Policy-based Privacy-Preserving Scheme for Primary Users in Database-driven Cognitive Radio Networks

Jianqing Liu[1], Chi Zhang[2], Haichuan Ding[1], Hao Yue[3], and Yuguang Fang[1]

[1]Dept. of Electrical and Computer Engineering, University of Florida, Gainesville, Florida 32611 USA,
Email: jianqingliu@ufl.edu, fang@ece.ufl.edu, dhcbit@gmail.com
[2]Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
University of Science and Technology of China, Hefei 230027, P. R. China,
Email: chizhang@ustc.edu.cn
[3]Department of Computer Science, San Francisco State University, San Francisco, CA 94132 USA,
Email: haoyue@sfsu.edu

***Abstract*—In cognitive radio networks (CRNs), spectrum database has been well recognized as an effective means to dynamically sharing licensed spectrum among primary users (PUs) and secondary users (SUs). In spectrum database, the protected incumbents (a.k.a. PUs) and the CRs (a.k.a. SUs) are required to register in database their *operational specifications* such as transmitting power, antenna height, time of operation and etc. so as to provide an up-to-date radio map for public queries and avoid possible interference. However, it poses potentially serious privacy problems especially when governmental and military systems participate in spectrum sharing through spectrum database. Most recent research works in database-driven CRNs, however, only focused on protecting user's location privacy but merely studied preserving PUs' operational specifications. In this paper, we propose a secure and privacy-preserving scheme using hidden policy-assisted attribute-based encryption technique to protect sensitive PUs' *operational privacy* without affecting database's accessibility and spectrum utilization efficiency. The security and performance analysis demonstrates that our scheme is secure and computationally efficient. Additionally, our policy-assisted scheme is practical and promising because of its consistency with FCC/NTIA's rule in spectrum regulation in database-driven CRNs.**

I. Introduction

In recent decades, the ever-increasing demand for higher data rates has motivated researchers to explore more efficient utilization of precious radio spectrum. Cognitive radio (CR) has been well-recognized as a powerful technique to improve spectrum utilization by allowing spectrum sharing among licensed users (a.k.a. primary users (PUs)) and unlicensed users (a.k.a. secondary users (SUs)) [1]. To enable dynamic spectrum access, SUs can either use spectrum sensing or make queries to spectrum database to determine locally available spectrum. However, in 2012, the FCC's rule eliminated spectrum sensing as a requisite capability for CR devices [2]. Instead, it enforces spectrum database query as a primary means of determining white space spectrum. Since then, FCC has certified several industrial entities (e.g. Google, LStelcom and etc.) as spectrum database administrators, which are required to house an up-to-date repository of incumbents, and in certain cases, protected access users (a.k.a. registered SUs) who register white space channels in database. Moreover, in 2012, FCC advocated sharing federal government (including military) spectrum in 3.5 GHz band, which is used by U.S. Department of Defense (DoD) for radar installations, with non-government systems [3].

It should be noted that when federal, possibly military spectra are incorporated into spectrum database, a serious breach of privacy may happen. Specifically, for conventional commercial incumbent systems such as TV stations, the leakage of their private information may not be a major issue. However, for sensitive incumbents such as military systems, their operational attributes such as identity, operating spectrum, antenna parameters (e.g. height, directivity), power level, geolocation and time of operations should be kept private [4]. The reason is that if not protected, due to curious database or malicious users through innocuous queries, PUs' operational specifications and activities could be disclosed which may result in compromise to military/federal systems. Nevertheless, such problem cannot be addressed by tightly controlling access to database, since all users need access to it to enable spectrum sharing. Thus, an intelligent method to obfuscate the private information needs to be designed.

Recently, most existing works on database-driven CRNs mainly focus on addressing user's location privacy issues by regarding spectrum database as a location-based service (LBS) [8]–[10]. However, the aforementioned privacy concerns have rarely been studied. To fill in such a gap, in this paper, we propose a scheme that can preserve sensitive PUs' operational privacy by hiding PUs among a large set of access users while revealing nothing of their operational specifications. To be specific, firstly, a trusted authority (e.g. FCC, or NTIA)

This work was partially supported by the US National Science Foundation under grant CNS-1343356.

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

issues encrypted control messages along with policies to spectrum database for access control of a specific spectrum at a certain time and area. Further, such policies are hidden in an access tree structure of **Or**- and **And**-gate so that adversaries cannot obtain the exact policy contents. Then, the trusted authority generates attribute credentials for access users (i.e. PUs/SUs) based on their operational specifications. After a public query/browse to spectrum database, a user uploads its attribute credential to database to request for a specific spectrum at a certain time and location. Then, database takes user's attribute credential to access tree structure to decrypt control messages, which for example maybe *"accept"*, *"deny"*, *"report to TA"* and *etc*. If the plaintext messages can be successfully obtained, database acts according to control messages; if not, database rejects user's request. Therefore, our scheme can achieve anonymity of sensitive PUs among a large group of access users and also obfuscation of their operational specifications. More importantly, the sensitive regulatory policies are also kept secure to spectrum database.

The reminder of this paper is organized as follows. Section II describes system model and security model. Section III introduces some essential preliminaries. Then our scheme is discussed in details in Section IV. The performance evaluation including security analysis and simulations are presented in Section V. Some related work is given in Section VI and finally, Section VII concludes the paper and outlines the future work.

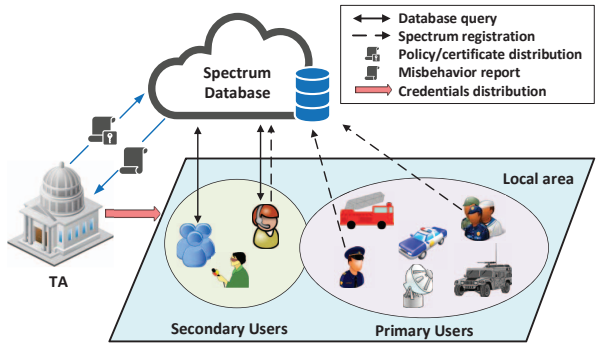

Fig. 1. System Model

II. System and Security Model

A. System Model

In our considered system as shown in Fig.1, there are two entities (i.e. TA and spectrum database (SD)) and access users, which respectively have the following properties.

- **TA**: A trusted authority (TA) can be served by FCC, NTIA or other federal technical centers. In this paper, TA is responsible for issuing encrypted control messages and hidden policies to spectrum database, generating attribute credentials for access users and monitoring misbehaving users through database feedback.
- **Spectrum Database (SD)**: A spectrum database (SD) is maintained by industrial companies that have enormous storage and computation capability such as Google. It hosts an up-to-date repository of geographic spectrum usage information. Additionally, it performs access control for any requested user.
- **User** (u): A user could be a general SU or a sensitive PU such as a military system. It first queries/browses spectrum database and then asks for attribute credential from TA. After that, user takes its credential to spectrum database to request for access to a specific spectrum at a certain location.

B. Security Model

TA is fully trusted in the system and will not be compromised by any attacker. PUs are trusted as well. Spectrum database is considered honest but curious because it is operated by industrial companies which maybe interested in collecting users' daily activities for commercial purposes such as advertisements (i.e. spams). More critically, malicious employees in database operating companies may have incentives to obtain sensitive PUs' operational specifications and activities. For example, employees could be lured by adversaries and sell such sensitive information of military systems to them, which may result in severe compromise to homeland security. Additionally, attackers could also come from public users who try to obtain PUs' operational specifications through innocuous queries to spectrum database.

III. Preliminaries

A. Bilinear Pairing

A bilinear paring is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$, where all groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are multiplicative cyclic groups of prime order p. g_1 is a generator of $\mathbb{G}_1$ and g_2 is a generator of $\mathbb{G}_2$. The pairing e has the following properties.

- **Bilinearity**: $e\left(u^a, v^b\right) = e(u, v)^{ab}$ for all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p^*$.
- **Non-degeneracy**: $e\left(g_1, g_2\right) \neq 1$ for $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$.
- **Computability**: $e\left(u, v\right)$ can be computed efficiently for any $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$.

The security of our scheme relies on discrete logarithm problem (DLP) and the external Diffie-Hellman (XDH) assumption which holds on MNT curves.

B. Ciphertext-policy Attribute-based Encryption (CP-ABE)

CP-ABE is used for cryptographical access control where every user receives a private key that corresponds to an individual set of attributes, each attribute attesting a certain property that the user has. The ciphertext is encrypted with a policy over these attributes in the form of boolean operations, and anyone whose attributes satisfy the policy can decrypt the cipertext [5]. In CP-ABE, there exists four fundamental algorithms and an access tree structure [5].

- **Setup**(1^κ): The algorithm takes a security parameter κ as input and generates the public key PK and master secret key MK.

- **KeyGen**(MK,L): The algorithm takes MK and an attribute list L as input and generates a private key SK_L associated with L.
- **Encrypt**(PK,M,Υ): The algorithm takes PK, control message M and access tree Υ as input and encrypts M to obtain a ciphertext CT.
- **Decrypt**(CT,SK_L): The algorithm takes CT and SK_L as input and returns M if the attribute list L satisfies the access tree Υ.
- **Access Tree** Υ: Given a policy, it can be represented by an access tree with $\vee$ (or) and $\wedge$ (and) gates and a set of leaf nodes that represent an attribute list. An toy example is shown in Fig.2 where the access tree is based on a policy specified on a certain location. For instance, TA may specify the following access structure at location loc_x for accessing the control message: $(v_1 \vee v_2) \wedge (v_3 \vee v_4)$. Due to space limitation, construction of access tree is omitted and interested readers are referred to [6] for more details.

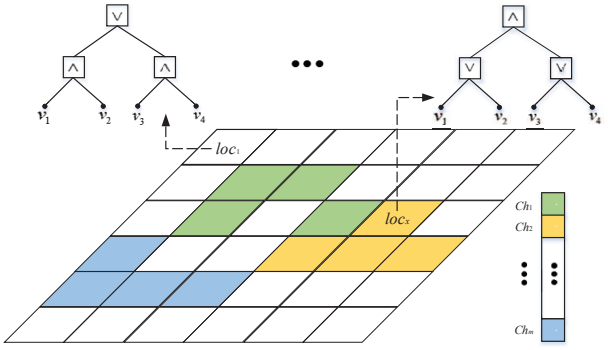

Fig. 2. Spectrum Map and Access Tree

IV. The Proposed Scheme

A. Overview

We first claim that the total attributes for a particular user can be categorised as $\{\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_i, ..., \mathbb{A}_n\}$ where $\mathbb{A}_i$ represents one specific type of attribute that can take possible values from a set $S_i = \{v_{i,1}, v_{i,2}, ..., v_{i,t}, ..., v_{i,n_i}\}$. For example, the antenna radiation pattern is an attribute which could be either omni-directional or uni-directional. For a particular user, we denote $L = [L_1, L_2, ..., L_i, ..., L_n]$ as its attribute list where $L_i \in S_i$. For the leaf nodes in access tree structure, we use notation $W = [W_1, W_2, ..., W_i, ..., W_n]$ to represent a policy issued by TA for a particular area. In order to preserve PUs' privacy on their operational attributes, we obfuscate the policy by letting $W_i \subseteq S_i$ to hide what subset W_i for each $\mathbb{A}_i$ is specified in each leaf node of the access tree. Then, the user of attribute list L that satisfies the policy W iff $L_i \in W_i$ for $1 \leq i \leq n$ can register in spectrum database without leaking their operational privacy. Such technique has two folds of benefits.

- Operational privacy can be protected using attribute credentials and through obfuscating policies while posing no computational overhead on user side.
- TA has more flexibility to control spectrum usage and meanwhile monitors misbehaving users by issuing various types of policies to spectrum database.

In the following discussion, we will elaborate detailed operations of our scheme.

B. System Initialization

TA firstly generates a tuple $G = [p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, e]$ and selects a random $\theta \in \mathbb{Z}_p^*$ as its private key TSK then computes ${g_2}^{\theta}$ as its public key TPK. For a particular user u, TA generates a private key USK_u as $\varepsilon_u \in \mathbb{Z}_p^*$ and calculates its corresponding public key UPK_u as ${g_2}^{\varepsilon_u}$. Additionally, TA picks a random number $\sigma \in \mathbb{Z}_p^*$ as the private key DSK for spectrum database and computes ${g_2}^{\sigma}$ as its public key DPK. A hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ is also defined. TA then publishes its public key TPK and securely delivers user' public key/private key pair, spectrum database's public key/private key pair to the corresponding user and entity, respectively.

C. Hidden Policy-assisted Attribute-based Encryption

In this paper, we use a cryptographic tool based on the conventional CP-ABE but with hidden policy property, which was proposed by Nishide et al in [7]. The detailed procedures are described as follows.

*1) **Setup**:* TA has the knowledge of operational attributes $\mathbb{A}$ of the user. It generates random values $\left\{a_{i,t} \in \mathbb{Z}_p^*\right\}_{1 \leq t \leq n_i}$ for i^{th} attribute ($1 \leq i \leq n$) which has n_i possible values. TA then computes $\left\{A_{i,t} = {g_1}^{a_{i,t}}\right\}_{1 \leq t \leq n_i}$. Additionally, TA chooses another two random numbers $\omega, \beta \in \mathbb{Z}_p^*$ and then computes $Y = e(g_1, g_2)^{\omega}$ and $B = {g_1}^{\beta}$. Afterwards, the policy public key (PPK) is published by TA as

$$\left\langle Y, B, p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e, \left\{\{A_{i,t}\}_{1 \leq t \leq n_i}\right\}_{1 \leq i \leq n}\right\rangle$$

while the corresponding attribute master key (AMK) $\left\langle \omega, \beta, \left\{\{a_{i,t}\}_{1 \leq t \leq n_i}\right\}_{1 \leq i \leq n}\right\rangle$ is kept securely at TA.

*2) **Encrypt**:* For a certain geographic area, TA chooses a control message M for access control which is then encrypted by the policy $W = [W_1, W_2, ..., W_i, ..., W_n]$, where W is the set of leaf nodes in the access tree Υ which in this paper is an **AND**-gate structure. TA then selects a random number $r \in \mathbb{Z}_p^*$ and sets $\widetilde{C} = MY^r$ and $C = B^r$. Also for $1 \leq i \leq n$, TA chooses random values $r_i \in \mathbb{Z}_p^*$ such that $r = \sum_{i=1}^{n} r_i$. TA then sets $C_{i,1} = g_1^{r_i}$ and computes $\{C_{i,t,2}\}_{1 \leq t \leq n_i}$ as follows:

- *well-formed*: $C_{i,t,2} = A_{i,t}^{r_i}$ if $v_{i,t} \in W_i$.
- *malformed*: $C_{i,t,2}$ is random if $v_{i,t} \notin W_i$.

Thus, ciphertext CT is given as

$$CT = \left\langle \Upsilon, \widetilde{C}, C, \left\{C_{i,1}, \{C_{i,t,2}\}_{1 \leq t \leq n_i}\right\}_{1 \leq i \leq n}\right\rangle$$

TA also generates a certificate $Cert_{TA,SD}$=$\langle DPK, SID, \delta_{TA,SD}\rangle$ for the spectrum database, where $\delta_{TA,SD} = Sig_{TSK}(DPK \| SID)$=$H(DPK \| SID)^{TSK}$ is TA's signature using its private key TSK, SID is the identity of

spectrum database and $\|$ is the concatenation of messages. Then, TA delivers the following information to spectrum database through a secure channel.

$$TA \rightarrow SD : Cert_{TA,SD}, CT$$

3) ***KeyGen****:* Suppose user u has the operational attribute list $L = [L_1, L_2, ..., L_n] = [v_{1,t_1}, v_{2,t_2}, ..., v_{n,t_n}]$. Such information should be uploaded to TA by each user. TA then picks up a random value $s_u \in \mathbb{Z}_p^*$ for user u and $\lambda_i \in \mathbb{Z}_p^*$ for each attribute $1 \leq i \leq n$. TA first computes $D_0 = g_2^{\frac{\omega+s_u}{\beta}}$ and then for $1 \leq i \leq n$, TA computes $[D_{i,1}, D_{i,2}] = \left[g_2^{s_u+a_{i,t_i}\lambda_i}, g_2^{\lambda_i}\right]$ where $L_i = v_{i,t_i}$. Thus, each user's attribute credential (AC_u) is given as

$$AC_u = \left\langle D_0, \{D_{i,1}, D_{i,2}\}_{1\leq i\leq n} \right\rangle$$

To protect a user's identity (e.g. call sign or MAC address) ID_u, TA generates PID_u as the pseudonym name for user u and securely stores ID_u-PID_u pair in its database in order to track misbehaving users whenever necessary. After that, TA generates a certificate $Cert_{TA,u} = \langle UPK_u, PID_u, \delta_{TA,u} \rangle$ for user u where $\delta_{TA,u} = Sig_{TSK}(UPK_u \| PID_u) = H(UPK_u \| PID_u)^{TSK}$ is TA's signature. Finally, TA delivers the following information to the user u through a secure channel.

$$\begin{aligned} TA \rightarrow u :& Cert_{TA,u}, PID_{u_j}, AC_u, \\ & Sig_{TSK}(PID_u), Sig_{TSK}(AC_u) \end{aligned}$$

4) ***Authentication****:* After User u obtains the credentials from TA, it first sends the following information to spectrum database for authentication

$$\begin{aligned} u \rightarrow SD :& Cert_{TA,u}, PID_u, cont_u \\ & Sig_{USK_u}(PID_u \| cont_u) \end{aligned}$$

where $cont_u$ is user u's attempted active area that reflects user's transmitting interference range. For instance, such area may include several square blocks as shown in Fig.2. Note that compared with traditional location report scheme, only registering protected/active contour in database to some extents preserves the geolocation privacy of user's transceiver pair. After receiving this message, spectrum database first verifies TA's signature on the certificate $Cert_{TA,u}$ by checking whether the equality $e(Sig_{TSK}(UPK_u \| PID_u), g_2) = e(H(UPK_u \| PID_u), TPK)$ holds.

If TA's signature is correct, spectrum database proceeds to verify u's signature using its public key UPK_u as included in the certificate $Cert_{TA,u}$. If u's signature is valid as well, spectrum database extracts the ciphertext CT at u's interested areas $cont_u$ and sends the following information to u. [1]

$$SD \rightarrow u : Cert_{TA,SD}, E_{UPK_u}\left(\left\{\{C_{i,t,2}\}_{1\leq t\leq n_i}\right\}_{1\leq i\leq n}\right),$$

$$Sig_{DSK}\left(E_{UPK_u}\left(\left\{\{C_{i,t,2}\}_{1\leq t\leq n_i}\right\}_{1\leq i\leq n}\right)\right)$$

where $\left\{\{C_{i,t,2}\}_{1\leq t\leq n_i}\right\}_{1\leq i\leq n}$ is the partial ciphertext information and it is encrypted by user u's public key UPK_u.

After receiving the message from spectrum database, user u first verifies TA's signature and then checks the validity of SD's signature. If results are correct, user u decrypts the partial ciphertext information using its private key USK_u.

5) ***Decrypt****:* After obtaining $\left\{\{C_{i,t,2}\}_{1\leq t\leq n_i}\right\}_{1\leq i\leq n}$, user u constructs a set $\{C'_{i,2}\}_{1\leq i\leq n}$ by letting $C'_{i,2} = C_{i,t_i,2}$ where $L_i = v_{i,t_i}$ for $1 \leq i \leq n$. Before sending registration request to spectrum database, user u needs to decide the access channel at area $cont_u$ through database queries. Next, user u sends the following information to spectrum database for registration.

$$\begin{aligned} & u \rightarrow SD : E_{DPK}\left(Cred_u \| \left\{C'_{i,2}\right\}_{1\leq i\leq n} \| ch_z\right), \\ & Sig_{USK_u}\left(E_{DPK}\left(Cred_u \| \left\{C'_{i,2}\right\}_{1\leq i\leq n} \| ch_z\right)\right) \end{aligned}$$

After receiving the message from user u, SD first verifies user' signature using its public key and then decrypts the message using its private key DSK if the previous verification is successful. After that, SD obtains $\{C'_{i,2}\}_{1\leq i\leq n}$ and uses it to reconstruct CT as $\langle \Upsilon, \widetilde{C}, C, \left\{C_{i,1}, C'_{i,2}\right\}_{1\leq i\leq n} \rangle$. Then, SD checks the validity of $Cred_u = \langle PID_u, AC_u, Sig_{TSK}(PID_u), Sig_{TSK}(AC_u) \rangle$ by verifying TA's signature. If it is valid, SD takes u's attribute credential AC_u along with ciphertext CT as input to **Decrypt**$(\cdot)$. If user u's attribute set satisfies the access policy, the message can be decrypted as follows

$$M = \frac{\widetilde{C}}{e(C,D_0)} \prod_{i=1}^{n} \frac{e(C_{i,1}, D_{i,1})}{e\left(C'_{i,2}, D_{i,2}\right)}$$

Correctness Proof.

$$\begin{aligned} & \frac{\widetilde{C}}{e(C,D_0)} \prod_{i=1}^{n} \frac{e(C_{i,1}, D_{i,1})}{e\left(C'_{i,2}, D_{i,2}\right)} \\ &= \frac{Me(g_1,g_2)^{wr}}{e\left(g_1^{\beta r}, g_2^{\frac{\omega+s_u}{\beta}}\right)} \prod_{i=1}^{n} \frac{e\left(g_1^{r_i}, g_2^{s_u+a_{i,t_i}\lambda_i}\right)}{e\left(g_1^{a_{i,t_i}r_i}, g_2^{\lambda_i}\right)} \\ &= Me(g_1,g_2)^{wr-r(\omega+s_u)} \prod_{i=1}^{n} \frac{e(g_1,g_2)^{r_i\left(s_u+a_{i,t_i}\lambda_i\right)}}{e(g_1,g_2)^{a_{i,t_i}r_i\lambda_i}} \\ &= Me(g_1,g_2)^{-rs_u} \prod_{i=1}^{n} e(g_1,g_2)^{r_i s_u} \\ &= Me(g_1,g_2)^{-rs_u} e(g_1,g_2)^{s_u \sum_{i=1}^{n} r_i} \\ &= Me(g_1,g_2)^{-rs_u+rs_u} \\ &= M \end{aligned}$$

□

Note that if the decryption is computed using an attribute credential AC_u that does not satisfy the access tree structure, then some values of $C'_{i,2}$ are random numbers instead of

[1]Different areas could have different access control policies which correspond to different ciphertext CT. Thus, spectrum database should send all the policy related information to user u. However, for simplicity and notational convenience, we only consider one ciphertext for area $cont_u$ in this paper.

$g_1^{a_{i,t_i} r_i}$, which will result in the failure of obtaining correct message M.

In light of this, if spectrum database confirms that user U's AC_u can correctly decrypt the access control message M, it obtains the control message M such as *"accept"*, *"deny"* or *"report to TA"*. Then, spectrum database will react accordingly. If a user is successfully registered, spectrum database updates its spectrum usage map and reveals nothing but user's occupied channel, protected contour and pseudonym in order to protect user's operational privacy against public queries. Moreover, in certain circumstances, TA can also issue regulation ciphertext policies to spectrum database to monitor the misbehaving registered users, for example, monitoring military spectrum usage information by registered users around The White House. Therefore, from this perspective, our proposed scheme is much flexible in administrating spectrum sharing and is also consistent with FCC's rule.

V. Performance Evaluation

A. Security Analysis

For the curious spectrum database, it is hard to reconstruct user's attribute list despite the fact that it has user's credentials AC_u. The reason is that AC_u is blinded by s_u which is a secret random value for any specific user. Besides, even though spectrum database has the cipthertext CT, under the XDH assumption, it is hard to know what subset W_i the TA specified for each attribute $\mathbb{A}_i$ in the ciphertext policy and thus unable to infer user's attribute from the policy. Furthermore, for external attackers, in order to decrypt the message M, they must recover $e(g_1, g_2)^{\omega r}$ which requires pairing C from ciphertext and D_0 from user's attribute credential. However, it is blinded by the value $e(g_1, g_2)^{r s_u}$. Collusion attacks among attackers won't help either since the blinding value s_u is randomized to the randomness from a particular user's AC_u. Above all, user's attribute credential, control message M and ciphertext policy are proven to be secure and user's operational privacy can thus be preserved.

B. Performance Analysis

1) Simulation Setup: We now provide the simulation results of computational cost of our scheme. The measurements were taken on a VM VirtualBox workstation with Intel i3 processor and 8G memory. The implementation is based on PBC Library [13] where a 224-bit MNT curve is used. The base field size is 224 bits and the discrete logarithm security level is 1344 bits. All the simulation results are averaged over 20 independent runs.

2) Simulations results: On the test machine, the PBC Library can generate random numbers in approximately 1.4ms, 9.4ms, 2.5ms and 0.2ms in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and $\mathbb{Z}_p^*$, respectively. Pairing operation takes 6.9ms and exponentiations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ take about 1.2ms, 9.5ms and 2ms, respectively. In light of this, the computational costs in generating private and public key pairs for users can be easily calculated based on above parameters. Thus, we will not elaborate it.

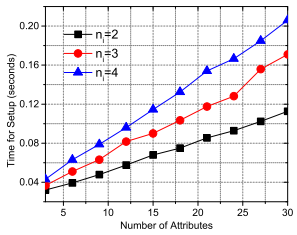

Fig. 3. Setup time

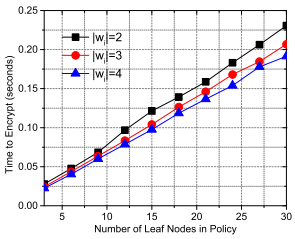

Fig. 4. Encryption time

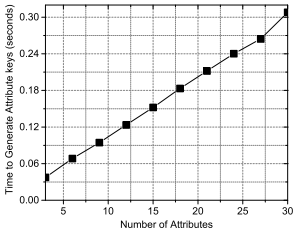

Fig. 5. Attribute credential generation time

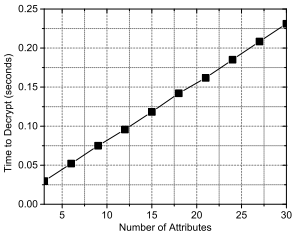

Fig. 6. Decryption time

Firstly, we examined the computational costs in the **Setup** process, which is executed in TA's side. The results are shown in Fig.3 where the **Setup** running time is almost linear with the number of attributes. On the other hand, for each attribute $\mathbb{A}_i$, there are n_i possible values. It can be observed that with the increase of n_i the computational costs increase given the same number of attributes. The reason is that TA needs to generate more random values $\{a_{i,t}\}$ and also more exponentiation operations to get $\{A_{i,t}\}$ for the increasing number of attributes or possible values for each attribute.

Then, TA needs to encrypt a message under a ciphertext policy W in the **Encrypt** process, whose computation costs are shown in Fig.4. In such graph, the number of leaf nodes in policy indicates the amount of attribute types that have been used in the access tree structure, while $|W_i|$ represents the number of values that were actually used in the leaf node i. The simulation is conducted given $n_i = 4$, which introduces different number of exponentiation and random selection operations for cases $|W_i| = 2, 3, 4$. For example, $\{C_{i,t,2}\}$ in the ciphertext CT consists of two *well-formed* values and two *malformed* values for the case of $|W_i| = 2$ while four *well-formed* values and no *malformed* values for the case of $|W_i| = 4$. Therefore, the computation cost decreases when $|W_i|$ increases due to the fact that random selection operation takes longer time than exponentiation operation. Besides, the encryption process runs in time almost linearly in the number of leaf nodes in policy.

At user's side, its attribute list L should be firstly sent to TA for generation of attribute credentials. This process is characterized by **Keygen** and its computation cost is displayed in Fig.5. As expected, the running time of generation of attribute keys is almost linear with respect to the number of

user's attributes. It can also be observed that the computational cost is very low even for large number of attributes, which is beneficial for light-weight portable devices.

After attribute keys are generated, the registered user sends them to spectrum database to decrypt the ciphertext CT in order to gain access to white space channels. The database then performs decryption as described in the process of **Decrypt**. The corresponding computational costs with respect to different number of attributes are shown in Fig.6. It can be seen that the running time is precisely linear in the number of attributes associated with number of issued attribute credentials. The reason is clear that larger number of attribute credentials causes increasing number of exponential and multiplication operations in the same scale. Additionally, the time cost in our test machine is very low for large number of attributes, let alone the industrial database with significant amount of computational power such as Google.

VI. Related Work

Recent researches of security and privacy issues in database-driven CRNs can be generally divided into three categories: preserving location privacy of SUs, protecting operational privacy of PUs, and securing database access protocol.

In [8], Gao et al identified that the spectrum utilization footprint could leak SUs' location privacy. They proposed a countermeasure that SUs utilized the most stable licensed channels with as few channel switches as possible to preserve SUs' location privacy. However, their scheme relies on the knowledge of PUs' information and may leak PUs' privacy, which becomes a serious issue when PUs are federal/military systems. To preserve PUs' operational privacy, Bahrak et al. [9] applied *k*-anonymity and proposed *buffer-time-slots* technique to obfuscate PUs' geolocation as well as operation time information in spectrum database. However, mapping PUs' operating area and time into a larger domain would inevitably impact the efficiency of spectrum utilization by SUs. Besides, even though PUs' operational specifications are obfuscated to a certain level, attackers can still obtain partial information which maybe enough for sophisticated adversaries to attack federal/military systems.

To address the problem of spectrum efficiency degradation in [9], Zhang et al in [10] applied differential privacy to guarantee certain level of location privacy for both PUs and SUs while achieving bilateral utility maximization. However, they only focused on preserving PUs' location privacy but ignoring PUs' other sensitive operational information. In [11], Robertson et al proposed a scheme where PUs and database collaborate to broadcast false spectrum utilization information to prevent adversary from detecting PUs' presence/absence information. Even though it excluded legitimate SUs and sacrificed spectrum utilization efficiency, they claimed that PUs' privacy could be preserved. However, sophisticated adversaries with sensing capabilities can still obtain PUs' actual operating activities.

Recent IETF draft [12] suggested using transport layer security (TLS) to address the threats to access protocols of spectrum database, including impersonation attacks, man-in-the-middle-attacks and etc. However, applying cryptography to preserving PUs' operational privacy has rarely been studied.

VII. Conclusion

In this paper, we presented a policy-based privacy-preserving scheme for PUs' operational specifications in spectrum database. On the one hand, our proposed scheme leveraged attribute-based encryption to generate attribute credentials based on users' operational specifications. On the other hand, a control message issued by TA for access control was encrypted by a hidden policy and can only be decrypted by the user of qualified attribute credential. The security analysis has demonstrated that our proposed scheme could protect PUs' operational privacy from curious database and malicious attackers while has no effect on the normal usage of spectrum database. The performance analysis has also shown that the proposed scheme has limited computational overhead. In future research, we will extend our work by elaborating the construction of access tree to obfuscate the policy to achieve the policy anonymity.

References

[1] J. Chen, Q. Yu, B. Chai, Y. Sun, Y. Fan, and X. Shen, "Dynamic channel assignment for wireless sensor networks: a regret matching based approach," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, January 2015, pp. 95-106.

[2] Federal Communications Commission, "Third Memorandum Opinion and Order," in *FCC 12-36*, May 2012. [Online]. Available: http://hraunfoss.fcc.gov/edocs public/attachmatch/FCC-12-36A1.pdf

[3] Federal Communications Commission, "Enabling innovative small cell use in 3.5 GHz band NPRM & order," in *FCC 12-148*, December 2012.

[4] J.-M. Park, J. Reed, A. Beex, T. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," in *Proc. IEEE*, vol. 102, no. 3, March 2014, pp. 270-281.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, May 2007, pp. 321-334.

[6] S. Mller, and S. Katzenbeisser, "Hiding the policy in cryptographic access control," in *Security and Trust Management*, 2011, pp. 90-105.

[7] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Proc. Appl. Cryptogr. Netw. Security*, LNCS 5037, S. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds., Berlin, Germany, 2008, pp. 111-129, Springer-Verlag.

[8] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *Proc. IEEE INFOCOM*, April 2013, pp. 2751-2759.

[9] B. Bahrak, S. Bhattarai, A. Ullah, J.-M. Park, J. Reed, and D. Gurney, "Protecting the primary users operational privacy in spectrum sharing," in *Proc. IEEE DySpan*, April 2014, pp. 236-247.

[10] Z. Zhang, H. Zhang, S. He, and P. Cheng, "Achieving bilateral utility maximization and location privacy preservation in database-driven cognitive radio networks," in *IEEE MASS*, October 2015, pp. 181-189.

[11] A. Robertson, J. Molnar, and J. Boksiner, "Spectrum Database Poisoning for Operational Security in Policy-Based Spectrum Operations," in *IEEE Military Communications Conference*, November 2013, pp. 382-387.

[12] Y. Wu, and Y. Cui, "Protocol to access white space database: security considerations," July 9, 2012. [Online]. Available: http://datatracker.ietf.org/doc/draft-wu-paws-secutity/

[13] https://crypto.stanford.edu/pbc/