robust against several types of malicious actors.

While we illustrate the benefit of the FPP in the context of pandemic contact tracing for mobile health applications, the protocol has broader applications. FPP is equally helpful for any nearby friend problem where individuals wish to share information (such as location) only with specific trusted parties. The canonical example is two friends trying to meet if nearby. But we need not restrict these protocols to strict definitions of "friends" and "nearby" in a physical sense. The entities could also be autonomous vehicles trying transfer packages in a mesh network. Or workers coordinating locations to organize a "bucket brigade" (Bartholdi III, Bunimovich, and Eisenstein 1999). And location need not be physical. Entities may want to connect opportunistically if nearby in other dimensions besides geolocation. We are interested in applying the FPP in other such contexts.

References

Acquisti, Alessandro, Christina Fong. 2020. An experiment in hiring discrimination via online social networks. *Management Science* **66**(3) 1005–1024.

Adjerid, Idris, Alessandro Acquisti, Rahul Telang, Rema Padman, Julia Adler-Milstein. 2016. The impact of privacy regulation and technology incentives: The case of health information exchanges. *Management Science* **62**(4) 1042–1063.

An, Yongdae, Seungmyung Lee, Seungwoo Jung, Howard Park, Yongsoo Song, Taehoon Ko. 2021. Privacy-oriented technique for covid-19 contact tracing (protect) using homomorphic encryption: Design and development study. *Journal of medical Internet research* **23**(7) e26371.

Aslan, Aycan, Tizian Matschak, Maike Greve, Simon Trang, Lutz M. Kolbe. 2023. At what price? exploring the potential and challenges of differentially private machine learning for healthcare. Tung X. Bui, ed., *56th Hawaii International Conference on System Sciences, HICSS 2023, Maui, Hawaii, USA, January 3-6, 2023*. ScholarSpace, 3277–3286. URL `https://hdl.handle.net/10125/103034`.

Bartholdi III, John J, Leonid A Bunimovich, Donald D Eisenstein. 1999. Dynamics of two-and three-worker "bucket brigade" production lines. *Operations Research* **47**(3) 488–491.

Brody, Richard G, Christine M Haynes, Hector Mejia. 2014. Income tax return scams and identity theft. *Accounting and Finance Research* **3**(1) 90.

Chatterjee, Sanjit, Koray Karabina, Alfred Menezes. 2009. A new protocol for the nearby friend problem. *IMA International Conference on Cryptography and Coding*. Springer, 236–251.

Cheng, Reynold, Yu Zhang, Elisa Bertino, Sunil Prabhakar. 2006. Preserving user location privacy in mobile data management infrastructures. *Privacy Enhancing Technologies: 6th International Workshop, PET 2006, Cambridge, UK, June 28-30, 2006, Revised Selected Papers 6*. Springer, 393–412.

Cheng, Yudan, Jianfeng Ma, Zhiquan Liu, Yongdong Wu, Kaimin Wei, Caiqin Dong. 2022. A lightweight privacy preservation scheme with efficient reputation management for mobile crowdsensing in vehicular networks. *IEEE Transactions on Dependable and Secure Computing* .

Cho, Hyunghoon, Daphne Ippolito, Yun William Yu. 2020. Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs. *arXiv preprint arXiv:2003.11511* .

Christin, Delphine. 2016. Privacy in mobile participatory sensing: Current trends and future challenges. *Journal of Systems and Software* **116** 57–68.

Cramer, Ronald, Rosario Gennaro, Berry Schoenmakers. 1997. A secure and optimally efficient multi-authority election scheme. *European transactions on Telecommunications* **8**(5) 481–490.

De Carli, Alessandro, M Franco, A Gassmann, Christian Killer, Bruno Rodrigues, E Scheid, D Schoenbaechler, Burkhard Stiller. 2020. Wetrace–a privacy-preserving mobile covid-19 tracing approach and application. *arXiv preprint arXiv:2004.08812* .

Dong, Wei, Vacha Dave, Lili Qiu, Yin Zhang. 2011. Secure friend discovery in mobile social networks. *Proceedings IEEE INFOCOM*. IEEE, 1647–1655.

Friedland, Gerald, Robin Sommer. 2010. Cybercasing the joint: On the privacy implications of geo-tagging. *HotSec*. 1–6.

Guo, Nan, Linya Ma, Tianhan Gao. 2018. Independent mix zone for location privacy in vehicular networks. *IEEE Access* **6** 16842–16850.

Jiang, Hongbo, Jie Li, Ping Zhao, Fanzi Zeng, Zhu Xiao, Arun Iyengar. 2021. Location privacy-preserving mechanisms in location-based services: A comprehensive survey. *ACM Comput. Surv.* **54**(1). 10.1145/3423165.

Lee, Hau L, Kut C So, Christopher S Tang. 2000. The value of information sharing in a two-level supply chain. *Management Science* **46**(5) 626–643.

Li, Lode. 2002. Information sharing in a supply chain with horizontal competition. *Management Science* **48**(9) 1196–1212.

Li, Long, Jianbo Huang, Liang Chang, Jian Weng, Jia Chen, Jingjing Li. 2023. Dpps: A novel dual privacy-preserving scheme for enhancing query privacy in continuous location-based services. *Frontiers of Computer Science* **17**(5) 175814.

Li, Nan, Guanling Chen. 2010. Sharing location in online social networks. *IEEE network* **24**(5) 20–25.

Ma, Meng, Ritu Agarwal. 2007. Through a glass darkly: Information technology design, identity verification, and knowledge contribution in online communities. *Information Systems Research* **18**(1) 42–67.

Missaoui, Chedy, Safa Bachouch, Ibrahim Abdelkader, Slim Trabelsi. 2018. Who is reusing stolen passwords? an empirical study on stolen passwords and countermeasures. *Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October 29–31, 2018, Proceedings 10*. Springer, 3–17.

Mitra, Sabyasachi, Sam Ransbotham. 2015. Information disclosure and the diffusion of information security attacks. *Information Systems Research* **26**(3) 565–584.

Mohibullah, Md, Md Zakir Hossain, Mahmudul Hasan. 2015. Comparison of euclidean distance function and manhattan distance function using k-mediods. *International Journal of Computer Science and Information Security (IJCSIS)* **13**(10) 61–71.

Narayanan, Arvind, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, Dan Boneh, et al. 2011. Location privacy via private proximity testing. *NDSS*, vol. 11. 1–17.

Paillier, Pascal. 1999. Public-key cryptosystems based on composite degree residuosity classes. *International conference on the theory and applications of cryptographic techniques*. Springer, 223–238.

Papageorgiou, Achilleas, Michael Strigkos, Eugenia Politou, Efthimios Alepis, Agusti Solanas, Constantinos Patsakis. 2018. Security and privacy analysis of mobile health applications: the alarming state of practice. *Ieee Access* **6** 9390–9403.

Rose, Chris, et al. 2011. The security implications of ubiquitous social media. *International Journal of Management & Information Systems (IJMIS)* **15**(1).

Sharma, Tanusree, Masooda Bashir. 2020. Use of apps in the covid-19 response and the loss of privacy protection. *Nature Medicine* **26**(8) 1165–1167.

Sharma, Tanusree, Hunter A Dyer, Masooda Bashir. 2021. Enabling user-centered privacy controls for mobile applications: Covid-19 perspective. *ACM Transactions on Internet Technology (TOIT)* **21**(1) 1–24.

Shubina, Viktoriia, Sylvia Holcer, Michael Gould, Elena Simona Lohan. 2020. Survey of decentralized solutions with mobile devices for user location tracking, proximity detection, and contact tracing in the covid-19 era. *Data* **5**(4) 87.

Šikšnys, Laurynas, Jeppe R Thomsen, Simonas Šaltenis, Man Lung Yiu, Ove Andersen. 2009. A location privacy aware friend locator. *International Symposium on Spatial and Temporal Databases*. Springer, 405–410.

Soomro, Tariq Rahim, Mumtaz Hussain. 2019. Social media-related cybercrimes and techniques for their prevention. *Appl. Comput. Syst.* **24**(1) 9–17.

Sun, Gang, Victor Chang, Muthu Ramachandran, Zhili Sun, Gangmin Li, Hongfang Yu, Dan Liao. 2017. Efficient location privacy algorithm for internet of things (iot) services and applications. *Journal of Network and Computer Applications* **89** 3–13.

Vicente, Carmen Ruiz, Dario Freni, Claudio Bettini, Christian S Jensen. 2011. Location-related privacy in geo-social networks. *IEEE Internet Computing* **15**(3) 20–27.

Vidros, Sokratis, Constantinos Kolias, Georgios Kambourakis. 2016. Online recruitment services: Another playground for fraudsters. *Computer Fraud & Security* **2016**(3) 8–13.

Vu, Khuong, Rong Zheng, Jie Gao. 2012. Efficient algorithms for k-anonymous location privacy in participatory sensing. *2012 Proceedings IEEE INFOCOM*. IEEE, 2399–2407.

Yang, Huijie, Pandi Vijayakumar, Jian Shen, Brij B Gupta. 2022. A location-based privacy-preserving oblivious sharing scheme for indoor navigation. *Future Generation Computer Systems* **137** 42–52.

Zhang, Jia-Dong, Chi-Yin Chow. 2014. Real: A reciprocal protocol for location privacy in wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing* **12**(4) 458–471.

Zhong, Ge, Ian Goldberg, Urs Hengartner. 2007. Louis, lester and pierre: Three protocols for location privacy. *International Workshop on Privacy Enhancing Technologies*. Springer, 62–76.