Poster: Privacy-Preserving Server-Driven Dynamic Spectrum Access System

Yanzhi Dou, Kexiong (Curtis) Zeng, and Yaling Yang
Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA, USA
{yzdou, kexiong6, yyang8}@vt.edu

ABSTRACT

Dynamic spectrum access (DSA) technique has been widely accepted as a crucial solution to mitigate the potential spectrum scarcity problem. As a key form of DSA, government is proposing to release more federal spectrum for sharing with commercial wireless users. However, the flourish of federal-commercial sharing hinges upon how privacy issues are managed. In current DSA proposals, the sensitive operation parameters of both federal incumbent users (IUs) and commercial secondary users (SUs) need to be shared with the dynamic spectrum access system (SAS) to realize efficient spectrum allocation. Since SAS is not necessarily operated by a trusted third party, the current proposals dissatisfy the privacy requirement of both IUs and SUs. To address the privacy issues, this paper presents a privacy-preserving SAS design, which realizes the complex spectrum allocation decision process of DSA through secure computation over ciphertext based on homomorphic encryption, thus none of the IU or SU operation parameters are exposed to SAS.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; K.4.1 [**Computer and Society**]: Public Policy Issues—*Privacy*

Keywords

server-driven DSA; privacy; homomorphic encryption

1. INTRODUCTION

The depletion of wireless spectrum has greatly promoted the advances in DSA. In DSA, unlicensed secondary users can opportunistically access licensed spectrum under the constraint that they do not impose harmful interference to incumbent users. Sharing between the government incumbents (i.e. federal or non-federal agencies) and commercial wireless broadband operators/users is one of the key forms of DSA that is recommended by the NTIA and FCC. Recommendations in the President's Council of Advisors on Science and Technology (PCAST) report has recommended to set up a spectrum access system (SAS) database to govern the spectrum sharing between federal incumbent users (IUs) and commercial secondary users (SUs) [5]. In a typical scenario in these SAS-driven systems, both IUs and SUs need to send their operation data to SAS to realize DSA.

One of the critical concerns in light of the increasing prospects of the SAS-driven spectrum sharing between federal government incumbent systems and non-government systems is the privacy issue. For national security reasons, operation information of government incumbent users is often classified data. For example, the IUs in 3.5 GHz DSA band include military and fixed satellite service licensees, whose operation data is highly sensitive. Similarly, secondary users' operation parameters may also be sensitive commercial secrets for their operators. Yet, to realize efficient spectrum access, current server-driven designs require IUs and SUs to send their operation data to SAS for spectrum allocation, which exposes the IUs and SUs to potentially severe privacy violation.

In this paper, we aim to solve the privacy issues of SUs and IUs for SAS by designing a privacy-preserving SAS system that realizes the spectrum allocation decision process of DSA while preserving users' privacy. The main challenge is that the spectrum allocation process needs to consider many factors and use complex signal propagation models. To solve the challenge, (1) we separate the parameters in a radio propagation model that need privacy protection from those that are open knowledge, and only the part of radio propagation model that involves private parameters need to be carried out securely over ciphertext. (2) We strategically disintegrate the spectrum allocation computation related to private operation parameters into a small set of primitives like addition, substraction and scalar multiplication that can be homomorphically evaluated by Paillier cryptosystem [4] efficiently. Under our design, SUs and IUs's interactions with the SAS follow the typical database interaction pattern (e.g. update, query, response and confirmation) and do not need to stay connected with the SAS system beyond the most basic handshakes. To the best of our knowledge, we are the first to be able to successfully realize practical secure spectrum allocation. Preliminary evaluation results are given, which show the performance of our secure SAS design is reasonable on real world data.

MobiCom'15, September 7–11, 2015, Paris, France.
ACM. ISBN 978-1-4503-3619-2/15/09
http://dx.doi.org/10.1145/2789168.2795161.

2. PRELIMINARIES

2.1 Paillier Cryptosystem

The design of our secure SAS system is based on Paillier cryptosystem [4], which is a public-key encryption scheme that efficiently supports additive, subtractive and scalar multiplicative homomorphic operations as follows:

$$\begin{cases} Dec(\widehat{m_1} \oplus \widehat{m_2}) = m_1 + m_2 \\ Dec(\widehat{m_1} \ominus \widehat{m_2}) = m_1 - m_2 \\ Dec(c \otimes \widehat{m_1}) = c \times m_1 \end{cases},$$

where m_1, m_2 are arbitrary messages, $\widehat{m}$ is the ciphertexts of message m after encryption, $\oplus, \ominus, \otimes$ are used to represent the homomorphic version of addition, substraction, and scalar multiplication operation respectively, c is a constant value in plaintext, and $Dec()$ denotes the decryption operation.

2.2 Reference Model of Traditional SAS

In this section, we describe a reference model of the traditional server-driven SAS, which serves as the basis for understanding our design henceforth. In the typical scenario, there are three parties involved, including IUs, SUs and SAS Server. SAS Server refers to a cloud-based spectrum management infrastructure that allocates spectrum resources while considering incumbent protection. SAS Server receives SUs' spectrum access request, calculates the availability of the spectrum requested by SUs, and returns the corresponding request response to SUs. To accurately determine the spectrum availability requested by SUs, SAS Server predicts the aggregate SU interference on all IUs using some radio propagation model based on the operation parameters provided by IUs and SUs. If the predicted interference on any IU exceeds a desirable threshold when the requesting SU is allowed to transmit, SAS Server will respond the requesting SU with a denial of access right. Otherwise, the requesting operation is permitted. Upon receiving a permission, an SU may transmit a confirmation to SAS, confirming its reception of the license.

3. SECURE SAS SYSTEM DESIGN

Assume that the SAS is semi-honest, and the goal of our scure SAS is to realize the same computation task as the traditional SAS while protecting the data privacy of both IUs and SUs from the semi-honest SAS.

3.1 Design Overview

As shown in Figure 1, our design includes four different parties: (1) a SAS Server for computing spectrum allocation, (2) an IU Aggregator that is responsible for keeping track of IUs and is operated by a trusted party of the IUs, (3) SUs in the service area of SAS Server, (4) Key Distributor. Key Distributor creates a group Paillier public/private key pair (pk_G, sk_G) and is trusted for keeping sk_G a secret only known to itself. In addition, each SU b has his own pair of Paillier public/private keys (pk_b, sk_b) and uploads pk_b to Key Distributor. Using pk_b and sk_G, Key Distributor can provide ciphertext conversion service to SAS Server, where it converts a ciphertext encrypted by pk_G to a ciphertext encrypted by pk_b so that the ciphertext can be decrypted by SU b. Note that Key Distributor is not fully trusted, instead it is only trusted in keeping sk_G secret. So messages sent to Key Distributor will be hidden by adding some blind factors first.

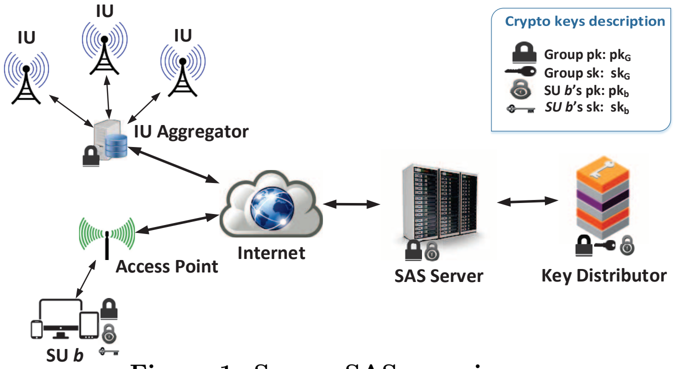

Figure 1: Secure SAS overview

3.2 Private Input Information to SAS

The core step in our SAS design is the secure computation of the aggregate of SU interference on IUs. In essence, this means that the aggregate RSS of all SUs on all IUs must be derived based on a certain radio propogation model and its required radio operation information. To compute the RSS, we adopt the highly-sophisticated and accurate Longley-Rice (L-R) model. L-R model is a complex model which takes as many as thirteen parameters as input, but in our problem, the only five parameter values that need to be considered are distance, the transmitter and receiver antenna heights, transmit power, and the terrain elevation profiles (frequency is a constant in single channel case). Observe that among the five parameters that we consider, only the transmitter and receiver configurations (e.g. locations, antenna heights, transmit power) are privacy related. The remaining factor – terrain information – is public knowledge that can be easily found on government terrain database like [1]. Thus, to use L-R model to compute if SU interference exceeds IU tolerance, the following private operation information needs to be transmitted to SAS Server:

(1) Private IU information transmitted from IU Aggregator to SAS: Interference threshold of each IU, location of each IU, and antenna height of each IU. The interference threshold represents the maximum amount of aggregate SU interference that an IU can tolerate.

To reduce computation overhead, we quantize all the possible values of location and antenna height into limited choices. Specifically, we divide the service area of a SAS Server into L small grids of equal size and express an IU's location using its grid number. In addition, we quantize IU receiver antenna height into H_I levels. Using the above quantization, a two-dimensional matrix $\mathbf{T} := \{T(l, h_I)\}_{L \times H_I}$ is used to represent the IU operation information that is transmitted to SAS. When an IU of antenna height h_I is located in geographic grid l, $T(l, h_I)$ equals the interference threshold of the IU. Otherwise, $T(l, h_I)$ is set to an extremely large number (e.g. 100W) that cannot be exceeded in realistic radio propagation scenarios, representing an infinite tolerance to interference. $\mathbf{T}$ is encrypted by pk_G before sending to SAS Server, and the generated ciphertexts is denoted as $\widehat{\mathbf{T}}$.

(2) Private SU information transmitted from an SU b to SAS: Maximum transmit power, location, and antenna height of SU b.

Similar to the IU case, a two-dimensional matrix $\mathbf{R}_b := \{R_b(j, h_S)\}_{L \times H_S}$ is used to represents the above SU operation parameters, where the SU transmitter antenna height is quantized into H_S levels. $R_b(j, h_S)$ equals SU b's maximum transmit power if SU b is located in grid j and has an antenna height of h_S. Otherwise, $R_b(j, h_S)$ equals 0, indicating no active transmission in such a configuration. $\mathbf{R}_b$ is encrypted by pk_G before sending to SAS Server, and the generated ciphertexts is denoted as $\widehat{\mathbf{R}_b}$.

3.3 Secure Dynamic Spectrum Allocation

With the up-to-date ciphertext of IU operation information (i.e. $\widehat{\mathbf{T}}$) from IU Aggregator, SAS Server can securely compute whether SU Interference exceeds IU tolerance level. SAS Server first precomputes a plaintext attenuation map $\mathbf{I} := \{I\left(l, j, h_I, h_S\right)\}_{L^2 \times H_I \times H_S}$ based on the public terrain information. Here the entry $I(l, j, h_I, h_S)$ represents the precomputed path attenuation from a transmitter with h_S antenna height in grid j to a receiver with h_I antenna height in grid l. Given $\mathbf{I}$, after SU b uploads $\widehat{\mathbf{R}_b}$ to SAS Server in its request for spectrum access, the interference that SU b imposes on an IU in grid l with antenna height h_I can be derived as:

$$\widehat{F_b}(l, h_I) := \oplus_{j,h_S}\left[I(l, j, h_I, h_S) \otimes \widehat{R_b}(j, h_S)\right], \qquad (1)$$

where $\oplus_{j,h_S}$ and $\otimes$ are the homomorphic version of the summation symbol $\sum_{j,h_S}$ and the scalar multiplication symbol respectively, and $\widehat{F_b}(l, h_I)$ is the ciphertext of the SU interference.

SAS Server then can compute if adding SU b will exceed the interference threshold of some IU. First, SAS Server maintains an interference budget matrix $\widehat{\mathbf{N}} := \{\widehat{N}(l, h_I)\}_{L \times H_I}$ of IUs. Each entry $\widehat{N}(l, h_I)$ holds the ciphertext of the amount of additional SU interference that can be tolerated at every grid. The initial value of the interference budget matrix $\widehat{\mathbf{N}}$ is set to be the same as the interference threshold matrix $\widehat{\mathbf{T}}$, which is provided by IU Aggregator. In essence, this means that if an IU with antenna height of h_I exist in grid l, the initial value of $\widehat{N}(l, h_I)$ equals the ciphertext of the interference threshold of the IU. If no such IU exists, $\widehat{N}(l, h_I)$ is a ciphertext of an extremely large power value.

Then, using (1), SAS Server can create an interference indicator matrix $\widehat{\mathbf{G}_b} := \{\widehat{G_b}(l, h_I)\}_{L \times H_I}$ as follows:

$$\widehat{G_b}(l, h_I) := \widehat{N}(l, h_I) \ominus \widehat{F_b}(l, h_I). \qquad (2)$$

The signs of all $\widehat{G_b}(l, h_I)$'s plaintexts indicate whether SU b should be allowed to operate. If there exists a $\widehat{G_b}(l, h_I)$ that holds the ciphertext of a non-positive value, it means that an IU with antenna height h_I exists in grid l and this IU runs out of its interference budget if SU b operates. In such a case, SU b is not allowed to operate and SAS does not compute a valid spectrum access license for SU b. Only when all $\widehat{\mathbf{G}_b}$ entries are ciphertext of positive values, SU b is safe to operate. In this case, SAS generates a valid license to give SU b spectrum access right. Such license computation is made by employing ciphertext conversion service provided by Key Distributor. On a high level, Key Distributor transfers the sign information contained in $\widehat{\mathbf{G}_b}$ (encrypted by pk_G) to a license (encrypted by pk_b). If there exists one $\widehat{\mathbf{G}_b}$ entry whose plaintext is non-positive value, an invalid license is generated. Otherwise, a valid license is generated.

SAS Server then sends a response back to SU b. The response includes the encrypted license and $\widehat{\mathbf{F}_b}$. Upon receiving the response, SU b finds out the availability of the spectrum by checking the validity of the license after decrypting using his own private key sk_b. Then SU b creates a matrix $\widehat{\mathbf{U}_b} := \{\widehat{U_b}(l, h_I)\}_{L \times H_I}$ by:

$$\widehat{U_b}(l, h_I) := \begin{cases} \widehat{F_b}(l, h_I) \oplus \widehat{0}(l, h_I), & \text{SU } b \text{ gets the valid license} \\ \widehat{0}(l, h_I), & \text{SU } b \text{ gets no valid license} \end{cases}, \qquad (3)$$

$\widehat{\mathbf{U}_b}$ is then sent to SAS Server along with SU b's confirmation message. SAS Server then updates the interference budget matrix by substracting $\widehat{\mathbf{U}_b}$ from $\widehat{\mathbf{N}}$ as follows:

$$\widehat{N}(l, h_I) \leftarrow \widehat{N}(l, h_I) \ominus \widehat{U_b}(l, h_I). \qquad (4)$$

Combining (3) and (4), we can see that, when SU b receives the license, the interference budgets (i.e. the plaintexts of $\widehat{\mathbf{N}}$ entries) are reduced by SU b's interference. When SU b does not get the license, interference budgets stay the same. In such a manner, $\widehat{\mathbf{N}}$ is gradually reduced as more SUs obtain their operation licenses. In addition, the homomorphic addition of $\widehat{0}(l, h_I)$ to $\widehat{F_b}(l, h_I)$ in (3) ensures that the ciphertext $\widehat{U_b}(l, h_I)$ stays different from the ciphertext $\widehat{F_b}(l, h_I)$ even when their plaintexts are the same. Thus, SAS Server cannot guess the spectrum allocation results by comparing $\widehat{U_b}(l, h_I)$ with $\widehat{F_b}(l, h_I)$.

4. PRELIMINARY EVALUATION RESULTS

To evaluate our SAS system, we set the service area to be a large 154.82 km^2 area in Washington DC. We employ L-R model provided by SPLAT! [2] to calculate the attenuation map $\mathbf{I}$ in this area. We feed high resolution SRTM3 [1] terrain elevation data to SPLAT!. We construct the Paillier cryptosystem in C programming language based on GMP library [3]. Notably, The length of security parameter n is set to be 2048 bits, so the Paillier cryptosystem we build has an equivalent security level of 112 bits. The SAS system is built based on the Paillier cryptosystem implementation. For parallelization implementation, we employ the POSIX Threads in C programming and we divide the computation in SAS into 16 concurrent threads, and each of two desktops runs 8 threads. Evaluation results show that the total time to process a request in SAS Server is around 3 minutes, and the total communication bandwidth overhead between SU and SAS Server in one handshake is 6.68 MB.

5. CONCLUSIONS

In this paper, we build a secure SAS system which performs DSA while preserving users' privacy. Our design coverts complex DSA process into limited computation types provided by Paillier homomorphic crypotosystem. The computation and communication overhead is both moderate for practical SAS service.

6. REFERENCES

[1] http://dds.cr.usgs.gov/srtm/version2_1/SRTM3/.

[2] http://www.qsl.net/kd2bd/splat.html.

[3] https://gmplib.org/.

[4] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT'99*, pages 223–238. Springer, 1999.

[5] PCAST. Report to the president realizing the full potential of government-held spectrum to spur economic growth. 2012.