Incorporating Obfuscation Techniques in Privacy Preserving Database-Driven Dynamic Spectrum Access Systems

Douglas M. Zabransky

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Masters of Science

in

Computer Engineering

Yaling Yang, Chair

Joseph G. Tront

Ryan M. Gerdes

August 7, 2018

Blacksburg, Virginia

Keywords: Dynamic Spectrum Access, Inference Attack, Spectrum Efficiency, Homomorphic Encryption

Incorporating Obfuscation Techniques in Privacy Preserving Database-Driven Dynamic Spectrum Access Systems

Douglas M. Zabransky

(ABSTRACT)

Modern innovation is a driving force behind increased spectrum crowding. Several studies performed by the National Telecommunications and Information Administration (NTIA), Federal Communications Commission (FCC), and other groups have proposed Dynamic Spectrum Access (DSA) as a promising solution to alleviate spectrum crowding. The spectrum assignment decisions in DSA will be made by a centralized entity referred to as as spectrum access system (SAS); however, maintaining spectrum utilization information in SAS presents privacy risks, as sensitive Incumbent User (IU) operation parameters are required to be stored by SAS in order to perform spectrum assignments properly. These sensitive operation parameters may potentially be compromised if SAS is the target of a cyber attack or an inference attack executed by a secondary user (SU). In this thesis, we explore the operational security of IUs in SAS-based DSA systems and propose a novel privacy-preserving SAS-based DSA framework, Suspicion Zone SAS (SZ-SAS), the first such framework which protects against both the scenario of inference attacks in an area with sparsely distributed IUs and the scenario of untrusted or compromised SAS. We then define modifications to the SU inference attack algorithm, which demonstrate the necessity of applying obfuscation to SU query responses. Finally, we evaluate obfuscation schemes which are compatible with SZ-SAS, verifying the effectiveness of such schemes in preventing an SU inference attack. Our results show SZ-SAS is capable of utilizing compatible obfuscation schemes to prevent the SU inference attack, while operating using only homomorphically encrypted IU operation parameters.

Incorporating Obfuscation Techniques in Privacy Preserving Database-Driven Dynamic Spectrum Access Systems

Douglas M. Zabransky

(GENERAL AUDIENCE ABSTRACT)

Dynamic Spectrum Access (DSA) allows users to opportunistically access spectrum resources which were previously reserved for use by specified parties. This spectrum sharing protocol has been identified as a potential solution to the issue of spectrum crowding. This sharing will be accomplished through the use of a centralized server, known as a spectrum access system (SAS). However, current SAS-based DSA proposals require users to submit information such as location and transmission properties to SAS. The privacy of these users is of the utmost importance, as many existing users in these spectrum bands are military radars and other users for which operational security is pivotal. Storing the information for these users in a central database can be an major privacy issue, as this information could be leaked if SAS is compromised by a malicious party. Additionally, malicious secondary users (SUs) may perform an inference attack, which could also reveal the location of these military radars. In this thesis, we demonstrate a SAS-framework, SZ-SAS, which allows SAS to function without direct knowledge of user information. We also propose techniques for mitigating the inference attack which are compatible with SZ-SAS

Dedication

This thesis is dedicated to my family, friends, and all those who have supported me throughout my life and academic career. I would not be where I am today without the guidance and love of my parents.

Acknowledgments

I would first like to thank my advisor, Dr. Yaling Yang, without whom it would have been impossible to complete this work. Her advice, guidance, and support have been instrumental in my success throughout my time as a graduate student. She has been a voice of support in my work while still challenging my ideas and holding me to high academic standards.

I would also like to thank my committee members, Dr. Tront and Dr. Gerdes. Your input has been invaluable, and the lessons I have learned whether in your classes, during our meetings, or while joining you at conferences will remain with me.

Finally, I would like to acknowledge my lab mates and friends, He Li, Chang Lu, Curtis Zeng, Dong Wang, Michael Littley, and Takondwa Kakusa. Your companionship and input during my time at Virginia Tech has helped me grow and kept me grounded.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The trend in commercial innovation has been towards inter-device communication. Developments in fields such as Internet of Things (IoT), embedded sensors, wearable devices, and vehicular communication systems are introducing novel solutions to modern problems which often require inter-device wireless communication. These innovations and the trends towards increased mobile and wireless Internet traffic [1] are resulting in concerns that the wireless spectrum will continue to become increasingly overcrowded and open spectrum bands for novel technologies will increase in scarcity. These concerns have led to an increase in research into alternative spectrum allocation methods to remedy this issue. One such proposed solution is Dynamic Spectrum Access (DSA), which has been a main focus of research in this field and shows promising preliminary results. However, security is an issue in DSA, particularly in the first generation of DSA systems, which consists of database-driven spectrum access system (SAS).

Several SAS-based DSA systems have been proposed, few of which have addressed the concern of incumbent user (IU) privacy. This lack of privacy is a major issue, as the bands which have been designated for DSA have IUs which consist primarily of government and military users, whose privacy is of the utmost importance. Two primary attack vectors have been identified for SAS-based DSA, direct database access through cyber attacks and indirect access through secondary user (SU) inference attacks.

In this chapter, we first introduce and justify DSA and subsequently define several security

and privacy issues currently inherent to the proposed DSA systems. We then introduce our contributions towards addressing these issues and finally we present the organization of the remainder of this thesis.

1.1 Dynamic Spectrum Access

Historically, spectrum allocation in the United States has been handled with static assignment by the Federal Communications Commission (FCC). These spectrum assignments are made via competitive auctions, in which access licenses for frequency bands are sold to the highest bidder. These licenses cover large geographic areas and grant exclusive access to the specified frequency bands for a predetermined amount of time, ranging from years to decades [2]. This inflexibility of static spectrum assignment is an unnecessary hurdle to modern innovation, as it is difficult for new users and technologies to access the limited spectrum resources while outdated legacy technologies can potentially hold exclusive licenses to spectrum. This is especially evident as the rate of growth of mobile and wireless Internet traffic continues to climb due to the growing adoption rate of IoT, multimedia streaming, and other wireless technologies. Figure 1.1 demonstrates the issue of overcrowding in the 3-5 GHz bands.

The inflexible nature of static spectrum assignment also often leads to underutilized dedicated frequency bands. For instance, the National Telecommunications and Information Administration (NTIA) performed a study on the occupancy of the 3.55-3.65 GHz band near San Diego Naval Base in California in 2012. This band has historically been reserved for military radiolocation and aeronautical radionavigation. The location was chosen because the naval base serves as a port for ships with highly sensitive radars which operate in the 3.55-3.65 GHz band, thus the results of the study will show the potential for spectrum

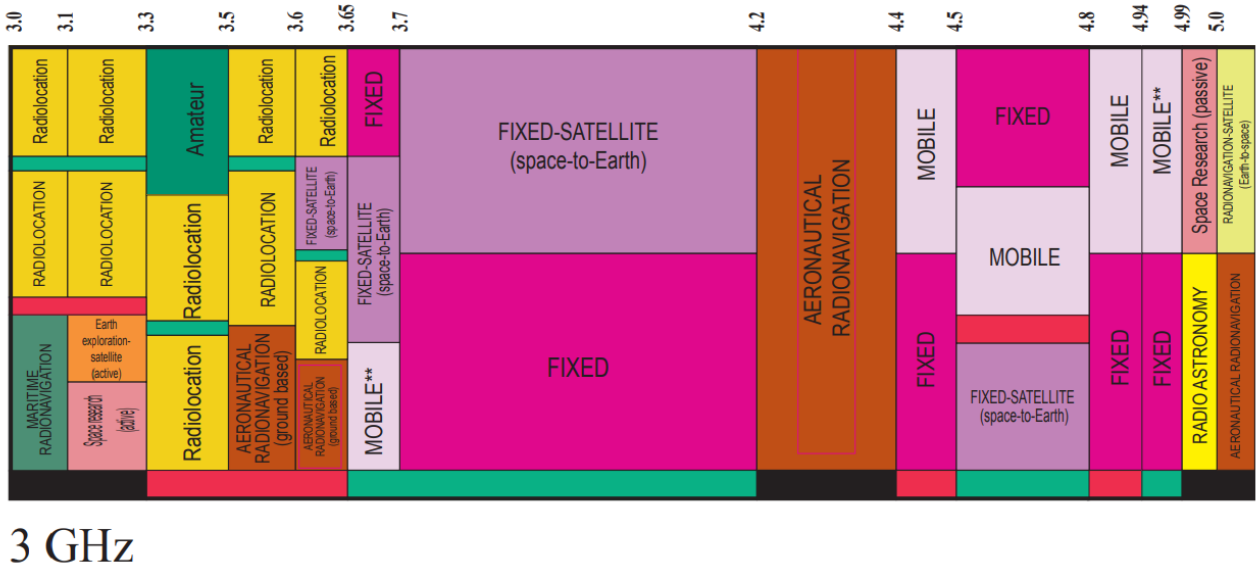


Figure 1.1: Static Spectrum Allocations in 3-5 GHz Bands [4]

sharing in a case in which the band is abnormally crowded when compared to the majority of areas in the United States which lack large naval shipyards. The report found the average band occupancy to range from only 7.5% to 36.6% and concluded there exists a 40.0% to 59.8% chance the band was completely unutilized [12].

DSA has been proposed to solve the issues inherent in static assignment and functions by dividing users into two classes, Incumbent Users (IUs), who are the legacy users and who should maintain continuous interference-free access to their licensed spectrum, and Secondary Users (SUs), who are previously unlicensed users attempting to opportunistically access underutilized spectrum. The main goal of DSA is to increase spectrum utilization, which can be accomplished by allowing SUs to access spectrum as long as their transmissions will not interfere with the operations of IUs. The NTIA study of the 3.55-3.65 GHz band shows the great potential of opening underutilized bands to spectrum sharing, but also highlights the need for interference limitations and security to be central considerations during the design of DSA systems, due to the sensitive nature of IU communications and IU location

information [12].

Interference regulation in DSA systems can be accomplished with either the spectrum sensing model or the database-driven SAS-based model. The spectrum sensing model is built around cognitive radio technology which allows SUs to detect the presence of IUs or the availability of unused channels. The SU can then modify its operating parameters based upon the informationg gained during the sensing stage. A promising advancement, collaborative spectrum sensing, uses the sensing results from multiple SUs at different positions in order to improve the probability of IU detection, decreasing the likelihood that SU transmissions will create interference [19].

Alternatively, the database-driven SAS-based DSA model is characterized by SUs receiving spectrum availability information by querying a central server. This server, known as a Spectrum Access System (SAS), contains a geolocation database with updated spectrum availability information for the managed spectrum channels. The server is able to determine whether an SU will interfere with IU activity because it maintains a database of IU operation information, including location, operating frequency, and interference threshold. It uses these operating parameters, a radio propagation model, and a spectrum access enforcement policy in order to determine whether an SU can operate in a given location. The spectrum access enforcement policies utilize either exclusion zones or protection zones. Exclusion zones are areas around an IU in which no SUs are permitted to operate and are determined using the previously mentioned propagation model and operating parameters. Protection zones are similar to exclusion zones, but allow users to transmit within the area as long as the total interference from all SUs does not exceed an interference threshold value established by the IU [7].

Recent proposals produced by the FCC and the President's Council of Advisors on Science and Technology (PCAST) have eliminated spectrum sensing as a requirement for cognitive

radio devices and recommended the database-driven DSA model. The FCC rules and regulations also specify that current DSA systems should use exclusion zones until a distributed system of sensors known as an environmental sensing capability (ESC) has been approved and deployed, in which case the exclusion zones can be replaced with protection zones [3]. For this reason, we focus on database-driven DSA systems employing exclusion zone enforcement in the remainder of this thesis.

1.2 Security and Privacy Concerns in Database-Driven Spectrum Sharing

As discussed in the previous section, IUs often consist of military and federal government users and the operational security (OPSEC) of such users is a primary concern in the deployment of DSA systems. The authors of [23] identify several operational attributes of IU systems that should not be revealed if the IU is operated by the military or federal government, including:

- Transmitter identity
- Geolocation
- Antenna parameters
- Transmit power
- Transmit protection contours
- Times of operation

Several of these parameters are clearly essential to the normal operation of SAS. Most proposed SAS frameworks require IUs to submit geolocation, transmit protection contours, antenna parameters, and times of operation as these parameters are needed in order to determine whether a querying SU will interfere with IU operation. These proposed frameworks typically assume SAS to be a trusted database and thus allow SAS to operate with these parameters in plaintext. However, the FCC has approved several private companies to develop SAS systems, which would result in the creation of plaintext databases of sensitive federal and military information stored on the servers of private companies. In an era in which data breaches are common, these newly created databases serve to create a novel attack vector which adversaries may utilize to efficiently gain information on all IUs associated with SAS by compromising this singular access point. The companies responsible for operating SAS may also be tempted to maintain a record of IU parameters for their own future research purposes. Therefore, SAS clearly cannot be assumed to be trustworthy, and thus new frameworks must be developed in order to maintain the OPSEC of IUs.

In addition to defending against attacks which attempt to directly access the database itself, SAS must also have some manner of defense from inference attacks. Inference attacks are a common consideration for any security-aware database-driven service, and are characterized by a user making seemingly innocuous queries to the database and using the trivial information gained from the results of these queries to infer more sensitive information which is not available directly from queries to the database. Inference attacks are an issue in all manner of database-driven services, including medical databases and location-based services [27, 31]. In SAS-based DSA systems, adversarial SUs may correlate the results of spectrum availability queries in order to infer the geolocation or transmit protection contour information of an IU by simply observing which areas are and are not available for transmission, and using this information to infer the exclusion zone boundaries or the geolocation of IUs.

There are a number of previous works which attempt to address privacy of IUs against either an untrusted SAS via operation on homomorphically encrypted parameters [13, 14, 20] or untrusted SUs performing an inference attack [9, 10, 30]. However, no previous work considers a scenario consisting of an untrusted SAS and the SU inference attack. Previously proposed inference attack countermeasures are inherently incompatible with operations on homomorphically encrypted parameters as they require SAS to have full knowledge of IU and SU operations. We address this gap in the literature with our contributions described in the remainder of this thesis.

1.3 Research Contributions

The contributions of this thesis are as follows:

- We propose a novel database-driven DSA framework, Suspicion Zone SAS (SZ-SAS), the first such framework which allows for obfuscation to be applied on a per-user or per-group basis based upon the query history of an individual user and the first such framework which protects against both the scenario of inference attacks in an area with sparsely distributed IUs and the scenario of untrusted SAS.
- We introduce a modified inference attack, showing a lower bound of privacy provided by non-obfuscated SAS responses than previously suspected.
- We provide and analyze multiple obfuscation techniques which are compatible with the proposed DSA framework.

Chapter 2

SZ-SAS: Privacy-Preserving SAS with Dynamic Obfuscation

In this chapter, we introduce a novel DSA framework, SZ-SAS, which maintains the operational security of IUs in the case of both untrusted SAS and SU inference attack execution. In Section 2.1, we discuss related work regarding privacy-preserving SAS designs. In Section 2.2, we discuss the cryptographic background upon which SZ-SAS has been built. And in Section 2.3, we discuss the system design and operation specifics of SZ-SAS.

2.1 Related Work

Research efforts on preserving the OPSEC of IUs typically fall into two categories: encryption and obfuscation. The authors of [13, 14, 20] present various frameworks based upon homomorphic proxy re-encryption and multi-party computation which allow SAS to perform calculations on encrypted sensitive IU parameters and enforce the spectrum access policy without direct knowledge of these parameters. These encryption-based schemes protect IUs from revealing their operating parameters directly to SAS, a feature which could prove to be important as the FCC allows SAS to be operated by private companies in order to increase scalability and efficient deployment of SAS [24].

Obfuscation techniques have been introduced as a solution to protect the geolocation infor-

mation of IUs against inference attacks. These obfuscation techniques include k-anonymity, k-clustering, random false positives, and perturbation with additive noise. Each of these techniques attempts to prevent inference attacks by replacing an actual response from SAS to an SU query with a false positive response, representing the existence of an IU in an area in which no IU actually exists. However, these techniques have an introduce an inherent trade-off between location privacy and spectrum utilization.

2.2 Cryptosystem Preliminaries

SZ-SAS utilizes the AFGH cryptosystem [8], which is a single-hop, unidirectional, non-transitive, collusion resistant, homomorphic proxy re-encryption scheme based upon bilinear maps. In this section, we discuss the mathematical basis of this cryptosystem, its construction and operation, the cryptographic assumptions upon which it was designed, and the relevant properties which are leveraged by SZ-SAS.

2.2.1 Mathematical Basis: Bilinear Maps

A bilinear map is a function, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ if the following conditions are satisfied:

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3$ are cyclic groups of prime order q.
- For all $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_q$, $e(g^a, h^b) = e(g, h)^{ab}$
- The map is non-degenerate, $e(g_1, g_2) \neq 1$

The AFGH cryptosystem utilizes mappings in which $\mathbb{G}_1$ and $\mathbb{G}_2$ are the same group, and thus $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$

2.2.2 The AFGH Homomorphic Proxy Re-encryption Scheme

SZ-SAS requires a cryptosystem in which SAS can perform operations securely on encrypted parameters and re-encrypt the result such that it can be decrypted by the querying SU. This functionality requirement necessitates the use of a scheme capable of performing unidirectional proxy re-encryption and either fully or partially homomorphic encryption. Partially homomorphic cryptosystems, such as AFGH, only allow operations in the ciphertext space using either addition or multiplication, but have significantly less overhead than fully homomorphic schemes which allow both operations to be performed. Because SZ-SAS will potentially be processing thousands of queries per minute, the computation and communication overhead per query will need to be minimal, thus relatively light-weight partially homomorphic AFGH scheme was chosen.

AFGH as proposed in [8] is homomorphic with respect to multiplication. However, our scheme requires an accumulator and so we must modify AFGH to become homomorphic with respect to addition. In order to accomplish this, we simply exponentiate the generator, Z of $\mathbb{G}_T$, by our plaintext prior to encryption as Z^{PT}. This will allow us to perform addition, but the discrete log problem (DLP) would need to be solved in order to recover the actual plaintext rather than Z^{PT}. Thus, we have built our system to operate without requiring the plaintext to be recovered from Z^{PT} in order to avoid the computations required by computing the DLP. We are also able to perform multiplication of a ciphertext by a plaintext value using exponentiation. We define the encrypted addition operation and the plaintext multiplication created by our modification as $\oplus$ and $\otimes$ for the remainder of this paper and describe the construction and operation of the additive AFGH cryptosystem below:

- **System Parameters**: $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a Type 1 bilinear map, g is a random generator of $\mathbb{G}$, and $Z = e(g, g)$ is a random generator of $\mathbb{G}_T$.

- **Key Generation**: Set a group secret key $SK_a \leftarrow_{\$} \mathbb{Z}_p^*$ and public key $PK_a = g^{SK_a}$ for all IUs. SU key generation will occur during the SU's registration with SAS and will generate individual secret keys, $SK_b \leftarrow_{\$} \mathbb{Z}_p^*$, and public keys, $PK_b = g^{SK_b}$, which will be unique to each SU.

- **Re-encryption Key Generation**: To re-encrypt a level 2 ciphertext which was originally encrypted with IUs' public key into a level 1 ciphertext which can be decrypted by SU b's secret key, a re-encryption key must be generated. This key is generated with SU b's public key, PK_b, and IUs' private key, SK_a, as $RK_{a\to b} = PK_b^{1/SK_a}$. This is equivalent to g^{SK_b/SK_a}.

- **Level 1 Encryption** $Enc_1(m, PK)$: To encrypt a message $m \in \mathbb{G}_T$ using public key PK_a, such that it can only be decrypted by SK_a, the ciphertext tuple C $= (Z^m Z^r, Z^{SK_a r}) = (Z^{m+r}, Z^{SK_a r})$ is computed. In order to compute $Z^{SK_a r}$, we utilize the bilinear map to compute $Z^{SK_a} = e(g, PK_a)$. Note that $r \leftarrow_{\$} \mathbb{Z}_p^*$.

- **Level 2 Encryption** $Enc_2(m, PK)$: To encrypt a message $m \in \mathbb{G}_T$ using public key PK_a, such that it can only be decrypted by SK_a or re-encrypted with $RK_{a\to b}$, the ciphertext tuple C $= (Z^m Z^r, PK_a^r) = (Z^{m+r}, PK_a^r)$ is computed. Note that $r \leftarrow_{\$} \mathbb{Z}_p^*$.

- **Level 1 Decryption** $Dec_1(C, SK)$: To decrypt a level 1 ciphertext tuple, C $= (\alpha, \beta)$, which has been encrypted with public key PK_a, compute $m' = Z^m = \alpha/\beta^{1/SK_a}$.

- **Level 2 Decryption** $Dec_2(C, SK)$: To decrypt a level 2 ciphertext tuple, C $= (\alpha, \beta)$, which has been encrypted with public key PK_a, compute $m' = Z^m = \alpha/e(\beta, g^{1/SK_a})$.

- **Re-encryption**: To re-encrypt a level 2 ciphertext tuple C $= (c_1, c_2)$, which has been encrypted with public key PK_a, into a level 1 ciphertext tuple which can be decrypted with secret key SK_b, use the re-encryption key $RK_{a\to b}$ to compute $c_2' =$

$e(c_2, RK_{a\to b}) = Z^{rb}$. The resulting re-encrypted level 1 ciphertext tuple is then C = (c_1, c_2').

As mentioned previously, our modified AFGH cryptosystem is capable of performing homomorphic addition of two ciphertexts and multiplication of a single ciphertext with a plaintext via homomorphic exponentiation. We describe these two operations below:

- **Homomorphic Addition *add(C,C')*:** given two input ciphertexts, $C = (c_1, c_2) = (Z^{m+r}, c_2)$ and $C' = (c_1', c_2') = (Z^{m'+r'}, c_2')$, which have been encrypted with the same public key and the same encryption level, the output of *add(C,C')* is $C_a = (c_1c_1', c_2c_2') = (Z^{m+m'+r+r'}, c_2c_2')$. When C_a is decrypted, the output will be $Z^{m+m'}$.

- **Multiplication by a Plaintext *mul(C,*α*)*:** given one input ciphertext, $C = (c_1, c_2) = (Z^{m+r}, c_2)$ and, one input plaintext, α, the output of *mul(C,*α*)* is $C_a = (c_1^\alpha, c_2^\alpha) = (Z^{m\alpha+r\alpha}, c_2c_2')$. When C_m is decrypted, the output will be $Z^{m\alpha}$.

We verify the correctness of these operations with the following propositions.

- **Proposition 1.** *add(C,C')* is a correct homomorphic addition algorithm given:

$$(sk, pk) \leftarrow KeyGen$$

$$M, M' \in \mathbb{G}_T$$

$$C_1 \leftarrow Enc_1(M, pk), C_2 \leftarrow Enc_2(M, pk)$$

$$C_1' \leftarrow Enc_1(M', pk), C_2' \leftarrow Enc_2(M', pk)$$

$$C_{add1} \leftarrow add(C_1, C_1'), C_{add2} \leftarrow add(C_2, C_2')$$

We can then perform $Dec_1(C_{add1}, sk) = Dec_2(C_{add2}, sk) = Z^{M+M'}$, for which the discrete log problem may be solved to recover $M + M'$.

Proof. Let $sk = a_1$ and $pk = g^{a_1}$. Based upon the details of additive AFGH, let $C_1 = (Z^{M+r_1}, Z^{r_1 a_1})$, $C_1' = (Z^{M'+r_2}, Z^{r_2 a_1})$, $C_2 = (Z^{M+r_3}, g^{r_3 a_1})$, $C_2' = (Z^{M'+r_4}, g^{r_4 a_1})$

Utilizing homomorphic addition, we have:

$$\begin{aligned} C_{add1} &= (Z^{M+r_1} \cdot Z^{M'+r_2}, Z^{r_1 a_1} \cdot Z^{r_2 a_1}) \\ &= (Z^{M+r_1+M'+r_2}, Z^{a_1(r_1+r_2)}) \\ C_{add2} &= (Z^{M+r_3} \cdot Z^{M'+r_4}, g^{r_3 a_1} \cdot g^{r_4 a_1}) \\ &= (Z^{M+r_3+M'+r_4}, g^{a_1(r_3+r_4)}) \end{aligned}$$

Therefore, when decrypting these results allows us to compute:

$$\begin{aligned} Dec_1(C_{add1}, a_1) &= (\frac{Z^{M+r_1+M'+r_2}}{Z^{a_1(r_1+r_2)\frac{1}{a_1}}}) = Z^{M+M'} \\ Dec_2(C_{add2}, a_1) &= (\frac{Z^{M+r_3+M'+r_4}}{e(g^{a_1(r_3+r_4)}, g^{\frac{1}{a_1}})}) = Z^{M+M'} \end{aligned}$$

□

- **Proposition 2.** *mul(C,α)* is a correct homomorphic multiplication algorithm given:

$$(sk, pk) \leftarrow KeyGen$$

$$M \in \mathbb{G}_T, \alpha \in \mathbb{Z}_p^*$$

$$C_1 \leftarrow Enc_1(M, pk), C_2 \leftarrow Enc_2(M, pk)$$

$$C_{m1} \leftarrow mul(C_1, \alpha), C_{m2} \leftarrow mul(C_2, \alpha)$$

We can then perform $Dec_1(C_{m1}, sk) = Dec_2(C_{m2}, sk) = Z^{M\alpha}$, for which the discrete log problem may be solved to recover $M\alpha$.

Proof. Let $sk = a_1$ and $pk = g^{a_1}$. Based upon the details of additive AFGH, let $C_1 = (Z^{M+r_1}, Z^{r_1 a_1})$, $C_2 = (Z^{M+r_2}, g^{r_2 a_1})$

Utilizing homomorphic multiplication, we have:

$$\begin{aligned} C_{m1} &= ((Z^{M+r_1})^\alpha, (Z^{r_1 a_1})^\alpha) \\ &= (Z^{\alpha M + \alpha r_1}, Z^{\alpha r_1 a_1}) \\ C_{m2} &= ((Z^{M+r_2})^\alpha, (g^{r_2 a_1})^\alpha) \\ &= (Z^{\alpha M + \alpha r_2}, g^{\alpha r_2 a_1}) \end{aligned}$$

Therefore, when decrypting these results allows us to compute:

$$Dec_1(C_{m1}, a_1) = (\frac{Z^{\alpha M + \alpha r_1}}{Z^{(\alpha r_1 a_1)\frac{1}{a_1}}}) = Z^{M\alpha}$$
$$Dec_2(C_{m2}, a_1) = (\frac{Z^{\alpha M + \alpha r_2}}{e(g^{\alpha r_2 a_1}, g^{\frac{1}{a_1}})}) = Z^{M\alpha}$$

□

Neither Proposition 1 nor Proposition 2 violate the security of the AFGH cryptosystem, as neither proposition impacts Theorem 3.1 in [8].

For simplicity, for the remainder of this thesis, we represent level 1 ciphertext tuples, $(Z^{m+r}, Z^{PK_a r})$, as $[\![m]\!]_1$ and level 2 ciphertext tuples, (Z^{m+r}, PK_a^r), as $[\![m]\!]_2$ for message m.

In addition to the homomorphic operations available in AFGH, this scheme has many attractive properties including unidirectionality, non-transitivity, collusion resistance, and single-hop proxy re-encryption. We define these properties below.

- **Unidirectionality:** This property allows for the generation of a re-encryption key, $RK_{a\to b}$, which only allows re-encryption to occur from a specified source user, a, to a specified destination user, b. It is not able to re-encrypt from b to a. Further, an adversary cannot compute $RK_{b\to a} = g^{SK_a/SK_b}$ simply by knowing $RK_{a\to b} = g^{SK_b/SK_a}$

- **Non-transitivity:** This property specifies that the proxy is unable to utilize valid re-encryption keys to generate new re-encryption keys for previously unauthorized users. Re-encryption keys may only be generated using a source secret key and a destination public key.

- **Collusion Resistance:** This property prevents an adversarial proxy from colluding with one user to determine the secret key of another user. Thus, SAS cannot collude with SUs to determine the IU secret key nor can SAS collude with IUs to determine SU secret keys.

- **Single-hop Proxy Re-encryption:** A single-hop proxy re-encryption scheme allows for a given ciphertext to be re-encrypted only once. In AFGH, a level 2 ciphertext is transformed to a level 1 ciphertext when it is re-encrypted, and level 1 ciphertext cannot be re-encrypted. Thus, AFGH is single-hop.

2.3 System Design

From a high-level view, SZ-SAS leverages homomorphic proxy re-encryption to encrypt operation parameters from IUs such that SAS has no direct knowledge of these parameters or the results of SU queries. Utilizing the homomorphic nature of the chosen cryptosystem, SZ-SAS uses a novel method for maintaining and utilizing an encrypted count of potentially suspicious queries made by each SU. Then, it restricts SUs which have exceeded a given IU's specified threshold of suspicious queries from querying the IU's actual E-Zone. Instead, it calculates their query results from obfuscated E-Zones, which hide the true geolocation of the IU in question.

In this section, we discuss the specifics of the SZ-SAS design, including the basic system model and descriptions of operations performed by each component of the system.

2.3.1 System Model

SZ-SAS is comprised of a 4-party SAS structure consisting of key manager, IUs, SAS, and SUs. The key manager handles generation of re-encryption keys for newly-registered SUs. The system is initialized by first properly initializing the AFGH scheme by generating the AFGH system parameters at which point the key manager generates the shared IU key pair and publishes the shared IU public key and AFGH system parameters.

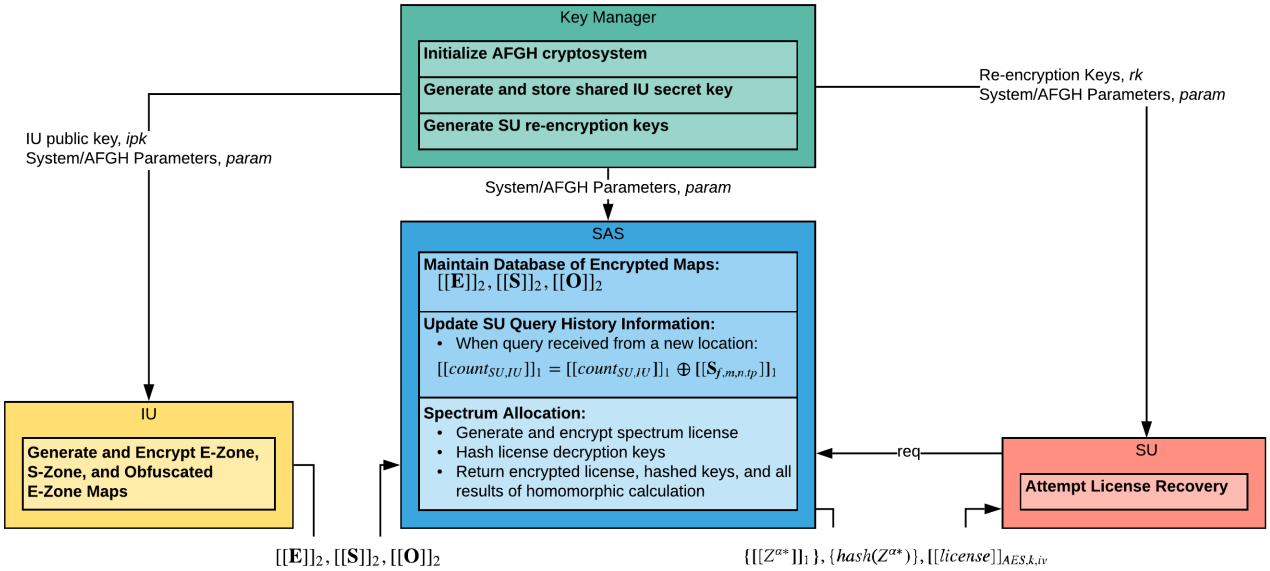


Figure 2.1: SZ-SAS System Framework

Key Manager Operations

The key manager generates re-encryption keys for each SU during the SU's initial registration with SAS. The key manager should either be controlled by a government entity or trusted-third party as re-encryption key generation requires knowledge of the shared IU secret key.

IU Operations

Each IU is responsible for the generation of its E-Zone, Suspicion Zone (S-Zone), and obfuscated E-Zone maps. Each of these maps is represented by an $M \times N$ matrix and contains information for the entire area covered by SAS. The construction of each of these maps is defined below.

- **Exclusion Zone Determination:** The IUs are responsible for calculation of the maximum allowable transmit power at which a co-channel SU can transmit without causing harmful interference from a given location. This is done using a Transmit Power Allocation (TPA) function, which depends on criteria such as the antenna properties of the IU and SU, the propagation path loss, P_L, between the IU and SU, the interference threshold, I_{th}, of the IU, and the terrain between the IU and SU. A simplified version of the TPA function can be constructed for our purposes, depending solely upon I_{th} and the value of P_L in a given cell, where P_L is calculated using a path loss model specified by the IU. We can then represent the TPA function as $T(P_L, I_{th})$ for simplicity.

  Using $T(P_L, I_{th})$ and various predetermined maximum transmit power level bins, the IU is able to create a series of exclusion zone maps. By evaluating $T(P_L, I_{th})$ for all grid locations covered by SAS at a fixed transmit power level, the IU can obtain an exclusion zone map for this transmit power level. Any cell in the M x N grid covered by SAS which has a path loss value less than the required path loss value of the IU is classified as an exclusion zone cell. Any SU transmissions in an exclusion zone cell at or above the specified power level will cause harmful interference to the IU and are thus prohibited. Any non-exclusion zone cell must satisfy the following inequality:

$$TP_{SU} - P_{L_{SU}} \leq I_{th} \tag{2.1}$$

Where TP_{SU} is the requested transmit power level, $P_{L_{SU}}$ is the result of the IU's selected propagation path loss model based upon the SU's location and transmit power level, and I_{th} is the interference threshold of the IU.

The transmit power level bins are used to categorize different SUs based upon their requested transmit power levels. When a request is received by SAS, SAS accesses the exclusion zone maps for each IU corresponding to the requested transmit power level and returns a response based upon these maps. In order for a valid license to be recovered by an SU for a specified transmit power level bin, the SU must be located in a non-exclusion zone cell for the specified combination of transmit power and location.

The series of E-Zone maps, each represented by a matrix $\mathbf{E}$, consist of values in Z_p. If channel f at grid location (m, n) and transmit power level (tp) is considered to be E-Zone by the IU, then the value of the E-Zone map $\mathbf{E}_{f,m,n,tp}$ is a random non-zero element picked from Z_p. Formally, we have

$$\mathbf{E}_{f,m,n,tp} \leftarrow^{\$} Z_p \setminus \{0\} \tag{2.2}$$

For all other elements, $\mathbf{E}_{f,m,n,tp}$ is set to 0. Formally,

$$\mathbf{E}_{f,m,n,tp} \leftarrow 0 \tag{2.3}$$

- **Suspicion Zone Determination:** Because the areas bordering the protection contour provide the most information to a SU performing an inference attack, the IU defines the cells adjacent to the protection contour as S-Zone cells and generates a series of maps, represented by matrix $\mathbf{S}$, corresponding to the contour of each E-Zone map. In our tests, we define the S-Zone cells as all E-Zone and all non E-Zone cells which border the protection contour for an IU. S-Zone cells can be adjusted by IUs, expanding or

reducing the size of the suspicion zone. If channel f at grid location m, n and and transmit power level (TP) is considered to be S-Zone by the IU, then the value of the S-Zone map $\mathbf{S}_{f,m,n,tp}$ is set to 1. Formally,

$$\mathbf{S}_{f,m,n,tp} \leftarrow 1 \tag{2.4}$$

For all other elements, $\mathbf{S}_{f,m,n,tp}$ is set to 0. Formally,

$$\mathbf{S}_{f,m,n,tp} \leftarrow 0 \tag{2.5}$$

- **Obfuscated E-Zone Determination:** Obfuscated E-Zone maps essentially are distorted and enlarged E-Zones and are generated using an obfuscation scheme as described in Section 3.4. These maps, represented by matrix $\mathbf{O}$, follow the same value assignment rules as E-Zone maps. The IU also determines a suspicious request threshold, represented by τ, which is the number of queries an SU is allowed to make from S-Zone areas. If τ is exceeded, the SU's queries to the E-Zone map are invalidated and queries must be fulfilled based upon the obfuscated E-Zone map. The values of the obfuscated E-Zone map are set based upon the rules described for standard E-Zone maps and shown in expressions 2.2 and 2.3.

- **Commitment of IU Parameters to SAS:** Each IU prepares these three categories of maps, encrypts each map value with the IU public key, and commits the resulting encrypted maps and the unencrypted threshold value to SAS. As discussed earlier, the encryption of these maps is essential as SAS can potentially be compromised and these maps may be used to derive sensitive parameters, such as the geolocation and operation times of the IU.

SAS Operations

SAS is responsible for performing three main operations: maintaining the database of all encrypted maps, $[\![\mathbf{E}]\!]_2$, $[\![\mathbf{S}]\!]_2$, $[\![\mathbf{O}]\!]_2$, updating SU query information, and performing spectrum assignment computations in response to SU requests.

- **Encrypted Map Database Maintenence:** When a new IU pairs with SAS, the IU commits $[\![\mathbf{E}]\!]_2$, $[\![\mathbf{S}]\!]_2$, $[\![\mathbf{O}]\!]_2$ and an unencrypted threshold value to SAS. If the IU's operation parameters change, or if the IU goes offline, it may wish to modify the values maintained by SAS, which would allow SUs more tranmission opportunities in the case of an IU going offline. The IU may then submit these new values, which will overwrite the previously stored parameters for this IU in SAS.

- **Updating SU Query Information**: SAS updates the encrypted number of suspicious queries made by each SU for each IU, represented by $[\![count_{SU,IU}]\!]_1$, whenever a query is initially received from an SU which is querying from a new location. When a query is received from location (m, n) for channel f at transmit power level (tp), SAS first determines if this is the querying SU's first query from this location. If it is, SAS re-encrypts $[\![\mathbf{S}_{f,m,n,tp}]\!]_2$ to $[\![\mathbf{S}_{f,m,n,tp}]\!]_1$ using the SU's re-encryption key and updates $[\![count_{SU,IU}]\!]_1 = [\![count_{SU,IU}]\!]_1 \oplus [\![\mathbf{S}_{f,m,n,tp}]\!]_1$. Otherwise, SAS does not update $[\![count_{SU,IU}]\!]_1$. This allows SAS to maintain an encrypted count of queries originating from grid locations which have been designated as potentially suspicious due to the potential information revealed in a SAS response to an SU performing an inference attack from this location.

  This operation is also essential in its ability to allow non-adversarial SUs to access the available spectrum without any applied obfuscation, which is critical because all obfuscation techniques will introduce some level of spectrum utility degradation. Ide-

ally, the obfuscation will only be applied to adversarial SUs without impacting the operation of trustworthy SUs; however, it is difficult or even impossible to determine from a single query whether or not an SU is trustworthy or adversarial. Thus, we use a query history to track the number of suspicious queries originating from each SU. Honest SUs which are not querying from multiple grid locations near the boundaries of an IU will not have their query history incremented. These SUs will then be able to derive their spectrum licenses based upon the unobfuscated E-Zone maps as described in the next several paragraphs.

- **Spectrum Assignmet** The spectrum assignment computation is the primary functionality of SAS. SAS must generate a spectrum license consisting of the SU's access information (expiration time, transmit power level, location, etc.) and a digital signature over this access information. This license will be transmitted to the SU in two possible situations. In the first situation, the SU has not exceeded the threshold of suspicious requests (i.e. $count_{SU,IU} < \tau$) and is not located in the E-Zone of any IU. In the second situation, the threshold has been exceeded and, thus, the SU must not be in the obfuscated E-Zone.

  Because SAS does not have direct knowledge of $\mathbf{E}$, $\mathbf{S}$, $\mathbf{O}$, it cannot directly determine whether the SU is located within an IU's E-Zone or obfuscated E-Zone. Therefore, the assignment is performed via a special process which leverages the properties of AFGH. First, SAS updates $[\![count_{SU,IU}]\!]_1 = [\![count_{SU,IU}]\!]_1 \oplus [\![\mathbf{S}_{f,m,n,tp}]\!]_1$ as described previously. Next, SAS generates the spectrum license, which must be encrypted in a manner such that it may only be decrypted if the spectrum request is valid, as in the two situations defined in the previous paragraph. In order to accomplish this, SAS utilizes a cascade encryption scheme and a specialized homomorphic calculation as described in the remainder of this section.

First, SAS generates a separate symmetric encryption key for each IU by selecting a random element α in $\mathbb{G}_T$, and uses this element to exponentiate the generator, Z, resulting in Z^α. Next, SAS hashes Z^α with a cryptographic hash function and the resulting hash is split into two bit strings, k and iv. The hash function used in this step will be referred to as the primary hashing function for the remainder of this thesis. Each IU's k and iv bit strings are then used as the secret key and initialization vector for a block cipher operating in a stream-like mode, such as CTR mode AES, and the license is sequentially encrypted by each k and iv pair via cascade encryption, resulting in $[\![license]\!]_{AES,k,iv}$. Each IU's Z^α and a series of threshold values from 0 to τ, denoted as ϵ_i for $i \in [0, \tau]$, are then encrypted to level 1 ciphertexts, $[\![Z^\alpha_{IU}]\!]_1$ and $[\![\epsilon_{i,IU}]\!]_1$, with the querying SU's AFGH public key. Next, $[\![\mathbf{E}_{f,m,n,tp}]\!]_2$, $[\![\mathbf{S}_{f,m,n,tp}]\!]_2$, and $[\![\mathbf{O}_{f,m,n,tp}]\!]_2$ are re-encrypted to level 1 ciphertexts with the SU's re-encryption key. SAS then performs the following homomorphic calculation for each IU and corresponding Z^α_{IU}:

$$\forall i \in [0, \tau] : [\![Z^{\alpha*}_{i,IU}]\!]_1 \leftarrow [([\![count_{SU,IU}]\!]_1 \oplus [\![\epsilon_i]\!]_1^{-1}) \otimes R] \oplus [\![\mathbf{E}_{f,m,n,tp}]\!]_1 \oplus [\![Z^\alpha_{IU}]\!]_1 \quad (2.6)$$

in which R is a random, large, negative nonce. The first portion of this calculation, $([\![count_{SU,IU}]\!]_1 \oplus [\![\epsilon_i]\!]_1^{-1}) \otimes R$, ensures that if the current value of $[\![count_{SU,IU}]\!]_1$ is greater than τ, every resulting decrypted $Z^{\alpha*}_{i,IU}$ will be equal to Z^α_{IU} distorted by a multiple of the random nonce, R. Additionally, if the query originated from a cell inside of the E-Zone, $[\![\mathbf{E}_{f,m,n,tp}]\!]_1 \oplus [\![Z^\alpha_{IU}]\!]_1$ will distort the result by the random value in G_T to which $\mathbf{E}_{f,m,n,tp}$ was initialized. If the current value of $[\![count_{SU,IU}]\!]_1$ is less than τ and the query originated from a cell outside of the E-Zone, there will exist one resulting $Z^{\alpha*}_{i,IU}$ that is equal to Z^α_{IU} for some i. SAS also produces one $[\![Z^{\alpha*}_{i,IU}]\!]$ for each IU based upon the obfuscated E-Zone map by simply calculating $[\![Z^{\alpha*}_{i,IU}]\!]_1 \leftarrow [\![\mathbf{O}_{f,m,n,tp}]\!]_1 \oplus [\![Z^\alpha_{IU}]\!]_1$.

When decrypted, this $Z^{\alpha*}_{i,IU}$ will equal Z^{α}_{IU} if and only if the SU is located outside of the IU's obfuscated E-Zone and will allow the SU to recover a license if it is outside of the obfuscated E-Zone. Thus, $[\![Z^{\alpha*}_{i,IU}]\!]_1$ equals Z^{α}_{IU} when the request is from either of the valid SU request situations.

SAS then hashes each Z^{α}_{IU} with a different cryptographic hash function than was used as the primary hash function, which we shall refer to as the secondary hash function, and returns this list of hashes, $[\![license]\!]_{AES,k,iv}$, and all resulting $[\![Z^{\alpha*}_{i,IU}]\!]_1$ rearranged in a randomized order, to the querying SU.

SU Operations

SUs are responsible for registering with SAS prior to submitting queries, querying SAS with the proper query parameters, and attempting recovery of spectrum licenses.

- **Registration with SAS:** Each SU must initially register with SAS by providing SAS with its operator identification, device identification, and geolocation information in the registration process specified by the FCC [5].

- **SU Query Format:** The registered SU may then query SAS. Each query should contain, at minimum: the SU's current location, requested channel, requested maximum transmit power, and FCC ID [11].

- **License Recovery:** SAS will respond to valid SU queries with an encrypted license $[\![license]\!]_{AES,k,iv}$, a list of all $[\![Z^{\alpha*}]\!]_1$ values produced by each IU represented by $\{[\![Z^{\alpha*}]\!]_1\}$, and a list of hashes of the $Z^{\alpha*}$ values used in encryption of the license represented by $\{hash(Z^{\alpha*})\}$. The SU then decrypts each $[\![Z^{\alpha*}]\!]_1$ using its private key and executes the secondary hash function on the resulting decrypted $Z^{\alpha*}$. If the result

of this hash exists in the list of hashes, the SU recognizes this hash corresponds to an IU's Z^{α}, recovers k and iv using the primary hash function, and removes it from the list. The SU then uses these k and iv values to remove one layer of encryption from $[\![license]\!]_{AES,k,iv}$. Once the $Z^{\alpha*}$ for each hash in the hash list has been found and the corresponding k and iv pair used to encrypt $[\![license]\!]_{AES,k,iv}$ has be recovered, the SU can successfully fully decrypt the license. If the SU hashes each $Z^{\alpha*}$ but is unable to find all hashes in the hash list, its spectrum request was invalid and thus it is unable to recover the license.

Correctness of SZ-SAS

The correctness property requires that when an SU is located in an E-Zone of any IU, its spectrum request cannot be approved, and thus an SU cannot receive a valid spectrum license. Additionally, when an SU which has exceeded the query threshold of any IU and is located within the obfuscated E-Zone for this IU, its spectrum request cannot be approved. The SZ-SAS functionality can be donated as a function f:

$$\texttt{license}^* := f([\![\mathbf{E}_{f,m,n,tp}]\!]_1, [\![\mathbf{O}_{f,m,n,tp}]\!]_1, \tau, [\![count_{SU,IU}]\!]_1, \texttt{req}), \tag{2.7}$$

where `req` is the information received from an SU during a spectrum request.

Definition 2.1. SZ-SAS is correct if it satisfies the following condition: For any input $([\![\mathbf{E}_{f,m,n,tp}]\!]_1, [\![\mathbf{O}_{f,m,n,tp}]\!]_1, \tau, [\![count_{SU,IU}]\!]_1, \texttt{req})$ to SZ-SAS, if the requested location (m, n) is within an IU's E-Zone, or $[\![count_{SU,IU}]\!]_1 > \tau$ and (m, n) is within an IU's obfuscated E-Zone, $\texttt{license}^*$ is invalid. Conversely, if the requested location (m, n) is outside of all IU's E-Zone, and $[\![count_{SU,IU}]\!]_1 < \tau$ for all IUs or (m, n) is outside of all IU's obfuscated E-Zone, $\texttt{license}^*$ is valid

Theorem 2.2. *The probability with which SZ-SAS is NOT correct is negligible.*

Proof. The correctness follows directly from the specification of the SZ-SAS protocols. $[\![count_{SU,IU}]\!]_1$ can be updated using the homomorphic addition specified in Section 3.1. Let (m, n) be the location of `req`. If (m, n) is located within an IU's E-Zone, then $\mathbf{E}_{f,m,n,tp} \leftarrow\$ Z_p \backslash \{0\}$ for this IU. Thus, in the situation in which a query originates from the E-Zone of an IU, $Z^{\alpha*}_{IU} = Z^{\alpha}_{IU} + Z_p \backslash \{0\} \neq Z^{\alpha}_{IU}$. In the second situation, if $[\![count_{SU,IU}]\!]_1 > \tau$, then $Z^{\alpha*}_{IU} = Z^{\alpha}_{IU} + R \neq Z^{\alpha}_{IU}$ where R is some random value in G_T. Additionally, if the query is also located in the obfuscated E-Zone of an IU, then $\mathbf{O}_{f,m,n,tp} \leftarrow\$ Z_p \backslash \{0\}$ for this IU. As a result, the $Z^{\alpha*}_{IU} = Z^{\alpha}_{IU} + Z_p \backslash \{0\} \neq Z^{\alpha}_{IU}$ for the $Z^{\alpha*}_{IU}$ associated with the obfuscated E-Zone map. As a result, for all i, $Z^{\alpha*}_{IU,i} \neq Z^{\alpha}_{IU}$, and as such, the key and IV for this IU will not be recoverable, and the license will not be successfully decrypted by the SU. Conversely, if (m, n) is located outside of an IU's E-Zone, then $\mathbf{E}_{f,m,n,tp} = 0$ for this IU and additionally if $[\![count_{SU,IU}]\!]_1 < \tau$, then $Z^{\alpha*}_{IU} = Z^{\alpha}_{IU} + \mathbf{E}_{f,m,n,tp} = Z^{\alpha}_{IU}$. Also, if (m, n) is outside of an IU's obfuscated E-Zone map, then $\mathbf{O}_{f,m,n,tp} = 0$ for this IU. As a result, the $Z^{\alpha*}_{IU} = Z^{\alpha}_{IU}$ for the $Z^{\alpha*}_{IU}$ associated with the obfuscated E-Zone map and the SU can recover the license.

□

2.3.2 Implementation and Overhead Discussion

SZ-SAS was implemented in C using the pairing-based cryptography (PBC) library [22]. In order to evaluate the overhead of the current implementation, we assume 100 IU's operating in a 156.25 km^2 region divided into a grid of 50×50 cells with side lengths of 250 meters. We also assume each IU submits a suspicious request threshold of 2, and E-Zone, S-Zone, and Obfuscated E-Zone maps for 5 maximum transmit power level bins. We summarize the relevant operation parameters in Table 2.1.

Area Covered by SAS	156.25 km^2
Side Length	250 m
Total Number of Cells	2500
Suspicious Request Threshold	2 queries
Transmit Power Level Bins	5
Number of IUs	100

Table 2.1: Overhead Simulation Parameters

As mentioned previously, we are concerned primarily with the communication and computation overhead of operations requiring direct involvement by SAS which could potentially create a bottleneck at scale. The operations which require significant computation by SAS and thus must be considered in our analysis include the calculation of spectrum license encryption keys using equation 2.6, and the encryption of the spectrum license. The computation involved in each of these operations is substantial primarily because the number of iterations each of these operations must undergo is directly correlated to the number of IUs associated with SAS. Thus, as our example involves a central SAS with 100 associated IUs, SAS must encrypt the SU's requested spectrum license 100 times and produce 100 sets of N encryption keys where N is the SU suspicious request threshold value. Additionally, these operations must be executed each time an SU request is processed by SAS. SAS may receive hundreds or even thousands of SU requests per minute, so minimizing the computation time required to handle each request is essential. We present the computation overhead, as indicated by execution time, for these essential SAS operations and operations performed by SU and IU in Table 2.2.

	Entity	**Overhead (ms)**
Encryption Key Generation	SAS	375.6213
License Encryption	SAS	6.9351
E-Zone Map Encryption	IU	3786.443
License Decryption	SU	202.9326

Table 2.2: Computation Overhead of Critical Processes in SZ-SAS

The operations for which communication overhead must be considered include the transmission of the spectrum license from SAS to SU, and the commitment of new encrypted E-Zone, S-Zone, and obfuscated E-Zone maps from the IUs to SAS. The spectrum license transmission overhead will potentially be large, as it scales with the number of IUs as mentioned previously. It is the critical component of each request response and includes one encrypted spectrum license and M sets of N encryption keys where M is the number of IUs associated with SAS and N is the SU suspicious request threshold value. The communication overhead associated with the commitment of each encrypted map is directly correlated to the number of grid locations covered by SAS, because each map is an array of AFGH ciphertext values which each represent the status of a particular grid location. Our experimental setup consists of a region divided into a grid of 50×50 cells; therefore, each encrypted map consists of 2500 ciphertext values. We present the communication overhead for these operations and operations performed by the SU in Table 2.3

	Source	**Destination**	**Overhead**
Spectrum License Transmission	SAS	SU	601.2 kB
Encrypted Map Commitment	IU	SAS	75 MB
Spectrum Request	SU	SAS	0.12 kB

Table 2.3: Communication Overhead of Critical Processes in SZ-SAS

It is evident that the encrypted map commitment process requires by far the most over-

head. This value is directly related to the number of cells covered by SAS and the number of transmit power level bins available, so it will scale directly with these two parameters. However, this map commitment will only occur either during initialization of the system or when an IU wishes to modify its E-Zone, S-Zone, and obfuscated E-Zone maps. These situations should be infrequent, so the high overhead of this particular operation should not have a major impact on the operation of the system.

2.4 Summary

In this chapter, we introduced SZ-SAS, a novel framework which focuses on preserving IU privacy in the scenario of untrusted SAS and untrusted SU. This framework leverages the additive version of the AFGH cryptosystem to allow SAS to apply obfuscation to individual users which have made queries from locations deemed to be suspicious while allowing honest users to access spectrum without the spectrum utility degradation introduced by obfuscation. In the next chapter, we will discuss obfuscation techniques which are compatible with SZ-SAS.

Chapter 3

Preserving Location Privacy of Incumbent Users Against Secondary Users

3.1 Introduction

The main focus of this thesis is the development of framework which which would allow obfuscation to be applied discriminantly to SU queries based upon individual or group SU query history within a homomorphically encrypted database. In this chapter, we demonstrate the necessity of obfuscation by introducing the inference attack, a technique which can be utilized by an adversarial SU to determine sensitive IU parameter information. We then examine several obfuscation methods, focusing on those which are compatible with the proposed framework and which could thwart the inference attack. We also introduce a novel obfuscation technique, envelopment by offset virtual IUs (OVIU).

3.2 Related Work

Location privacy has grown has a field of research in recent years with the increased ubiquity of location-based services (LBS). As mentioned previously, inference attacks are a primary concern in the security analysis of LBS. First identified as a concern in SAS-based DSA systems in [9], adversarial SUs may execute an inference attack to correlate the results of spectrum availability queries in order to infer the sensitive operation parameters of associated IUs. A specific inference attack algorithm for determining the geolocation of an IU is also introduced in [9] and has led to research efforts which have focused on countering this attack via obfuscation.

Location obfuscation in SAS-based DSA involves the replacement of strategically selected query responses with false positives, which are introduced in such a manner that an adversarial SU may incorrectly infer the location of an IU. Researchers have proposed many obfuscation techniques of varying complexity, including k-anonymity, k-clustering, perturbation with additive noise, perturbation with transfiguration, optimal obfuscation, and randomized transmit inhibition [9, 10, 28, 30], with varying levels of success. Several of these studies also identify and discuss the trade-off between IU privacy and SU spectrum utilization, which is an issue inherent to all location obfuscation techniques.

k-Anonymity

k-anonymity was introduced in [25] as a method to allow access to individualized data whilst maintaining the anonymity of users to which the data belongs. This obfuscation technique is successful if an attempt to determine the identity of a user results in at least k potential indistinguishable possibilities. k-anonymity has previously been applied to location based services [15, 29] using various cloaking techniques, thus its application to IU privacy in SAS-

based DSA is fairly straightforward. With regards to location, k-anonymity is satisfied if a user's location is indistinguishable from at least $k-1$ other users' locations.

In [9, 30], the authors propose a method to employ k-anonymity by grouping all n IUs associated with SAS into $\lceil \frac{n}{k} \rceil$ groups of k IUs which are nearest to one another. The algorithm then replaces each group of IUs with a single virtual IU, which has an E-Zone which envelops all k E-Zones of the associated IUs.

The authors of [9] note that this obfuscation scheme may result in significant loss to SU spectrum utility, as sparsely distributed IUs may result in dramatically enlarged E-Zones. They attempt to address this utility loss with another obfuscation technique, known as k-clustering.

k-Clustering

This obfuscation technique is similar to k-anonymity, except with a difference in the method of grouping. k-anonymity forms $\lceil \frac{n}{k} \rceil$ groups of k IUs, whereas k-clustering forms k groups of variable size. The algorithm begins by forming a group between the two nearest IUs and replacing those two IUs with a new virtual IU. This process of combining IUs and combining clusters continues while there are more than k clusters. The author adds that at this point, other obfuscation strategies such as perturbation with additive noise or perturbation with transfiguration may also be applied [9].

Analyzing this obfuscation technique, there is one readily apparent flaw. k-anonymity can be implemented such that each group is guaranteed to contain k users. However, in k-clustering, an IU which is located far from any potential neighbors will never be joined into a cluster. For example, in a 2-clustering scenario, the algorithm may group all IUs except for a singular outlying IU together as illustrated in figure 3.1. This would result in no added

security benefit to this outlying IU and leave it vulnerable to an SU inference attack.

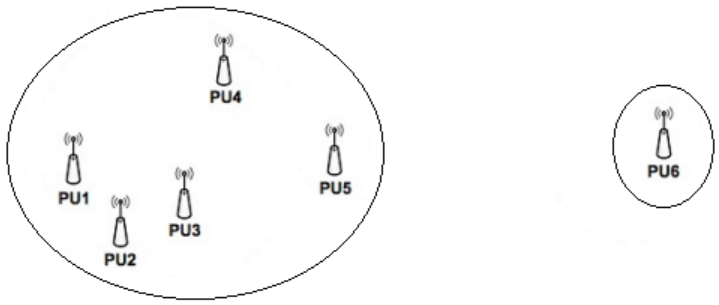


Figure 3.1: k-clustering scenario with vulnerable outlier

Enlarging the Exclusion Zone (Perturbation with Additive Noise)

An additional obfuscation scheme based upon the introduction of additive noise is described in [9]. In order to add obfuscation based upon this scheme, SAS simply introduces negative random noise to the transmit power level, P_k resulting in a decreased maximum transmit power level, $P'_k = P_k + \epsilon_k$, where ϵ_k is the additive noise value. SAS may then determine E-Zone information based upon this new maximum transmit power level. In this scheme, this additive noise is initialized to a random value and then held constant at this value for all queries, causing the scheme to be analogous to enlarging the exclusion zone of an IU as proposed in [28]. The authors also mention that this noise value must be negative, as a positive noise value would result in $P'_k > P_k$, allowing SUs to transmit at maximum transmit power levels which could cause interference for the IUs.

Optimal Obfuscation

This obfuscation technique introduces variable levels of additive noise to each query, generated by maximizing the IU's location privacy while attempting to minimize the loss in

spectrum utilization. In this obfuscation strategy, SAS maintains a history of the information revealed to each SU in the SU's query responses. It then uses this information, the SU's current querying location, and the perturbation with additive noise technique to respond to future queries without revealing more information or causing too great of spectrum utility degradation [10].

Perturbation with Transfiguration

This technique attempts to mask any symmetry in E-Zones by transforming a previously symmetrical E-Zone boundary into an N-sided polygon with an irregular shape. Symmetrical E-Zones are the result of basic propagation models, which produce circular E-Zones of a constant radius because they ignore irregularities such as terrain information. However, this strategy cannot be utilized if E-Zones are generated using more sophisticated propagation models, such as the Longley-Rice model [17].

Randomized Transmit Inhibition (RTI)

This obfuscation scheme replaces responses to SU queries with false positives according to a defined probability of replacement, p_{fp}. This strategy can replace responses with false positives for queries originating either indiscriminately from any location or from a particular area of interest, such as the E-Zone boundaries of an IU [28].

3.3 Location Inference Attack and Countermeasures

In this section, we first define the system model and threat model for our location inference attack. We then discuss the location inference algorithm introduced in [9] and subsequently

introduce our variation of this algorithm, which is focused on the detection of a single IU existing in a given geographic area. We then compare the two inference attack algorithms, showing the number of queries required to locate a given IU is lower than previously expected, proving the necessity of leveraging obfuscation to mitigate such an attack.

3.3.1 The System Model

As was established in the previous chapter, SAS covers an $M \times N$ geographic area and the system follows the same database access protocol as described in Chapter 2. We focus on a single IU and a single SU operating within the region maintained by SAS for our testing, but the techniques described can be applied to scenarios consisting of multiple IUs and multiple SUs. The single IU scenario is of particular interest, because sparsely distributed IUs are especially vulnerable to inference attacks. IUs with connected exclusion zones can potentially mask one another's precise location due to their exclusion zone overlap, IUs with disjointed exclusion zones lack this potential innate protection.

3.3.2 Threat Model

We assume that there exists a singular honest-but-curious mobile SU with the ability to query SAS throughout the entirety of the region covered by SAS, this SU shall henceforth be referred to as the adversary. The adversary's goal is to determine the grid location of a stationary IU using only information gained from the responses received from SAS. We assume the attacker has knowledge of $T(P_L, I_{th})$, the propagation model used by the IUs to calculate P_L, and I_{th}. We also assume the adversary has side knowledge indicating the existence of at least one IU operating on a channel of interest served by SAS.

3.3.3 Location Inference Algorithms

Inference algorithms have been previously developed which can generate a series of probability values which represent the adversary's confidence that an IU exists in the corresponding grid location. These algorithms are robust, and focus on locating multiple IUs while being able to potentially ignore false positive responses and is useful when we are attempting to evaluate the efficacy of proposed obfuscation schemes. However, a major focus of this study is the minimum privacy level of any single incumbent user, and thus we propose a separate algorithm with the focus of locating a singular incumbent user lacking the ability to inject false positive responses. This algorithm is introduced below.

First-Detected IU Inference Algorithm

We first define a Bernoulli random variable, $R_{xy}^{(k)}$, which represents the probability of an Incumbent user existing in grid location $g(x, y)$ on channel k. Based upon the properties of the Bernoulli distribution, $P(R_{xy}^{(k)} = 1) = p_{xy}^{(k)}$ and $P(R_{xy}^{(k)} = 0) = 1 - p_{xy}^{(k)}$. Because the adversary has knowledge that there exists at least one IU on a channel of interest in the area covered by SAS, and is only attempting to locate a single IU, the IU is equally likely to be located in any of the grid locations covered by SAS. Thus, we initialize $p_{xy} = \frac{1}{MN}\ \forall\ g(x, y)$ on the given channel.

The adversary then queries the database from a chosen location, first for TP_2 and potentially again for TP_1 if the query for TP_2 resulted in acquisition of an invalid license. Based upon the responses received for the queries from the chosen location, the adversary then updates the value p_{xy} for all affected $g(x, y)$. The adversary's inference regarding IU location for each possible combination of query responses is as follows:

- **Case 1 (Valid License for TP_2):** This first case implies that there are no IUs

operating in any cells with path loss values less than P_{L2} relative to the query location. Because $P_{L2} < P_{L1}$, any cells that have path loss values less than P_{L2} also have path loss values less than P_{L1}, so the adversary does not need to query again for TP_1. Because there are no IUs in any cells with $P_L < P_{L2}$, the adversary sets $p_{xy} = 0$ for all $P_{L_{xy}} < P_{L2}$. The adversary then adjusts p_{xy} for each remaining non-zero p_{xy} to reflect the current number of possible IU locations. As all non-zero locations are equally likely to contain the IU, $p_{xy} = \frac{1}{(MN)-n_{p0}}$, where n_{p0} is the total count of $g(x, y)$ with $p_{xy} = 0$.

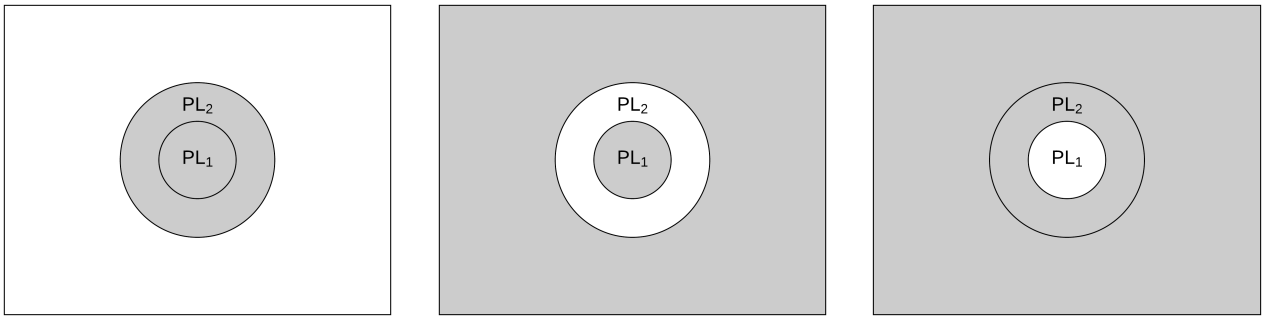
- **Case 2 (Invalid License for TP_2, Valid License for TP_1):** In this case, the adversary's query for TP_2 implies that there is an IU operating in some cell with path loss value less than P_{L2} relative to the queried location. The adversary then queries for TP_1 and is able to infer that there is no adversary operating in a cell with path loss value less than P_{L1}. Thus, there exists an adversary in a cell with $P_{L1} < P_{L_{xy}} < P_{L2}$ with respect to the queried location.

  The adversary uses this information to update the affected values of p_{xy} by first setting $p_{xy} = 0$ for all $P_{L_{xy}} > P_{L2}$ or $P_{L_{xy}} < P_{L1}$. The adversary then adjusts all remaining non-zero p_{xy} using the same logic as in the first scenario.

- **Case 3 (Invalid License for both TP_2 and TP_1):** This final case indicates the existence of an IU operating in a cell with $P_{L_{xy}} < P_{L_1} < P_{L_2}$. The adversary sets $p_{xy} = 0$ for all $P_{L_{xy}} > P_{L_1}$ and uses the same logic as in the first scenario to update all remaining non-zero p_{xy}.

3.3.4 Location Privacy Metrics and Algorithm Comparison

The two inference attack algorithms presented in this thesis can be best compared by determining the number of queries required to locate the IU with some degree of certainty, which



(a) Case 1 (b) Case 2 (c) Case 3

Figure 3.2: Examples of inferences made by the adversary based upon each case of query result in a location inference attack

can be quantified either as the value of p_{xy} for the IU's actual location or as the calculated incorrectness (IC) as defined in [26]. The IC value represents the distance between the actual and inferred location of the IU.

$$IC = \sum_{x=1}^{M} \sum_{y=1}^{N} p_{i,j} d(i,j) \tag{3.1}$$

where $p_{i,j}$ is the inferred probability an IU exists in cell (i,j) and $d(i,j)$ is the euclidean distance from (i,j) to the actual IU location. For the remainder of this thesis, all presented incorrectness values have been normalized by dividing the incorrectness value with the maximum possible incorrectness. The maximum incorrectness is the incorrectness prior to any queries, when $p_{i,j} = \frac{1}{MN}$ for all cells.

	IC <0.01	**p(x,y) >0.90**
Standard Algorithm	142.12022	161.18045
Modified Algorithm	115.75138	111.24599

Table 3.1: Comparison of standard inference attack algorithm presented in [9] and our modified inference attack algorithm.

In order to compare these two algorithms, we simulate an adversary which queries from randomly selected cells and uses the results of each query to update p_{xy} with each algorithm

as it attempts to locate a singular IU which exists at location $(25, 25)$ of the 50×50 grid covered by SAS. We average the results of 1000 trials in order to mitigate the inherent randomness introduced by this adversary's querying technique. Our results, as presented in Table 3.1, show that the modified inference attack algorithm allows an adversary to locate an IU with fewer queries than the standard inference attack algorithm. We also present a heatmap, which allows us to visualize the p_{xy} for each grid location during the execution of an inference attack in Figures 3.3 and 3.4.

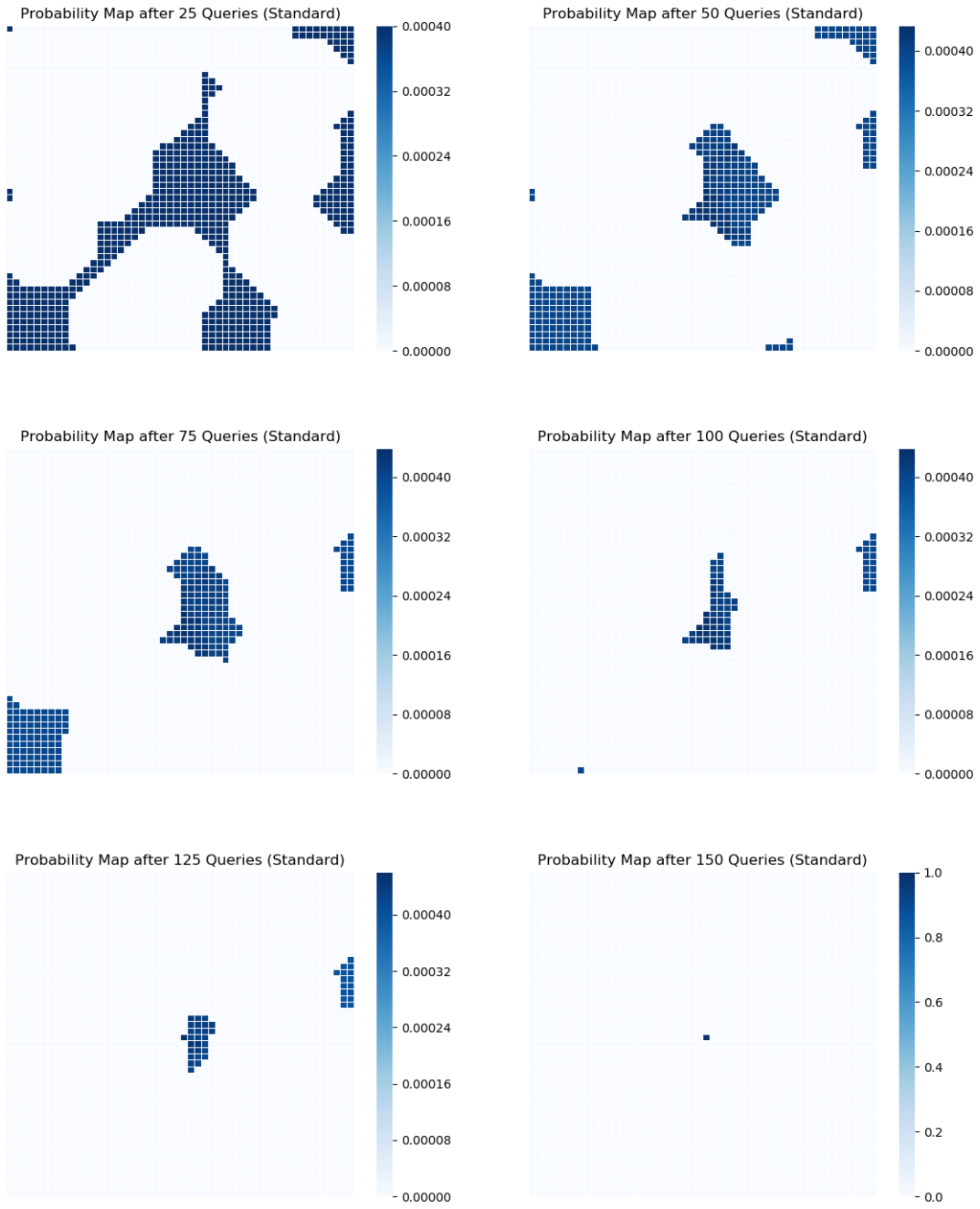


Figure 3.3: Heatmap of p_{xy} values during a standard inference attack

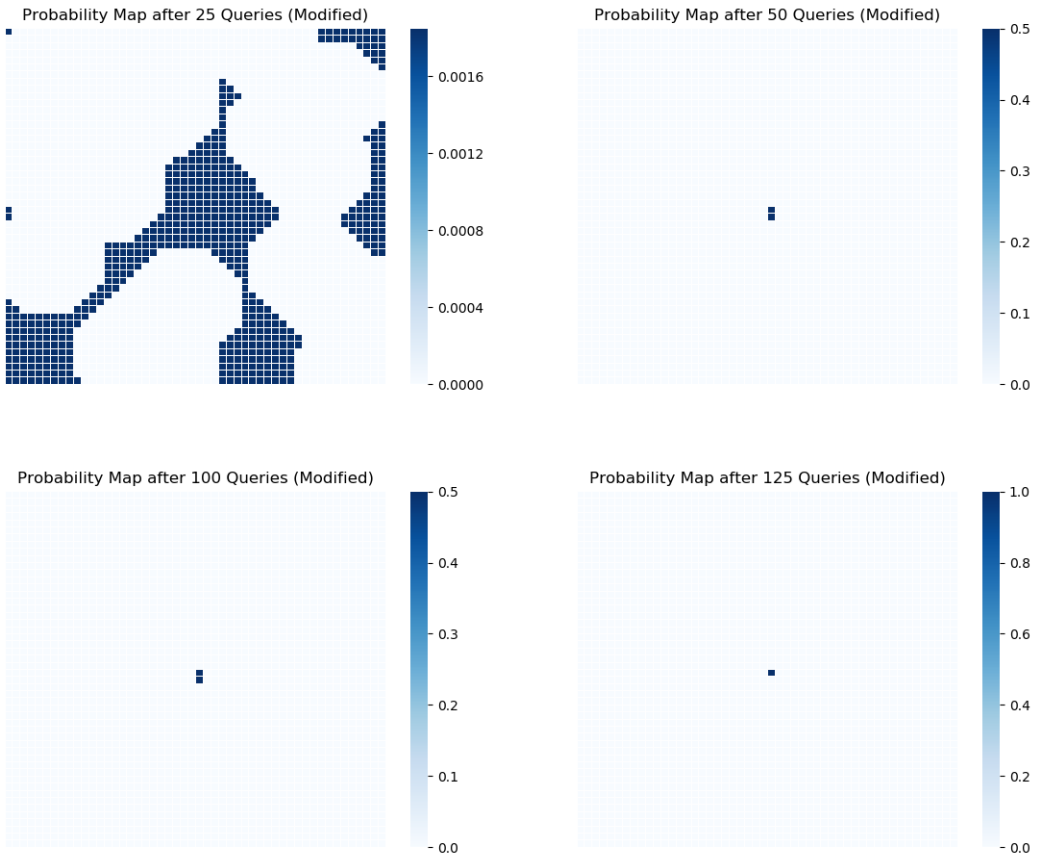


Figure 3.4: Heatmap of p_{xy} values during a modified inference attack

3.4 Determining SZ-SAS Compatible Obfuscation Schemes

Previous works, such as those discussed in Section 3.2, have focused entirely on obfuscation techniques which could be applied directly by SAS. These obfuscation techniques require SAS to have intimate knowledge of all operating parameters of each IU and the results of SU queries. Due to the end-to-end encryption of our system spanning IU to SU, SAS does not have this information. This limitation requires obfuscation schemes to meet the following criteria:

- Schemes requiring obfuscation maps must be able to calculate these obfuscation maps individually by each IU prior to commitment to SAS.
- The scheme can only restrict SU queries either indiscriminately or based upon each SU's query location history.

Techniques such as k-anonymity and k-clustering [9] rely on SAS to form groups, or clusters, of IUs. Thus, these schemes fail to meet the first proposed criteria.

The optimal obfuscation strategy proposed in [10] relies on the ability of SAS to dynamically compute the information revealed to an adversary executing an inference attack. This obfuscation scheme clearly fails to meet both of our criteria, as SAS must have access to the IU operation parameters in order to dynamically determine and limit the information revealed to an adversary.

In the following subsections, we analyze three obfuscation techniques which are compatible with SZ-SAS, including randomized transmit inhibition, perturbation with additive noise, and envelopment by virtual offset IUs, and discuss any modifications needed to utilize these schemes with SZ-SAS.

3.4.1 Randomized Transmit Inhibition (RTI)

In order to apply RTI in SZ-SAS, SAS can add a nonce to the SU key calculation formula according to the probability of replacement, p_{fp}, causing the affected keys to become invalidated. It is important to note that p_{fp} can affect any key, whether or not it was valid prior to the addition of the nonce, and p_{fp} is therefore not to be confused with the probability of replacing a valid key with a false positive.

This strategy can replace responses with false positives for queries originating either from an

area that has been predefined by the IU and committed to SAS, or indiscriminately from any location. The modified SU key calculation formula for these two cases is shown in equations 3.2 and 3.3 respectively.

$$\forall i \in [0, \tau] : [\![Z^{\alpha*}_{i,IU}]\!]_1 \leftarrow [([\![count_{SU,IU}]\!]_1 \oplus [\![\epsilon_i]\!]_1^{-1}) \otimes R] \oplus [\![\mathbf{E}_{f,m,n,tp}]\!]_1 \oplus [\![Z^{\alpha}_{IU}]\!]_1 \oplus [\![R_{fp}]\!]_1 \quad (3.2)$$

$$\forall i \in [0, \tau] : [\![Z^{\alpha*}_{i,IU}]\!]_1 \leftarrow [([\![count_{SU,IU}]\!]_1 \oplus [\![\epsilon_i]\!]_1^{-1}) \otimes R] \oplus [\![\mathbf{E}_{f,m,n,tp}]\!]_1 \oplus [\![Z^{\alpha}_{IU}]\!]_1 \oplus [\![\mathbf{R}_{f,m,n,tp}]\!]_1 \quad (3.3)$$

Where R_{fp} is a random nonce and $[\![\mathbf{R}_{f,m,n,tp}]\!]_1$ is the value of a map committed by the IUs which contains the value of a random nonce for locations of interest and 0 for all other locations. The results of these formulas will then be used to replace the results of formula 2.6 at a rate consistent with p_{fp}

3.4.2 Enlarging the Exclusion Zone

In our work, the IUs are responsible for submitting obfuscation maps. The IU can create obfuscated maps using this scheme by simply decreasing their interference threshold by some value and then calculating the obfuscation map using the same process as when it calculates an actual exclusion zone map. This will result in enlarged E-Zones, which could potentially obfuscate the true location of the IU.

3.4.3 Envelopment by Offset Virtual IUs (OVIU)

This novel obfuscation scheme covers the entirety of the actual IU's E-Zone area with E-Zones which have been generated by virtual IUs. In order to simulate generate a virtual IU, we first select a random x and y location offset and add this offset to the IU's true location to create a new location for the virtual IU. We then apply enlarge the virtual IU's exclusion zone using the same technique as in Section 3.4.2. We repeat this process k times, to generate k distinct virtual IUs. We then adjust the location offset and exclusion zone enlargement until the true IU's exclusion zone is completely enveloped by the newly generated virtual IU exclusion zones. This results in a new E-Zone map which obfuscates the E-Zone map of the original IU, but is not centered around the IU's true location.

3.5 Performance of Compatible Obfuscation Techniques

We conducted a series of simulations testing each of the proposed obfuscation schemes. In these simulations, we consider an area covered by SAS consisting of a 12.5 km $\times$ 12.5 km region divided into 50 $\times$ 50 cells with grid side lengths of 250 meters. Within this region, there exists a singular IU at location $(25, 25)$.

We generate Exclusion Zone Maps using the the ECC-33 propagation loss model [6] with realistic military radar interference thresholds as described in [21] and SU transmit power and antenna height properties as defined by the FCC [18]. Of particular importance, we have I_{th} = -105 dBm, TP_1 = 23 dBm, TP_2 = 30 dBm, and f = 3.55 GHz.

As in Section 3.3.4, we simulate a random adversary utilizing the standard inference attack algorithm and average the results of 1000 trials for each simulation. We first simulate a naive adversary, which does not attempt to leverage information regarding the obfuscation

schemes. However, this scenario essentially assumes security by obscurity. A more realistic scenario is one in which the adversary attempts to compromise the location privacy of the IU using information about the potential obfuscation schemes employed. We simulate this scenario as well, and refer to this adversary as the strategic adversary.

3.5.1 Randomized Transmit Inhibition (RTI)

The simulation results for a naive adversary conducting an inference attack on an E-Zone map which has been obfuscated using the RTI scheme are presented in Figure 3.5.

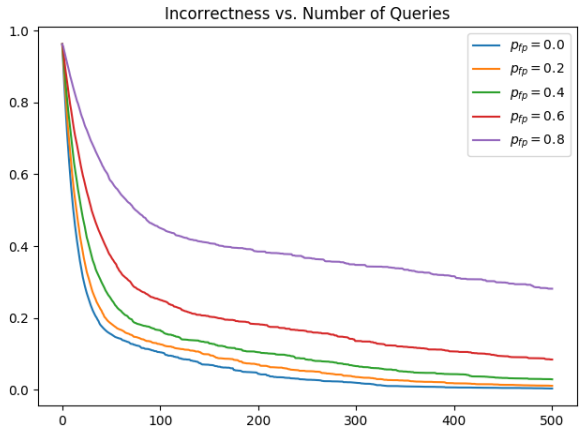


Figure 3.5: Location privacy of RTI against naive adversary

Table 3.2 presents the incorrectness value following the execution of the inference attack by a naive adversary averaged over all 1000 trials. These results appear to illustrate that at higher p_{fp} values, the adversary will be unable to infer the location of an IU, particularly around $p_{fp} = 0.8$.

However, in order to mitigate the obfuscation introduced by RTI, an adversary can filter out false positive responses by simply querying multiple times from each location it visits and ignoring responses which indicate the existence of an IU unless every response from a given

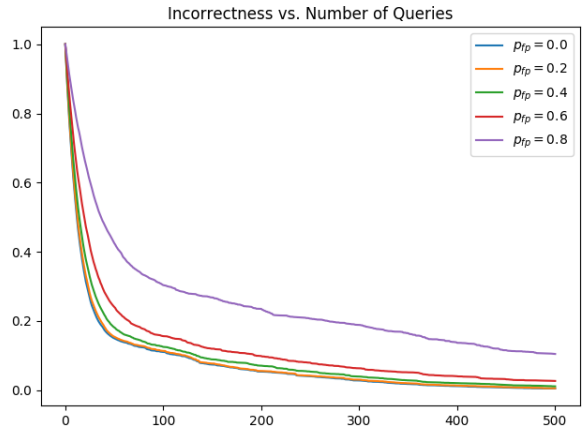


Figure 3.6: Location privacy of RTI against strategic adversary $n = 2$

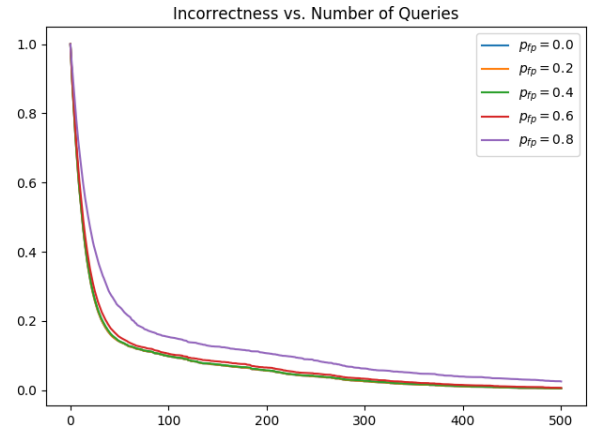


Figure 3.7: Location privacy of RTI against strategic adversary $n = 5$

location indicates the existence of an IU. We simulated two strategic adversaries, one which queries each location twice, and one which queries each location five times. Their results are displayed in Figures 3.6 and 3.7. Tables 3.3 and 3.4 present the incorrectness value following the execution of the inference attack by a naive adversary averaged over all 1000 trials for the strategic adversary.

p_{fp}	IC
0.0	0.0033268633646328922
0.2	0.01081518355202428
0.4	0.02892036037782027
0.6	0.0840254187079344
0.8	0.2812790501940863

Table 3.2: Incorrectness of naive adversary against RTI

p_{fp}	IC
0.0	0.004972927802358861
0.2	0.005638240722767167
0.4	0.011178501157565721
0.6	0.026743223723742143
0.8	0.10475184113540945

Table 3.3: Incorrectness of a strategic adversary against RTI ($n = 2$)

p_{fp}	IC
0.0	0.004546076882455241
0.2	0.004546425015609954
0.4	0.00486759287327197
0.6	0.006430687986600399
0.8	0.024911629955400555

Table 3.4: Incorrectness of a strategic adversary against RTI ($n = 5$)

It is evident that the RTI obfuscation scheme provides little protection against an adversary

which is filtering potential false positives, even at higher p_{fp} values. Thus, although this scheme is compatible with SZ-SAS, it may not provide the desired IU privacy protection.

3.5.2 Enlarging the Exclusion Zone

Next, we simulate a naive adversary conducting an inference attack on an E-Zone map which has been obfuscated using the EEZ scheme with several additive noise values, including $\epsilon =$ 0, 5, 10, 15, and 20 dBm. The average trend in incorrectness during execution of 1000 trials of the inference attack by a naive adversary is presented in Figure 3.8.

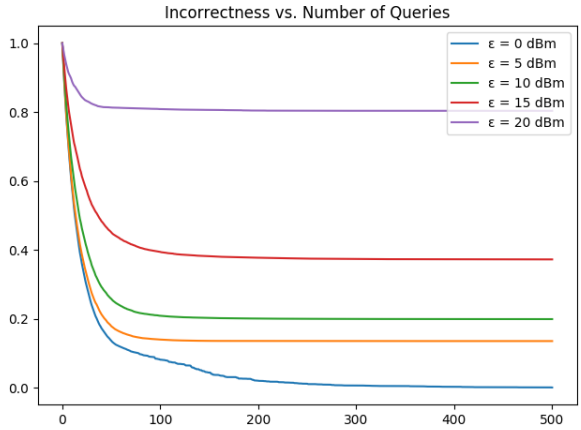


Figure 3.8: Location privacy of EEZ against naive adversary

We note that after approximately 100 queries, the adversary is unable to further reduce the value of IC. Additionally, Table 3.5 presents the incorrectness value following the execution of the inference attack by a naive adversary averaged over all 1000 trials. These results appear to illustrate that increasing the additive noise, ϵ, will result in situations in which the adversary will be unable to infer the location of an IU beyond the stabilized IC value which is reached after approximately 100 queries.

ϵ	IC
0 dBm	0.0014562236111755227
5 dBm	0.13606529281548715
10 dBm	0.1997603678872978
15 dBm	0.37296142606321414
20 dBm	0.8038758282063957

Table 3.5: Incorrectness of naive adversary against EEZ

δ	IC
0 dBm	0.11255450503958538
5 dBm	0.05542989439565684
10 dBm	0.027160212627258387
15 dBm	0.0

Table 3.6: Incorrectness of strategic adversary against EEZ at $\epsilon = 15$ dBm

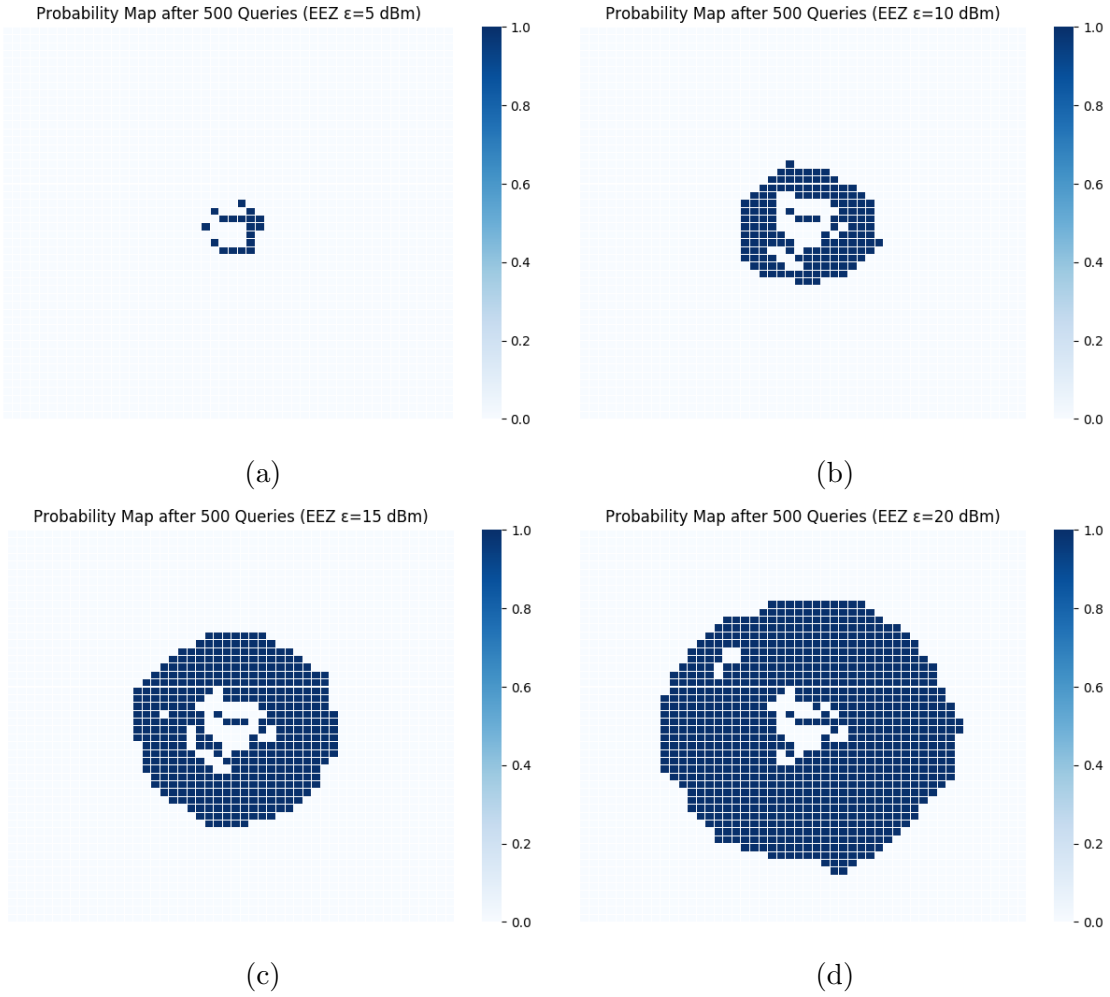


Figure 3.9: Heatmap of p_{xy} from a naive adversary against an IU employing EEZ obfuscation

Figure 3.9 visualizes the p_{xy} values for the entire grid at the conclusion of a single instance of the inference attack. The adversary may recognize the application of EEZ obfuscation due to the contiguous areas of high p_{xy} values. This adversary may then attempt to remove the EEZ obfuscation by adjusting the inference attack to remove the additive noise introduced by EEZ. This adversary can perform several inference attacks concurrently based upon only one set of queries by analyzing the query responses at several I_{th} values, which can be calculated in order to negate ϵ by adding a subtracting a noise value, represented by δ, from I_{th}.

Next, we simulate this strategic adversary by first producing an E-Zone map which has been obfuscated with EEZ at $\epsilon = 15$ dBm for an IU located at (25,25) and then performing the inference attack with several newly calculated I'_{th} values, in which $I'_{th} = I_{th} - \delta$. Heatmaps visualizing the p_{xy} for all cells following the execution of an inference attack with $\delta = 0$, 5, 10, and 15 are presented in Figure 3.10.

As expected, when the adversary adjusts I'_{th} to account for the noise introduced by EEZ, as in Figure 3.10d, the adversary is able to negate this obfuscation. The adversary can simply evaluate the query responses received during an inference attack for a multitude of I'_{th} values until it is able to effectively locate the obfuscated IU.

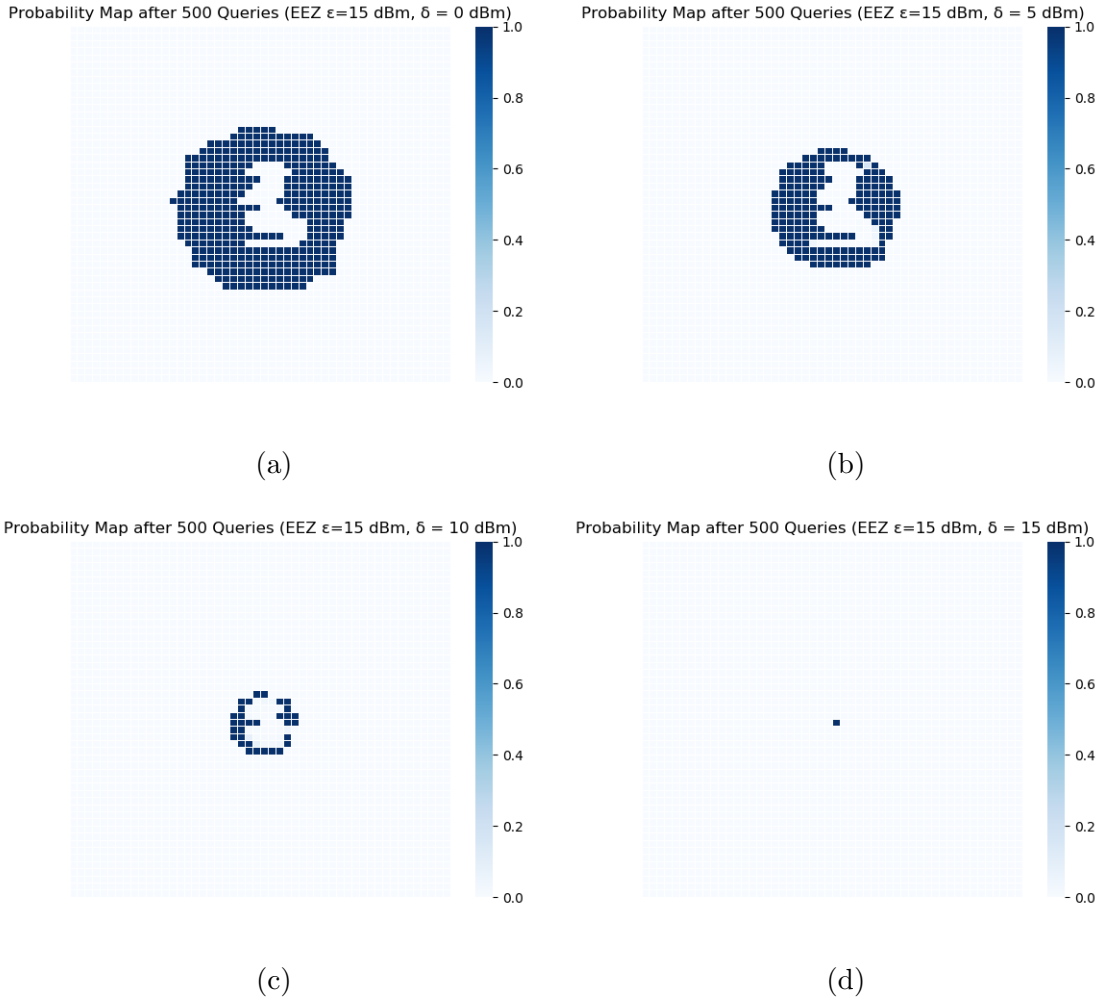


Figure 3.10: Heatmap of p_{xy} from a strategic adversary executing an inference attack against an IU employing EEZ obfuscation

3.5.3 Envelopment by Offset Virtual IUs (OVIU)

Finally, in order to evaluate OVIU, we simulate a naive adversary conducting an inference attack on an IU which has been obfuscated using the OVIU scheme with two virtual IUs. First, two sets of x and y-coordinate offsets were randomly generated. In our simulation, these two offsets resulted in virtual IUs centered at (22, 28) and (27, 22). We then obfuscate each virtual IU using the EEZ obfuscation scheme such that these virtual IUs will produce an

E-Zone map which completely covers the E-Zone cells of the original IU. Thus, we obfuscate each virtual IU with the EEZ obfuscation scheme at $\epsilon = 5$, 10, and 15 dBm. The average trend in incorrectness during execution of 1000 trials of the inference attack by a naive adversary is presented in Figure 3.11 for each of these ϵ values.

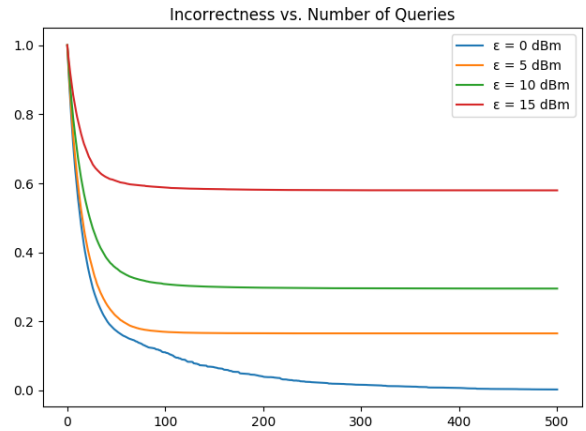


Figure 3.11: Location privacy of OVIU against naive adversary

ϵ	IC
0 dBm	0.0028803411050649947
5 dBm	0.16534184673473434
10 dBm	0.2951511469653312
15 dBm	0.5796339525806437

Table 3.7: Incorrectness of a naive adversary against OVIU

δ	IC
0 dBm	0.33993922745293764
5 dBm	0.3274084698913646
10 dBm	0.22511575696199818
15 dBm	0.0736760509749309

Table 3.8: Incorrectness of a strategic adversary against OVIU at $\epsilon = 15$ dBm

We note that once again, after approximately 100 queries, the adversary is unable to further reduce the value of IC. Figure 3.11 and Table 3.7 both demonstrate similar results to those presented in Figure 3.8 that increasing the additive noise, ϵ, will result in situations in which the naive adversary will be unable to infer the location of an IU beyond the stabilized IC value which is reached after approximately 100 queries.

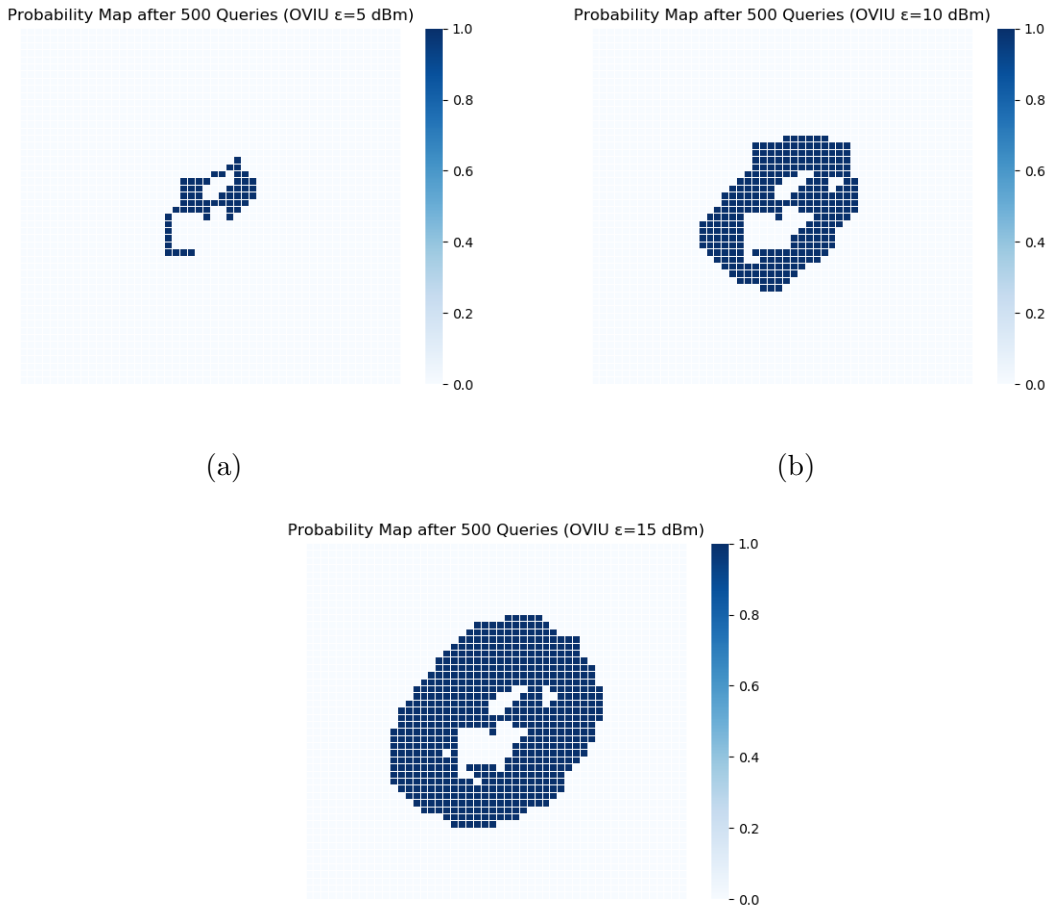


Figure 3.12: Heatmap of p_{xy} from a naive adversary against an IU employing OVIU obfuscation

However, the strategic adversary can attempt to filter the noise introduced by EEZ as in Section 3.5.2. We simulate this strategic adversary in a similar manner, by first producing an E-Zone map which has been obfuscated with OVIU for an IU located at (25,25), resulting in two virtual IUs centered at (22, 28) and (27, 22) with EEZ obfuscation of $\epsilon = 15$ dBm. We then perform the inference attack with several newly calculated I'_{th} values. The resulting heatmaps for $\delta = 0$, 5, 10, and 15 are presented in Figure 3.13.

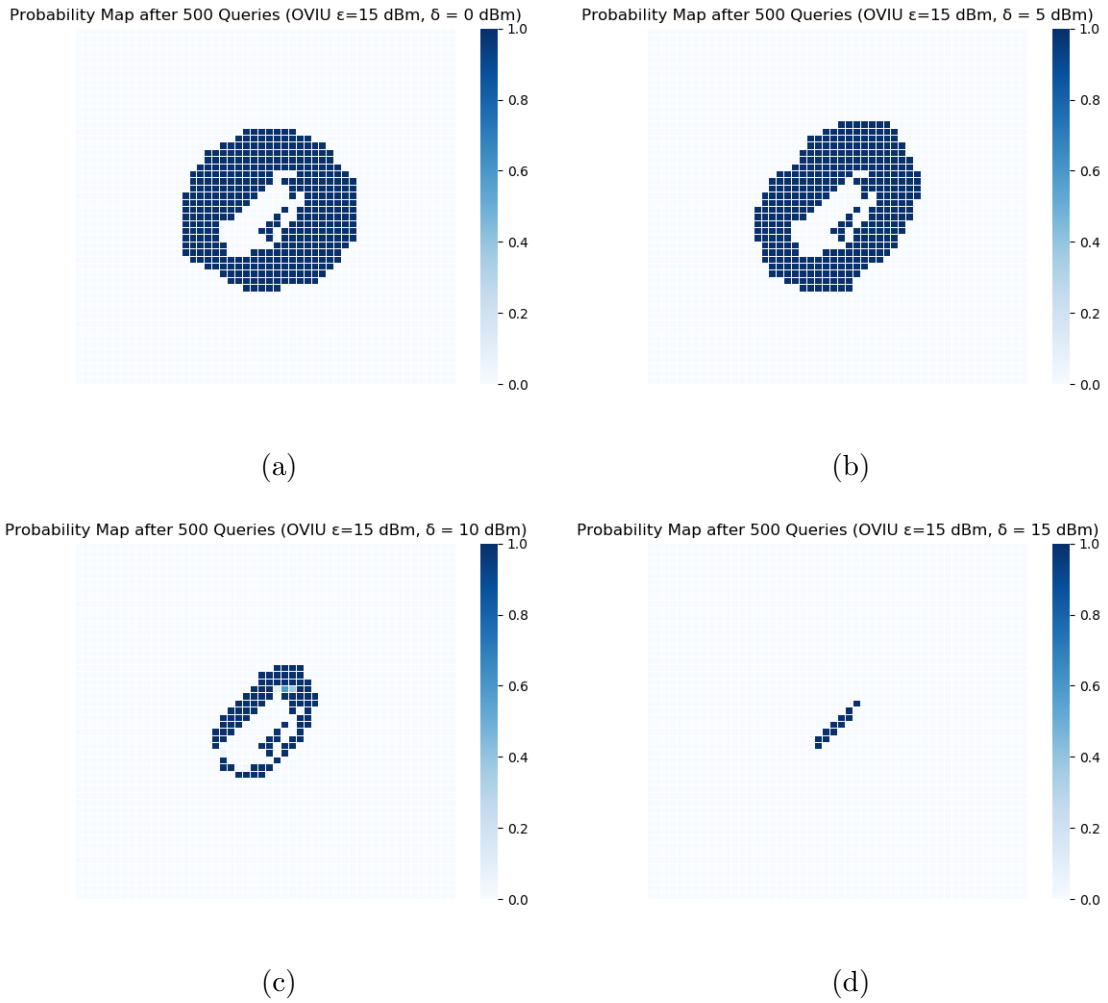


Figure 3.13: Heatmap of p_{xy} from a naive adversary against an IU employing OVIU obfuscation

In this case, when the adversary adjusts I'_{th} to account for the noise introduced by EEZ, as in Figure **??**, the adversary is able to localize the virtual IUs to a general area. However, because these virtual IUs are located a random distance in a random direction from the true location of the IU, the adversary is unable to accurately locate beyond a general area of possible locations near the virtual IUs. Thus, this obfuscation scheme could potentially be utilized to maintain the location privacy of IUs within the SZ-SAS framework.

Chapter 4

Future Work and Conclusions

4.1 Future Work

In future works, SZ-SAS can be utilized as a building block when designing SAS-based DSA systems. Future works may also develop and analyze other SZ-SAS-compatible obfuscation methods, as our list of proposed obfuscation schemes is certainly not exhaustive. More effective and spectrum-efficient [16] obfuscation schemes which could be developed and implemented within SZ-SAS in order improve IU location protection. Additionally, as light-weight fully homomorphic cryptosystems are developed, AFGH may be replaced by one such cryptosystem in SZ-SAS, which would allow researchers more flexibility when designing novel obfuscation schemes. Reimplementing SZ-SAS with a fully homomorphic cryptosystem could also the privacy of SUs to be preserved in addition to the location privacy provided to IUs.

4.2 Conclusions

In this thesis, we discussed operational security issues present in SAS-based DSA designs. In order to address these security concerns, we introduced SZ-SAS, a novel SAS-based DSA framework which is the first such framework to protect sensitive IU parameters from both untrusted SUs and untrusted SAS. Additionally, we proposed modifications to the SU inference

attack and demonstrate the effectiveness of such an attack. Our simulations demonstrated the necessity of obfuscation in order to prevent the inference attack.

Finally, we discussed obfuscation techniques which are compatible with SZ-SAS and performed inference attack simulations to determine their ability to mitigate the inference attack. We determined SZ-SAS is compatible with obfuscation schemes which meet a set of criteria, defined in Section 3.4. The results of our simulations indicate some of these obfuscation schemes have potential vulnerabilities, which allow a smart attacker to mitigate the obfuscation, whereas other schemes provide IUs with protection from the SU inference attack.

Bibliography

[1] The zettabyte era: Trends and analysis. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.

[2] Fcc auctions. http://wireless.fcc.gov/auctions/default.htm?job=auctions_home, .

[3] Acronyms and definitions pertaining to commercial operations in the u.s. 3550-3700 mhz band. https://www.wirelessinnovation.org/fcc-definitions, .

[4] United states frequency allocations: The radio spectrum. https://www.ntia.doc.gov/files/ntia/publications/january_2016_spectrum_wall_chart.pdf.

[5] Small entity compliance guide: Amendment of the commissions rules with regard to commercial operations in the 3550-3650 mhz band. Technical Report FCC 15-47, September 2015.

[6] V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellars, and M. G. Brown. Comparison of empirical propagation path loss models for fixed wireless access systems. In *2005 IEEE 61st Vehicular Technology Conference*, volume 1, pages 73–77 Vol. 1, May 2005. doi: 10.1109/VETECS.2005.1543252.

[7] M. Altamimi, M. Weiss, and M. McHenry. Enforcement and spectrum sharing: Case studies of federal-commercial sharing. 09 2013.

[8] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*,

9(1):1–30, February 2006. ISSN 1094-9224. doi: 10.1145/1127345.1127346. URL http://doi.acm.org/10.1145/1127345.1127346.

[9] B. Bahrak, S. Bhattarai, A. Ullah, J. M. J. Park, J. Reed, and D. Gurney. Protecting the primary users' operational privacy in spectrum sharing. In *2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pages 236–247, April 2014. doi: 10.1109/DySPAN.2014.6817800.

[10] S. Bhattarai, P. R. Vaka, and J. M. J. Park. Thwarting location inference attacks in database-driven spectrum sharing. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1, 2017. doi: 10.1109/TCCN.2017.2785770.

[11] V. Chen, S. Das, L. Zhu, J. Malyar, and P. McCann. Protocol to Access White-Space (PAWS) Databases. RFC 7545, May 2015. URL https://rfc-editor.org/rfc/rfc7545.txt.

[12] M. G. Cotton and R. A. Dalke. Spectrum occupancy measurements of the 35503650 megahertz maritime radar band near san diego, california. Technical Report TR 14-500, National Telecommunications and Information Administration, January 2014.

[13] Y. Dou. *Toward Privacy-Preserving and Secure Dynamic Spectrum Access*. phdthesis, Virginia Polytechnic Institute and State University, 2018. URL https://vtechworks.lib.vt.edu/handle/10919/81882.

[14] Y. Dou, H. Li, K. Zeng, J. Liu, Y. Yang, B. Gao, and K. Ren. Preserving incumbent users' privacy in server-driven dynamic spectrum access systems. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 729–730, June 2016. doi: 10.1109/ICDCS.2016.40.

[15] A. Gkoulalas-Divanis, P. Kalnis, and V. S. Verykios. Providing k-anonymity in location based services. *SIGKDD Explor. Newsl.*, 12(1):3–10, November 2010. ISSN 1931-0145. doi: 10.1145/1882471.1882473. URL http://doi.acm.org/10.1145/1882471.1882473.

[16] F. Hessar and S. Roy. Capacity considerations for secondary networks in tv white space. 14, 03 2013.

[17] S. Kasampalis, P. Lazaridis, Z. Zaharis, A. Bizopoulos, S. Zettas, and J. Cosmas. Comparison of longley-rice, itm and itwom propagation models for dtv and fm broadcasting. In *16th International Symposium on Wireless Personal Multimedia Communications (WPMC 2013)*, 06 2013.

[18] I. Keltz. Spectrum issues recent rulemakings. https://transition.fcc.gov/bureaus/oet/ea/presentations/files/oct15/2-TCB-Rulemaking-UpdateOctober-2015.pdf.

[19] H. Li, X. Cheng, K. Li, C. Hu, N. Zhang, and W. Xue. Robust collaborative spectrum sensing schemes for cognitive radio networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(8):2190–2200, Aug 2014. ISSN 1045-9219. doi: 10.1109/TPDS.2013.73.

[20] H. Li, Y. Dou, C. Lu, Y. Yang, J. Park, and D. Zabransky. Preserving incumbent user location privacy in 3.5ghz with environmental sensing capability. 2018.

[21] G. Locke and L. Strickling. An assessment of the near-term viability of accommodating wireless broadband systems in the 1675-1710 mhz, 1755-1780 mhz, 3500-3650 mhz, and 4200-4220 mhz, 4380-4400 mhz bands. Technical report, Federal Communications Commission, October 2010.

[22] B. Lynn. *On the Implementation of Pairing-Based Cryptosystems*. phdthesis, Stanford University, 2007. URL https://crypto.stanford.edu/pbc/thesis.pdf.

[23] J. M. Park, J. H. Reed, A. A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak. Security and enforcement in spectrum sharing. *Proceedings of the IEEE*, 102(3):270–281, March 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2301972.

[24] P. Powell. Shared commercial operations in the 3550-3650 mhz band. Technical Report 2015-14494, Federal Communications Commission, June 2015.

[25] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 08 1998.

[26] R. Shokri, G. Theodorakopoulos, J. Y. Le Boudec, and J. P. Hubaux. Quantifying location privacy. In *2011 IEEE Symposium on Security and Privacy*, pages 247–262, May 2011. doi: 10.1109/SP.2011.18.

[27] J. Vaidya, B. Shafiq, X. Jiang, and L. Ohno-Machado. In *AMIA Joint Summits on Translational Science*, page 262, March 2013.

[28] P. Vaka. Security and performance issues in spectrum sharing between disparate wireless networks. mastersthesis, Virginia Polytechnic Institute and State University, 2017. URL https://vtechworks.lib.vt.edu/handle/10919/77960.

[29] W. Wang and Q. Zhang. *Location Privacy Preservation in Cognitive Radio Networks*. Springer Publishing Company, Incorporated, 2014. ISBN 3319019422, 9783319019420.

[30] L. Zhang, C. Fang, Y. Li, H. Zhu, and M. Dong. Optimal strategies for defending location inference attack in database-driven crns. In *2015 IEEE International Conference on Communications (ICC)*, pages 7640–7645, June 2015. doi: 10.1109/ICC.2015.7249548.

[31] H. Zhou, Y. Wu, S. Zhong, Z. Luo, and X. Wang. Preventing location-based inference attack in location based services. In *2012 International Conference on Industrial Control and Electronics Engineering*, pages 1587–1590, Aug 2012. doi: 10.1109/ICICEE.2012.418.