When Smart TV Meets CRN: Privacy-preserving Fine-grained Spectrum Access

Chaowen Guan*, Aziz Mohaisen*, Zhi Sun†, Lu Su*, Kui Ren* and Yaling Yang‡
*Department of Computer Science and Engineering
University at Buffalo, SUNY, Buffalo, NY 14260
Email: {chaoweng, mohaisen, lusu, kuiren}@buffalo.edu
†Department of Electrical Engineering
University at Buffalo, SUNY, Buffalo, NY 14260
Email: zhisun@buffalo.edu
‡The Bradley Dept. of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061
Email: yyang8@vt.edu

***Abstract*—Dynamic spectrum sharing techniques applied in the UHF TV band have been developed to allow secondary WiFi transmission in areas with active TV users. This technique of dynamically controlling the exclusion zone enables vastly increasing secondary spectrum re-use, compared to the "TV white space" model where TV transmitters determine the exclusion zone and only "idle" channels can be re-purposed. However, in current such dynamic spectrum sharing systems, the sensitive operation parameters of both primary TV users (PUs) and secondary users (SUs) need to be shared with the spectrum database controller (SDC) for the purpose of realizing efficient spectrum allocation. Since such SDC server is not necessarily operated by a trusted third party, those current systems might cause essential threatens to the privacy requirement from both PUs and SUs. To address this privacy issue, this paper proposes a privacy-preserving spectrum sharing system between PUs and SUs, which realizes the spectrum allocation decision process using efficient multi-party computation (MPC) technique. In this design, the SDC only performs secure computation over encrypted input from PUs and SUs such that none of the PU or SU operation parameters will be revealed to SDC. The evaluation of its performance illustrates that our proposed system based on efficient MPC techniques can perform dynamic spectrum allocation process between PUs and SUs efficiently while preserving users' privacy.**

I. Introduction

Today's TV white space model of spectrum sharing in the ultra high frequency (UHF) TV band allows regionally unused channels to be repurposed for unlicensed-style secondary access. When a geographical region has no primary broadcaster on a particular channel, that channel is said to be "TV white space (TVWS)," which is then made available for transmission by secondary users under current regulatory frameworks, including those in the U.S. [1] and the U.K. [27]. Unfortunately, the large number of over-the-air TV broadcasters in many populated areas yields extremely limited white space availability [26]. Nonetheless, the number of viewers watching TV via UHF is dwarfed in practice by those watching TV via satellite or cable. Furthermore, recent data shows the severe under-utilization of this spectrum, with vast regions in the range of TV transmitters having no active TV receivers on multiple channels even at peak TV viewing times. To alleviate this under-utilization of UHF, Zhang and Knightly proposed WATCH [36] (for Wifi in Active TV CHannels), a system to enable secondary WiFi transmission in active TV channels while simultaneously protecting active primary TV receivers. In contrast to previous TVWS models which calculate exclusion zones, in which secondary transmissions are not allowed and transmit power is set to zero based on transmitting TV channels and their corresponding tower locations [1], WATCH introduced a dynamically computed exclusion zone characterized as the union of locations where secondary user transmit power must be reduced in order to protect active TV receivers.

A major concern in the above dynamic spectrum sharing system between primary TV receivers (aka primary users, or PUs) and secondary users (SUs) is privacy: TV receivers might not want others to know what channels they are receiving, where the operation data can be sensitive (§III-D). Similarly, SUs' operational parameters (viz. antenna height, transmit power, etc.) may also be sensitive operator data, since SU's location can be derived from those parameters when combined with public terrain knowledge. To realize efficient dynamic spectrum access, WATCH requires PUs and SUs to send their operation data (receiver channel from PU and location and antenna height from SU) to one central Spectrum Database Controller (SDC) for spectrum allocation, which in turn exposes the PUs and SUs to a potential privacy violation. For example, this central SDC system is not necessarily trustworthy, since it may be operated by untrusted third parties. Furthermore, even when the operator of the SDC system is trusted, the SDC may be breached by adversaries (viz. insiders and outsiders) that are becoming prevalent today. In such scenario, an adversary will have access to all PU and SU operation information. How to protect PU and SU operation privacy from SDC (and other potential adversaries) becomes a central challenge that we believe might deter the wide adoption of this dynamic

spectrum sharing technology.

To address this privacy issue, one can turn to the general secure multiparty computation (MPC) protocols or fully homomorphic encryption (FHE) schemes. While these approaches are appealing, they are still far from practical for most real-world applications and special solutions should be developed for special cases for efficiency, as Goldreich pointed out in [22]. Thus, one may try to simply apply much more efficient *somewhat homomorphic encryption* (SWHE) scheme. However, it is a nontrivial task to employ existing SWHE schemes in an efficient and secure manner. For instance, encrypting every single data that will be transmitted through the connection using those schemes will cause unacceptable delay, since SU needs to encrypt the data every time it wants to request access to the WiFi transmission. Another feature of the dynamic sharing system that makes applying SWHE schemes nontrivial lies in the required numeric comparisons between different ciphertexts. One may simply use a SWHE scheme that supports secure subtraction over ciphertexts, but the SDC still cannot know the comparison result due to the semantic security of a secure SWHE scheme, and thus SDC cannot fulfill the spectrum allocation. Allowing SDC to learn the comparison result, on the other hand, will contradict the desired security goal (i.e., semantic security).

To this end, we fundamentally address this privacy issue by designing a privacy-preserving spectrum sharing system. We design and demonstrate PISA, a system that addresses the privacy between TV signal and WiFi Transmission. The proposed system can perform dynamic spectrum allocation between TV receivers and SUs while preserving users' privacy. Our scheme is based on an efficient SWHE scheme, and it guarantees that no snooping entities, including the SDC itself, can obtain any sensitive information about PU and SU operation data during the dynamic spectrum allocation process.

Contributions. Our contributions can be summarized as follows:

1) We outline a critical privacy issue with the existing state-of-the-art system, WATCH, and motivate for a privacy-preserving spectrum allocation.
2) We propose PISA, a system to address this privacy issue. PISA utilizes an efficient somewhat homomorphic encryption based on the Paillier cryptosystem.
3) We provide an extensive evaluation, using experiments using a prototype, to demonstrate that PISA meets its requirements and goals. Compared to the generic fully homomorphic encryption techniques, PISA provides privacy-preserving spectrum allocation at a practical cost.

Organization. In section II we review related work. In section III we formalize the problem statement. In section IV we introduce the protocol description. In section VI we evaluate PISA. In section VII we sum up with concluding remarks and outline various directions of future work.

II. Related Work

Privacy-preserving Spectrum Sharing. To protect SU's location privacy against untrusted spectrum access system (SAS), Gao *et al.* [17] used private information retrieval (PIR), a cryptographic primitive that allows a user to retrieve records from a database without revealing which records are retrieved. Their main idea is to divide the service area of SAS into grids where each grid's spectrum availability information is maintained by SAS [17]. An SU then uses PIR to retrieve the spectrum availability information of its grid from the database. However, Gao *et al.* only considered the protection of SU's location and did not address PU's privacy issue. Unlike TV white space where the PU's operation data (e.g., TV location) can be public and needs no protection, in many other scenarios involving federal-commercial (public-private) sharing the PU's privacy is a more critical concern.

Bahrak *et al.* [7] identified a novel attack on PU's operation privacy utilizing malicious SUs in SAS. Their idea is that a malicious SU can determine the types and locations of a PU in a given region of interest by sending seemingly innocuous queries to SAS. To counter the attack, an obfuscation technique is used to hide information revealed by SAS, which leads to a certain level of privacy assurance.

Computing on Encrypted Data. How to outsource computations while preserving their privacy is a question that has been raised frequently. Standard solutions rely on encrypting data, which would perfectly solve the privacy issue. However, requirements for standard encryption schemes disallow most functionalities: we cannot perform any computations on the encrypted data. A solution to this is *homomorphic encryption* (HE), first introduced by Rivest, Adleman and Dertouzos in [31]. With HE, the untrusted server can carry out computations on ciphertext and generate an encrypted result which, when decrypted, matches the result of operations performed on the plaintext.

HE schemes that allow simple computations on encrypted data are known for a long time, including [23], [15], [30], [9]. Until recently, obtaining HE schemes that supports any functionality, viz. FHE, was a major open problem. In a breakthrough work, Gentry [19] gave the first construction of an FHE scheme. However, his construction was very complex. Since then FHE has been an area of very active research, and there has been much informal discussion in the industry as to whether FHE is implementable and practical. While the initial solution may not have been practical, subsequent developments produced considerably simpler FHE schemes [34], [32], [11], [20], [10] Unfortunately, the computation overhead incurred in those general constructions is still far from practical. This, in turn, resulted in excluding general FHE from adoption for many problems [35] and sparked interests in special case solutions for efficiency

reasons. The scenario being considered in this paper is one of those cases.

III. Preliminaries

We introduce WATCH (§III-A), the design goals (§III-B), system model (§III-C), and the scope of private and public data used for enhancing the performance of our scheme (§III-D).

A. The WATCH System

Settings. WATCH [36] involves three entities: SDC, primary (active) TV receiver (PU) and secondary transmitter (SU). The SDC serves as the coordinator between SUs and active PUs to ensure the secondary transmissions are not interfering with active PUs. In particular, WATCH does not explicitly disallow secondary transmissions in certain areas. Instead, it divides the region into blocks and computes the maximum SU Effective Isotropic Radiation Power (EIRP) for each block. Secondary transmission requests will be disallowed only in blocks where the maximum SU EIRP is zero. With the public knowledge of the information of transmit power and location of TV transmitters, WATCH further requires that the SDC also collects the location and channel reception of active PUs.

WATCH dynamically controls and updates the interference threshold of the channel used by the TV receiver. Hereafter we denote the minimum required TV signal strength by $S^{PU}_{sv\_min}$, TV signal signal-to-interference-plus-noise ratio (SINR) by Δ_{TV_SINR}, the path-loss of secondary signals by $h(\cdot)$, and the maximum path-loss over a certain distance by $h_{max}(\cdot)$.

Receiving a Certain Channel. Whenever a TV receiver i becomes active in channel c, the SDC performs the following:

1) Compute the distance d^c within which the SU's EIRP needs to be updated as follows:

$$\Delta_{TV\_SINR} + \Delta_{redn} = \frac{S^{PU}_{sv\_min}}{S^{SU}_{max} \cdot h_{max}(d^c)}, \quad (1)$$

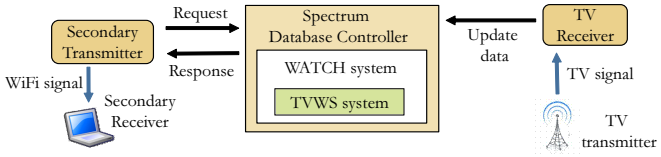
where an additional Δ_{redn} is added to represent the aggregate interference from multiple SU's.

2) Update the maximum SU EIRP $S^{SU}_{c,j}$ for c to ensure that

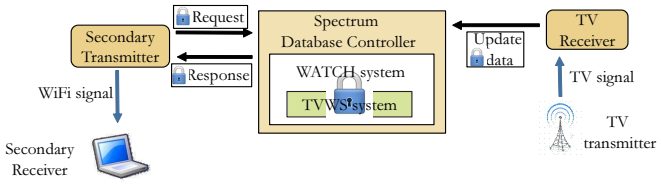
$$S^{SU}_{c,j} \leq \frac{S^{PU}_{c,i}}{(\Delta_{TV\_SINR} + \Delta_{redn}) \cdot h(d^c_{i,j})}, \quad (2)$$

where each block j is within d^c from i and $S^{PU}_{c,i}$ is the mean TV signal strength at receiver i in channel c, which is computed by the L-R irregular terrain model [29].

Note that d^c is only related to the channel and the maximum SU EIRP is limited to S^{SU}_{max}. Per [36], $S^{PU}_{sv\_min}$ and Δ_{TV_SINR} within the TV service area can be obtained from the legacy standards, e.g., the ATSC DTV standard [2].



(a) WATCH system overview



(b) Privacy-Preserving WATCH

Figure 1: WATCH and Privacy-Preserving WATCH's designs.

Switching. When a receiver is turned off or switched to another channel, all $S^{SU}_{c,j}$ within distance d^c are updated either to a larger value restricted by another active TV receiver or to S^{SU}_{max}; the latter happens if all receivers within d^c are switched to other than c or turned off. All SU's are required to provide their information to the SDC in order to acquire the transmission parameters, as in the traditional TVWS systems.

Zhang and Knightly [36] also designed WATCH-IC and CAT, which are out of the scope of our paper, and the interested reader may refer to [36] for further details.

As shown in Figure 1a, the primary TV receivers (PUs) first perform initializing and update the SDC with their operation data, such as locations and interference sensitivity thresholds. Then, any SU that needs the spectrum must send SDC a request for spectrum access along with its operation data, including its location and antenna heights. Upon receiving those operation data, SDC computes whether the SU interference to any PU will exceed the PU's precomputed interference sensitivity threshold when this SU starts operating: the SDC computes (1) and (2) and denies SU's transmission request if (2) does not hold; it grants transmission otherwise.

B. Design Goal and Threat Model

To realize efficient spectrum sharing, WATCH requires PUs and SUs to send their operation data to SDC for spectrum allocation, which exposes PUs and SUs to privacy violation. The SDC is not necessarily trustworthy as it may be operated by untrusted third parties, giving an adversary access to all PU and SU operation information by controlling SDC.

In our threat model, we assume that SDC is not fully trusted by both PUs and SUs. Thus, the goal of our system is to realize the same function as WATCH while protecting

Parameters: Let public key be (n, g), and secret key be (λ, μ).
Encryption: 1. Let m be a message to be encrypted where $m \in \mathbb{Z}_n$. 2. Select random $r \in \mathbb{Z}_n^*$. 3. Encrypt m and by computing $\widetilde{m} = E(m, r) = g^m \cdot r^n \bmod n^2$.
Homomorphic Properties: Addition: $D(\widetilde{m_1} \oplus \widetilde{m_2}) = D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n$ Subtraction: $D(\widetilde{m_1} \ominus \widetilde{m_2}) = D(E(m_1, r_1) \cdot E(m_2, r_2)^{-1} \bmod n^2) = m_1 - m_2 \bmod n$ Scalar Multiplication: $D(m_2 \otimes \widetilde{m_1}) = D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 \cdot m_2 \bmod n$

Figure 2: Paillier cryptosystem

the data privacy of both the PUs and SUs. We assume a threat model where the SDC is honest-but-curious (i.e. semi-trusted), which means that all parties in the system exactly follow PISA's design steps but may attempt to infer private operation data of PUs and SUs from the information communicated to them. More formally, the goals that PISA is designed to protect the privacy of:

- PU's channel reception information.
- SU's operational parameters (viz. location and EIRP) and the decision on transmission requests from SDC.

To design such a system, we employ the homomorphic properties of Paillier cryptosystem [30] (c.f. Figure 2 for a brief description). We separate the sensitive parameters that need privacy protection from what can be publicly revealed. Only sensitive private parameters are encrypted while other computations can be done on data of plaintext form, which provides significant computation and communication improvements. Then, we decompose the computation on private parameters into a set of basic arithmetic operations that can be homomorphically computed. By doing that, all private inputs to the system from PU and SU are in ciphertext form such that they cannot be decrypted by the SDC or any other snooping identity without valid decryption keys.

C. System Model and Operation

Entities. Our system includes four entities (Figure 3): SDC for computing spectrum allocation, PUs and SUs in the service area of SDC, and a Semi-trusted Third Party (STP).

In PISA, STP creates a global Paillier public/private key pair (pk_G, sk_G). The STP is trusted for keeping sk_G as a secret only known to itself. Each SU i has her own pair of Paillier public/private keys (pk_i, sk_i) and uploads pk_i to STP. Using pk_i and sk_G, the STP can provide key conversion service to SDC Server. Specifically, the STP helps SDC convert a ciphertext encrypted by pk_G to another ciphertext encrypting the same plain message over pk_i so that the resulting ciphertext is decrypted by SU i. Anyone can retrieve pk_G and SU Paillier public keys (e.g. pk_i for SU i) from the STP.

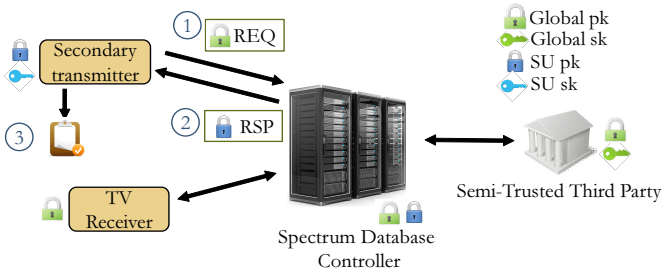


Figure 3: secure WATCH overview

System operation. All PUs will update the SDC server with their operation information every time they change channels. To ensure privacy, PUs encrypt their operation information using pk_G before sending the update to SDC. Then, when an SU i needs to obtain the spectrum access right, our proposed system works as follows (also as shown in Figure 3):

1) SU i prepares and transmits a request for the secondary spectrum access to SDC server by including the ciphertext of its operation information encrypted using pk_G.
2) Upon receiving SU i's request for a license (transmission permission), the SDC server and STP perform a secure computation over the encrypted SU and PU information to determine whether the interference on any PU exceeds a desirable threshold when the requesting SU i is allowed to transmit. A response is generated and sent to SU i.
3) The response includes a ciphertext encrypted by SU i's own public key pk_i. After getting the response, SU i decrypts the response message and should be able to learn if its request for WiFi transmission is permitted or not.

In 2) above, the SDC utilizes the key conversion capability of the STP to generate a response to SU i based on the private spectrum computation results. Since SDC Server does not know sk_G, both the PU and SU operation information and the secure computation results stay encrypted and remain private during the computation. In 3), the SU i can also obtain a properly digitally signed secondary transmission license from the decrypted response message when permission is granted. The license will contain the SU i's operation parameter specification and will be properly signed with a digital signature for preventing potential forging or tampering attempts. Furthermore, only the owner SU i of a valid decryption key sk_i is able to decrypt the response message to obtain the license. In the entire process, all private PU and SU operation data and the intermediate and final computation results stay in ciphertext format and are never exposed to any other part of this proposed system, including the SDC server and the STP. The design mainly leverages the

homomorphic properties of the Paillier cryptosystem [30].

D. Private Input Data and Public Data

The main facilitator of our efficiency is the separation of data into public and private, which we discuss below.

Private data. To prevent the aggregate SU interference from harming PUs, operation data of PUs and SUs need to be provided to the system to predict received signal strength from SUs to PUs, which is then compared with PUs interference thresholds, i.e., (2). From SU, the location, transmitter power, antenna gain, and line loss information are required. From PU, the channel reception information and interference threshold are required as private data.

Public data. Other parameters are set as pre-determined values from regulation, standards, and pre-computation and hence need no privacy protection. Also, terrain information is public knowledge that is easily found on government terrain database like [3], [33]. The location of a TV receiver is usually fixed, and thus is public; as pointed out in [36], registration of TV location is already required in countries like Norway [16].

Input format and representation. Per [36], we quantize the service area of the SDC server into B small blocks. Since PU and SU have different private input data to the system, we pre-process them in two different ways. For PU, we quantize the channel reception information into C slots. For SU, we quantize its transmitter power PT, antenna gain GA and line-loss LS, and compute $EIRP = PT + GA - LS$.

Using this quantization, an PU i's private input data is represented by a two-dimensional matrix $\mathbf{T}_i := \{t_i(c,i)\}_{C\times B}$. When the PU receiver i is located in geographic block i and switches to channel c , $t_i(c,i)$ equals an integer representation of the mean TV signal strength in mW at the TV receiver i in channel c. Otherwise, $t_i(c,i)$ is set to 0. For SU j's private operation data, we use a two-dimensional matrix $\mathbf{S}_j = \{s_j(c,b)\}_{C\times B}$. If SU j is located in block b, $s_j(c,b)$ equals SU j's EIRP $S^{SU}_{c,j}$ for channel c in integer form (e.g. in the unit of mW). Otherwise, $s_j(c,b)$ equals 0, indicating no active transmission in such a configuration.

To ensure PU and SU data privacy, each entry of $\mathbf{T}_i$ and $\mathbf{S}_j$ are encrypted by pk_G before sending them to SDC. For notation simplicity, given any plaintext m, we denote its ciphertext created using pk_G by $\widetilde{m}$. $\widetilde{\mathbf{M}}$ denotes the encryption of the matrix $\mathbf{M}$ by pk_G. Hence, PU i submits $\widetilde{\mathbf{T}}_i$ to the SDC server in its update message and SU j transmits $\widetilde{\mathbf{S}}_j$ to the SDC server in its request for secondary access.

IV. Protocol Design

For secure computations at the SDC server side, we first outline the computation steps in the plaintext domain (§IV-A) then show how to realize these computation steps homomorphically in the ciphertext domain to preserve privacy (§IV-B). We conclude the section with the final system operation (§IV-B).

A. Spectrum Computation in Plaintext

The spectrum computation in the plaintext form consists of initialization, update from the PU, and request for transmission from SU. In the following, we describe each of those steps.

1) Initialization: In this step, the SDC server precomputes the maximum SU EIRP e for each block with the information of transmit power and location of TV transmitters and the location and channel reception of active TV receivers using some pathloss model (such as the Extended Hata sub-urban model [5]), and define $\mathbf{E} = \{E_S(c,b)\}_{B\times C}$ where $E_S(c,b)$ is the maximum SU EIRP for block b and channel c.

2) Update from PU: Every time a PU receiver is turned off or switched to another channel, the SDC server aggregates all PU inputs to create $\mathbf{T}'$ as

$$\mathbf{T}' = \sum_{i\in \text{ all } PUs} \mathbf{T}_i, \tag{3}$$

where $T'(c,b) := \sum_{i\in \text{ all } PUs} T_i(c,b), \forall\ (c,b) \in (C,B)$. Note that if a PU receiving channel c exists in block i, $T_i(c,b)$ equals the mean TV signal strength $S^{PU}_{c,i}$ at TV receiver i in channel c; otherwise, $T_i(c,b) = 0$. For notation simplicity, here we can assume only one PU in each block. For the case having multiple PUs in one block, we can simply create one $T_i(c,b)$ entry for each PU. Since a block is normally of size $10m \times 10m$ (as pointed out in [36]), the number of PUs in one block would be just a small constant. The SDC server then initializes an interference budget matrix $\mathbf{N} := \{N(b,c)\}_{B\times C}, \forall (b,c) \in (B,C)$ as

$$N(c,b) = \begin{cases} T'(c,b), & \text{if } T'(c,b) \neq 0 \\ E_S(c,b), & \text{if } T'(c,b) = 0 \end{cases}, \tag{4}$$

3) Transmission Request from SU: Whenever SU j needs a WiFi permission, it prepares a request by computing

$$F_j(c,i) = S^{SU}_{c,j} \cdot h(d^c_{i,j}), \tag{5}$$

where $d^c_{i,j}$ is the distance between PU in block i and SU in block j. Since the TV receiver i's location is fixed and registered, $d^c_{i,j}$ can be pre-computed.

Upon receiving a transmission request from SU j, the SDC server computes the interference that SU j imposes on the PU receiving c in block i as:

$$R_j(c,i) = F_j(c,i) \cdot (\Delta_{TV\_SINR} + \Delta_{redn}), \tag{6}$$

for all PU i within distance d^c from SU, where d^c is derived from (1). With matrix $\mathbf{R}_j = \{R_j(c,i)\}_{C\times B}$, SDC then subtracts SU j's interference from $\mathbf{N}$ to create an interference indicator matrix $\mathbf{I}_j$ as

$$\mathbf{I}_j = \mathbf{N} - \mathbf{R}_j. \tag{7}$$

Decision on transmission request. Note that in the original WATCH system, SDC will only simply send out a notification, instead of an actual transmission permission license,

since that original work did not consider security issues. To protect SU's privacy, a system should as well prevent SDC from knowing the decision on SU's transmission request. To do that, we introduce an actual transmission permission license, which is typically defined as a digital signature. This license will be encrypted and sent out by SDC whether SU is granted permission or not, but SU can retrieve the valid permission license only when it is allowed to transmit (c.f. §IV-B for more details). Therefore, the decision on SU's transmission request will be made according to the following:

- Some entries in $\mathbf{I}_j$ are less than or equal to 0. In this case, the interference budget for some PU with configuration (c, i) is exceeded. Thus, the SDC server denies SU j's WiFi transmission request and will not return a valid transmission permission license.
- All entries in $\mathbf{I}_j$ are greater than 0. In this case, all channel receptions of PUs are safe. Thus, the SDC server permits SU j's WiFi transmission request and returns a valid transmission license.

We can see that, regardless of whether SU j receives a valid transmission permission license or not, the interference budgets stay the same because the situation of multiple SUs is handled by the value Δ_{redn}. In such a manner, the system will keep updating the parameters, and the feedback loop ensures that the PUs are finally protected and $\mathbf{N}$ becomes stable (c.f. [36] for more details). Note that the initialization step does not require any private input data. Hence, it can be carried out in the plaintext domain. For the other steps, the computations will directly or indirectly use the private input data of PUs and SUs. Therefore, as shown in §IV-B they need to be carried out over encrypted data.

B. The Proposed System

In this section, we will give the details of how to carry out the computation steps from §IV-A in a privacy-preserving way. A systematic description of how the protocol works in one round as a protocol is illustrated by Figure 4 and Figure 5. Recall that in this system, the location of TV receiver is part of public knowledge. Figure 4 gives the steps which TV receiver i takes to inform SDC when it is switching to channel c. Figure 5 shows how a transmission request from SU will be processed among SU j, SDC and STP. Those formulas referred in the two figures will be explained below.

Update from PU. (see Figure 4) Given all PUs' private input t_i, formula (3) can be realized over ciphertexts by straightforward homomorphic addition as

$$\widetilde{\mathbf{T}'} = \oplus_{i\in \text{ all } PUs} \widetilde{\mathbf{T}_i}, \tag{8}$$

where $\mathbf{T}_i := \{T_i(c,i)\}_{C\times B}$ and $\oplus_{i\in \text{ all } PUs}$ is the homomorphic version of $\sum_{i\in \text{ all } PUs}$.

From §IV-A2, this matrix $\mathbf{T}'$ determines the encrypted version of matrix defined by formula (4). However, to realize

TV receiver i:
Assume that it is receiving channel c in block i,
1) Set $T(c,i) = S^{PU}_{c,i}$, $T(k,i) = 0, \forall k = 1, \cdots, C, \quad k \neq c$.
2) Encrypt $T(k,i), \forall k = 1, \cdots, C$ using public key pk_G.
3) Send SDC the generated ciphertexts $\widetilde{T}(1,i), \cdots, \widetilde{T}(C,i)$.

SDC:
Upon receiving $\{\widetilde{T}(1,i), \cdots, \widetilde{T}(C,i)\}$ from receiver i,
4) Compute (8), (9), (10) to get the encrypted interference budget matrix $\widetilde{\mathbf{N}}$.

Figure 4: Channel Reception Update

SU j:
1) Compute (5) and then encrypt to get $\widetilde{\mathbf{F}_j} = \{F_j(c,i)\}_{C\times B}$.
2) Send $\widetilde{\mathbf{F}_j}$ to SDC.

SDC: Upon receiving $\widetilde{\mathbf{F}_j}$ from SU j,
3) Calculate (11) using homomorphic scalar multiplication to get $\widetilde{\mathbf{R}_j} = \{R_j(c,i)\}_{C\times B}$.
4) Compute (12) to get the encrypted interference indicator matrix $\widetilde{\mathbf{I}_j}$.
5) Input matrix $\widetilde{\mathbf{I}_j}$ to (14) to obtain $\widetilde{\mathbf{V}_j} = \{V_j(c,i)\}_{C\times B}$, and then forward the matrix $\widetilde{\mathbf{V}_j}$ to STP.

STP: Upon receiving $\widetilde{\mathbf{V}_j}$ from SDC,
6) Decrypt each $V_j(c,i)$ and set new value $X_j(c,i)$ as defined by formula (15).
7) Encrypt each $X_j(c,i)$ using SU j's public key to get $\widetilde{X_j}^{pk_j}(c,i)$.
8) Define a matrix $\widetilde{\mathbf{X}_j} = \{\widetilde{X_j}^{pk_j}(c,i)\}_{C\times B}$ and send this matrix back to SDC.

SDC: Upon receiving $\widetilde{\mathbf{X}_j}$ from SDC,
9) Compute (16) to derive $\widetilde{Q_j}^{pk_j}(c,i)$ which is ensured to satisfy (13).
10) Generate a signature SG_j and then encrypt it using SU j's public key pk_j to get $\widetilde{SG_j}^{pk_j}$.
11) Calculate formula (17) to obtain $\widetilde{G_j}^{pk_j}$ which is a ciphertext of a valid signature when SU j is allowed for transmission, and then forward $\widetilde{G_j}^{pk_j}$ to SU j.

Figure 5: Transmission Permission Request

formula (4) is not trivial since determining the equality of $T'(c,b)$ and 0 in ciphertext domain is a tricky integer comparison problem. Some of the existing methods [13], [12], [18] require the involved integers to be encrypted bit by bit. Consequently, this will make the rest computations involving $T'(c,b)$ extremely complex and time-consuming. (Those methods will also need multiple rounds of communications, which is not desirable in this context.) We completely avoid the overhead of secure integer comparison by utilizing the following method.

First, we adjust the PU's way of computing its $T_i(c,i)$, assuming $T_i(c,i)$ is a positive integer. If the PU in block i is

receiving channel c, we create $W_i(c,i) = T_i(c,i) - E_S(c,i)$. The other entries of $W_i(c,i)$ are set to 0. The PU submits $\widetilde{W_i} = \{W_i(c,i)\}_{C\times B}$ to the SDC server. Then, in the SDC server we compute the following instead of formula (7)

$$\widetilde{\mathbf{W'}} = \oplus_{i\in \text{ all } PUs} \widetilde{\mathbf{W}_i}. \quad (9)$$

The SDC sever then encrypts the matrix $\mathbf{E} = \{E_S(c,i)\}_{C\times B}$ (defined in §IV-A1) to obtain $\widetilde{\mathbf{E}}$. Note that in this matrix, each entry $E_S(c,i)$ is the maximum SU EIRP for block b and channel c. Finally, we homomorphically compute:

$$\widetilde{\mathbf{N}} = \oplus_{i\in \text{ all } PUs} \widetilde{\mathbf{W'}} \oplus \widetilde{\mathbf{E}}. \quad (10)$$

Accordingly, we can use (9) and (10) to realize (4) without the need to perform any secure integer comparison.

Transmission Request from SU. (Figure 5) Whenever an SU j wants to transmit data using WiFi, it prepares and encrypts its request; besides computing (5), SU j also needs to encrypt the resulting values. Then formula (6) can be easily realized in ciphertext domain by Paillier's homomorphic operations:

$$\begin{aligned} \widetilde{\mathbf{R}_j} &= \{\widetilde{R_j}(c,i)\}_{C\times B} \\ &= \{\widetilde{F}(c,i) \otimes X\}_{C\times B}), \quad (11) \\ \widetilde{\mathbf{I}_j} &= \widetilde{\mathbf{N}} \ominus \widetilde{\mathbf{R}_j}, \quad (12) \end{aligned}$$

where, from (5), $X = \Delta_{TV\_SINR} + \Delta_{redn}$.

However, to decide if an SU is allowed to transmit is not a simple task when using encrypted information provided by SU. First, one has to find out the sign of all entries in $\widetilde{\mathbf{I}_j}$, which requires secure integer comparisons. Second, if $\widetilde{\mathbf{I}_j}$ indicates that an SU will not cause significant interference to any TV receiver, we have to generate a valid transmission permission license for this SU. This requires specifying the operation parameters of SU and the permission license to be properly digitally signed to prevent tampering attempts. Also, to ensure privacy of SU, whether the SU obtains the license or not, the specification of SU's operation parameters in the license should also be kept private. We solve these challenges by a two-step approach using the key conversion in STP as follows:

• **Step (1).** SDC in this step creates a ciphertext $\widetilde{\mathbf{Q}_j}^{pk_j} = \{\widetilde{Q_j}^{pk_j}(c,i)\}_{C\times B}$, whose plaintext has the property:

$$Q_j(c,i) := \begin{cases} 0, & \text{if } I_j(c,i) > 0 \\ -2, & \text{if } I_j(c,i) \le 0 \end{cases}, \quad (13)$$

Here $Q_j^{pk_j}(c,i)$ is encrypted by the individual Paillier public key pk_j of SU j, instead of the group public key. The key conversion is realized by leveraging STP, who possesses the group private key sk_G and can decrypt any message encrypted by pk_G. Specifically, the SDC server first sends the following ciphertext for all pairs (c,i) to the STP:

$$\widetilde{V_j}(c,i) := \epsilon(c,i) \otimes \left[\left[\alpha(c,i) \otimes \widetilde{I_j}(c,i)\right] \ominus \widetilde{\beta}(c,i)\right]. \quad (14)$$

In (14) $\alpha(c,i), \beta(c,i)$ are one-time large positive random integers generated for each pair (c,i), and $\alpha(c,i) > \beta(c,i)$ holds for any (c,i). $\epsilon(c,i)$ is randomly selected in {-1, 1}. $\alpha(c,i)$ and $\beta(c,i)$ are used for hiding the sign of $I_j(c,i)$ when $V_j(c,i)$ is exposed in the decryption phase. It is not hard to see that the plaintext of $\epsilon(c,i) \otimes \widetilde{V_j}(c,i)$ has the same sign as the plaintext of $\widetilde{I_j}(c,i)$, yet it is difficult for the STP to know the value and sign of $\widetilde{I_j}(c,i)$'s plaintext by only knowing $\widetilde{V_j}(c,i)$. Thus, the operation in (11) prevents any potential leakage of spectrum allocation information to the STP with the security guarantee provided by the underlying Paillier cryptosystem.

The STP decrypts $\widetilde{V_j}(c,i)$ and creates $X_j(c,i)$ as follows

$$X_j(c,i) := \begin{cases} 1, & \text{if } V_j(c,i) > 0 \\ -1, & \text{if } V_j(c,i) \le 0 \end{cases}. \quad (15)$$

The STP then encrypts $X_j(c,i)$ using SU j's individual Paillier public key pk_j to create $\widetilde{X_j}^{pk_j}(c,i)$ and sends back $\widetilde{\mathbf{X}_j} = \{\widetilde{X_j}^{pk_j}(c,i)\}_{C\times B}$. Note that since only SU j knows the secret key sk_j, the SDC server cannot derive the plaintext of $\widetilde{\mathbf{X}_j}$.

Upon receiving $\widetilde{\mathbf{X}_j}$, the SDC server then computes

$$\widetilde{Q_j}^{pk_j}(c,i) := \left[\epsilon(c,i) \otimes \widetilde{X_j}^{pk_j}(c,i)\right] \ominus \widetilde{1}^{pk_j}(c,i), \quad (16)$$

where $\epsilon(c,i)$ is the one used in (14). According to (14) and (15), we can see that the plaintext of $\widetilde{Q_j}^{pk_j}(c,i)$ satisfies (13).

• **Step (2).** In this step, we grant or deny SU j's request for transmission by homomorphic computation using $\widetilde{\mathbf{Q}_j}^{pk_j} = \{\widetilde{Q_j}^{pk_j}(c,i)\}_{C\times B}$. Concretely, the SDC server first creates a license for SU j to transmit. The license includes the identity of SU j, the identity of license issuer (e.g., the SDC server), and $\widetilde{\mathbf{S}_j}$, which is the ciphertext of SU j's operation parameters that are submitted in its transmission request. The SDC server uses a typical digital signature algorithm (e.g., RSA, DSA, etc.) to generate a signature SG_j of the license. Using the SU j's public key pk_j, the SDC server encrypts SG_j as $\widetilde{SG_j}^{pk_j}$. It then computes

$$\widetilde{G_j}^{pk_j} := \widetilde{SG_j}^{pk_j} \oplus \left[\eta \otimes \left(\oplus_{c,i} \widetilde{Q_j}^{pk_j}(c,i)\right)\right], \quad (17)$$

where η is a one-time large random integer. In essence, $\widetilde{G_j}^{pk_j}$ holds the ciphertext of the valid license signature SG_j when all $Q_j(c,i)$ are 0. From (13), this case can only happen when all $I_j(c,i)$ are positive numbers, indicating that all channel receptions of PUs are safe. If any of the $Q_j(c,i)$ is not 0, SU j should not be granted to transmit since some PU's interference threshold will be exceeded. In this case, (17) makes sure that G_j becomes an invalid signature $SG_j + \eta'$, where η' is some random number related to η.

The SDC server then sends the transmission permission license along with $\widetilde{G_j}^{pk_j}$ back to SU j in response to j's transmission request. Upon receiving $\widetilde{G_j}^{pk_j}$, SU j decrypts to obtain G_j. If SU j attains a valid signature (i.e. when $G_j = SG_j$), SU j knows that it can perform WiFi transmission. Otherwise, SU j's transmission request is denied. The above approach ensures that SU j can obtain a properly signed transmission permission license only when its operation does not disturb PUs. Also due to the unforgeability of a secure digital signature scheme and the bijective property (Lemma 3 in [30]) of Paillier cryptosystem, a dishonest SU cannot forge a valid license.

V. Security Analysis

[30] proved that Paillier cryptosystem is semantically secure in Theorem 15. Our proposed system achieves the same security level as the underlying Paillier cryptosystem does. This can be summarized in a more formal language as follows.

Lemma V.1. *Assuming that 1) Paillier cryptosystem is semantically secure, 2) random blinding factors are properly generated, and 3) the semi-honest SDC and STP are non-colluding, the proposed system PISA executes the WATCH system in a privacy-preserving way.*

Proof: It can be proved using the composition theorem [19] under the semi-honest model by analyzing the security of each step in PISA system. Note that all the computations done by SDC are performed over ciphertexts. This means, if STP is not colluding with SDC, then SDC cannot learn any information of the private SU operational parameters or the PU's channel reception, due to the semantic security of Paillier cryptosystem [30]. In the key conversion step, even though STP can obtain the underlying plaintext $V_j(c,i)$ (in 15), the privacy is still be perserved as long as knowing $V_j(c,i)$ only gives STP negligible advantage in distinguishing between $I_j(c,i)$'s and random guesses [28]. This can be achieved by utilizing proper random blinding factors $\epsilon(c,i), \alpha(c,i)$, and $\beta(c,i)$. With those random factors, we can obfuscate the true value and sign of $I_j(c,i)$. ■

VI. Evaluation and Experiment

We introduce the evaluation of PISA using both simulation and real-world experiments based on a prototype. As for the performance of the original WATCH system, please refer to [36].

A. Simulation Results

Evaluation criteria. To evaluate PISA, we use computation and communication overhead as criteria. We evaluate the initialization and request processing on the SDC server upon receiving a WiFi transmission request from an SU, and the time and storage consumption of the Paillier cryptosystem.

Settings. Table I shows the evaluation settings. We use 60-bit integer representation, which satisfies FCC regulation [1], [4] and SPLAT [6]. We implement a software prototype of PISA using the GMP library [24] of arbitrary precision integer. NIST's recommendations [8] indicate a security level of 112 bits, which is achieved by setting n to 2048 bits. We compile our code on a Dell desktop with Intel Core i5-2400 CPU running at 3.10GHz with 4GB of RAM.

Table I: Parameter Settings

Number of PUs	100
Number of blocks	600
Number of channels	100
Bit length of integer representation	60

Benchmark. Table II shows the benchmark of Paillier cryptosystem in the aforementioned settings, measuring the operation time (average of 30 iterations) and storage. We found that the total time for preparing a transmission request by SU is about 4 mins, which can be pre-computed offline for efficiency. We also notice $\approx$ 3 mins for processing this request. We note that the SDC is simulated using an off-the-shelf desktop, without any optimizations. However, in reality, an SDC would normally utilize a much more powerful hardware and can process the transmission request much faster. Thus, to request a transmission permission from SDC, SU just needs to spend a reasonably small amount of time (here $\approx$ 7 mins) to prepare and wait until he can get the permission. Compared to generic methods based on fully homomorphic encryptions, this is acceptable and practical: recent implementations [20], [14], [25] of fully homomorphic encryption schemes still require daunting overheads in terms of time and storage consumption. (Even computing AES circuit [21] over encrypted data will take ≈ 5.8 seconds and will use ≈ 21 MB of memory per 128-bit input message. c.f. [21] for formal details).

Table II: Benchmark of Paillier cryptosystem (n is 2048-bit)

Public key size	4096 bits
Secret key size	4096 bits
Plaintext message size	2048 bits
Ciphertext size	4096 bits
Encryption	30.378 ms
Decryption	21.170 ms
Homomorphic addition	0.004 ms
Homomorphic subtraction	0.073 ms
Homomorphic scale (100-bit constant)	1.564 ms
Homomorphic scale	18.867 ms

Evaluation results. Figure 6 shows the evaluation results of PISA. Whenever an SU j wants to make a WiFi transmission request, it prepares a request by encrypting her private operational parameter $\{F(c,i)\}_{C\times B}$ as specified by (5) in §IV-A3 with $C = 100$ and $B = 600$, which takes about 221s. This preparation phase can be *precomputed*, since SU

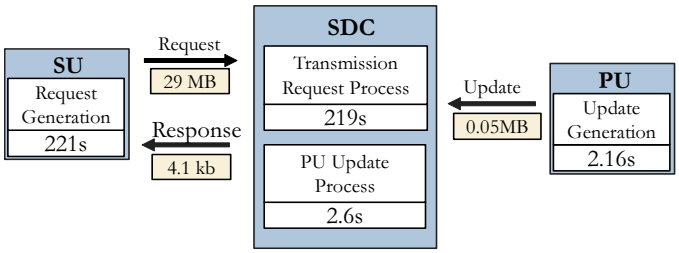


Figure 6: System Evaluation

will not change its own configurations frequently. Also a portion of the encrypted data is encryptions of 0, because SU only needs to include the values of $F(c,i)$'s, where i is within a certain distance from SU, which is d^c based on (1). The resulting ciphertext $\widetilde{\mathbf{F}} = \{\widetilde{F}(c,i)\}_{C\times B} C \times B$ is about 29 MB. SU j then sends this request to the SDC server. Note that although SU will not change its operational parameters frequently, it might want the ciphertext that encrypts the same parameters to look different. In order to achieve that, SU can simply multiply the pre-stored ciphertexts by r^n with a new randomly selected r (as seen in Figure 2). This will be a relatively much cheaper procedure than re-constructing the request, since for each entry in the submitted matrix it only involves one multiplication (which takes the same amount of time as homomorphic addition as shown in Table II).

It is worth to point out that the 221s time overhead incurred during request preparation can be dramatically reduced to $\approx \mathbf{11}$s if SU will use the same configuration for all the channels it is requesting, that is, one configuration for all the 100 channels in this simulation. This is because SU can use the strategy mentioned above to randomize the encrypted parameters, which involves only one multiplication per ciphertext.

One might be also concerned that if the PUs switch the channels frequently, then those PUs need to send update requests to the SDC very often and thus causing SUs to request new transmission accesses very frequently; while the update requests from PUs do not happen often in practice. As pointed out in [36] (§3.3) channels are categorized into physical and virtual channels, and each physical channel can contain several virtual channels. Hence, when a PU is switching between virtual channels but staying in the same physical channel, it does not need to notify the SDC. The update request is required only when the PU switches between physical channels. Per [16], TV viewers switch between virtual channels about 2.3-2.7 times per hour on average, and the rate of switching between physical channels is much lower.

After receiving SU j's request, the SDC server spends about 219s to process this request. Concretely, the SDC server computes (11), (12), (14), (16) and (17), and it will generate a response to SU j of size about 4.1kb, which is the size of one ciphertext in Paillier cryptosystem with $n = 2048$ as shown in Table II. In order to update, PU sends the encrypted form of $\widetilde{\mathbf{W}_i}$ as explained in §IV-B, which is of size ≈0.05MB. Note that the size of the encrypted data sent by PU is independent of the number of blocks in the service area and it grows linearly with only the number of channels that PU receives. This is particularly true because the location of TV receiver can be fixed and registered, which is already required in some countries as pointed out in [16]. With the update request from PU, the SDC server needs to perform formulas (9) and (10), which takes about 2.6s for each update.

SU's location privacy vs time trade-off. In the simulation above, we consider preventing the SDC from knowing any information related to SU's location. This is, during each transmission request, SU needs to submit an encrypted matrix of size proportional to the area size, which requires homomorphic operations over the encrypted data associated with the entire area. Thus, one way to reduce the computational overheads incurred during request generation is to relax the level of the desired privacy of SU's location. If the SDC is allowed to know within which specified area the SU is located, then the SU just needs to compute an encrypted matrix of size equal to the number of blocks in the specified area. For instance, in our simulation setting, one SU in the north part of the entire (600-block) map wants to transmit and the SDC is allowed to know that this SU is located somewhere in the north. To prepare such request, SU computes an encrypted 100×300 (instead of 100×600) matrix and consequently SDC just needs to handle this smaller size matrix for processing the request. Note that the homomorphic operations are applied to each entry one by one in the encrypted matrix. Therefore, it is easy to see that the relation between the privacy of SU's location and request preparation/processing time is asymptotically linear. That is, the request preparation/processing time grows linearly as the protection level on SU's location increases, and it will reach the maximum value when considering the complete protection of SU's location.

B. Real Experiments

Experiment Setup. To verify the validity PISA, we setup an experiment environment with Soft Define Radio (SDR, consisting of two Ettus USRP N210 devices as SU_1 and SU_2, Ettus USRP X310 as PU, and a DELL E6520 lap-top as the SDC). All the USRP devices work on the 2.4GHz with IEEE 802.11g. We choose channel 6 (Center frequency 2.437GHz, bandwidth 22MHz) for our experiment; this channel has the minimum interference based on our test, although any channel that has such characteristics could be used. All equipments are shown in Figure 7. With the software GNU Radio on SDC we can visually monitor all the signal received by SUs and PU.

In this experiment, we consider the following scenarios. 1) PU is not occupying channel 6, and SU_1 and SU_2 are

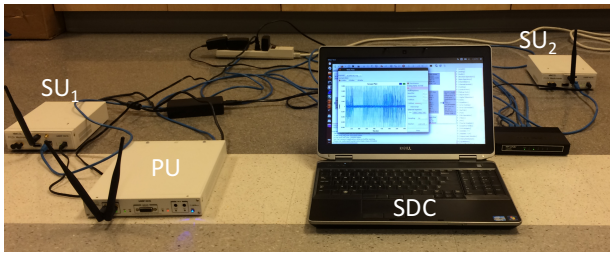


Figure 7: The Experiment Environment.

transmitting signals through channel 6. 2) PU wants to use channel 6, and thus PU sends a feedback signal to SDC. 3) SU_1 and SU_2 prepare and send out the transmission requests to SDC. 4) After processing the transmission requests from SU_1 and SU_2, SDC grants the transmission permission to the one which will not cause significant inference to PU.

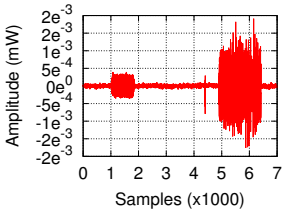


Figure 8: Signals from SU_1/SU_2.

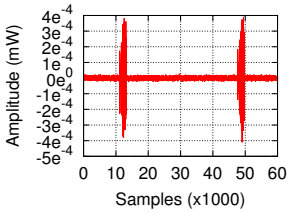


Figure 9: Request from SU.

Scenario 1. At the beginning of the experiment, SU_1 and SU_2 are transmitting on channel 6 while we are monitoring the channel using PU. As a result, Figure 8 shows that two packets were sent from SU_1 and SU_2 within about 0.35ms, which are received by PU with sample rate 20MHz. The waveforms received by PU shown in Figure 8 are of two different amplitudes (which means that the signal strength or power). It is easy to see that this difference stems from the fact that the distance of the two SUs from PU is not equal.

Scenario 2. Figure 10 shows that SU_1 and SU_2 occupy the channel and transmit signals on it. When PU wants to use the channel, it sends an update message (using the channel) to the SDC right at the moment it starts to use this channel. SDC notifies SU_1 and SU_2 of PU's usage and requests them to stop transmitting so that PU can occupy the channel.

Scenario 3. After PU starts using the channel, SU_1 and SU_2 want to request sharing the channel with PU (SU wants to transmit in active PU channel). As in Figure 11, SU_1 and SU_2 send transmission requests to SDC for transmission permission on the channel, and then the SDC sends back acknowledgement messages to notify them of the reception of requests. Whether they will be allowed to transmit or not will be the decision made by SDC in the next scenario.

Scenario 4. Upon receiving two SUs' transmission requests,

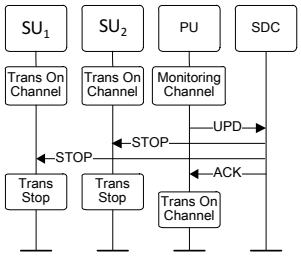


Figure 10: Update from PU.

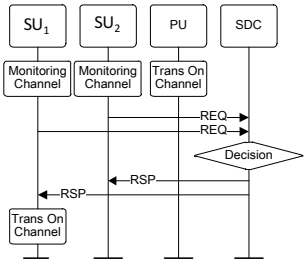


Figure 11: Request from SU.

the SDC processes the requests and decides if they, operating with the submitted configuration, will cause interference to PU's usage on the channel. If the average signal strength of SU is not excessively strong (lower than a threshold associated with PU's) so that a noticeable interfere to PU's usage will occur, the SDC will allow SU to transmit on channel as well. In Figure 9, we can see that SDC lets one of the SUs transmit. In this experiment, SU_2 is allowed to transmit and sends out about 11 packets within 20ms (sample rate is 20MHz).

The four scenarios above together depict a typical situation where SUs want to transmit after PU updates the channel he is receiving. In our experiment, two SUs are requesting to transmit over a specified channel, and SDC only allows the one whose transmission will not have significant impact on PU's reception to transmit. This verifies the validity of the proposed PISA.

VII. Conclusion

We design PISA, a protocol for dynamic spectrum allocation while preserving users' privacy. PISA uses the homomorphic properties of the Paillier cryptosystem by decomposing the complex spectrum allocation process into basic arithmetic computation types that can be performed homomorphically. We implemented and conducted experiments to evaluate PISA, showing its feasibility in a real-world application.

In the future, we will relax the assumption on the STP in PISA. Specifically, we will pursue a model that does not involve an STP, and a protocol that requires less communication rounds and latency, compared with PISA, which requires STP.

Acknowledgement

This work was supported in part by the U.S. National Science Foundation under Grants CNS-1547223, CNS-1547366, and CNS-1547908 and by the Global Research Lab. Program of Korea National Research Foundation under grant NRF-2016K1A1A2912757.

REFERENCES

[1] —, "Second report and order and memorandum opinion and order in the matter of unlicensed operation in the tv broadcast bands additional spectrum for unlicensed devices below 900 mhz and in the 3 ghz band," FCC, 2008.

[2] ——, "Atsc. atsc digital television standard - part 2: Rf/transmission system characteristics," Advanced Television Systems Committee Report, December 2011.

[3] ——, "U.S. Geological Survey," Online, March 2016.

[4] ——, "DTV Reception Maps," Online, March 2016.

[5] ——, "CPTE. Extended Hata and Hata-SRD Models." Online, March 2016.

[6] ——, "SPLAT," Online, March 2016.

[7] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. Park, J. Reed, and D. Gurney, "Protecting the primary users' operational privacy in spectrum sharing," in *DYSPAN*, 2014.

[8] E. Barker and Q. Dang, "Sp 800-57. recommendation for key management, part 3," in *NIST Special Publication 800-57 Part 3, Revision 1*. Gaithersburg, MD, United States: NIST, 2015.

[9] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *TCC*, 2005.

[10] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *CRYPTO*, 2012.

[11] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *CRYPTO*, 2011.

[12] O. Catrina and S. De Hoogh, "Improved primitives for secure multiparty integer computation," in *SCN*, 2010.

[13] I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft, "Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation," in *TCC*, 2006.

[14] Y. Doroz, Y. Hu, and B. Sunar, "Homomorphic AES Evaluation using NTRU," Cryptology ePrint Archive, Report 2014/039, 2014.

[15] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *CRYPTO*, 1984.

[16] B. Ellingsæter, H. Bezabih, J. Noll, and T. Maseng, "Using tv receiver information to increase cognitive white space spectrum," in *DYSPAN*, 2012.

[17] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *IEEE INFOCOM*, 2013.

[18] J. Garay, B. Schoenmakers, and J. Villegas, "Practical and secure solutions for integer comparison," in *PKC*, 2007.

[19] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[20] C. Gentry, S. Halevi, and N. P. Smart, "Fully homomorphic encryption with polylog overhead," in *EUROCRYPT*, 2012.

[21] ——, "Homomorphic evaluation of the aes circuit," Cryptology ePrint Archive, Report 2012/099, 2012, http://eprint.iacr.org/.

[22] O. Goldreich, "Secure multi-party computation," Manuscript., 1998.

[23] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *STOC*, 1982.

[24] T. Granlund and the GMP development team, *GNU MP: The GNU Multiple Precision Arithmetic Library*, 6th ed., 2015.

[25] S. Halevi and V. Shoup, "HElib - An Implementation of homomorphic encryption," Cryptology ePrint Archive, Report 2014/039, 2014.

[26] K. Harrison, S. M. Mishra, and A. Sahai, "How much white-space capacity is there?" in *IEEE DySPAN*, 2010.

[27] H. R. Karimi, "A framework for calculation of tv white space availability subject to the protection of dtt and pmse," in *PIMRC*, 2013.

[28] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2014.

[29] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "Senseless: A database-driven white spaces network," *IEEE TMC*, 2012.

[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT'99*. Springer.

[31] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of secure computation*, 1978.

[32] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *PKC*, 2010.

[33] U.S. Geological Survey, "SRTM3," Online, March 2016.

[34] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *EUROCRYPT*, 2010.

[35] A. Yao, "How to generate and exchange secrets," in *FOCS*, 1986.

[36] X. Zhang and E. W. Knightly, "Watch: Wifi in active tv channels," in *MobiHoc*, 2015.