Towards Privacy-Preserving Aggregation for Collaborative Spectrum Sensing

Yunlong Mao, Tingting Chen, Yuan Zhang, Tiancong Wang, and Sheng Zhong

***Abstract*— Collaborative spectrum sensing has become increasingly popular in cognitive radio networks to enable unlicensed secondary users to coexist with the licensed primary users and share spectrum without interference. Despite its promise in performance enhancement, collaborative sensing is still facing a lot of security challenges. The problem of revealing secondary users' location information through sensing reports has been reported recently. Unlike any existing work, in this paper we not only address the location privacy issue in the collaborative sensing to be against semi-honest adversaries, but also take malicious adversaries into consideration. We propose efficient schemes to protect secondary users' reports from being revealed in the aggregation process at the fusion center. We rigorously prove that our privacy-preserving collaborative sensing schemes are secure against attacks from both the fusion center and secondary users. We also evaluate our schemes extensively and verify its efficiency and feasibility.**

***Index Terms*— Location privacy, privacy preserving, collaborative sensing, cognitive radio.**

I. Introduction

With the development of wireless communication and the proliferation of mobile devices in recent years, dynamic spectrum allocation has been considered as an effective way to remedy the shortage of spectrum. Cognitive radio networks in particular have been proposed to make dynamic spectrum allocation possible and increase the efficiency of resource utilization. In cognitive radio networks, unlicensed (secondary) users can sense spectra and tune their transmitters to available channels, which are premised on the fact that their communication does not bring interference to the users with licenses (primary users) [2]. For the reason that primary users have no obligation to help secondary users allocate the spectrum, secondary users need to cognitively sense the spectrum to avoid interference with existing primary users.

In order to effectively avoid interference in cognitive radio networks (CRNs), collaborative sensing has been used to detect the existing communication of primary users (PUs) [3]. In particular, each secondary user (SU) measures the received signal strength (RSS). Then it either forwards the RSS, as a report, to a centralized fusion center, or sends its local decision on whether the licensed communication exists to the fusion center (FC) after analyzing the RSS. The FC collects all the reports from participating SUs and draws a joint conclusion. If the spectrum is idle, the FC will coordinate the SUs to access the available channels. In this way, the spectrum that is not being used by PUs can be fully utilized.

With the increasing popularity of the collaborative spectrum sensing (CSS), some security concerns have been raised. For example, if the reports sent by SUs are altered by an attacker, it may lead to a wrong sensing result at the FC and cause an interference. Even more seriously, collaborative sensing is facing the challenge that secondary users can be malicious and deliberately submit fake or invalid sensing reports. To address these issues, much research work [4]–[7] has been done.

Recently, a new privacy issue, location privacy for SUs in CRNs, has attracted people's attention. Related work [8], [9] has shown that in spectrum sensing, SUs' location information is highly correlated to the RSS after the propagation from PUs to the SUs. Hence attackers can utilize reports to explore SUs' location information. As the first remedy of location privacy issue, Shuai [9] proposes a cryptographic scheme to enable SUs to conceal reports from attackers. A very similar privacy issue has been studied by Zhaoyu [10], [11] in database-driven CRNs. In database-driven CRNs, SUs query a central database to obtain spectrum availability information, then attackers can infer users' location by finding overlapping areas of spectrum the SU has used, which should be studied separately because this is not the model (i.e. "sensing and aggregation") that we are concerned about.

Beyond that, existing work has considered limited attack scenarios and trust models. Shuai's scheme [9] assumes that the FC cannot be more than curious, i.e., it must truthfully perform report aggregation although it wants to reveal SUs' privacy. Zhaoyu's scheme [11] should be performed in semi-honest CRNs with trusted central databases. The differences between our schemes and existing schemes are shown in Table.I. As shown in the table, Shuai's scheme is more relevant to ours, but Shuai's scheme causes significant overhead because of keys management of pairwise encryption.

Manuscript received October 6, 2016; revised January 3, 2017; accepted February 3, 2017. Date of publication February 13, 2017; date of current version March 14, 2017. The work of Y. Zhang was supported by the National Natural Science Foundation of China under Grant NSFC-61402223. This work was supported in part by the National Natural Science Foundation of China under Grant NSFC-61425024 and Grant NSFC-61300235, in part by the Jiangsu Province Double Innovation Talent Program, and in part by the Grant NSFC-61321491. This paper was presented in part at MSWiM'15 [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Stefan Katzenbeisser.

Y. Mao, Y. Zhang, and S. Zhong are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China, and also with the Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China (e-mail: njucsmyl@163.com; zhangyuan@nju.edu.cn; zhongsheng@nju.edu.cn).

T. Chen is with the Computer Science Department, California State Polytechnic University, Pomona, CA 91768 USA (e-mail: tingtingchen@cpp.edu).

T. Wang was with Nanjing University, Nanjing 210023, China. He is now with the University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: go.tcwang@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2668219

TABLE I
ATTACK SCENARIOS AND TRUST MODELS OF DIFFERENT SCHEMES

scheme	Zhaoyu's	Shuai's	Ours
can be malicious?	No	Yes	Yes
can collude with FC?	No	Yes	Yes
can FC be malicious?	No	No	Yes

In our previous work [1], the proposed scheme is more efficient than Shuai's scheme. But both of these two schemes have not considered robustness of schemes. In this paper, we have considered a more robust and more practical approach with threshold cryptosystem to ensure the quality of service, which will take a little more overhead as its cost. Security analysis and experimental results have shown this newly proposed sensing scheme is robust against malicious attacks and nodes' failures. This new scheme makes significant improvement of our previous work.

To protect location privacy of SUs in CRNs, two schemes, with emphasis on efficiency and robustness respectively, will be introduced in this paper. The first scheme, Collaborative Spectrum Sensing with Derivative ElGamal (CSS_{DE}), can protect SUs' location privacy against semi-honest attackers. Then we extend it to defend against malicious attackers (assuming that SUs and the FC can deviate from original protocols). But when we consider scheme's robustness, both Shuai's scheme and our first scheme will be sensitive to SU's malfunction. Hence, we trade some efficiency to design our second scheme, Collaborative Spectrum Sensing with Threshold Cryptosystem (CSS_{TCS}), to be more robust when facing attacks and SUs' malfunction.

Particularly, in CSS_{DE}, we leverage an efficient and flexible cryptographic tool as a key component. Then we carefully design CSS_{DE} with zero-knowledge proof for malicious model. In CSS_{TCS}, we build threshold cryptosystem to defend against attackers and design the system to be failure-resilient. Both of the two schemes secure SUs' location privacy against the FC and other SUs. To summarize, the contributions of this paper are as follows.

- We study the location privacy issue in collaborative spectrum sensing in both semi-honest model and malicious model. We propose two efficient schemes with emphasis on efficiency and robustness respectively, to protect SUs' reports from being revealed in aggregation process.
- We prove that our privacy-preserving sensing schemes are secure against attacks from the FC and SUs in semi-honest model and in malicious model.
- We extensively evaluate the performance of our schemes and verify their efficiency.

The rest of this paper is structured as follows. In Section II, we provide a general introduction of models. In Section III we introduce CSS_{DE} and provide both security analysis and complexity analysis. In Section IV we propose an approach in an entirely malicious model as the extension of CSS_{DE}. In section V, we consider scheme's robustness and introduce CSS_{TCS}. In Section VI we describe simulation experiments we have performed to verify our schemes' feasibility and efficiency. We conclude the paper in Section VII.

II. PRELIMINARY

We will have a brief review of the collaborative sensing model. Aiming at the location privacy issue, we use a classical collaborative sensing model, and then based on this model, our attack models consist of a semi-honest model and a restricted malicious model. The attack scenarios under each model will be introduced. The last subsection is an introduction of a novel cryptographic technique that we have used.

A. Collaborative Spectrum Sensing

We use a centralized cognitive radio model [12], which has a central control unit, known as the FC, to coordinate the work of each SU in the network and hold the right to make decisions regarding each affair. The whole process of CRNs consists of two main parts, collaborative spectrum sensing and spectrum allocation. Our work focuses on the first part, so we will put the details of spectrum allocation aside.

Here is the CRN model that we consider. Each node (including the FC and SUs) in CRN has a set of fully functional radio equipment and every two nodes can establish direct communication. No node has motility. In this CRN, SU set U_s consists of n users $U_s = \{s_1, s_2, \ldots, s_n\}$. There is only one PU U_p concerned, and the channels set $C = \{c_1, c_2, \ldots, c_m\}$ consists of all channels that U_p occupies. The FC is denoted by F. SU s_i's sensing report in U_p's channel c_j, in which $j \in [1, m]$, is denoted by r_i^j, and if we just look at a certain channel every time, we can just use r_i instead. We use R to denote global sensing result the FC gives in the end of sensing.

Now we define a round of collaborative sensing (which will be referred to as round hereafter). The FC confirms participating SUs and assigns the target channel c_j. Once a new round begins, all participants sense the channel c_j, and send their sensing reports r_i^j containing RSS to the FC. When spectrum sensing completes, the FC must give a final global sensing result R^j based on aggregation of SUs' reports. Various methods are available to detect the PU's signal [13]. Generally, we choose the method based on RSS, which follows the distribution below [14]:

$$r_i^j \sim \begin{cases} N(n_0, \frac{n_0^2}{M}), & H_0 \\ N(p_i^j + n_0, \frac{(p_i^j + n_0)^2}{M}), & H_1 \end{cases} \quad (1)$$

In the formula above, we denote s_i's sensing report by r_i^j and n_0 is the additive white Gaussian noise (AWGN). p_i stands for the s_i's received signal power from the primary transmitter on channel c_j. M is the signal sample number. Let H_0 be channel's idle state, and H_1 be channel's busy state. The final result that the FC gives can be described as:

$$R^j = \sum\nolimits_{i=1}^{n} \omega_i r_i^j, \quad (2)$$

In the formula above, ω_i is the weight of s_i's sensing result. We just use equal gain combination (EGC), setting all weights as 1 [14]. Then R^j is the statistical result of the channel c_j.

B. Attack Models

In a semi-honest model, all the parties must follow original sensing protocols but they can keep their own intermediate results. In the restricted malicious model, loosely speaking, only SUs can be malicious. Malicious users may submit arbitrary reports to disturb collaborative sensing result. As for a malicious model, both the FC and SUs could be malicious, who can behave beyond prescribed protocols and nobody can predict their next move. But please note that honest users must be the majority [15] in all situations.

The *attacker* we define against is the one who wants to acquire SU's location information. Either the FC or any SU could be an attacker. We allow attackers to collude. That means a malicious SU can collude with other SUs or the FC. The only assumption is that in CSS_{DE}, if the FC is an attacker, it cannot collude with the Helper (which is to be introduced at the beginning of section III), and they cannot be malicious at the same time. This assumption will be removed in the extension of CSS_{DE}. Due to the little difference in SUs' maliciousness between the semi-honest model and the restricted malicious model, to be succinct, we use the semi-honest model as default setting.

We consider attackers use the same method as in Shuai's paper to get users' location information, and here we briefly describe it. Generally, we consider one attacker s_a in the set of U_s, who casts covetous eyes on location information of some $s_d \in U_s$. First of all, s_a collects as much as possible sample locations' information. Then, s_a classifies the RSS sample data of each region into m classes using the input from two channels, and obtains each cluster's central value. Finally, s_a eavesdrops on s_d's sensing reports in the two channels, and calculates their distance with each cluster's central value. If s_a finds that s_d's distance with cluster k is the minimum distance, then s_a can regard s_d's location the same as cluster k's.

C. Definition of Security

Here we define proper notions of security. Intuitively, we want SUs to know nothing from our schemes, and want the FC to know only a random permutation of all SUs' sensing results. We formalize the above idea using standard cryptographic terms as follows. Let $I = \{1, \ldots, n\}$ be the index set of SUs and $\mathbf{r} = (r_1, \ldots, r_n)$ denotes sensing results from all SUs. Let $\rho(\mathbf{r})$ be a uniformly random permutation of $\mathbf{r}$.

Definition 1 (Security Against Secondary Users): We say a CSS is secure against all SUs in the sense that it reveals **nothing other than the total number of SUs** to all SUs if, given any R and a security parameter t, for each $i \in \{1, \ldots, n\}$, there exists a probabilistic polynomial-time simulator S_i for every probability

$$\{S_i(r_i, n, t)\} \stackrel{c}{\equiv} \{CSS\_View_{s_i}(R, t)\},$$

where $CSS\_View_{s_i}(R, t)$ denotes the view of SU i while it runs the sensing scheme with R being all SUs' sensing results. Here, a user's view consists of its own coin flips and all messages from other participants that it sees in the scheme. The notation $\stackrel{c}{\equiv}$ denotes *computational indistinguishability* (please refer to [16] for a precise definition) of two *probability ensembles* [16]. Intuitively, this definition states that what a SU sees from the scheme can be efficiently simulated by a simulator given that this user's private input, the total number of SUs and a public security parameter are the only inputs. Therefore, we can conclude that the CSS reveals nothing to all SUs. Similarly, we can define the security against the FC as follows.

Definition 2 (Security Against the Fusion Center): We say a CCS is secure against the FC in the sense that it reveals only **a random permutation of all SUs' sensing results** if, given any R and a security parameter t, there exists a probabilistic polynomial-time simulator S_{FC} for every probability

$$\{S_{FC}(\rho(R), t)\} \stackrel{c}{\equiv} \{CSS\_View_{FC}(R, t)\},$$

where $CSS\_View_{FC}(R, t)$ denotes the view of the FC.

D. Derivative ElGamal Encryption

ElGamal encryption algorithm is a classic asymmetric key encryption algorithm. Its encryption result is determined by not only plain text and public key, but also a random integer from encoder. In CSS_{DE}, we use a derivative algorithm of ElGamal encryption [17]. In addition, we modify it to apply to multiple parties. Choose a big prime with form of $p = 2q + 1$, where q is another big prime. Denote a quadratic residues generator in Z_p^* by g, $g \neq 1$. In this scheme, considering that every node in the network including the FC may be untrusted, we separate receiver party's private key into two parts, x_1 and x_2. Both of them are chosen from Z_q randomly, and kept by the receiver. Combine x_1, x_2 to get keys by calculating $x \equiv x_1 + x_2 \ mod \ q$ and $y = g^x \ mod \ p$. Now, we have (p, g, y) as the public key, and (p, g, x) as the private key. Anyone who wants to send a message m with encryption can randomly choose an integer k from Z_q, encrypt plain text m into (g^k, my^k), then send it to the receiver. The receiver firstly decrypts the cipher text with one part of private key x_1 by calculating $my^k g^{-kx_1}$, and then it can get the original message from calculating $my^k g^{-kx_1} g^{-kx_2}$ in another part.

III. Collaborative Spectrum Sensing With Derivative ElGamal

The goal of scheme CSS_{DE} is to ensure that SUs will not expose their location privacy during the process of collaborative sensing in CRNs. But, considering the FC may be an attacker, a preprocessing is needed to protect original data before FC's aggregation. SUs should anonymize their reports, so that the FC cannot match each report with its source. SUs could do this by self-organizing or through a trusted third-party. In order to be more efficient and avoid involving a trusted third-party as much as possible, a SU will be selected to be an assistant *Helper* to prevent attacks from the FC. The Helper, a new role in CSS_{DE}, can be played by any SU. In other words, the Helper is a special SU who assists the FC with the aggregation. Except that, the Helper has the same equipment as other SUs. We can use many existing methods to select a SU as the Helper [13], [18], such as a voting algorithm. Since a SU, as the Helper, will cost more energy to do computation, this role can be played in turn. Many incentive

Algorithm 1 procedure of CSS_{DE}

F, H:
randomly pick $p,\ q,\ p = 2q + 1$, p is l-bit length;
choose one generator of Z_p^* as g ;
H randomly chooses x_1 in Z_q, F randomly chooses x_2 in Z_q;
$x \equiv x_1 + x_2\ (mod\ q)$;
$y = g^x\ mod\ p$, $y_1 = g^{x_1}\ mod\ p$, $y_2 = g^{x_2}\ mod\ p$;

s_i:
foreach $s_i \in U_s$ **do**
  randomly chooses k_i in Z_q;
  $\bar{r_i} \leftarrow (g^{k_i}\ mod\ p, r_i y^{k_i}\ mod\ p)$;
  s_i sends $\bar{r_i}$ to H;
end

H:
re-randomized permuting $(\bar{r_1}, \bar{r_2}, \ldots, \bar{r_n}) \rightarrow (\hat{r_1}, \hat{r_2}, \ldots, \hat{r_n})$;
foreach $\hat{r_i} = (\hat{r_{i,1}},\ \hat{r_{i,2}})$ **do**
  $\tilde{r_{i,2}} \leftarrow \frac{\hat{r_{i,2}}}{(\hat{r_{i,1}})^{x_1}}\ mod\ p$;
end
H sends $((\hat{r_{1,1}}, \tilde{r_{1,2}}), (\hat{r_{2,1}}, \tilde{r_{2,2}}), \ldots, (\hat{r_{n,1}}, \tilde{r_{n,2}}))$ to F;

F:
foreach $i \in [1, n]$ **do**
  $r_i' \leftarrow \frac{\tilde{r_{i,2}}}{(\hat{r_{i,1}})^{x_2}}\ mod\ p$;
end
F aggregates all of the r_i' to get the final sensing result R.

schemes developed for CRNs [19], [20] can be easily applied to compensate for the energy cost, which is out of our concern. As for the location attacks based on physical layer, it is beyond our discussion. Since this type of attack can be widely found in various kinds of networks instead of just aiming at cognitive radio's location privacy, it deserves separate research, and there are many effective methods to solve it, such as [21].

A. Procedure Of CSS_{DE}

CSS_{DE} consists of four steps: initializing, encrypting reports, Helper's decrypting, FC's decrypting and aggregating. Generally, we randomly choose a SU as the Helper for one round before the sensing starts. Since the Helper can also be untrusted, to avoid the situations where the Helper is watched, or the Helper is an adversary itself, we re-randomize permuting the combination of users' sensing reports. However, this will not cause any effect on the final aggregating result. And we will prove CSS_{DE}'s correctness in the end of this subsection.

We use the cryptographic tool in the following way. The receiver's two parts in derivative algorithm are the FC and the Helper, both of which hold part of the private key respectively. All SUs who want to submit sensing report in a sensing round need to encrypt his report with the public key. Then reports are sent to the Helper who will decrypt reports with his part of the private key and re-randomize permuting the match of reports and sources. The FC will get anonymous sensing data by decrypting report with his part of the private key. Our algorithm's specific flow is shown in Alg.1.

Here are some explanations of the algorithm. First of all, a secure length parameter l should be determined, and it can be set with firmware. A generator g of Z_p^* is randomly chosen. The Helper and the FC respectively generate random integer $x_1,\ x_2 \in Z_q$, and x_1, x_2 can not be exposed to others. Then let the Helper and FC work together to get x [1] and y, y_1, y_2. The public key can be sent to all SUs through broadcasting. Once s_i finishes local sensing, s_i encrypts the sensing report with the public key (p, g, y), and sends the encrypted report to the Helper instead of directly to the FC. The Helper re-randomizes permuting sensing reports received, decrypts reports with its part of the private key x_1 and sends result to the FC. The FC decrypts reports from the Helper with another part of private key x_2 to get the original sensing report, do the final aggregation work and announce global collaborative sensing result R.

In both the semi-honest model and the restricted malicious model, an SU attacker can obtain nothing about other SUs' locations even if he colludes with the FC or the Helper. Similarly, if the FC or the Helper is an attacker, he cannot obtain anything about any SU's location except those he colludes with. To keep our statement coherent, we put all these proofs in Theorem.5-7.

Theorem 3: Scheme CSS_{DE} keeps the correctness of collaborative sensing result.

Proof: Recall that the FC receives SUs' reports and give aggregation by $F(\mathbf{r})$. Let $A(\mathbf{r})$ denote CSS_{DE}'s execution. If we can prove that $F(\mathbf{r}) = F(A(\mathbf{r}))$, we can ensure the correctness. The Helper has $\bar{r_i} = (g^{k_i} \mod p, r_i y^{k_i}\ mod\ p)$, $\forall r_i \in \mathbf{r}$, then after randomly permuting, we assume that $\hat{r_j} = \bar{r_i} = (\bar{r_{i,1}}, \bar{r_{i,2}})$. Then the Helper partially decrypts $\hat{r_j}$ by $\tilde{r_{j,2}} = \frac{\bar{r_{i,2}}}{(\bar{r_{i,1}})^{x_1}}\ mod\ p$. Now we have $(\bar{r_{i,1}},\ \tilde{r_{j,2}})$ as input for the FC. Finally, the FC calculates $r_i{}' = \frac{\tilde{r_{j,2}}}{(\bar{r_{i,1}})^{x_2}}\ (mod\ p) = \frac{\bar{r_{i,2}}}{(\bar{r_{i,1}})^{x_1+x_2}}\ (mod\ p)$, where $x_1 + x_2 = x\ (mod\ q)$, and $r_{i,1} = g^{k_i}\ (mod\ p), r_{i,2} = r_i y^{k_i}\ (mod\ p)$, then $r_{i,1}^x = y^{k_i}\ (mod\ p)$, $r_i{}' = \frac{r_i y^{k_i}}{y^{k_i}}\ (mod\ p) = r_i$. $\mathbf{r}'$, the set of $r_i{}'$ is exactly the same set as $\mathbf{r}$. And the FC can give the same result because the aggregation is unrelated to the permutation of reports [22].

Unlike semi-honest SUs, a malicious SU may falsify his sensing report r_m in uncertain ways. But no matter what content is in r_m, the FC can still give the same result as long as the number of malicious SU's false reports is below the aggregation's threshold which is usually set as half the number of SUs [22]. Therefore, if less than half SUs are malicious, CSS_{DE}'s correctness can be kept. ■

B. Security Analysis

Here we formally prove the security of CSS_{DE}. After the Helper is introduced, we should take a new attack scenario into consideration, an SU attacker colluding with the Helper. First of all, we define the security requirement for the Helper which is similarly to the security requirements for SUs and the FC.

[1] we recommend to obtain x by introducing a trusted third party. However if the trusted third party is not available, we can still obtain x with cryptographic protocols easily.

Definition 4 (Security Against the Helper): We say a collaborative sensing scheme is secure against the Helper in the sense that it reveals **nothing other than the total number of SUs** to the Helper if, given any R and a security parameter t, there exists a probabilistic polynomial-time simulator S_H for every probability

$$\{S_H(n,t)\} \stackrel{c}{\equiv} \{CSS\_View_H(R,t)\},$$

where $CSS\_View_H(R,t)$ denotes the view of the Helper.

Theorem 5: Scheme CSS_{DE} is secure against SUs.

Proof: Recall that our definition of security against SUs states what an SU sees from the scheme can be efficiently simulated by a simulator given that only the total number of SUs and his own sensing result are the inputs. According to CSS_{DE}, s_i's view consists of three parts: s_i's internal coin flips cfs, encrypted sensing report $(\bar{r_j})_{j\in I\setminus\{i\}}$ that are sent from other users to the Helper (s_i could know these by eavesdropping the communication between other SUs and the Helper), and the half-decryption results $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ sent from the Helper to the FC (s_i could know these by eavesdropping the communication between the Helper and the FC). Now we construct a simulator S_i as follows.

Given inputs n, t, S_i runs CSS_{DE} alone and uses the coin flips cfs^* to simulate cfs. Also, S_i computes $\bar{r_j}^*$ ($j \in I \setminus \{i\}$) by running the key generation algorithm of Elgamal with security parameter t to generate a random encryption key and uses it to encrypt 1. In addition, S_i uses $(\bar{r_j}^*)_{j\in I\setminus\{i\}}$ to simulate $(\bar{r_j})_{j\in I\setminus\{i\}}$. Similarly, S_i computes n random encryptions of 1 (denoted by $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$) and uses them to simulate $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$.

Clearly, distributions of cfs^* and cfs are the same. Also, due to the multi-messages indistinguishability [16] of Elgamal encryption, $(\bar{r_j})_{j\in I\setminus\{i\}}$ and $(\bar{r_j}^*)_{j\in I\setminus\{i\}}$ are computationally indistinguishable. In addition, it is easy to verify that $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ are n Elgamal encryptions using encryption key y_2, thus $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$ and $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ are computationally indistinguishable according to the multi-messages indistinguishability of Elgamal encryption.

It is easy to see: 1)cfs is independent from $(\bar{r_j})_{j\in I\setminus\{i\}}$ and $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$. 2)$cfs^*$, $(\bar{r_j}^*)_{j\in I\setminus\{i\}}$ and $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$ are pairwise independent. Due to the uniformly random permutation and re-randomization on the ciphertexts performed by the Helper, it can be proved that $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ are random encryptions of a random permutation of all users' sensing results and are independent of $(\bar{r_j})_{j\in I\setminus\{i\}}$. Therefore, we know cfs, $(\bar{r_j})_{j\in I\setminus\{i\}}$ and $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ are pairwise independent, and the two ensemble distributions of $(cfs, (\bar{r_j})_{j\in I\setminus\{i\}}, ((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I})$ and $(cfs^*, (\bar{r_j}^*)_{j\in I\setminus\{i\}}, ((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I})$ are computationally indistinguishable. ■

Theorem 6: Scheme CSS_{DE} is secure against the Helper.

Proof: Recall the definition of security against the Helper requires that the Helper knows nothing other than the total number of SUs. We prove this by constructing a simulator S_H as follows.

According to CSS_{DE}, Helper's view consists of two parts: his internal coin flips cfs and encrypted sensing reports $(\bar{r_j})_{j\in I}$. Given inputs n, t, S_H runs CSS_{DE} alone and uses the internal coin flips cfs^* to simulate cfs. It is easy to see that the two distribution ensembles are the same. Also, S_H simulates each $\bar{r_j}$ with a random encryption of 1 generated by running the key generation algorithm of Elgamal with security parameter t to generate a random encryption key and using it to encrypt 1. Due to the multi-messages indistinguishability of Elgamal, the joint distribution of n random encryptions of 1 is indistinguishable with $(\bar{r_j})_{j\in I}$. In addition, it is easy to see that the distribution of coin flips and distribution of the encryption results are independent. Therefore, we can conclude that the ensemble of the coin flips and encryptions generated by S_H are computationally indistinguishable to the Helper's view. ■

Theorem 7: Scheme CSS_{DE} is secure against the FC.

Proof: Recall our definition of security against the FC requires that the FC knows nothing other than a random permutation of all users' sensing reports. We prove this by constructing a simulator S_{FC} as follows.

According to CSS_{DE}, the FC's view consists of two parts: the encrypted sensing results $(\bar{r_j})_{j\in I}$ sent from other users to the Helper (the FC can know these by eavesdropping the communication between SUs and the Helper), and the half-decryption results $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ sent from the Helper to the FC. Given a random permutation $\rho(R)$, S_{FC} generates $(\bar{r_j}^*)_{j\in I}$, $|\rho(R)|$ random encryptions of 1, to simulate $(\bar{r_j})_{j\in I}$ similarly as S_i simulates $(\bar{r_j})_{j\in I\setminus i}$. Again, the computationally indistinguishability follows the multi-message indistinguishability of Elgamal encryption. To simulate $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$, S_{FC} computes $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$ by encrypting $\rho(R)$ using encryption key y_2. The computationally indistinguishability between $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ and $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$ follows the uniformly randomness of the permutation performed by the Helper. Clearly $(\bar{r_j}^*)_{j\in I}$ and $((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}$ are independent. Same as what we have showed in the proof of the security against users, $(\bar{r_j})_{j\in I}$ and $((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}$ are indpendent. Therefore, $\{(\bar{r_j}^*)_{j\in I}, ((\hat{r_{j,1}}^*, \tilde{r_{j,2}}^*))_{j\in I}\}$ and $\{(\bar{r_j})_{j\in I}, ((\hat{r_{j,1}}, \tilde{r_{j,2}}))_{j\in I}\}$ are computationally indistinguishable. ■

C. Complexity Analysis

If spectrum sensing spends more time than the limitation, it may cause the sensing result to be invalid. So it is necessary to analyse the computation complexity of CSS_{DE}. In the first part of algorithm, where the Helper and the FC generate the encrypting model cooperatively, the process can be finished in an invariable time $O(k_1)$. As for the encrypting processes, every SU can do encryption individually. Besides, some fast algorithms of exponent arithmetic can ensure that user's process finishes in another invariable time $O(k_2)$. As for the Helper, the total time of re-randomize permuting process and partly decrypting can be equivalent to $O(n)$. Similarly, FC's decrypting time and aggregating time can be equivalent to $O(n)$. It is evident that CSS_{DE}'s overhead depends on the amount of SUs. Normally, a cognitive radio network can not contain so many SUs to result in an unacceptable overhead which means that CSS_{DE}'s overhead is acceptable.

IV. The Extension of CSS_{DE}

In a semi-honest model and a restricted malicious model, malicious users' effect can be wiped off by voting or statistics.

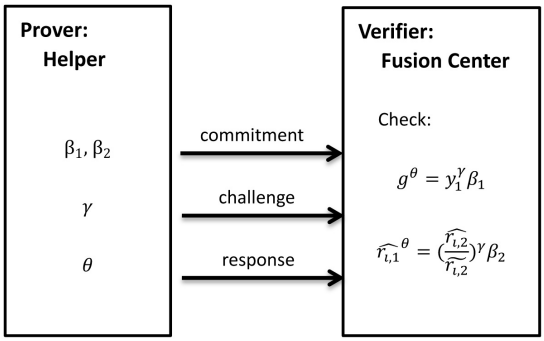


Fig. 1. Non-interactive zero-knowledge proof flow.

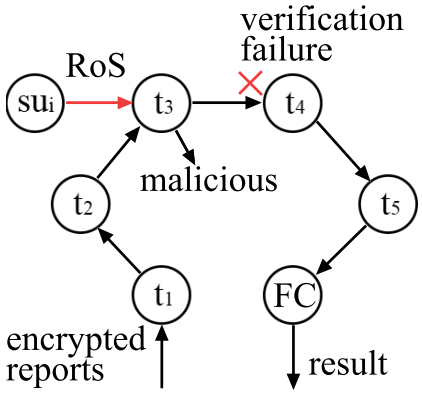


Fig. 2. Illustration of threshold system and failure resilience.

However, if we take a look at the entirely malicious model, where anyone, including the Helper and the FC, can turn into a malicious attacker, CSS_{DE} will probably be disrupted. For example, if a malicious Helper broadcasts part-decrypted reports with re-permutation clues, the FC can easily obtain users' privacy. Aiming to extend CSS_{DE} to be more general and robust, we want to find an effective way to solve the problems emerged with malicious Helper. In fact, we are faced with two questions: how can the FC verify the Helper's identity, and how can the FC trust that the reports the Helper sends are faithfully recorded instead of arbitrarily reports? But after all, we should remember that the FC may be untrusted, so we cannot reveal any information of the Helper in the communication. Thus, we should let the FC obtain no knowledge about Helper's privacy and SUs' privacy except those which are already included in encrypted and re-permuted reports.

In order to solve the two questions we mentioned above, we need to introduce Fiat-Shamir heuristic [23], a paradigm of non-interactive zero-knowledge proof, into CSS_{DE}. The core idea is letting the Helper use non-interactive zero-knowledge proof to prove that he has private key x_1 and the reports he sends are not arbitrary to the FC, using non-interactive zero-knowledge proof. The proof flow can be illustrated with Fig.1.

As the prover, the Helper should prove $\log_{\hat{r_{i,1}}} \frac{\hat{r_{i,2}}}{\tilde{r_{i,2}}} = \log_g y_1$ to the verifier, the FC. The Helper needs to pick α uniformly random from the quadratic residue in Z_p^*, then the Helper gets $\beta_1 = g^{\alpha}$, $\beta_2 = \hat{r_{r,1}}$ as the commitment in standard zero-knowledge proof (ZKP) [24]. A hash function H modeled as a random oracle is needed, and H is a cryptographic hash function whose range is Z_q. So that the Helper can get $\gamma = H(g,\ y_1,\ \hat{r_{i,1}},\ \tilde{ri},1,\ \beta_1,\ \beta_2)$, as challenge in ZKP. The last step is to get $\theta = \gamma x_1 + \alpha$ as response. Then the Helper sends $(\beta_1,\ \beta_2,\ \gamma,\ \theta)$ to the FC, who checks whether $g^{\theta} = y_i^{\gamma}\beta_1$ and $\hat{r_{i,1}}^{\theta} = (\frac{\hat{r_{i,2}}}{\tilde{r_{i,2}}})^{\gamma}\beta_2$ hold. If so, then the FC accepts the proof of the Helper.

V. Collaborative Spectrum Sensing With Threshold Cryptosystem

Our first scheme CSS_{DE} is capable of privacy-preserving sensing in fully functional CRNs with the help of the Helper. But this is under the assumption that the FC cannot collude with the Helper. Under some conditions, this may be not guaranteed. Even in the best situation, if we choose the Helper randomly, there will still be a possibility of choosing a compromised SU as the Helper. To this end, we want to achieve an even more robust scheme without the Helper. Specifically, what we want is a more robust scheme by normalizing the trust distribution of SUs in CRNs.

Shuai's scheme [9] uses pairwise keys encryption which has a normalized trust distribution. However, this encryption scheme has significant disadvantages, such as complex setting up and expensive key maintaining. In a worse situation, Shuai's scheme will be faced with single-point failure because of lacking instantaneous re-keying and key revocation if any SU is disconnected or colluded with the FC. Our CSS_{DE} scheme may suffer single-point failure too because the role of Helper introduces an unbalanced trust distribution. Although we can select another SU to be the Helper after the failure of the prior one, this will lose one round of sensing or reveal some SUs' privacy. To guarantee a more robust privacy-preserving CSS, we leverage a Threshold Cryptosystem (TCS) and propose scheme CSS_{TCS}.

A. Construction of CSS_{TCS}

Threshold cryptography is a powerful encryption scheme which makes it possible to keep system secure as long as the number of malicious users is under a certain threshold (i.e. the adversary corrupts a minority of users). We leverage a non-interactive threshold cryptosystem which can be secure against chosen-cipher attacks (CCA). An illustration of our CSS_{TCS} can be found in Fig.2.

The core component of CSS_{TCS} is the construction of our hash proof system with publicly verifiable and simulation-sound proofs. We implement these proofs following the recommendation of [25], which produces a general framework allowing to construct non-interactive CCA-Secure threshold cryptosystem with adaptive security.

Key Generation: To construct a (t, n)-threshold cryptosystem, a threshold hash proof system Π^{THPS} should be initialized first. We choose to construct Π^{THPS} in a group $\mathbb{G}$ of composite order N, products of two big primes. Then our hardness assumption is subgroup decision problem in composite order groups. Details of this hash proof system's

initializing is not concerned in this work, and please refer to [25] if interested. We recommend to have key generating stage done by certificate agents or a trusted third party. $(pk, \{sk_i\}_{i=1}^n, \{vk_i\}_{i=1}^n)$ should be generated in the following way:

1) Say $N = p1\,p2$, where $p1$, $p2$ are large primes, subject to security parameter λ. Pick generator g from $\mathbb{G}_{p1}$, u, v from $\mathbb{G}$, x from $\mathbb{Z}_N$, set $X = g^x \in \mathbb{G}_{p1}$, randomly.
2) Randomly generate polynomial $P[X] \in \mathbb{Z}_N[X]$ of degree $t-1$, satisfying $P(0) = x$. Compute $Y_i = g^{P(i)} \in \mathbb{G}_{p1}$, $\forall i \in \{1, \ldots, n\}$.
3) Select a pairwise independent hash function H.
4) Select a one-time signature scheme $\Sigma = (Gen, Sig, Ver)$.
5) Define private keys $sk_i = P(i) \in \mathbb{Z}_N$ for each $i \in \{1, \ldots, n\}$, verifying keys $vk_i = Y_i \in \mathbb{G}_{p1}$ for each $i \in \{1, \ldots, n\}$. Then public key $pk = (\mathbb{G}, N, g, X, u, v, H, \Sigma)$.

Encrypted Collaborative Sensing: Based on the assumption that malicious SUs should be no more than half amount of SUs, in our CRN model if we want to construct a (t, n)-threshold where at least t parties should be selected to decrypt the ciphertext, then we should set $t = \lfloor n/2 + 1 \rfloor$ to guarantee the security, where n is the amount of SUs. All SUs should generate their reports using pk in the following way:

1) Generate signature key sk_Σ and verifying key vk_Σ of Σ.
2) Compute $\Phi = g^r \in \mathbb{G}_{p1}$, where r is randomly chosen from $\mathbb{Z}_N$. Blind original report m as $m' = m \oplus H(X^r)$.
3) Generate a proof $\pi_{\mathbb{G}_{p1}} = (u^{vk_\Sigma} v)^r$.
4) Encrypt sensing report $C = (vk_\Sigma, m', \Phi, \pi_{\mathbb{G}_{p1}}, \sigma)$, where $\sigma = \Sigma.Sig(sk_\Sigma, (m', \Phi, \pi_{\mathbb{G}_{p1}}))$.

Verification And Decryption: Generally, we choose a random subset T of all SUs, $|T| = t$, to decrypt reports. Reports $\{m_i\}_{i=1}^n$ from each SU are sent to s_1 and delivered to $s_2, \ldots, s_t \in T$ successively. Order of T can be assigned randomly because SUs are treated equally. Then for each $s_i \in T$, reports should be verified and decrypted as follow:

1) Verify whether C holds that $\Sigma.\mathrm{Ver}(vk_\Sigma, (m', \Phi, \pi_{\mathbb{G}_{p1}}), \sigma) = 1$ and $e(\Phi, u^{vk_\Sigma} v) = e(g, \pi_{\mathbb{G}_{p1}})$ where function e is a bilinear map. If verification passes then continue, otherwise go to failure resilience phase.
2) Generate $K_i = \Phi^{sk_i}$ and $\pi_{K_i} = \epsilon$ which is an empty string.
3) Anonymize reports by re-randomized permutation.
4) Deliver reports and (K_i, π_{K_i}) to the next SU in T.

Aggregation: For the FC, all reports and $\{(K_i, \pi_{K_i})\}_{i=1}^t$ will be received. Then the FC should do aggregation as follow:

1) Verify whether C holds that $\Sigma.\mathrm{Ver}(vk_\Sigma, (m', \Phi, \pi_{\mathbb{G}_{p1}}), \sigma) = 1$ and $e(\Phi, u^{vk_\Sigma} v) = e(g, \pi_{\mathbb{G}_{p1}})$ where function e is a bilinear map. If verification passes then continue, otherwise go to failure resilience phase.
2) Verify whether it holds that $K_i \in \mathbb{G}, vk_i \in \mathbb{G}, \pi_{K_i} = \epsilon$ and $e(g, K_i) = e(\Phi, vk_i)$. If it holds then continue, otherwise go to failure resilience phase.
3) Combine all shared private key as $K = H(\prod_{i \in T} K_i^{\Delta_{i,T}(0)})$ where Δ is the statistical distance. Then original report m_i can be recovered by $m_i' \oplus K^{-1}$.

B. Security Analysis

Based on similar analysis of security with CSS_{DE}, we can show that CSS_{TCS} is also secure against adversaries. Without the existence of Helper role, views of adversaries can be categorized into conventional SUs, threshold SUs, the FC. Considering that the FC and conventional SUs have the same views respectively as they are in the CSS_{DE} scheme, the security of their views can be proved with the same simulation as CSS_{DE}.

Theorem 8: Our CSS_{TCS} scheme is secure against conventional SUs and the FC assuming that Σ is a strongly unforgeable one-time signature. Our threshold hash proof system Π_{THPS} is constructed by following the definition of [25]. Then the security of CSS_{TCS} can be proved directly from Theorem.1 of [25].

Hence, what we need to prove specifically for CSS_{TCS} is the security of threshold SUs. The definition of the security against threshold SUs can be derived from the security against conventional SUs.

Definition 9 (Security Against Threshold SUs): We say a collaborative sensing scheme is secure against any threshold SU in the sense that it reveals **nothing other than the total number of SUs his own sensing result** to the threshold SU if, given any R and a security parameter t, there exists a probabilistic polynomial-time simulator S_T for every probability

$$\{S_T(n, t)\} \stackrel{c}{\equiv} \{CSS\_View_T(R, t)\},$$

where $CSS\_View_T(R, t)$ denotes the view of the threshold SU.

Theorem 10: Scheme CSS_{TCS} is secure against threshold SUs.

Proof: According to CSS_{TCS}, the view of $s_t \in T$ consists of four parts: s_t's internal coin flips cfs, encrypted sensing report $(vk_\Sigma, m', \Phi, \pi_{\mathbb{G}_{p1}}, \sigma)$ that are sent from SUs to the threshold, partially verified reports $\Phi^{sk_{t-1}}$ and $\pi_{K_{t-1}}$ sent from the prior threshold SU, and at most $t-1$ secret keys which are obtained by the worst SUs' colluding case. Now we construct a simulator S_T as follows.

Given inputs n, t, S_T runs CSS_{TCS} and uses the coin flips cfs^* to simulate cfs. The zero-knowledge proof and the verification of C satisfies zero-knowledge property so that the adversary cannot get any additional information. $\Phi^{sk_{t-1}}$ is asymmetric encryption. It is semantic secure. $\pi_{K_{t-1}}$ is an empty string. Hence, the only uncertain view is $t-1$ secret keys. Generally, we assume these $t-1$ secret keys as $f(1), f(2), \ldots, f(t-1)$. Recall that all n secret keys are randomly generated. Contents of these n secret keys are independent from encrypted reports. Besides, these n secret keys can reveal nothing about decryption information because of the property of Shamir secret sharing. Since cfs, $(vk_\Sigma, m', \Phi, \pi_{\mathbb{G}_{p1}}, \sigma)$, $\Phi^{sk_{t-1}}$, $\pi_{K_{t-1}}$ and $f(1), f(2), \ldots, f(t-1)$ are pairwise independent, if $f(1), f(2), \ldots, f(t-1)$ and $(vk_\Sigma, m', \Phi, \pi_{\mathbb{G}_{p1}}, \sigma)$ are regenerated randomly, views of adversary should be still indistinguishable with his views before. Hence, the views of $s_t \in T$ computationally indistinguishable. ■

C. Failure Resilience

CSS_{TCS}'s security can be guaranteed, but the robustness can be threatened in the malicious model. Generally, if there is any verification failure caused by any malicious SU, the whole scheme will be restarted. This will cause lose of one-round collaborative sensing. In the worst case, our CRN will be malfunctioning if malicious SUs keep disturbing aggregation in this way.

In order to achieve a more robust scheme in the malicious model, and to make the most of threshold cryptosystem, we design CSS_{TCS} to have an important feature of failure resilience. Recall that $\lfloor n/2+1 \rfloor$ SUs are randomly chosen to share private key. Once a verification fails at some SU, let's say $s_j \in T$, although we cannot tell whether s_{j-1} has colluded with prior SUs, we can determine that s_{j-1} is malicious and misbehaving. Then we take s_{j-1} out from potential options of T until this round of sensing ends and replace s_{j-1} with another candidate by uniformly random selection. The whole processing is illustrated in Fig.2. If there is a failure again in subsequent nodes in T, then the misbehaving SU will be replaced through retry of selection (RoS) until no malicious SUs exist in T. Since there is more honest users than malicious ones in CRN model in one round of sensing, our retry of selection can be done in $O(n)$.

Considering that SUs are usually active in some fixed areas in some time slot, we can introduce reputation system (RepSys) to make RoS more efficient and accurate. Some reputation systems such as [26], [27] are very powerful, but we here adapt a compact and classic reputation model to keep CSS_{TCS} efficient and low-cost. Our CSS_{TCS} with RepSys works in the following way. For any SU $s_j \in T$, if its verification fails, then its previous SU in T, s_{j-1} will be punished by adding *one untrusted token* to s_{j-1}'s token indicator. Another SU whose indicator is less than any other SUs in $U_s \backslash T$ will be selected into T. SUs in T will give another try to reveal encrypted reports. When another round of sensing begins, nodes in T will be selected according to users' reputation. Feasibility and practical performance of these two ways to achieve failure resilience will be evaluated in the next section.

VI. Evaluation

Since our schemes' security has been proved and the overhead of the sensing procedure is crucial [28], we perform a series of simulation experiments to evaluate our schemes' efficiency. We use Ubuntu 14.10 operating system, intel i3-4130 CPU, 2GB installed RAM. We implement our schemes with the help of CRE-NS3 [29], which is a cognitive radio extension of ns-3. Since our work focuses on collaborative sensing and aggregation, we ignore other cognitive radio's models in the simulation except necessary components. We modify CRE-NS3 and add our schemes mainly to the spectrum sensing and decision models.

Before an attacker seeks SU's location, necessary preparation is the collection of sample locations' information. Generally, we assume that every SU's location is sampled by the attacker. In order to be scalable for more SUs, we deploy SUs in grid and keep them equidistant, which can be illustrated with Fig.3. Each SU has 802.11g standard wifi MAC with a rate of 54 Mbps. SUs can establish direct communication with each other. According to the research of optimal sensing interval [28], we set the sensing interval to be 150ms. All of our simulating timer starts at the beginning of sensing and ends at the end of sensing decision. All results are average of 100 repeats. Sample positions' location information is recorded and associated with signal strength. In each position, the attacker records the results of 100 rounds collaborative sensing on two channels of PU. Then the attacker binds the central values with positions' labels. The central values of sample positions in two dimensions on channels are recorded for further use.

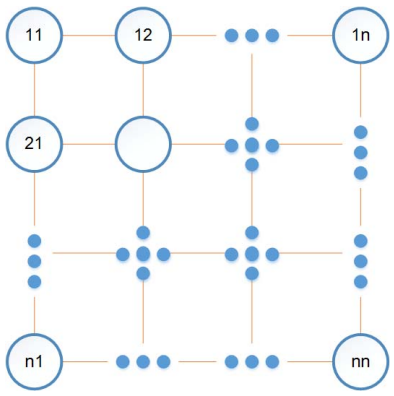


Fig. 3. SUs' locations in CRN.

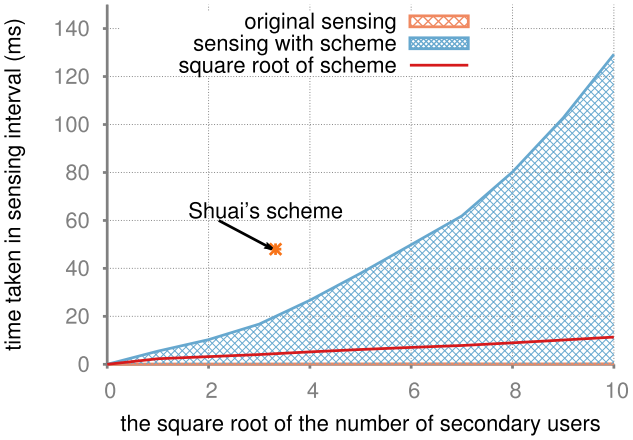


Fig. 4. Running time of CCS_{DE} during the interval.

A. CCS_{DE}

We have measured running time of collaborative sensing for different scales of SUs. Every time, we enlarge the SU set and add enough SUs on grid in a square area. We get every scale's average time to generate Fig.4. Even when the number of SUs is around 100, CCS_{DE} can still work with feasible overhead, which is much less than interval 150ms. Recall that there are three parties in our CRN: SU, the Helper, and the FC. For specific concerns, we measure running time on each party in the CRN. Because there are plural SUs, we record all SUs' running time in a round, and calculate the average for

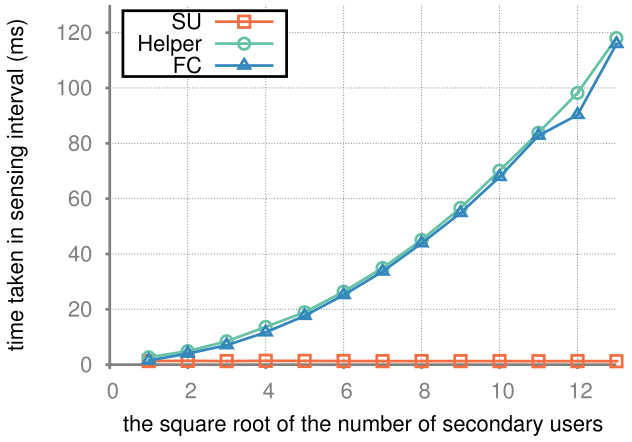


Fig. 5. Running time of CCS_{DE}'s each party.

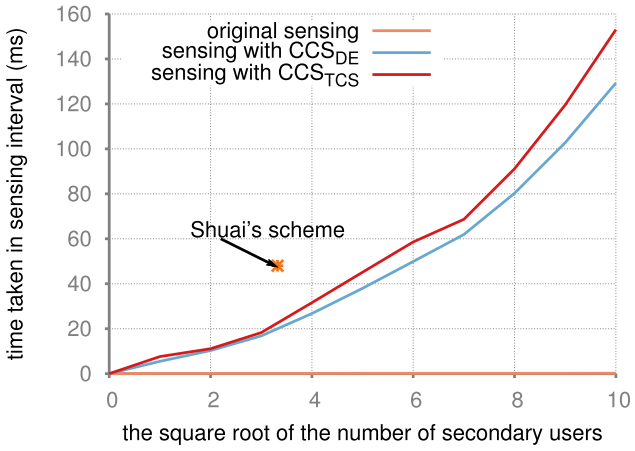


Fig. 7. Running time of CCS_{TCS} with different scales.

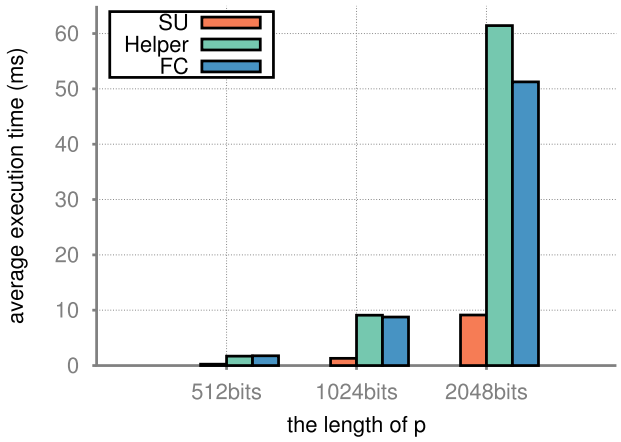


Fig. 6. Average running time with different l.

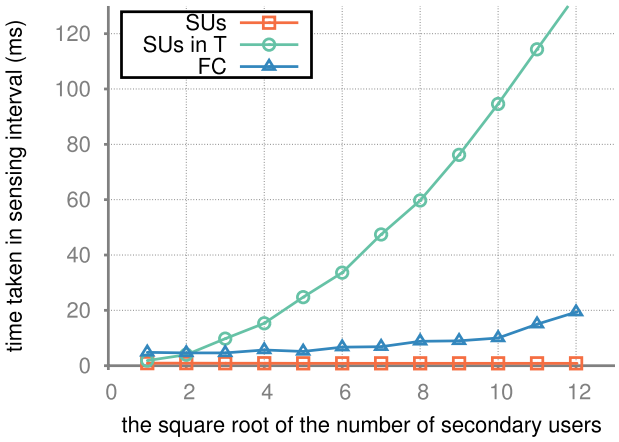


Fig. 8. Running time of CCS_{TCS}'s each party.

each round. From Fig.5, it is obvious that the Helper costs the most of time and has a great proportion on the total running time. And it is reasonable that Helper's running time and FC's running time grows linearly due to the increasing number of SUs, with abscissa being the square root of the number of SUs. The comparing result preliminary shows that when the scheme is applied to the network, a large portion of running time depends on the Helper's efficiency. So if a high-performance node was selected to be the Helper, the total running time it spends would decrease sensibly.

In the experiments above, we use 1024 bits as default set of the length of l, which is the security parameter of CCS_{DE}. To be comprehensive, we measure running time of different lengths of l. In this situation, we set number of SUs as 10. As shown in Fig.6, CCS_{DE} is feasible for commonly used lengths of p. In Shuai's work [9], first they evaluate the computation complexity of their scheme, then they use cryptographic benchmarking data to evaluate the total computation time as roughly 48ms for one aggregation, when the security parameter has 1024 bits and the CRN has 10 SUs. As for CCS_{DE}, we have analyzed that computation complexity is $O(n)$ in Section III. We use ns-3 network simulation tool under the same conditions as Shuai's to evaluate our scheme CCS_{DE}. The average computation time in the same setting is about 20ms for one aggregation. The comparing can be found in Fig.4. And it is obvious that CCS_{DE} can be more feasible in the massive users environment.

B. CCS_{TCS}

We have measured overhead of CCS_{TCS} for different scales of SUs too. To compare with CCS_{DE}, we use the same setting with CCS_{DE}'s experiments. Considering the robustness of CCS_{TCS}, the overhead shown in Fig.7 is acceptable. To the same reason, we evaluate running time of each party in CCS_{TCS}. As shown in Fig.8, parties in CCS_{TCS} have a different pattern with those in CCS_{DE}. The most expensive operations happen on SUs in set T. This is reasonable because SUs in set T have verification, decryption and permutation to do. On the other hand, SUs in T have reduced FC's computing complexity.

In order to verify the feasibility of failure resilience, we set statistic frequency of SUs' malicious behavior to follow a normal distribution $\mathcal{N}(50, 17)$, which means that malicious users are more likely to misbehave while normal users have very small possibility to misbehave. We set 101 SUs to have

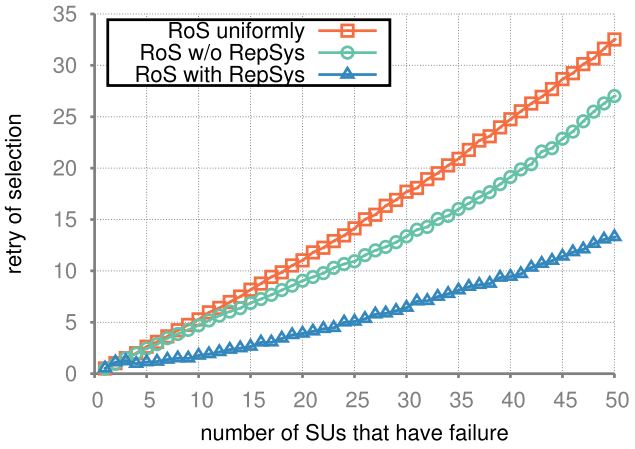


Fig. 9. Number of retrying when SUs have failure with CCS_{TCS}'s failure resilience.

potential failure but only fixed amount of SUs have failure at the same time. As shown in Fig.9, CCS_{TCS} can recover sensing by limited retry. Especially, our CCS_{TCS} with RepSys is very effective in retry of selection (RoS). One important observation is that RoS time is irrelevant with amount of SUs but malicious users ratio. That means, even in a large CRN, CCS_{TCS} can still work out just like in a small CRN.

VII. Conclusion

As the research of cognitive radio continues to improve, and with its outstanding dynamic spectrum accessing, it may well replace the traditional radio in the future. This paper studies the location privacy existing in collaborative spectrum sensing process of CRNs. We formalize privacy issue in both semi-honest model and malicious model. And we take a series of simulating experiments to prove our schemes' validity and we discuss its feasibility by analysing spectrum sensing results. We regard CCS_{DE} as an efficient privacy-preserving scheme for spectrum sensing, which can be extended into a solution for malicious situation when necessary. As for CCS_{TCS}, we think it can be adapted in more general situations for collaborative work and data aggregation when security and robustness are valued more than efficiency. For example, some applications like cooperative localization in wireless sensor networks or cooperative state estimation in smart grid networks.

On the other hand, we are also working on more pertinent schemes for collaborative sensing and more features of collaborative sensing will be utilized. In this way, we hope that we can improve the efficiency or security of our schemes in future work.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable time and efforts in reviewing this paper. Their insightful suggestions helped in improving the quality of this paper significantly.

References

[1] Y. Mao, T. Chen, Y. Zhang, T. Wang, and S. Zhong, "Protecting location information in collaborative sensing of cognitive radio networks," in *Proc. 18th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Nov. 2015, pp. 219–226.

[2] FCC. *Spectrum Inventory Table*, accessed on Oct. 2015, [Online]. Available: http://www.fcc.gov/oet/info/database/spectrum/

[3] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 4. Jun. 2006, pp. 1658–1663.

[4] R. Chen, J. M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 25–37, Jan. 2008.

[5] W. Wang, H. Li, Y. Sun, and Z. Han, "CatchIt: Detect malicious nodes in collaborative spectrum sensing," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2009, pp. 1–6.

[6] L. Duan, A. Min, J. Huang, and K. Shin, "Attack prevention for collaborative spectrum sensing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1658–1665, Sep. 2012.

[7] G. Ding *et al.*, "Robust spectrum sensing with crowd sensors," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3129–3143, Sep. 2014.

[8] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *IEEE Wireless Commun.*, vol. 19, no. 6, pp. 106–112, Dec. 2012.

[9] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 729–737.

[10] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy leaking from spectrum utilization information in database-driven cognitive radio network," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 1025–1027.

[11] 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*, Apr. 2013, pp. 2751–2759.

[12] H. Rifa-Pous and J. Rifa, "Spectrum sharing models in cognitive radio networks," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, 2011, pp. 503–510.

[13] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.

[14] A. Min, K. Shin, and X. Hu, "Secure cooperative sensing in IEEE 802.22 WRANs using shadow fading correlation," *IEEE Trans. Mobile Comput.*, vol. 10, no. 10, pp. 1434–1447, Sep. 2011.

[15] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proc. 19th Annu. ACM Symp. Theory Comput.*, New York, NY, USA, 1987, pp. 218–229.

[16] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[17] S. Zhong, "Privacy, integrity, and incentive-compatibility in computations with untrusted parties," Ph.D. dissertation, Dept. Comput. Sci., Yale Univ. New Haven, CT, USA, 2004.

[18] D. Cabric, S. M. Mishra, D. Willkomm, R. Brodersen, and A. Wolisz, "A cognitive radio approach for usage of virtual unlicensed spectrum," in *Proc. 14th IST Mobile Wireless Commun. Summit*, Jun. 2005.

[19] N. H. Tran, D. H. Tran, L. B. Le, Z. Han, and C. S. Hong, "Load balancing and pricing for spectrum access control in cognitive radio networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1035–1040.

[20] M. Abdelraheem, M. El-Nainay, and S. Midkiff, "Spectrum occupancy analysis of cooperative relaying technique for cognitive radio networks," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, Feb. 2015, pp. 237–241.

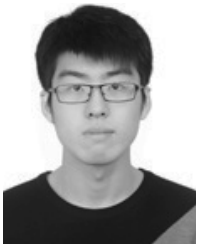
[21] T. Wang and Y. Yang, "Location privacy protection from RSS localization system using antenna pattern synthesis," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2408–2416.

[22] D. Teguig, B. Scheers, and V. Le Nir, "Data fusion schemes for cooperative spectrum sensing in cognitive radio networks," in *Proc. Military Commun. Inf. Syst. Conf. (MCC)*, Oct. 2012, pp. 1–7.

[23] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," *Adv. Cryptol.-CRYPTO*, vol. 263, pp. 186–194, Aug. 1986.

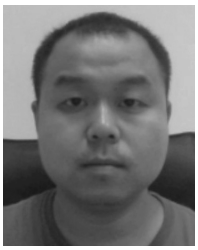
[24] U. Feige A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, 1988.

[25] B. Libert and M. Yung, "Non-interactive cca-secure threshold cryptosystems with adaptive security: New framework and constructions," in *Proc. Theory Cryptogr. Conf.*, 2012, pp. 75–93.
[26] S. Tadelis, "The economics of reputation and feedback systems in E-commerce marketplaces," *IEEE Internet Comput.*, vol. 20, no. 1, pp. 12–19, Jan. 2016.
[27] C. Zhu, H. Nicanfar, V. C. M. Leung, and L. T. Yang, "An authenticated trust and reputation calculation and management system for cloud and sensor networks integration," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 118–131, Jan. 2015.
[28] X. Xing, T. Jing, H. Li, Y. Huo, X. Cheng, and T. Znati, "Optimal spectrum sensing interval in cognitive radio networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2408–2417, Sep. 2014.
[29] A. Al-Ali and K. Chowdhury, "Simulating dynamic spectrum access using ns-3 for wireless networks in smart environments," in *Proc. 11th Annu. IEEE Int. Conf. Sens., Communi. Netw. Workshops (SECON Workshops)*, Jun. 2014, pp. 28–33.

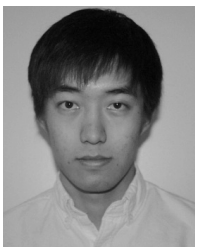


Yunlong Mao received the B.S. degree in computer science from Nanjing University, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include data privacy, security, and computer networks.

Tingting Chen received the B.S. and M.S. degrees in computer science from the Department of Computer Science and Technology, Harbin Institute of Technology, China, in 2004 and 2006, respectively, and the Ph.D. degree from the Computer Science and Engineering Department, State University of New York at Buffalo, in 2011. She is currently an Assistant Professor with the Computer Science Department, California State Polytechnic University, Pomona. Her research interests include data privacy and economic incentives in wireless networks.



Yuan Zhang received the B.S. degree in automation from Tianjin University in 2005, the M.S.E. degree in software engineering from Tsinghua University in 2009, and the Ph.D. degree in computer science from the State University of New York at Buffalo in 2013. His current research interests include security, privacy, and economic incentives.



Tiancong Wang received the bachelor's degree from Nanjing University in 2015. He is currently pursuing the master's degree with the University of Waterloo, with a focus on researching on bioinformatics. He was with COSEC during his undergraduate.

Sheng Zhong received the B.S. and M.S. degrees from Nanjing University in 1996 and 1999, respectively, and the Ph.D. degree from Yale University in 2004, all in computer science. He is interested in security, privacy, and economic incentives.