Privacy-Preserving Secure Neighbor Discovery for Wireless Networks

Ahmed Mohamed Hussain
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
ahmed.hussain@ieee.org

Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

***Abstract*—Traditional Neighbor Discovery (ND) and Secure Neighbor Discovery (SND) are key elements for network functionality. SND is a hard problem, satisfying not only typical security properties (authentication, integrity) but also verification of direct communication, which involves distance estimation based on time measurements and device coordinates. Defeating relay attacks, also known as "wormholes", leading to stealthy Byzantine links and significant degradation of communication and adversarial control, is key in many wireless networked systems. However, SND is not concerned with privacy; it necessitates revealing the identity and location of the device(s) participating in the protocol execution. This can be a deterrent for deployment, especially involving user-held devices in the emerging Internet of Things (IoT) enabled smart environments. To address this challenge, we present a novel Privacy-Preserving Secure Neighbor Discovery (PP-SND) protocol, enabling devices to perform SND without revealing their actual identities and locations, effectively decoupling discovery from the exposure of sensitive information. We use Homomorphic Encryption (HE) for computing device distances without revealing their actual coordinates, as well as employing a pseudonymous device authentication to hide identities while preserving communication integrity. PP-SND provides SND [1] along with pseudonymity, confidentiality, and unlinkability. Our presentation here is not specific to one wireless technology, and we assess the performance of the protocols (cryptographic overhead) on a Raspberry Pi 4 and provide a security and privacy analysis.**

***Index Terms*—Privacy-Preserving, Wireless Security, Secure Neighbor Discovery, Homomorphic Encryption, Privacy, Anonymity**

I. Introduction

Mobile networks are integral to our daily lives, offering essential services through various applications, building on the growth of 5G technologies [2], and the IoT, or more specifically, Internet of Vehicles (IoV), the Internet of Medical Things (IoMT), or the Internet of Drones (IoD). These networks need security, but they also affect the privacy of the users they interact with. A critical aspect of these technologies is Neighbor Discovery (ND), enabling devices to discover and communicate with other nearby devices [3]. ND protocols, however, are susceptible to eavesdropping, spoofing, Man-In-The-Middle (MITM) attacks, and relay or wormhole attacks [3], [4]. Secure Neighbor Discovery (SND) addresses these vulnerabilities with the use of cryptographic primitives and device distance estimation based on time and/or location measurements [3]. Essentially, two nodes executing an SND protocol measure their distance in two ways, including exchanging their coordinates and authenticating each other.

Although SND can provably defeat attacks, notably including relays, curious observers learn which device is a neighbor of which and the locations of nearby devices that run SND. Even if neighbor-to-neighbor encryption were applied, curious yet honest peers running themselves the SND protocol could learn the location of their peers.

The challenge is to both have SND and prevent the disclosure of identity and location information. Existing SND solutions are not concerned with privacy dimension [3]. Existing security solutions, IPsec, TLS, (but also SEND, WPA2/3, and cellular 4/5G LTE security) do not solve the SND problem – they all provide authentication, integrity, some access control, and confidentiality. In other words, they provide only Authenticated Neighbor Discovery (AND) with a level of privacy protection but not SND.

Integrating an SND protocol with existing security protocols that offer confidentiality is not easy and would duplicate or, at best, complex functionality with pairwise security associations (e.g., an IPsec device to device tunnel and AH and ESP) and run the SND protocol with them. On the other hand, AND can be combined with privacy-enhancing approaches, e.g., by changing/randomizing the MAC and IP address of the device each time it connects to an Access Point (AP) [5], [6] or more recent proposals that re-randomize addresses per packet [7]. None of these approaches are concerned with SND. Nonetheless, the notion of ephemeral and changing identities can be useful. Overall, special care is needed in any composition of security protocols to ensure the sought properties are satisfied—in particular, SND correctness and availability—after the privacy-enhancing/preserving elements are added. Therefore, focusing on SND solutions, identities and locations are left vulnerable during the SND process.

We close this gap by introducing the first Privacy-Preserving Secure Neighbor Discovery (PP-SND) protocol to ensure secure and private node discovery for next-generation networks. Homomorphic Encryption (HE) conceals data as it allows performing computations with encrypted data [8]. HE, together with pseudonymous authentication, are the building blocks for PP-SND.

The main objective of our protocol is to prevent learning about the identity of the nodes participating in the protocol execution and their locations. This ensures (i) secure and pseudonymous SND, specifically addressing the challenge of an *honest-but-curious* adversary that executes the SND protocol with the intent to obtain sensitive data (location and identity) of other devices/users. (ii) effectively preventing internal and external adversaries from being able to identify, extract, or link information eavesdropped during the protocol execution by any of the devices in their neighborhood.

Contribution. In this paper, we identify and present the requirements for SND to be privacy-preserving. Then, we introduce the first PP-SND protocol that conceals the device identity and location when performing SND. Additionally, we evaluate the PP-SND protocol performance overhead and provide security & privacy analysis to demonstrate feasibility and robustness.

Paper Organization. Sec. II discusses the classical Two-Party SND, the adversary model, and variants of protocols solving the problem. Sec. III discusses the adversarial model relevant to privacy. Sec. IV presents the problem statement and the properties required for PP-SND. Sec. V presents the PP-SND protocol. In Sec. VI, we provide a performance evaluation and the protocol security & privacy analysis. Sec. VII discusses relevant related literature, before we conclude and discuss future work in Sec. VIII.

TABLE I: Notations used throughout the paper.

Symbol	Description
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{E}$	Alice, Bob, and Eve
ToF	Time of Flight
$d_{ToF}()$	Distance based on ToF
$d_{\text{loc}}()$	Distance based on location
$d_{HE}()$	Distance based on Homomorphic Encryption (HE) computation
ϵ	Distance error threshold
$cert$	Certificate
$PNYM$	Pseudonym
Δ	Processing time
ppk	Paillier public key
psk	Paillier secret (private) key
$\mathcal{R}_{SND}$	Secure Neighbor Discovery range
x, y	latitude, longitude
X, Y	Encrypted latitude, Encrypted longitude

II. Two-Party Secure Neighbor Discovery

Neighbor Discovery (ND) is a fundamental network protocol mechanism that enables devices to identify each other's presence/proximity [3]. Here, we focus on communication neighborhood, that is, the ability to communicate directly, essential for initializing and maintaining local network communications. ND mechanisms typically rely on broadcasting or multicasting discovery messages that may include the device's medium access control (MAC) address and possibly other identifiers (e.g., IP address) and its attributes.

Authenticated Neighbor Discovery (AND) introduces cryptographic mechanisms for securing ND messages. Digital signatures or keyed-hash Message Authentication Codes (HMACs) ensure that messages originate from the sending node and have not been tampered with in transit. The SEcure Neighbor Discovery (SEND), specified in RFC 3971 [9], utilizes Cryptographically Generated Addresses (CGAs) and a Public Key Infrastructure (PKI) to authenticate the sender and protect the integrity of the messages.

AND can thwart device impersonation and ND forgery/modification attacks. However, it cannot alone ensure that two devices that exchanged authenticated messages, whose integrity and freshness can be verified, are indeed neighbors, that is, can directly communicate with each other. To clarify this, consider a scenario with two legitimate, honest devices, Alice ($\mathcal{A}$) and Bob ($\mathcal{B}$), attempting to establish a neighbor relationship through an authenticated ND protocol. In Fig. 1a, $\mathcal{A}$ and $\mathcal{B}$ are not communication neighbors but an adversary, Eve ($\mathcal{E}$), positioned physically between $\mathcal{A}$ and $\mathcal{B}$, captures the authenticated discovery message from $\mathcal{A}$ and relays it to $\mathcal{B}$, and vice versa, without altering the content. Despite the messages being authenticated and their integrity and freshness being verifiable, $\mathcal{E}$'s intervention falsely convinces $\mathcal{A}$ and $\mathcal{B}$ of their direct neighbor relationship, as the protocol lacks the means to verify the physical source of the communication.

Extending the aforementioned simple relay attack, a remote relay attack involves more sophisticated adversarial capabilities, potentially leveraging a network of malicious nodes. In this scenario, in Fig. 1b, $\mathcal{A}$ and $\mathcal{B}$ are located in distant network segments, far beyond each other's direct communication range, i.e., not within the communication neighborhood. The adversary, controlling a set of nodes $\{\mathcal{E}_1, \mathcal{E}_2\}$, creates a relay chain that captures the discovery message from $\mathcal{A}$ and forwards it through the other malicious node to $\mathcal{B}$, and vice versa. The relayed messages integrity and authenticity are maintained, misleading $\mathcal{A}$ and $\mathcal{B}$ into believing they are neighbors, i.e., capable of direct communication.

Secure Neighbor Discovery (SND) [3], [10] extends *Authenticated* ND by combining cryptography and distance estimates based on Time of Flight (ToF) measurements and geographical coordinates. This way, SND authenticates nodes executing the protocol, ensuring the integrity (and possibly the confidentiality, if such a feature were added) of the discovery message(s), and can establish whether nodes are in direct communication, hence, protecting against relay attacks. A formally proven secure SND that utilizes time or location-based distance estimate, or both combined for distance estimation, is presented in [1]. The authors considered the adversarial model (depicted in Fig. 1a and 1b) permits an external adversary to perform message relaying with a minimum relaying delay constraint, Δ_{relay}.

In a typical Two-Party SND setup, two honest nodes, $\mathcal{A}$ and $\mathcal{B}$ in a wireless network, provisioned with cryptographic credentials, are legitimate network participants. In other words, the SND protocol does not consider internal adversarial nodes (with credentials to participate yet deviate from the protocol operation). These nodes can be part of a larger network infrastructure where secure communication is essential, and they could be stationary or mobile, depending on the type of network topology. The SND ensures that a node can

(a) (b)

Fig. 1: $\mathcal{A}$ and $\mathcal{B}$ are unable to discover one another due to (a) a challenging propagation environment, an obstacle, or (b) being far from each other. The adversary $\mathcal{E}$ is a relay with one or two devices, relaying messages between $\mathcal{A}$ and $\mathcal{B}$ and misleading them that they are neighbors while they are not.

discover its communication neighbors, even in the presence of adversaries. Two classes of proven secure SND protocols exist: Time-based (T) protocols and Time-and-Location-based (TL) protocols [1], operating in a Beaconing (B) or Challenge-Response (CR) mode. Fig. 2 and 3 illustrate the sequence of messages and operations for B and CR-TL protocols.

Time-based SND protocols rely on the precise measurement of message propagation time, converting the time a message takes to travel between two devices into a distance estimate. B-T protocols require synchronized clocks among devices; if the measured distance does not exceed a predefined threshold, devices are within direct communication range. However, T protocols guarantee SND if the relay adversary, Δ_{relay} is above a particular value [1].

Time-and-Location-based SND protocols incorporate both precise timing and geographical location information, combining the device geographical distance with the distance estimated based on ToF, offering a stronger basis for verifying neighbor relationships. Specifically, TL protocols compare the ToF estimated distance with the geographical distance obtained from secure and accurate location information, achieving SND for any $\Delta_{\text{relay}} > 0$ [1].

***B* and *CR* Variants** are variants T and TL protocols. Beacon-based (*B*) protocols are designed for efficiency, requiring only a single message for SND while participants maintain synchronized clocks to estimate the ToF. The *B* protocols are useful in scenarios where minimal communication overhead is desired. On the other hand, Challenge-Response (*CR*) protocols involve a two-message exchange (a challenge followed by a response) for ranging (distance estimation). Unlike *B* protocols, *CR* ones do not require participants' synchronized clocks for ToF estimation.

A. Specifics and Comparison of Variants

T protocols, relying solely on timing information to estimate distance, can be deceived by fast relaying of messages, misleading a node into believing it is closer than it actually is. TL protocols incorporate both timing and precise geographical location information to verify the proximity of the nodes. By comparing the measured ToF of the signals with the calculated distance based on known locations, TL protocols can accurately determine whether the responding node is within the expected range. This dual verification makes TL protocols resilient against sophisticated, low-delay relay attacks, unlike T protocols that are effective only when the Δ_{relay} is above a certain threshold.

For the *B*-TL (Fig. 2) SND protocol, $\mathcal{A}$ sends a beacon message received by $\mathcal{B}$. The beacon typically contains a timestamp t_1 and geographical location. This message is received by $\mathcal{B}$ at t_2 and $(t_2 - t_1)$ is used to verify if ToF matches the expected distance based on the geographical proximity (computed based on the included location of $\mathcal{A}$ and own location).

The *CR*-TL protocol has $\mathcal{A}$ send a timed challenge to $\mathcal{B}$, which must respond within a minimal delay. The *CR* serves as a ranging message exchange and does not involve computationally non-negligible operations to have an accurate ToF measurement (with a low constant for responding). The ToF and geographical distances are expected to be practically the same. The underlying requirement is that $\mathcal{A}$ and $\mathcal{B}$ have trustworthy location data, while Line of Sight (LoS) is necessary for availability.

An SND protocol is considered complete if the following properties are satisfied [1]:

P1 – Correctness ensures that if an honest device (node) is declared neighbor at some time t, the node must indeed be a communication neighbor at that time ($\mathcal{A}$ being able to receive messages from $\mathcal{B}$).

P2 – Availability implies that the protocol ascertains neighbor relationships for every distance within a specified ND range (R), i.e., $\mathcal{R}_{\text{SND}}$ is $\leq R$, the radio/datalink nominal range.

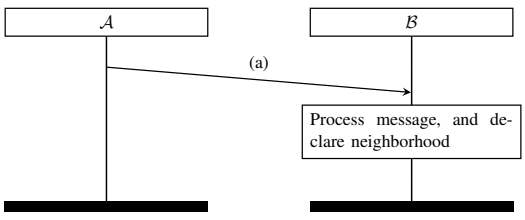


Fig. 2: TL Beacon-based SND protocol variant. Where message (a) is Beacon Message contains ⟨time, location⟩. This variant requires both participants to have synchronized clocks.

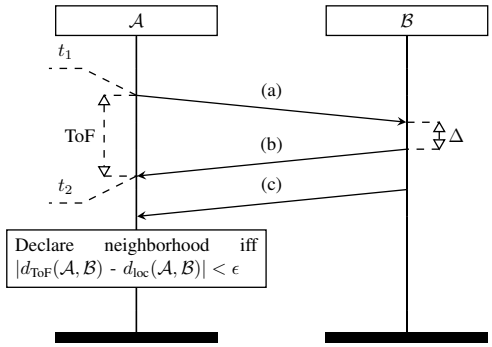


Fig. 3: *CR*-TL SND Protocol variant. (a) is $\mathcal{A}$'s challenge $\langle \text{time}, \text{n}_1 \rangle$, (b) is $\mathcal{B}$'s response $\langle \text{time}, \text{n}_2 \rangle$, and (c) contains $\langle \text{time}, \text{location}(\mathcal{B}), \text{auth}_{\mathcal{B}}(\text{n}_1, \text{n}_2, \text{location}(\mathcal{B})) \rangle$. This variant does not require clocks to be synchronized.

III. Adversarial Model

We consider both internal and external adversaries, sophisticated yet with realistic adversarial capabilities. The adversary is fundamentally a relay, $\mathcal{E}$, with one or two devices, as described in Sec. II, that attacks SND protocols executed by two honest nodes. We extend this adversarial model, depicted in Fig. 4, considering first external adversaries (that is, nodes that are not provisioned with cryptographic credentials) that seek to extract/infer information about other nodes.

1) Eavesdroppers, logging SND messages to collect information on neighbor relationships node identities (for all SND protocols) and node locations (for *TL* SND protocols).
2) Honest-but-curious nodes that actively initiate the SND protocol to either measure their distance to other nodes (in case of CR protocols) or have them reveal their location (for *CR-TL* protocols).

We also consider external adversaries targeting the protocol availability by:

3) Injecting forged messages for ToF measurements (for *CR* protocols)
4) Initiating the SND protocol with broadcast messages to engage nearby nodes and exhaust their resources (computation, power, bandwidth).

Then, we depart from the two-party SND model, considering *honest-but-curious* internal nodes, i.e., provisioned with cryptographic credentials and entitled to execute SND. They execute the SND protocol correctly but:

5) They eavesdrop on SND protocol executions by other nodes, similar to external eavesdroppers.
6) They initiate the SND protocol, as authorized to do, but at a high rate, with the intent to complete it, establish neighbor relationships repeatedly, and maintain fine-grained distance (*CR-T* protocols) and location (*CR-TL* protocols) data for their peers.

For *B* protocols, adversaries face inherent limitations due to the design of the protocol itself. As these protocols involve sending beacon messages at pre-determined, often rapid intervals, attackers cannot force SND nodes to act more quickly/frequently than they are meant to. When *B* protocols are executed, an eavesdropper faces significant challenges in determining who is a neighbor of whom. As beacons are typically broadcasted and do not require responses from receiving nodes, an eavesdropper cannot easily determine whether any two nodes are actually communicating directly or merely listening to the same broadcasts.

The rate of execution of SND protocols can be either a parameter selectable by the protocol designers or dynamically decided by the nodes based on the network's state and requirements. This flexibility enables the network to adapt to varying conditions, optimizing both performance and security. However, it is essential to find the optimal SND rate with the potential risk of information disclosure: a higher SND rate, while providing more frequent updates and potentially higher security, also means that more information about node presence and proximity is broadcasted more often, possibly leveraged by an eavesdropper. If the protocol definition includes a minimum period τ_{SND} for SND execution, this sets a boundary that nodes, especially *honest-but-curious* ones, will not exceed. Adhering to τ_{SND} prevents nodes from initiating SND processes (too) frequently, thus not only conserving network resources but also minimizing exposure of network topology (or node interactions). The ability of the adversary to relay fast is important - if the relay is fast enough, T protocols can be defeated. On the other hand, the lower the communication range for the SND nodes, the harder it is to have adversaries with technical capabilities to perform attacks successfully. The so-called distance-decreasing attacks [11] at the physical layer are not within the scope of this work.

IV. Problem Statement

While we aim for SND, as presented in Sec. II, we extend the adversarial model to consider adversaries that target SND and seek to exfiltrate node information based on the SND execution. Hence, we build on top of the proven secure SND protocol, with the aim to design a protocol that enables secure and privacy-preserving SND, mitigating the risk posed by *honest-but-curious* internal adversaries, and external adversaries (as defined in Sec. III). Our objective is to have a protocol that thwarts learning about the identity of the nodes participating in the protocol execution and their locations, effectively preventing internal and external adversaries from being able to identify, extract, or link information eavesdropped during the protocol execution by any of the nodes.

Properties of a Privacy-Preserving Secure Neighbor Discovery. In addition to **P1** and **P2** (discussed in Sec. II), an SND protocol is considered privacy-preserving if the following properties are also met:

P3 – Pseudonymity[1] (or pseudonymous authentication) refers to the ability of participating nodes to hide their true identi-

[1]Pseudonymity can be augmented to anonymity (always along with security, notably authenticity, and integrity) with the use of anonymous authentication in schemes such as in [12]. This would require a different protocol design in our context, and it is beyond the scope of this work.

(a) (b)

Fig. 4: Extended Adversarial Model. (a) External or *honest-but-curious* (internal) adversary eavesdropping communications between $\mathcal{A}$ and $\mathcal{B}$ while they execute SND protocol. $\mathcal{E}$ eavesdrops messages exchanged to learn about $\mathcal{A}$ and $\mathcal{B}$. (b) External adversary $\mathcal{E}_1$ attempting to initiate the SND protocol with $\mathcal{A}$ or *honest-but-curious* $\mathcal{E}_2$ executing the SND protocol at a high rate, both with the intent to learn (identity and location) about or track neighboring nodes.

ties/credentials during the communication process from external adversaries, *honest-but-curious* participants, and honest n. des (neighbors), while still having their messages authenticated. In particular, each node uses a typical ephemeral identity and credential, termed pseudonym, and a corresponding ephemeral private key, that cannot be linked/connected to the actual identity and long-term credential and private key of the node.

P4 – Confidentiality refers to the protection of sensitive information, notably the location of nodes executing the SND protocol. External nodes cannot obtain it, and it is not revealed to the other protocol participants.

P5 – Unlinkability refers to the ability of a device to participate in the protocol without its identity linked across multiple protocol executions. It should be impossible for any observer or another participating device to determine whether two SND executions by the same node were indeed so.

V. Proposed Solution

The PP-SND protocol relies on pseudonyms to conceal the identity of the nodes participating in the protocol execution and provides authentication. Additionally, distance computations employ privacy-preserving techniques to preserve location information. This section presents the considered setup and overall design (Sec. V-A), and the core elements, namely, pseudonymous authentication (Sec. V-B), privacy-preserving distance estimation (Sec. V-C), of the PP-SND protocol. We then present the design in a nutshell (Sec. V-D) and the protocol operation and its phases (Sec. V-E).

A. Considered Setup and Design Choices

Considered Setup. We consider two nodes (devices), $\mathcal{A}$ and $\mathcal{B}$, which could be deployed as a part of any general infrastructure where secure communication is essential. $\mathcal{A}$ and $\mathcal{B}$ are equipped with radios that support a nominal communication range $\mathcal{R}$ and the SND range, $\mathcal{R}_{SND} < \mathcal{R}$, a more conservative range (device distance) for which SND is executed, and concludes with respect to $\mathcal{R}_{SND}$ whether $\mathcal{A}$ and $\mathcal{B}$ are neighbors. While $\mathcal{A}$ and $\mathcal{B}$ can be located at specific (latitude, longitude) coordinates, i.e., stationary, our setup acknowledges the possibility of mobility. Assuming that mobility does not change significantly within the protocol execution, $\mathcal{R}_{SND}$ can be chosen so that the maximum relative node speed does not cause a displacement close to $\mathcal{R} - \mathcal{R}_{\text{SND}}$.

SND Protocol Choice. *CR*-SND protocols require an exchange of a challenge and a response between the participating nodes, inherently leading to solutions that provide mutual authentication and SND since both parties actively participate. In this work, we design the PP-SND protocol based on the *CR*-TL SND protocol. It necessitates an exchange between nodes, naturally utilizing both ToF and precise geographical location coordinates. *CR*-TL protocols are particularly effective against sophisticated relay attacks.

B. Pseudonymous Authentication

Long Term Certificates (LTCs) and Long Term Certification Authority (LTCA) are essential for accountability within a Public Key Infrastructure (PKI). The LTCA is responsible for registering entities, such as devices or users, and issuing one LTC. Using their LTC and their corresponding long-term private key, devices can obtain anonymous tokens from the LTCA, using, in turn, the tokens to anonymously request pseudonyms. Pseudonyms (PNYM) are ephemeral certificates [12], [13], from a Pseudonym Certificate Authority (PCA). The pseudonyms are anonymized, unique user/entity representations, replacing the LTC, in the form of ⟨Pseudonym Provider ID, Pseudonym Lifetime, Public Key (pk), Pseudonym Provider Signature⟩; the PNYM Provider ID refers to the PCA whose signature is used to validate the pseudonym and establish trust. *Pseudonym Lifetime* indicates how long the pseudonym is valid. *Public Key* (pk) together with the corresponding private key are generated by the device, with pk sent via a certificate signing request to the PCA. Similar to the IoV PCA and pseudonymous authentication, each node gets K pseudonyms, each valid for a specific lifetime τ [14].

The PP-SND protocol relies on the use of a modified pseudonym structure that includes two additional elements, namely, the Paillier public key [15] addition to the aforementioned public key. Given the design in [12], [13], and for clarity, we refer to this as the ECDSA public key. The new pseudonym becomes ⟨Pseudonym Provider ID, Pseudonym Lifetime, Paillier Public Key (ppk), ECDSA public Key, Pseudonym Provider Signature⟩.

C. Privacy-Preserving Distance Estimation

Coordinate-based Distance Estimation can be performed using homomorphic computations on the node encrypted co-

ordinates [16], [17]. This ensures that no participating nodes executing the protocol learn about one another's coordinates. A notable implementation in [18] utilizes HE to estimate if an entity lies within a "Geofence". This is done by leveraging the *Paillier* cryptosystem [15] homomorphic properties: it provides Partially Homomorphic Encryption (PHE), allowing for the additive combination of encrypted values. Specifically, given two plaintexts, m_1 and m_2, their respective encryptions, $Enc(m_1)$ and $Enc(m_2)$, can be multiplicatively combined to yield $Enc(m_1+m_2)$. Decryption of this result reveals the sum of $m_1 + m_2$. Furthermore, Paillier PHE facilitates subtraction by computing the modular inverse of $Enc(m_2)$ and then multiplying it with $Enc(m_1)$, thereby obtaining $Enc(m_1 - m_2)$. Additionally, scalar multiplication is achievable, allowing for the calculation of $Enc(m_1 \cdot m_2)$ through the exponentiation of $Enc(m_1)^{m_2}$.

Initially, for the PP-SND protocol, $\mathcal{B}$ encrypts its coordinates using $\mathcal{A}$'s ppk. It performs HE operations (mainly subtraction) between the encrypted $\mathcal{A}$ lat and lng and its own. This allows only $\mathcal{A}$ to decrypt the difference using its psk and compute the Euclidean distance, in addition to the distance based on $\text{ToF}_{\mathcal{AB}}$, to verify if it is within range or not. This step is essential for maintaining privacy, as neither party reveals their actual coordinates.

Time of Flight-based Distance Estimation computes the ToF, with $\mathcal{A}$ initiating a local timer at t_1 once it starts the discovery process. Upon the reception of the response packet from node $\mathcal{B}$ at time t_2, the timer stops. The time difference, $\delta = t_2 - t_1$, is used to compute $d_{\text{ToF}} = \frac{c \cdot \delta}{2}$, where c (3×10^8) is multiplied by δ, the Round Trip Time (RTT), divided by 2 to estimate the distance.

D. Design in Nutshell

The PP-SND protocol enables secure and privacy-preserving neighbor discovery in wireless networks. Each entity generates Paillier and Elliptic Curve Digital Signature Algorithm (ECDSA) key pairs for HE operations and signing/verification, respectively. Initially, entity $\mathcal{A}$ broadcasts an advertisement message containing its identifier, an authentication token, and a pseudonym to nearby devices, maintaining pseudonymity and integrity.

During the ranging phase, $\mathcal{A}$ sends a message to $\mathcal{B}$ with its identifier and a nonce, initiating a timer to calculate the ToF. $\mathcal{B}$ responds with its nonce, upon receipt $\mathcal{A}$ stops its timer, allowing $\mathcal{A}$ to estimate the distance based on ToF. To authenticate the ranging exchange, $\mathcal{B}$ sends a message back to $\mathcal{A}$, an authenticator of the exchange, also attaching its pseudonym. $\mathcal{A}$ verifies the authenticator and checks if the $d_{\text{ToF}}(\mathcal{A}, \mathcal{B})$ is within a predefined secure range, $\mathcal{R}_{\text{SND}}$.

In the coordinate exchange and distance calculation step, $\mathcal{A}$ encrypts its coordinates using its Paillier public key, ppk, and sends them to $\mathcal{B}$ along with a signed message. $\mathcal{B}$ uses $\mathcal{A}$'s ppk to encrypt its coordinates and calculate the HE difference, it then sends the encrypted coordinate difference to $\mathcal{A}$. $\mathcal{A}$ decrypts using its Paillier secret key, psk, the differences and computes the Euclidean distance, essentially $d_{\text{loc}}(\mathcal{A}, \mathcal{B})$ without having any access to loc($\mathcal{B}$). Finally, $\mathcal{A}$ declares $\mathcal{B}$ as a neighbor if ToF and HE-based distances are within an acceptable low threshold, i.e., $< \epsilon$.

Combining the aforementioned components results in a secure and privacy-preserving ND. Indeed, ToF measurements and homomorphically encrypted location coordinates ensure that distance is derived without exposing the actual location information. Moreover, the distance between the communicating entities does not provide precise information on where the devices are, not even direction.

E. Protocol Operation Phases

Recall that each entity in the current PNYM lifetime has two sets of keys before engaging in the protocol: Paillier key pair (ppk, psk) for HE operations and ECDSA key pair for signing and verifying. The PNYM includes Paillier public key (ppk) and ECDSA public key. Additionally, $auth_{\mathcal{A}}$ and $auth_{\mathcal{B}}$ are digital signatures computed with the node's current private key (corresponding to the current PNYM public key). Fig. 5 depicts the protocol messages sequence.

First, an initialization/advertisement message is transmitted by $\mathcal{A}$, which initiates the protocol by broadcasting a message to all nearby devices. This message (a) includes its identifier $\mathcal{A}$, an authentication $auth_{\mathcal{A}}(n_1)$, and $PNYM_{\mathcal{A}}$. This broadcast announces $\mathcal{A}$'s presence while keeping its identity hidden. Moreover, $auth_{\mathcal{A}}(n_1)$ ensures authenticity, and integrity, proving that $\mathcal{A}$ is a legitimate participant.

Second, $\mathcal{A}$ sends a follow-up message (b), initiating ranging, containing $\mathcal{A}$'s identifier and the nonce n_1. This step initiates a timer (at t_1) on $\mathcal{A}$, to calculate the ToF that will be used for $d_{\text{ToF}}(\mathcal{A}, \mathcal{B})$. $\mathcal{A}$ waits for $\mathcal{B}$'s response. $\mathcal{B}$ checks if $h(n_1)$, using n_1, equals the field in message (a) and aborts the protocol run otherwise. Then $\mathcal{B}$ replies (c) with a nonce n_2, and $\mathcal{A}$ stops the timer (at t_2) upon receipt, and computes d_{ToF}. This exchange enables $\mathcal{A}$ to estimate the distance based on ToF. $\mathcal{B}$ sends a follow-up message (d) that contains $auth_{\mathcal{B}}(n_1, n_2)$, and its $PNYM_{\mathcal{B}}$. This enables $\mathcal{A}$ to verify if $\mathcal{B}$ is a legitimate and honest node. $\mathcal{A}$ then checks if $d_{\text{ToF}}(\mathcal{A}, \mathcal{B}) < \mathcal{R}_{SND}$ to either continue the execution of the protocol or terminates.

Once $\mathcal{A}$ verifies the $d_{\text{ToF}}(\mathcal{A}, \mathcal{B}) < \mathcal{R}_{SND}$, it encrypts its coordinates (lat, lng) using its Paillier public key, $ppk_{\mathcal{A}}$, and sends it to $\mathcal{B}$ along with the $auth_{\mathcal{A}}(message)$ (message refers to the entire payload). This is the message (e) in the exchange. $\mathcal{B}$ then uses $ppk_{\mathcal{A}}$ to encrypt its own coordinates and perform HE computations on the encrypted data to obtain the coordinate differences. $\mathcal{B}$ then sends a message (f) containing the encrypted differences ($diff\_lat, diff\_lng$), and $auth(message)$. $\mathcal{A}$ decrypts the difference using its own $psk_{\mathcal{A}}$ and estimate the euclidean distance. Finally, $\mathcal{B}$ is declared a neighborhood iff $|d_{\text{ToF}}(\mathcal{A}, \mathcal{B}) - d_{\text{HE}}(diff\_lat, diff\_lng)| < \epsilon$. The sequence of message exchanges is summarized as follows:

EncCoord (Algorithm 1) and **HEC** (Algorithm 2) are used to encrypt coordinates using a given ppk and perform HE difference, respectively. **EncCoord** initially uses Algorithm 3 to normalize the input values (degree) to achieve better accuracy.

(a) $\mathcal{A} \rightarrow * : \langle \mathcal{A}, h(n_1), auth_{\mathcal{A}}(n_1), PNYM_{\mathcal{A}} \rangle$

(b) $\mathcal{A} \rightarrow * : \langle \mathcal{A}, n_1 \rangle$ # *Ranging message*

(c) $\mathcal{B} \rightarrow \mathcal{A} : \langle \mathcal{B}, \mathcal{A}, n_2 \rangle$ # *Ranging message*

(d) $\mathcal{B} \rightarrow \mathcal{A} : \langle \mathcal{B}, \mathcal{A}, n_2 + 1, auth_{\mathcal{B}}(\mathcal{A}, n_1, n_2 + 1), PNYM_{\mathcal{B}} \rangle$

(e) $\mathcal{A} \rightarrow \mathcal{B} : \langle \mathcal{A}, \mathcal{B}, X_{\mathcal{A}}, Y_{\mathcal{A}}, auth_{\mathcal{A}}(X_{\mathcal{A}}, Y_{\mathcal{A}}) \rangle$

(f) $\mathcal{B} \rightarrow \mathcal{A} : \langle \mathcal{B}, \mathcal{A}, diff\_lat, diff\_lng, auth_{\mathcal{B}}(diff\_lat, diff\_lng) \rangle$

Final Step $\mathcal{A}$: Declare neighborhood iff $|d_{\text{ToF}}(\mathcal{A}, \mathcal{B}) - d_{\text{HE}}(diff\_lat, diff\_lng)| < \epsilon$

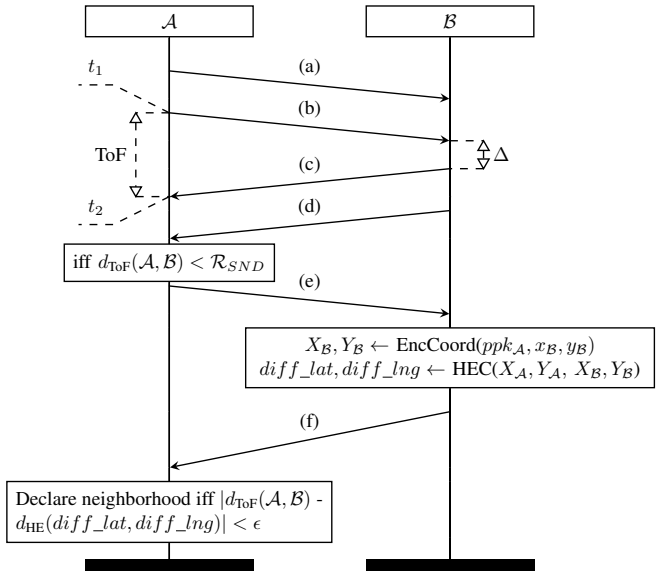


Fig. 5: PP-SND Protocol.

Algorithm 1: Encrypt Coordinates (EncCoord)

Input: $public\_key$, lat, lng
Output: $encrypted\_lat$, $encrypted\_lng$
$normalized\_lat \leftarrow normalize\_deg(lat)$;
$normalized\_lng \leftarrow normalize\_deg(lng)$;
$encrypted\_lat \leftarrow public\_key.encrypt(normalized\_lat)$;
$encrypted\_lng \leftarrow public\_key.encrypt(normalized\_lng)$;
return $(encrypted\_lat, encrypted\_lng)$;

Algorithm 2: Homomorphically Compute the Difference Between Input and Participant Coordinates (HEC)

Input: $encrypted\_input\_lat$, $encrypted\_input\_lng$, $my\_encrypted\_lat$, $my\_encrypted\_lng$
Output: $diff\_lat$, $diff\_lng$
$diff\_lat \leftarrow encrypted\_input\_lat - my\_encrypted\_lat$;
$diff\_lng \leftarrow encrypted\_input\_lng - my\_encrypted\_lng$;
return $(diff\_lat, diff\_lng)$;

Algorithm 3: Normalize Degrees for Encryption

Input: deg, is_lat
Output: $normalized\_deg$
if is_lat **then**
  $normalized\_deg \leftarrow deg + 90$;
end
else
  $normalized\_deg \leftarrow deg + 180$;
end
$normalized\_deg \leftarrow normalized\_deg \cdot normalize\_factor$;
$normalized\_deg \leftarrow round(normalized\_deg)$;
return $normalized\_deg$;

VI. Performance Evaluation and Security & Privacy Analysis

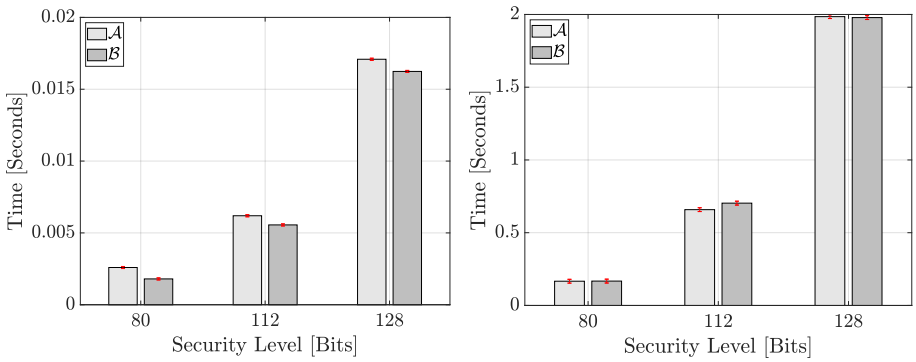
This section provides details on SND and PP-SND cryptographic performance overhead evaluated on Raspberry Pi (Sec. VI-A). Additionally, achieved security and privacy properties are analyzed (Sec. VI-B).

A. SND and PP-SND Performance Evaluation

The protocol implementation is done using Python 3.11. We selected the *Paillier* cryptosystem [15], known for its multiplicative and additive homomorphic properties (as earlier discussed in Sec. V-B) with the use of the *phe* 1.5.0 python library [19]. Additionally, the gmpy2 [20] extension is used as it supports fast multiple-precision arithmetic to speed up the HE computations. ECDSA was employed for digital signatures, specifically utilizing the *brainpoolP256r1* curve parameters [21]. The signing and verifying keys are 256 bits each. To assess the efficiency and performance, we benchmark the cryptographic overhead imposed through a $10,000$ simulations, i.e., emulated executions of the protocol—with the processing by the actual device. This is done using different key sizes, i.e., 1024, 2048, and 3072 bits, corresponding to different security levels, i.e., 80, 112, and 128 bits, recommended by NIST [22]. The execution time mean value is then computed, in addition to the 95% Confidence Interval (CI). We evaluated the protocol on a Raspberry Pi 4, running a 4 cores Broadcom BCM2711 Quad-core Cortex-A72 (ARM v8) CPU, with a clock speed of 1.8 GHz, and 8 GB of RAM.

Fig. 6 illustrates the overall performance of the two protocols, i.e., SND and PP-SND, for $\mathcal{A}$ and $\mathcal{B}$ as a function of security level. Fig. 6a shows the SND performance. Indeed, increasing the key size increases computation overhead and, hence, more processing time. In all executions, $\mathcal{A}$ takes more time than $\mathcal{B}$, due to verifying the authenticated message of $\mathcal{B}$ and computing the distance.

Similarly, Fig. 6b shows the PP-SND performance. Unlike SND, the PP-SND results in higher overhead, due to the time needed to perform the HE computation, i.e., encrypting the coordinates of $\mathcal{A}$ and $\mathcal{B}$, performing the subtraction on the encrypted values, and decrypting the difference to estimate the distance. Hence, the computation overhead of $\mathcal{A}$ and $\mathcal{B}$ is similar. $\mathcal{A}$ encrypts its own coordinates using $ppk_{\mathcal{A}}$. Similarly, $\mathcal{B}$ does the same to its coordinates using $ppk_{\mathcal{A}}$ (Algorithm 1),



(a) (b)

Fig. 6: Evaluation of the cryptographic overhead for each protocol, (a) SND, and (b) PP-SND, as a function of different security levels.

and then perform subtraction (Algorithm 2). Additionally, $\mathcal{A}$ decrypts the final difference. These operations significantly increase the execution time. Furthermore, increasing the security level (thus the key size) increases processing time.

Although the PP-SND adds significant overhead compared to the SND execution time, it ensures the protocol participants have a secure and privacy-preserving SND without leaking any information about the involved parties.

B. Security & Privacy Analysis

SND Protocol. The proven secure SND protocol in [1] utilizes ToF measurements and location information to prevent potential adversaries from relaying or replaying messages between nodes in the network. The presented PP-SND protocol leverages ToF measurements and uses HE techniques for the coordinate-based distance estimation, ensuring that they are (within $\mathcal{R}_{\text{SND}}$) and, most important, within ϵ of each other, fulfilling **P1** and **P2**.

Correctness (P1) $\mathcal{A}$ and $\mathcal{B}$ are declared neighbors if and only if the estimated distance using ToF and $d_{\text{loc}}(\mathcal{A},\mathcal{B})$, while $\mathcal{R}_{\text{SND}}$ is the maximum SND range: $d_{\text{loc}}(\mathcal{A},\mathcal{B}) \sim d_{\text{ToF}}(\mathcal{A},\mathcal{B}) \leq \mathcal{R}_{\text{SND}}$, then $\mathcal{A}$ and $\mathcal{B}$ are considered neighbors.

Availability (P2) is achieved if for any d such that $0 < d \leq \mathcal{R}_{\text{SND}}$, there exists a mechanism that ensures: $\exists\, d_{loc}(\mathcal{A},\mathcal{B})$ calculated using ToF and HE such that $\{d_{\text{ToF}}(\mathcal{A},\mathcal{B}) \wedge d_{\text{HE}}(\mathcal{A},\mathcal{B})\} \approx d_{loc}(\mathcal{A},\mathcal{B})$ enabling the protocol to identify a node as a neighbor within the operational range. Let $\mathcal{A}$ and $\mathcal{B}$ be two nodes participating in the protocol, with ToF and HE privacy-preserving distance estimations. Let $\delta_1 \doteq |d_{\text{ToF}}(\mathcal{A},\mathcal{B}) - d_{loc}(\mathcal{A},\mathcal{B})|$ be the ToF estimate error and $\delta_2 \doteq |d_{\text{HE}}(\mathcal{A},\mathcal{B}) - d_{loc}(\mathcal{A},\mathcal{B})|$ be the HE estimate error. Since we know that δ_1 is negligible [1], $\delta_1 = \delta_2$, then P1 and P2 hold.

Pseudonymity and Identity Protection. The use of pseudonyms plays a key role in hiding user identities. By substituting real identifiers with temporary, dynamically changing pseudonyms, the PP-SND protocol hides the real identity of the communicating parties. This approach effectively prevents adversaries from linking these pseudonyms to the actual user. Hence, satisfying both properties **P3** and **P5**.

Let $\mathcal{N} = \mathcal{A},\mathcal{B},\ldots$ denote the set of nodes participating in the protocol. Let $\mathcal{P} = P_{\mathcal{A}}, P_{\mathcal{B}},\ldots$ denote the set of pseudonyms corresponding to the nodes in $\mathcal{N}$. For any node $\mathcal{A} \in \mathcal{N}$, $P_{\mathcal{A}}^{(t)}$ denotes the pseudonym used by $\mathcal{A}$ at time t.

Pseudonymity (P3) is preserved since each entity uses a sequence of pseudonymous identities (certificates) that cannot be connected to its actual long-term identity (LTC). Any observer (including adversaries or other protocol participants) cannot link different pseudonyms to the same entity, thereby preserving pseudonymity. By design, not even the issuer of the pseudonyms—the PCA—knows which LTC they were issued for. This ensures that the PCA cannot link a pseudonym back to the LTC or the long-term identity of the entity. As a result, unless the node itself uses its long-term private key and LTC, no information about the entity's identity is revealed.

Unlinkability (P5) is achieved by utilizing a new pseudonym at each protocol execution instance with the corresponding cryptographic keys. Any observer, protocol participant, or adversary cannot determine whether this new pseudonym belongs to the same node as any previously observed pseudonym, based on the included cryptographic keys and operations; i.e., syntactic unlinkability. Each pseudonym $P_{\mathcal{A}}^{(t)}$ is generated to be independent of the previous one $P_{\mathcal{A}}^{(t-1)}$, and all $P_n^{(t)}, \forall n \in \mathcal{N}$, are issued with the same validity period, thereby preventing linkability through pattern analysis or identifier correlation. Furthermore, we consider semantic unlinkability [14], which extends beyond the identifiers to include the context and content of the messages exchanged. Semantic unlinkability ensures that even if the content of the messages is analyzed, an adversary cannot infer that different messages are related or originated from the same node. The content of the PP-SND messages is encrypted (homomorphically), thus not leaving any useful information to the observer or participant to link interactions once the pseudonym and the identity of the device and its public key change (transition to the next identity/pseudonym lifetime). This could have been the case if, for example, location were visible and the adversary could

infer based on mobility patterns [23]. Additionally, in a static setup, the computed distance can lead the *honest-but-curious* to infer that two sessions could have been executed with the same peer node.

In each session, a node can select a new pseudonym with new private keys not linked to its real identity or any previous pseudonym/keys it has used. The protocol implementation is expected to ensure that all device identifiers across the protocol stack are changed at the time of the pseudonym/identity transition. For example, this can be done in modern platforms by changing the MAC address and obtaining a new IP address.

Moreover, observing, recording, or analyzing the pseudonyms used in multiple sessions provides no useful information to link these sessions to each other or the real identity of a node. Thus, even if $\mathcal{E}$ could observe all communication sessions, the lack of any deterministic pattern or link between the pseudonyms across sessions makes it computationally impractical to achieve meaningful linkage.

Confidentiality (P4): The content, notably location, remains inaccessible to adversaries. The exchange of information computed via HE ensures that an adversary cannot deduce the location of the communicating parties. HE allows for operations to be performed on encrypted data, ensuring that the outcome remains encrypted and can be only decrypted by the intended recipient, i.e., the node that holds the corresponding psk to its ppk.

Let $C_{\mathcal{A}} = (lat_{\mathcal{A}}, lon_{\mathcal{A}})$ and $C_{\mathcal{B}} = (lat_{\mathcal{B}}, lon_{\mathcal{B}})$ represent the geographical coordinates of $\mathcal{A}$ and $\mathcal{B}$, respectively. $E_{ppk}(x)$ and $D_{sk}(x)$ denote the encryption and decryption of x using the Paillier public key ppk and the private key psk, respectively. $E_{ppk_{\mathcal{A}}}(CD_{\mathcal{AB}}) = (E_{ppk_{\mathcal{A}}}(lat_{\mathcal{A}}) - E_{ppk_{\mathcal{A}}}(lat_{\mathcal{B}}), E_{ppk_{\mathcal{A}}}(lon_{\mathcal{A}}) - E_{ppk_{\mathcal{A}}}(lon_{\mathcal{B}}))$ and $E_{ppk_{\mathcal{B}}}(CD_{\mathcal{BA}}) = (E_{ppk_{\mathcal{B}}}(lat_{\mathcal{B}}) - E_{ppk_{\mathcal{B}}}(lat_{\mathcal{A}}),\ E_{ppk_{\mathcal{B}}}(lon_{\mathcal{B}}) - E_{ppk_{\mathcal{B}}}(lon_{\mathcal{A}}))$ represents the homomorphically computed coordinates difference between $\mathcal{AB}$, and $\mathcal{BA}$, respectively.

The Paillier Cryptosystem supports additive homomorphic encryption, where for any two plaintexts a and b [15]: $E_{ppk}(a) \times E_{ppk}(b) \equiv E_{ppk}(a+b) \mod n^2$, where n is part of the public key in the Paillier Cryptosystem, the distance function $d_{sk_{\mathcal{A}}}(CD_{\mathcal{AB}})$ enables $\mathcal{A}$ to decrypt the computed difference, using its Paillier psk, and compute the Euclidean distance between itself and $\mathcal{B}$. Similarly, $d_{sk_{\mathcal{B}}}(CD_{\mathcal{BA}})$ enables $\mathcal{B}$ to decrypt the computed difference, using its psk, and to compute the Euclidean distance between itself and $\mathcal{A}$.

By encrypting $C_{\mathcal{A}}$ and $C_{\mathcal{B}}$ with the Paillier public key of the participating entities, we ensure the encrypted coordinates, $E_{ppk}(C_{\mathcal{A}})$ and $E_{ppk}(C_{\mathcal{B}})$, are not disclosed and that arithmetic operations can be computed homomorphically without decryption.

The properties of the Paillier Cryptosystem ensure that without access to either entity's private key psk, it does not reveal any information about the exchanged C. The operation on these encrypted values, namely, distance computation, remains secure and private. The encrypted difference is decrypted, and the distance is computed, then compared with the ToF-based distance measurement, $d_{\text{ToF}}(\mathcal{A}, \mathcal{B})$, without revealing the actual locations:

$$D_{sk_{\mathcal{A}}}(E_{ppk_{\mathcal{A}}}(CD_{\mathcal{AB}})) = d_{\text{loc}}(C_{\mathcal{A}}, C_{\mathcal{B}}) \approx d_{\text{ToF}}(\mathcal{A}, \mathcal{B})$$

This approach guarantees that data will not reveal the location or movement patterns of the parties involved. Therefore, fulfilling properties **P1**, **P3**, **P4** and **P5**.

Limitations. It is important to note that this analysis does not extend to adversaries capable of localizing devices based on physical layer features, such as Radio Signal Strength (RSS), or those that fingerprint devices based on unique characteristics of their radio equipment [24], [25]. Such methods require additional countermeasures beyond the scope of the PP-SND protocol.

VII. Related Work

The notion of **Secure Neighbor Discovery** was first introduced by [10], in the context of secure route discovery—however, that property was closer to authenticated ND. Poturalski et al. [26] explored solutions for SND [3] by proposing protocols robust against relay attackers. Additionally, in [1], they presented a comprehensive framework for the security assessment of ND protocols within wireless networks. They presented formal proofs of the security properties of four SND protocols. By integrating time-and-location information into the protocol design, they showed that SND can be achieved against fast-relaying adversaries.

Distance-bounding Protocols prevent distance fraud, relevant in our context if one of the SND participants were adversarial (recall: we consider only honest but curious internal nodes). Brands and Chaum [27] pioneered these protocols to establish practical upper bounds on physical distances between parties by measuring single-bit challenge-response delays. Hancke and Kuhn [28] extended this concept to RFID systems, developing a protocol that uses Ultra-Wide-Band (UWB) pulse communication. Their low-power, asynchronous approach proves particularly effective for passive RFID tokens in noisy environments, ensuring tokens remain within verified distances from authenticators. In [29], the authors addressed implementation challenges by developing a prototype that processes signals in less than 1 ns using Challenge Reflection with Channel Selection (CRCS). This achievement limits adversarial provers from falsely appearing closer than 15 cm to the verifier, while avoiding the demodulation delays inherent in traditional XOR and comparison methods.

Privacy-Preserving Location Estimation. Several studies have explored privacy-preserving location estimation using HE. Hallgren et al. [17] developed InnerCircle, a decentralized protocol that enables proximity detection without a trusted third party. Their approach combines Secure Multi-party Computation (SMC) and PHE to let users verify their relative distance while keeping their exact locations private. Similarly, Zhong et al. [16] proposed three protocols—Louis, Lester, and Pierre—for privacy-aware proximity detection in Location-based Services (LBS). While Louis relies on a semi-trusted third party, Lester and Pierre operate in a fully decentralized

manner. All three protocols use HE to enable users to check their friends' proximity without exposing location data to any central authority.

VIII. Conclusion and Future Work

We presented the first Privacy-Preserving Secure Neighbour Discovery (PP-SND) protocol, where the participating entities' information, notably real identity and location, is anonymized and encrypted, making it impossible for participating entities to learn about one another. We defined the properties required to have a PP-SND protocol that is both functional and privacy-preserving. We presented the overall protocol architecture, functions, and operations. Our findings demonstrate the protocol feasibility to provide secure and privacy-preserving device discovery, useful in ubiquitous computing scenarios. As for future work, we plan to implement and test the protocol across various wireless technologies. Moreover, we plan to test different cryptosystems that enable homomorphic operations on data and ensure their applicability across these technologies.

Acknowledgment

This work is supported in parts by the Swedish Research Council and the Knut and Alice Wallenberg Foundation.

References

[1] Poturalski et al., "Formal Analysis of Secure Neighbor Discovery in Wireless Networks," *IEEE Transactions on Dependable and Secure Computing (IEEE TDSC)*, vol. 10, no. 6, pp. 355–367, November 2013.

[2] U. Pingle, "5G Means Big Changes for Large Public Venues in 2023 and Beyond," https://www.commscope.com/blog/2023/5g-means-big-changes-for-large-public-venues-in-2023-and-beyond/, Jan. 2023.

[3] Papadimitratos et al., "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking," *IEEE Communications Magazine*, vol. 46, no. 2, Feb. 2008.

[4] A. AlSa'deh and C. Meinel, "Secure neighbor discovery: Review, challenges, perspectives, and recommendations," *IEEE Security & Privacy*, vol. 10, no. 4, 2012.

[5] Hugon et al., "RoMA: Rotating MAC Address for privacy protection," in *Proceedings of ACM SIGCOMM Poster and Demo Sessions*, 2022. [Online]. Available: https://doi.org/10.1145/3546037.3546055

[6] Martin et al., "A study of mac address randomization in mobile devices and when it fails," *arXiv preprint arXiv:1703.02874*, 2017.

[7] H. Jin and P. Papadimitratos, "Over-the-air runtime wi-fi mac address re-randomization," in *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, May 2024. [Online]. Available: https://doi.org/10.1145/3643833.3656122

[8] Acar et al., "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3214303

[9] J. Kempf, J. Arkko, B. Zill, and P. Nikander, "SEcure Neighbor Discovery (SEND)," RFC 3971, Mar. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc3971

[10] P. Papadimitratos and Z. J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," in *IEEE International Symposium on Applications and the Internet - Workshop on Security and Assurance in Ad Hoc Networks*, Orlando, FL, USA, Jan. 2003.

[11] Clulow et al., "So near and yet so far: Distance-bounding Attacks in Wireless Networks," in *Security and Privacy in Ad-Hoc and Sensor Networks: Third European Workshop, ESAS*. Springer, 2006.

[12] Khodaei et al., "Scaling Pseudonymous Authentication for Large Mobile Systems," in *ACM Conference on Security & Privacy in Wireless and Mobile Networks (ACM WiSec)*, Miami, FL, USA, May 2019.

[13] H. Jin and P. Papadimitratos, "Scaling VANET Security through Cooperative Message Verification," in *IEEE Vehicular Networking Conference (IEEE VNC)*, Kyoto, Japan, Dec. 2015.

[14] M. Khodaei and P. Papadimitratos, "Cooperative Location Privacy in Vehicular Networks: Why Simple Mix-zones are not Enough," *IEEE Internet Of Things Journal*, vol. 8, no. 10, pp. 7985–8004, May 2021.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, ser. EUROCRYPT'99, May 1999.

[16] Zhong et al., "Louis, Lester and Pierre: Three Protocols for Location Privacy," in *Privacy Enhancing Technologies*, N. Borisov and P. Golle, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[17] Hallgren et al., "Innercircle: A parallelizable decentralized privacy-preserving location proximity protocol," in *13th Annual Conference on Privacy, Security and Trust (PST)*, 2015, pp. 1–6.

[18] N. Doiron, "Crypto and privacy village - geolocation and homomorphic encryption," DEF CON, 2018. [Online]. Available: https://doi.org/10.5446/39868

[19] C. Data61, "Python paillier library," https://github.com/data61/python-paillier, 2013.

[20] GMP. General Multiprecision PYthon project. Accessed April 2023. [Online]. Available: https://gmpy2.readthedocs.io/en/latest/

[21] Chen et al., "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters," National Institute of Standards and Technology, Tech. Rep., 2023.

[22] Damien Giry. Cryptographic Key Length Recommendation. Accessed April 2023. [Online]. Available: https://www.keylength.com/en/4/

[23] Buttyán et al., "SLOW: A Practical pseudonym changing scheme for location privacy in VANETs," in *IEEE Vehicular Networking Conference (VNC)*, Tokyo, Japan, Oct. 2009.

[24] Lin et al., "Wireless device identification based on radio frequency fingerprint features," in *IEEE International Conference on Communications (ICC)*. Dublin, Ireland: IEEE, Jun. 2020.

[25] S. Al-Hazbi, A. M. Hussain, S. Sciancalepore, G. Oligeri, and P. Papadimitratos, "Radio Frequency Fingerprinting via Deep Learning: Challenges and Opportunities," in *2024 International Wireless Communications and Mobile Computing (IWCMC) Security Symposium*, Ayia Napa, Cyprus, May 2024, pp. 1–6.

[26] Poturalski et al., "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," in *ACM Symposium on Information, Computer and Communications Security (ACM ASIACCS)*, Tokyo, Japan, March 2008, pp. 189–200.

[27] S. Brands and D. Chaum, "Distance-Bounding Protocols," in *Advances in Cryptology — EUROCRYPT '93*, 1994.

[28] G. Hancke and M. Kuhn, "An rfid distance bounding protocol," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, Sep. 2005, pp. 67–73.

[29] K. B. Rasmussen and S. Capkun, "Realization of RF Distance Bounding," in *USENIX Security Symposium*, Washington, DC, Aug. 2010. [Online]. Available: https://www.usenix.org/conference/usenixsecurity10/realization-rf-distance-bounding