Providing Efficient and Secure Cooperative Spectrum Sensing for Multi-Channel Cognitive Radio Networks

by

Behzad Kasiri Mashhad

A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg

Abstract

The focus of this thesis is on cooperative spectrum sensing and related security issues in multi-channel cognitive radio networks (MCCRNs). We first study the channel assignment for cooperative spectrum sensing in MCCRNs to maximize the number of available channels. In centralized implementation, a heuristic scheme is proposed along with a greedy scheme to reduce the reported information from the cognitive radios (CRs). In distributed scenario, a novel scheme with multi-round operation is designed following the coalitional game theory. Next, we focus on the physical layer security issues for cooperative spectrum sensing in MCCRNs, caused by Byzantine attacks. New counterattacks are proposed to combat attacks comprising coalition head and CRs as Byzantine attackers, which target to reduce the number of available channels for sensing in distributed MCCRNs. First, a new secure coalition head selection is proposed, by using statistical properties of the exchanged SNRs in the coalitions. Then, an iterative algorithm is proposed to block out attackers, if they continue attacking the system. The important problem of key management is considered next, and an energy-efficient identity-based and a certificate-based distributed key management schemes are proposed. First, a new elliptic curve cryptography (ECC)-based distributed private key generation scheme is proposed to combat the single point of failure problem along with novel distributed private key generator (DPKG) selection schemes to preserve security and energy-efficiency. Because of its importance in the proposed identity-based key management scheme, we further propose a low-complexity DPKG assignment, based on multi-objective programming, which can capture DPKG fairness in addition to energy-efficiency. Finally, a more powerful and intelligent distributed cooperative Byzantine attack on the proposed multi-channel cooperative spectrum sensing is proposed, where attackers collude by applying coalitional game theory to maximize the number of invaded channels in a distributed manner. As a remedy, a hierarchical identity-based key management scheme is proposed, in which CRs can only play on a certain number of requested channels and channel access for sensing is limited to the honest CRs selected in the coalitional game. Simulation results show that the proposed schemes can significantly improve cooperative spectrum sensing and secure the system against Byzantine attacks.

Keywords: Multi-channel cognitive radio networks, cooperative spectrum sensing, coalitional game theory, Byzantine attacks, key management, identity-based cryptosystem.

Acknowledgments

First and foremost, I would like to thank God for all blessings that He has given to me in my life.

I would like to offer my special thanks to my advisors, Dr. Jun Cai and Dr. Attahiru S. Alfa, for their great inspiration, excellent vision, patient guidance and continuous support during my PhD studies, which made it possible for me to accomplish this research and produce the original contributions that comprise this thesis.

In addition, I would like to thank all my examining committee members, including external examiner, Dr. Yu Cheng of Illinois Institute of Technology, and internal examiners, Dr. Pradeepa Yahampath and Dr. Xikui Wang, for dedication of their invaluable time and expertise, and assessment of my thesis. Their insightful comments and suggestions have considerably improved the quality of this thesis. I am thankful to Dr. Yang Wang for serving as the chair of final Ph.D. oral examination.

I am grateful to my colleagues and fellow researchers in the Communication and Network Engineering Research (CNER) Group. Particularly, I would like to thank Dr. Weiwei Wang for his priceless time, constructive discussions, valuable feedback and excellent cooperation during his time at the University of Manitoba.

I would also like to give my sincere thanks to my wife for her invaluable support, encouragement and genuine dedications, along with my parents who have dedicated their lives to me. Without them, this work would not have been possible. I am truly blessed and thrilled to have them in my life.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada IRC and Discovery Grants.

Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

AI-DPKG	All-in DPKG
APML	A Mathematical Programming Language
ATT	Attacker
AWGM	Additive White Gaussian Noise
BF-IBE	Boneh Franklin-Identity-base Encryption
CA	Certificate Authority
CBCKM	Certificate-based Channel Key Management
CDH	Computational Diffie-Hellman
CHA	Coalition Head Attack
CMSS	Cooperative Multi-Channel Spectrum Sensing
CR	Cognitive Radio
CRN	Cognitive Radio Network
CPE	Consumer premise equipment
CSS	Cooperative Spectrum Sensing
DDH	Decisional Diffie-Hellman
DoS	Denial of Service
DPKG	Distributed Private Key Generator
DPL	Discrete Logarithm Problem
EB-DPKG	Energy-based DPKG
ECC	Elliptic Curve Cryptography
ECDL	Elliptic Curve Discrete Logarithm

GDH	Gap Diffie-Hellman
HLA	Higher Level Authority
IBC	Identity-based Cryptography
IBCKM	Identity-based Channel Key Management
ID	Identity
IND-CCA	Indistinguishability under Chosen Ciphertext Attack
JTRS	Joint Tactical Radio System
LB	Lower Bound
MBA	Multi-Channel Byzantine Attack
MCCRN	Multi-Channel Cognitive Radio Network
MCG	Multi-Channel Game
MMAP	Many-to-Many Assignment Problem
NiMH	Nickel Metal Hydride
NIST	National Institute of Standards and Technology
OFDM	Orthogonal Frequency Division Multiplexing
PC	Primary Channel
PKG	Private Key Generator
PKI	Public Key Infrastructure
POMDP	Partially Observable Markov Decision Process
PU	Primary User
RKRL	Radio Knowledge Representation Language
RSA	Rivest, Shamir, and Adelman
SB-DPKG	Security-based DPKG
SBS	Secondary Base Station
SNR	Signal-to-Noise Ratio
SSDF	Spectrum Sensing Data Falsification
UB	Upper Bound

WRAN	Wireless regional area network
WiMax	Worldwide inter operability for microwave access
ZKP	Zero-Knowledge Protocol

Chapter 1

Introduction

1.1 Background and Motivation

The long-established approach for spectrum management, in which a license is assigned to each operator to work within a particular frequency band, is extremely inflexible. Since most of the radio spectrum has already been allocated, it is turning out to be very hard to find unoccupied bands for introducing new services or improving established ones. Motivated by growing attention to wireless services and significantly enlarged demands for radio spectrum, cognitive radio (CR), which allows much more efficient spectrum utilization by dynamic spectrum access, has become an impending asset for upcoming wireless systems to alleviate the spectrum scarcity problem [1, 2]. The basic idea of cognitive radio is to allow unlicensed users to make use of a licensed band. With the intention of enabling access to vacant licensed spectrum, a CR has to check licensed bands and opportunistically occupy them as long as no primary signal is detected. When the primary user (PU) appears, CRs have to vacate the licensed bands to avoid interference to the PUs.

1.1.1 Cognitive radio classification

CRs can be categorized in three different classes.

i) *Ontological CR*: Ontological CRs are the only CRs that utilize learning and reasoning, which make them the most intelligent CRs [3–5]. According to [3], to address the existence of entities and their relationships, and the ways of classifying them based on differences and similarities (as principles of ontological reasoning), Radio Knowledge Representation Language (RKRL) is used. Such ontologies simplify the reasoning engine to understand the environment and make decisions. Ontological CRs do not employ any pre-specified logic to take actions. Instead, they use their own reasoning by considering past cognition cycles.

ii) *Procedural CRs*: Adaptation in this class of CRs is done by observation and applying some fixed algorithms, also called *if-else* rules, [6]. As an instance, the authors in [7] proposed a dynamic frequency selection, where a genetic algorithm is used for adaptation. The outputs of these algorithms are proper actions depending on variety of inputs. Procedural CRs have a deterministic work flow, as their actions can be predicted based on input observations. Therefore, Procedural CRs are less intelligent than Ontological CRs.

iii) *Policy radios*: These radios do not have any learning or reasoning engine, and are the least intelligent CRs. They operate under a range of rules, called the radios policy [5, 8]. In order to select proper rules to follow, policy radios should take into account different parameters of domain knowledge, e.g. the constraints imposed by PU in the specified spectrum, environment, and locations. Such rules are meant to be insurance for PU, by restricting CRs to have minimum interference with PUs. In addition, these rules can be programmed by the operators or implemented in the manufacturing process. However, rules may change, when either device, regulation or regulatory party (licensed network) changes. Thus, policy radios may face regulatory issues.

1.1.2 Cognitive radio networks architectures

Similar to many networks, CRNs fall into two categories, depending on the availability of the core network.

i) *Centralized CRN*: In centralized CRN, a central authority, called secondary base station (SBS), manages the network. For example, a network comprising of a few CRs connected to an access point is a centralized CRN. SBS works as a fusion center to decide on the availability of PCs [9]. Therefore, the SBS has to minimize the interference with PUs. Each CR reconfigures its parameters, based on the decisions made by the SBS.

ii) *Ad hoc CRN*: When a centralized authority does not exist, each CR has to make decisions on the presence of PU and adapt to the environment, independently. This type of CRNs are called ad hoc CRN. In ad hoc CRNs, CRs can either communicate with other CRs, by dynamically using vacant frequency bands or applying existing protocols, e.g. WiFi. As an example, cognitive maritime wireless ad hoc network is proposed in [10] so that maritime users can utilize vacant licensed bands opportunistically. Cooperative schemes play an important role in ad hoc CRNs. It is because local observations of a CR cannot help it beware of the consequence of its action all over the network. Therefore, cooperative schemes can bring a global information of the entire network to each CR [11].

In order to utilize PCs, CRN has to check the their availability [12]. Therefore, each CR has to sense the spectrum, called local sensing. In centralized CRNs, CRs send their local sensing results to SBS, which is a fusion center to decide on the presence of PUs. CRs may send sensing data, e.g., raw energy values, or binary vectors, 0 as absence and 1 as presence of PU, so that SBS can use soft-fusion or hard-fusion rules to make decision, respectively, while in ad hoc CRN, each CR has to fuse its neighbours' local sensing results to make its final decision. The process of sharing local sensing results needs cooperation among CRs, which is referred to as cooperative spectrum sensing.

1.1.3 IEEE Standards for cognitive radio networks

There are two famous standards supporting CRNs, i.e. SCC41 (also known as P1900) and IEEE 802.22. However, some indirectly related standards have been studied for years. One of main concentrations of these standards is coexistence. In many applications, mobile devices have to be able to coexist with others in the same spectrum. For example, different protocols and standards may be used over unlicensed spectrum, e.g. IEEE 802.11 and IEEE 802.15. Power control and dynamic frequency selection are among coexistence techniques, which are similar to the ones developed for CRNs.

IEEE standards for CRNs started by the evolution of coexistence. Early coexistence standards, e.g. IEEE 802.16.2 and IEEE 802.15.2, let interference mitigation through manual coordination, as the cognitive engine. The next generation of standards implemented automation, by considering power control and dynamic frequency selection, e.g. IEEE 802.11h, IEEE 802.16a and IEEE 802.15.4. After that, the main standards for CRs have been developed, which are discussed as follows.

i) *IEEE SCC41*: The focus of this standard is on dynamic spectrum access networks and is co-sponsored by IEEE Electromagnetic Compatibility and Communications Societies. The issues that are considered in SCC41 are network and interference management, and information sharing. The SSC41 considers software defined radio as enabling technology for CRs [13]. The SCC41 has developed policy-based network management for dynamic spectrum access among WiFi, 3G and 4G, and worldwide inter operability for microwave access (WiMax) networks [14].

ii) *IEEE 802.22*: The FCC announced the use of unlicensed access to the analog TV bands in May 2004 [15]. Afterwards, the IEEE 802 Standards committee initiated the 802.22 working group [16] on wireless regional area networks (WRANs) with a air interface based on CR for use by unlicensed users in VHF and UHF (54862 MHz) spectrum, without interference. IEEE 802.22, which put together a significant work on physical and MAC layers, and cognitive domain, is the first standard based on cognitive radios with allocated spectrum. The IEEE 802.22-based networks are cellular networks, which include consumer premise equipments (CPEs) and a base station. CPEs send their local sensing results to the base station for data fusion and decision making. Then, the base station allocates the vacant channels to the associated CPEs.

The IEEE 802.22 standard lets neighbouring cells, which have any coverage overlap, share the frequency band (called self-coexistence) by inter-base station dynamic resource sharing mechanisms. In on-demand spectrum contention, each base station contends for the shared channels [17]. If it wins the contention, other base stations have to switch and leave the channel.

1.1.4 Applications of cognitive radio networks

Besides the aforementioned IEEE standards, CRNs can be useful in different applications. Here we mention some of them as follows.

- *Medical applications*: CRs can be used for emergency medical services to prevent or respond to incidents, and by patients to access emergency services quickly. By using cognitive ID tags for each patient, vital signs of patient such as temperature, pressure and blood oxygen can be simultaneously monitored. In case of any abnormality detection, cognitive ID tags can transmit the signals to clinicians and control services so that they can respond rapidly for diagnosing or treating patients [18].

- *Traffic Management*: In order to cope with traffic problem in congested areas, traffic flow, automatic traffic reports, advices and alternative routes can be transmitted to navigation system of vehicles by traffic management center. Moreover, the cognition capability can be used to monitor and forecast the traffic model, which can be used to determine the duration of red or green signals [18].

- *Rescue*: In situations when the location information of CRs are needed, CRs can find the location of other CRs in trouble by using GPS capability. Moreover, short range signaling on vacant channels can be used as a beacon for the person in trouble [19].

- *Crisis Management*: CRNs can provide communications by deploying ad hoc connections during disasters when normal communications infrastructure has been degraded or destroyed [18]. These networks provide a means for communication in order to reduce the impact of possible future disasters.

- *Mining*: In mining accidents, CRs can select proper transmission parameters and adapt to the situation and environment so that a reliable connection can be established between the outsiders and those who are in the mine [18].

- *Meteorology*: Cognitive sensors can be deployed in a specific region to intelligently collect and share meteorology data, e.g. measurements of heat and humidity, without any human involved, and hand over the information to the central control system so that a more precise meteorology model can be obtained [18].

- *Tactical networks*: One of the most important areas of CR applications is tactical networks. Cognitive radio capabilities can contribute to a narrow-band tactical communications and bring intelligence and environment adaptability in already environmentally-dependent tactical networks. There has been some efforts in this area, e.g. Joint Tactical Radio System (JTRS), and jamming and anti-jamming, that try to apply CR concepts [19].

1.1.5 Cooperative spectrum sensing

The performance of spectrum sensing is usually described by the miss-detection probability (the probability that an occupied PC is declared vacant) and false alarm probability (the probability that a vacant PC is declared to be occupied). The former determines the interference to PUs, and the latter is related to the degradation of the achievable capacity of CRs. Thus, to improve the capacity of CRs with little harm to PUs, the developed sensing algorithms should guarantee that both parameters are constrained over the sensed channel. Due to the limitation on the sensing capacity, the channel fading, and the potential hidden terminal problem, the channel sensing done by a single CR may not meet the required sensing performance, which results in a new technology, called cooperative spectrum sensing (CSS) [20, 21]. Specifically, in CSS, several CRs cooperate with each other to sense one channel simultaneously and with the aid of fusion center, a final decision is made based on all sensing results via fusion rules, such as OR-rule [21], AND-rule [22], or Counting-rule [23]. In the literature, most of the research on CSS focused on single channel systems where all CRs sense the same channel together. Although the sensing performance is guaranteed by CSS, sensing one PC by all CRs is ineffective to improve the capacity of CRs. It is due to the fact that after spectrum sensing, all CRs can only utilize one PC for communication, and thereby, co-channel interference among CRs restricts the capacity improvement. In practice, with the popularity of multi-channel systems, such as orthogonal frequency division multiplexing (OFDM) systems, multiple channels can be sensed by CRs. Thus, it is more important to find more channels satisfying the required sensing performance by CSS and then assign them to different CRs to avoid co-channel interference. In other words, suitably grouping CRs for multi-channel sensing becomes important. Recently, some researchers have begun to study this area. For example, in [24], a partially observable Markov decision process (POMDP) based sensing scheme was proposed to sense part of system channels at the same time, and [25] designed a receiver-aided multi-channel spectrum sensing scheme. However, both schemes are discussed under an additive white Gaussian noise (AWGN) channel and thereby CSS was not considered. In [26], the authors studied how to maximize throughput of CRs for discrete and continuous sensing times in multi-channel CRNs (MCCRNs), while, their scheme was designed for soft-decision fusion only. In addition, the details of channel assignments for sensing was not mentioned. In [27], sensor allocation and quantization schemes in MCCRNs were considered in a centralized fashion. However, the assumptions of error-free reporting channels and assigning equal number of sensors to each PC may not be practical and may result in the reduction on the total number of PCs sensed.

The works mentioned above belong to the narrow-band spectrum sensing, where each sensed channel is sufficiently narrow, and each CR cannot sense large range of frequency spectrum in the system. Recently, a new direction, named wide-band spectrum sensing, has drawn attention for multi-channel spectrum sensing [28]. In this field, the number of channels each CR can sense matches the number of PCs in the system. Thus, the researches are focused on designing detectors to improve the sensing performance and reduce the complexity [29–31]. However, in a real system, the number of PCs can be more than the number of channels each CR can sense. Therefore, assigning different channels to CRs for spectrum sensing becomes a practical issue which has not been covered by wide-band spectrum sensing yet.

1.1.6 Security in cognitive radio networks

Although the CSS can improve the sensing performance of CRNs under channel fading, the request on the cooperation among CRs makes the network more vulnerable to attacks. Therefore, security is one of the major concerns in CRNs [32]. There are some general security requirements in CRNs, e.g. confidentiality, authentication, integrity, etc., [33], which are common in networks. In addition, *Availability* is an exclusive security requirement for CRNs. *Availability* is defined as accessibility of PUs and CRs to the spectrum. This accessibility means that PUs should not be interfered by any unlicensed user, when using a licensed band, and CRs can utilize licensed spectrum bands, when PUs are absent.

The motives of attackers in CRNs can be classified as follows [34].

i) *Selfish attacks*: In this type of attack, the attacker pretends to have higher priority, by changing the transmission parameters and making other CRs believe it is PU, to utilize the spectrum. Therefore, the attacker can individually use the spectrum. CRNs are susceptible to selfish attacks, since the performance of CRs is degraded by improving the performance of the attacker.

ii) *Malicious attack*: In malicious attacks, the attacker prevents other CRs from utilizing a licensed spectrum, leading to *denial of service* (DoS). In DoS attacks, attackers send false local sensing results indicating that the PU is present. While attackers' local sensing results are used for the final decision making, DoS attacks can drastically decrease the availability, by decreasing available spectrum band, resulting in a significant performance degradation [35].

1.1.7 Effects of attacks on cognitive radio networks

Attacks can have two types of effects on CRs.

i) *Direct attack*: The goal of direct attack is DoS. For instance, attackers manipulate the sensing process to make honest CRs believe that PUs are present and prevent them to access the available spectrum. As another example, attackers may use jamming signals on a vacant PC to interfere with honest CRs.

ii) *Induced attack*: The goal of induced attacks is usually to cause regulation and policy issues. Therefore, CRN may face serious outcomes, by violating legal agreements. For instance, any unauthorized channel access induced by attackers, which leads to policy violations falls into this category.

1.1.8 Security in multi-channel cognitive radio networks

One of the important examples of direct attacks that causes DoS on ontological and procedural CRs, is Byzantine attack (BA), where attackers (ATTs) are part of the CRN as compromised CRs. In BAs, ATTs can send false spectrum sensing information to the fusion center by manipulating their own local sensing results, called spectrum sensing data falsification (SSDF) attack [36]. As a result, fusion center makes a wrong decision and spectrum sensing process fails. Although its importance, in literature, few works have been done in this area. Authors in [37] and [38] proposed several methods to combat attacks by eliminating ATTs in a time window from CRN. However, it is assumed that the fusion center is aware of the presence of ATTs and knows the exact number of ATTs. In addition, all CRs have the same sensing capabilities (same probability of detection and false alarm). These assumptions may not be true in practice. In [39], an ATT detection method was proposed based on conditional frequency check statistics considering Markovian spectrum model. However, the work follows the same uniform assumption on CR sensing capacities. Moreover, attack costs to ATTs (such as attack time, attack resources, etc.) are usually missed in the design of attacks. In reality, if the attack cost is not worth compared to the goal of attack, ATTs may withdraw. Besides, the works in the literature considered single-channel CRNs, where there exist only one fusion center and one PC. However, the cooperative spectrum sensing in MCCRNs raises new challenges in security. For example, in distributed cooperative spectrum sensing, fusion center is selected from CRs. Therefore, each CR needs to participate in cooperative sensing over multiple PCs and may be selected as the fusion center as well. Thus, simply adopting traditional single-channel CRN security methods may impose a considerably high processing time and signalling overhead, leading to high energy consumption at CRs. This can be one of the objectives of ATTs, which may not be limited to modification of local sensing decisions. Furthermore, if there is no limitation on coalition formation and sensing, ATTs can easily collude and spread themselves over as many PCs as possible in order to maximize their attack efficiency.

In MCCRNs, in addition to physical layer security mechanisms, key management plays an important role in protecting the network from malicious attacks by managing cryptographic keys for granting access permission to CRs before spectrum sensing and sharing.

In literature, key management has been widely discussed for ad hoc networks [40]. Among all solutions, Elliptic curve cryptography (ECC)-based key management [41] is a promising one, which can achieve energy-efficiency. Compared to Rivest, Shamir and Adleman (RSA) cryptosystem [42], ECC offers equivalent security level with smaller key sizes, faster computing, and savings in power and bandwidth [43]. These characteristics make ECC attractive to networks with limited resources [41]. To provide systems with further lightweight security, identity-based cryptography (IBC), initially proposed by A. Shamir [44], can be applied in ECC-based systems. IBC reduces the communication, computation, and memory costs, and is more suitable for applications in wireless communications. In an IBC system, a client chooses an arbitrary string, called ID, as its public key. Its private key is created by binding ID with a system master secret owned by a central trusted authority, called private key generator (PKG). However, the request on the central trusted authority makes the traditional IBC unsuitable for ad hoc networks, because in such networks, no initial trust exists among users. Moreover, the single point of failure, due to the presence of PKG, dramatically increases the risk to the system security.

One of the solutions is to distribute the power of PKG among all users [45]. PKG distribution can be costly for CRs that serve as distributed private key generator (DPKG), due to their limited energy resources. Moreover, it imposes a huge communication overhead in the system. Therefore, DPKG selection and resource balancing of CRs become important so that CRs with high remaining energy are selected as DPKGs. There have been some attempts to solve key escrow problem, e.g. [46], [47] and [48]. However, they have some limitations. In [46], a trusted offline authority is needed. A general scheme is proposed in [47], which lacks details and still needs trusted entity for bootstrapping. A key issuing scheme is proposed for Boneh-Franklin scheme in [48], by using a combination of users and PKG. However, the problem of single trusted authority exits as PKG is required. Moreover, some key issues, such as DPKG selection and energy consumption problem, have not been taken into account in the literature.

Besides the aforementioned challenges in general ad hoc networks, key management in MCCRNs raises new requirements. For example, in MCCRNs, PCs may come from different operators so that rather than a general key, key management scheme needs to distribute unique keys for each PC. Till now, to the best of our knowledge, our work is the first to address the development of energy-efficient key management suite specifically for MCCRNs.

1.2 Contributions

The objective of this research is to develop effective cooperative spectrum sensing and security mechanisms for MCCRNs, so as to bring reliable communications for future wireless networks. The major contributions of this research can be listed as

1) Cooperative multi-channel spectrum sensing (CMSS): we first propose efficient CMSS mechanisms for both centralized and distributed implementations to enhance utilization by improving the availability.

2) Secure cooperative multi-channel spectrum sensing: we study the security of the proposed CMSS schemes, by emphasizing on the availability, as a unique challenge in MCCRNs. First, Physical layer security is considered against DoS attacks, initiated by Byzantine attackers and new effective countermeasures are proposed. Then, the higher

layer security is considered and new lightweight key management schemes are proposed to properly manage cryptographic keys in MCCRNs. Finally, security against cooperative Byzantine attacks, which impose distributed DoS is studied and a new hierarchical security protocol is proposed to mitigate such attacks.

The detailed contributions of this research are discussed in the followings.

In this research, first of all, we study CMSS in MCCRNs, in both centralized and distributed fashions. In centralized CMSS, two schemes are proposed in order to maximize the number of available channels with low computational complexity. Here, the available channel is defined as one which satisfies sensing performance constraints on both false alarm and miss-detection probabilities. One scheme is called heuristic centralized scheme, which considers the effects of the number of candidate coalitions for each channel, the number of channels associated with the coalition, and the miss-detection probability. In order to relive the undesirable overhead by transmission of signal-to-noise ratio (SNR) reports over all channel in the heuristic centralized scheme, we introduce a second mechanism, called greedy centralized scheme, which can carry out the channel assignment round by round and can limit the number of SNR reports in each round. The distributed CMSS employs the coalitional game theory in order to figure out the best coalition formation structure. The proposed scheme is implemented round-by-round. In each round, for all channels that have not been assigned to anyone for sensing, each CR selects no more than K PCs with the best channel states in terms of SNR. Then, CRs over each selected channel play the coalitional game simultaneously. After the stable coalitional structure is formed, according to the coalition property in terms of miss-detection and false alarm probabilities, each selected channel is assigned to the best coalition. The simulation results verify that both centralized and distributed schemes can significantly increase the number of *available channels* with different complexity levels.

After devising CMSS schemes for MCCRNs, we focus on their security issues. We start with physical layer threats and introduce two new Byzantine attacks (BAs), which aim at degrading the spectrum utilization. The first attack considers the scenario that the coalition head (H_d), as the fusion center, plays ATT's role, called coalition head attack (CHA). Obviously, this can be a significant threat to the network, since the ATT makes the final decision for the whole coalition. In the second attack, called multi-channel Byzantine attack (MBA), ATT plays as a coalition member and sends falsified local spectrum sensing result to mislead the fusion center to make an incorrect decision on the presence of PU. To combat these attacks, two countermeasures are then proposed. For CHA, the probability that coalition head becomes ATT is derived and a new coalition head selection criterion is proposed to prevent selection of an ATT as the coalition head. For MBA, the probability that a CR can be an ATT is introduced as a reputation factor for coalition formation in CMSS and is used to remove ATTs from the final coalition. Simulation results show that the proposed counterattacks can successfully eliminate a significant number of malicious coalition heads and can significantly increase the number of available channels in the presence of ATTs.

In order to manage cryptographic keys for providing security mechanisms to be paired with the physical layer security, a comprehensive bottom-up energy-efficient ID-based key management is proposed for MCCRNs based on the extended Boneh-Franklin identity-based encryption (IBE) [49]. The proposed scheme is based on ECC. To deal with key escrow problem, distributed private key generation is adopted. Different from traditional schemes, in our proposed mechanism, only a group of CRs are selected as DPKGs to significantly decrease the energy consumption and communication overhead in the network. Therefore, instead of deterministic security threshold (when all CRs serve as DPKGs), a location based probabilistic security threshold is proposed to determine the necessary number of DPKGs, by considering the fact that in many civil and almost all tactical scenarios, the probability of compromise is closely related to the location of the users which may be captured and their secret shares are exploited. After determining the number of DPKGs, a security-based DPKG selection along with an energy-based DPKG selection algorithms are proposed, by emphasizing on the security, and the location and energy level of each CR to balance the energy resources among CRs and increase the lifetime of CRs, respectively.

We then propose a distributed private key generation based on ECC to distribute the verifiable shares of PKG master key among DPKGs, while imposing minimum cryptographic overhead. In the proposed algorithm, each DPKG generates its master key share for the other DPKGs along with its commitment for verification. Upon receiving all shares, each DPKG verifies the collected master key shares and generates its master key share. In the proposed key management, each CR holds multiple IDs, including its original ID and IDs corresponding to each requested PC, as opposed to a single ID in traditional IBC systems. Thus, each CR sends its channel key request along with its original ID, as its ID on the demanded PC, to DPKGs in order to obtain the corresponding private key shares. After that by applying threshold secret sharing [50], CR is able to construct its private keys for all demanded PCs. In addition to the proposed ID-based key management scheme, we propose a certificate-based distributed key management scheme based on ECC for the scenarios requiring certificates. The certificate authority (CA) selection is similar to DPKG selection. The proposed scheme has two phases, called general certificate request and channel certificate request. In both phases, the certificate distribution is based on threshold secret sharing. In general certificate request, each CR obtains its general certificate for use in MCCRN, and in channel certificate request, each CR requests its certificate for each demanded PC. Moreover, practical wireless environment and system design parameters (e.g., the most accurate power consumption model including the circuit power, which changes with rate and transmit power) are taken into account to evaluate our proposed schemes. Simulation results show that by maintaining the security bound for DPKG, the proposed DPKG selection can significantly decrease overhead compared to the case, where all CRs serving as DPKG, and is close to that of using single PKG. In addition, it is shown that the proposed DPKG selection can balance the resources of CRs and extend the lifetime of MCCRN. As for the proposed identity-based scheme, it greatly decreases the traffic overhead and increases the average battery life of CRs, making it more energy-efficient than the certificate-based schemes, while ECC certificate-based scheme outperforms all other certificate-based schemes.

To further address the DPKG assignment problem, we propose a new energy-efficient DPKG assignment. The objective is to assign DPKGs to CRs such that the total consumed energy is minimized, while achieving fairness among CRs, i.e., each CR serves as DPKG to almost the same number of DPKGs it gets service from. The problem is modeled as a nonlinear multi-objective optimization problem. A new interactive algorithm is proposed to transfer the problem to a single objective problem by defining a weighted-sum function of the objectives. In the proposed algorithm, the weights are adjusted based on DPKG assignment in each round. We study the optimal solution and bounds of the DPKG assignment problem as the core of the proposed interactive algorithm. Then, a new DPKG assignment algorithm is proposed to facilitate the implementation and decrease the complexity. Simulation results show that the DPKG assignment algorithm performs near optimally and can substantially decrease the total power consumption in the network, compared to the random DPKG selection, while cutting down the computation time of the optimal solution considerably. In addition, both optimal solution and DPKG assignment algorithm can effectively improve the DPKG fairness compared to the optimal scenario without fairness considerations at a marginal increase of total power consumption.

We study the collusion of Byzantine attackers against the proposed coalitional game for CMSS as a powerful distributed DoS attack. A new cooperative Byzantine attack is identified first, which can maximize the number of PCs under attack. The proposed attack is based on coalitional game theory and is performed in two phases, where a different game is played by ATTs in each phase. In phase 1, all ATTs play a coalitional game among themselves so that a minimum number of necessary ATTs can be allocated to each PC. In phase 2, ATTs join honest CRs as regular players and play another coalitional game to form coalitions for sensing.

In order to make the network immune against the proposed cooperative attack, a key management scheme is proposed based on IBC [44]. The selection of ID-based key management results from the fact that it is lightweight and does not require certificate authorities compared to public key infrastructure. The proposed scheme has hierarchical structure consisting of two levels. The first level controls the rights of CRs for participating the game on each PC and is called *game access key management*. In this level, several CRs are first selected as DPKGs. Then, each CR sends its ID along with the vector of the selected PCs as its request to DPKGs. Upon receiving the requests, each DPKG computes its share of the private key for the CR on the requested PC. After gathering all shares, the CR applies threshold secret sharing [50] to compute its private key over each requested PC. In the second level, *sensing key management* is carried out to control sensing permission in each coalition. Compared to the first level, in the second level, all CRs in one coalition serve other members as trusted authorities, which are defined as higher level authorities (HLAs). The final private key for sensing is the combination of the private keys distributed by DPKGs in lower level and HLAs in higher level. At last, we evaluate the performance of the proposed attack and counterattack through simulation and Scyther, a cutting-edge security verification tool [51], respectively. The evaluation results show that the proposed attack can significantly decrease the number of potential channels with a small attack cost, and the proposed key management scheme is effective against the proposed attack and can satisfy different security properties, such as confidentiality, integrity, authentication, etc. It is also shown that the proposed key management scheme is energy-efficient compared to its certificate-based counterpart, especially for multi-channel systems.

The rest of this thesis is organized as follows. Cooperative multi-channel spectrum sensing is studied in Chapter 2. Chapter 3 presents attacks and counterattacks for cooperative multi-channel spectrum sensing. In Chapter 4, new energy-efficient key management schemes are proposed for multi-channel cognitive radio networks. Chapter 5 studies optimal and near optimal DPKG assignment multi-channel cognitive radio networks. In Chapter 6, a distributed cooperative attack on the multi-channel spectrum sensing is proposed along with an ID-based key management scheme as remedy, followed by conclusions and future work in Chapter 7.

Chapter 2

Cooperative multi-channel spectrum sensing

In this chapter, cooperative multi-channel spectrum sensing is discussed. First, system model is described. Then, new centralized and distributed cooperative multi-channel spectrum sensing schemes are proposed to increase the number of *available channels* and improve utilization, followed by the simulation results.

2.1 System model

Consider an MCCRN consisting of M PUs and N CRs which are deployed randomly in a given geographic area. Both centralized and distributed setups are taken into account. In centralized setup, a SBS exists as a central controller, while in distributed setup, all CRs work in an ad hoc manner. In the system, each PU is assigned to one PC so that there are total M PCs available in the system. Due to constraints on hardware and energy consumption, each CR can only sense up to K $(K < M)$ channels at a time [24]. For simplicity, the energy detection is adopted as in [21]. Let PUj occupy channel j ($j \in \{1, \cdots, M\}$). With Rayleigh fading, the sensing performance of CR$_i$ on channel j in terms of miss-detection probability, $p_m^{i,j}$, and false alarm probability, $p_f^{i,j}$, can be obtained, respectively, as [21]

$$
\begin{aligned}
p_m^{i,j} = 1 - [e^{-\frac{\lambda}{2}} \sum_{n=0}^{\tau-2} \frac{1}{n!}(\frac{\lambda}{2})^n + (\frac{1+\gamma_{i,j}}{\gamma_{i,j}})^{\tau-1} \\
\times(e^{-\frac{\lambda}{2(1+\gamma_{i,j})}} - e^{-\frac{\lambda}{2}} \sum_{n=0}^{\tau-2} \frac{1}{n!}(\frac{\lambda\gamma_{i,j}}{2(1+\gamma_{i,j})})^n)],
\end{aligned}
\tag{2.1}
$$

$$
p_f^{i,j} = \frac{\Gamma(\tau, \frac{\lambda}{2})}{\Gamma(\tau)}, \tag{2.2}
$$

where τ and λ are the time bandwidth product and the energy detection threshold, respectively. $\Gamma(\cdot,\cdot)$ is the incomplete gamma function and $\Gamma(\cdot)$ is the gamma function. $\gamma_{i,j}$ is the average SNR of the signal from PUj to CR$_i$, which can be calculated by $\gamma_{i,j} = \frac{P_{PU}\theta_{i,j}^{-\alpha}}{N_0}$. Here, P_{PU} is the transmission power of PU, $\theta_{i,j}$ is the distance between CR$_i$ and PUj, α represents the path loss exponent and N_0 denotes the noise power. Without loss of generality, τ and λ are set to be same for each CR. Thus, $p_f^{i,j}$ in (2.2) becomes a constant for all CRs on each PC and we use p_f instead of $p_f^{i,j}$ for notation simplification.

In the system, CMSS is applied. In CMSS, CRs are first grouped in coalitions, each of which is assigned to sense one PC, with the aid of secondary base station (SBS) in centralized setup or collaboratively in distributed setup. Thus, each CR can belong to multiple coalitions, which sense different channels. Then, each CR senses the associated PCs and submits the sensing results through the pre-defined common control channel to fusion center (the SBS in centralized setup or the coalition head in distributed setup), where the final decision is made based on the fusion rule. The best fusion rule selection depends on network deployment (e.g., the detection threshold, the channel model, and the number of cooperative users). Under the Rayleigh fading channel, it has been shown that OR-rule (i.e., the PC is considered to be idle if and only if all CRs reported so) has better sensing performance compared to others [23, 52, 53]. Thus, OR-rule is adopted in this thesis. We further let the reporting error probability, p_e, be the same for all CRs. However, our scheme can be extended easily for the scenario with different reporting error probabilities. Define $\mathscr{C}_j$ as the coalition sensing PC$_j$ and $|\mathscr{C}_j|$ as the number of elements in set $\mathscr{C}_j$. Then, for centralized setup, the miss-detection and false alarm probabilities of CMSS can be, respectively, calculated as [21]

$$Q_m^j = \prod_{i \in \mathscr{C}_j} [p_m^{i,j}(1-p_e) + (1-p_m^{i,j})p_e], \tag{2.3}$$

$$\begin{aligned} Q_f^j &= 1 - \prod_{i \in \mathscr{C}_j} [(1-p_f)(1-p_e) + p_f p_e] \\ &= 1 - [(1-p_f)(1-p_e) + p_f p_e]^{|\mathscr{C}_j|}. \end{aligned} \tag{2.4}$$

We can similarly derive the equations of the miss-detection and false alarm probabilities for distributed setup as

$$Q_m^{\mathcal{I},j} = p_m^{I,j} \prod_{i \in \mathscr{C}_j, i \neq I} [p_m^{i,j}(1-p_e) + (1-p_m^{i,j})p_e], \tag{2.5}$$

$$\begin{aligned} Q_f^{\mathcal{I},j} &= 1 - (1-p_f) \prod_{i \in \mathscr{C}_j, i \neq I} [(1-p_f)(1-p_e) + p_f p_e] \\ &= 1 - (1-p_f)[(1-p_f)(1-p_e) + p_f p_e]^{|\mathscr{C}_j|-1} \end{aligned} \tag{2.6}$$

where CR $\mathcal{I}$ denotes the coalition head. Obviously, the selection of coalition head only affects $Q_m^{\mathcal{I},j}$. Thus, in order to optimize the sensing performance, the CR which can minimize $Q_m^{\mathcal{I},j}$ should be chosen as the coalition head, i.e., $\mathcal{I}^* = \underset{\mathcal{I} \in \mathscr{C}_j}{\text{argmin}} Q_m^{\mathcal{I},j}$. Note that in practice, the coalition head selection can be implemented after each CR broadcasts its own p_m and p_f. Since the centralized and distributed scenarios are discussed separately, we use the same notations for both scenarios without introducing any confusion and then define $Q_m^j = Q_m^{\mathcal{I}^*,j}$ and $Q_f^j = Q_f^{\mathcal{I}^*,j}$.

Note that the difference between equation pairs of (2.3) and (2.4), and (2.5) and (2.6) comes from the fact that under distributed setup, there is no need for the coalition head to report the sensing results. To limit the interference to PUs and maintain high spectrum efficiency of CRs, Q_m^j and Q_f^j are constrained as $Q_m^j < \overline{Q_m}$ and $Q_f^j < \overline{Q_f}$, where $\overline{Q_m}$ and $\overline{Q_f}$ are two predefined thresholds. The determination of these two thresholds depends on the application and is out of the scope of this thesis. Then, from (2.4) and (2.6), the number of CRs in a coalition for sensing any PC$_j$ in both centralized and distributed scenarios, respectively, should satisfy

$$|\mathscr{C}_j| \leq \lfloor \frac{\log(1-\overline{Q_f})}{\log[(1-p_f)(1-p_e)+p_f p_e]} \rfloor = \mathcal{C}_{max}^{Cent} \tag{2.7}$$

$$|\mathscr{C}_j| \leq \lfloor \frac{\log(1-\overline{Q_f}) - \log(1-p_f)}{\log[(1-p_f)(1-p_e)+p_f p_e]} + 1 \rfloor = \mathcal{C}_{max}^{Dist} \tag{2.8}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation, and $\mathcal{C}_{max}^{Cent}$ and $\mathcal{C}_{max}^{Dist}$ denote the maximum number of CRs in each coalition for centralized and distributed scenarios, respectively. Our aim is to maximize the number of *available channels* to improve the spectrum efficiency of CRs. Notice that, different from [54] which considers one channel only, the system discussed in this paper addresses multi-channel scenario and requires multiple coalitions to sense different channels. Define an allocation matrix $\mathscr{X} = (x_{ij})_{N\times M}$ where $x_{ij} = 1$ indicates that channel j is allocated to CR$_i$ for spectrum sensing, otherwise $x_{ij} = 0$. Then, the general optimization problem can be formulated as

$$\max_{\mathscr{X}} \quad \sum_{j=1}^{M} \mathscr{I}_j(Q_m^j) \tag{2.9}$$

$$s.t. \quad \sum_{j=1}^{M} x_{ij} \leq K, i \in \{1, \cdots, N\} \tag{2.10}$$

$$\sum_{i=1}^{N} x_{ij} \leq \mathcal{C}_{max}, j \in \{1, \cdots, M\} \tag{2.11}$$

$$x_{ij} \in \{0, 1\}, \tag{2.12}$$

where $\mathscr{I}_j(Q_m^j)$ is an indicator function, i.e.,

$$\mathscr{I}_j(Q_m^j) = \begin{cases} 1, & Q_m^j < \overline{Q_m} \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

$\mathcal{C}_{max}$ is $\mathcal{C}_{max}^{Cent}$ or $\mathcal{C}_{max}^{Dist}$ for centralized or distributed setup. The constraint in (2.10) means that the maximum number of PCs assigned to CR$_i$ for sensing is K, and the inequation (2.11) defines the limitation on the number of CRs for sensing channel j, which is equivalent to $Q_f^j < \overline{Q_f}$.

2.2 Upper bound of the optimization problem

Obviously, the optimization problem defined in (2.9) is a nonlinear integer programming problem and is NP-hard [56–58]. Thus, in this section, we try to derive the performance upper bound by relaxing the original problem in (2.9).

First, a linear discrete function $\mathscr{T}_j(x_{ij})$ is derived such that $\mathscr{I}_j(Q_m^j) < \mathscr{T}_j(x_{ij})$. We rewrite (2.3) as

$$Q_m^j = \prod_{i=1}^{N} [p_e + p_m^{i,j}(1-2p_e)]^{x_{ij}}. \tag{2.14}$$

In (2.14), if CR$_i$ does not sense PC$_j$, $x_{ij} = 0$ and it has no contribution to Q_m^j; otherwise, $x_{ij} = 1$. Let $c_{ij}^a = p_e + p_m^{i,j}(1-2p_e)$. By applying natural logarithm, $Q_m^j < \overline{Q_m}$ in (2.13) can be written as

$$\ln(\prod_{i=1}^{N} {c_{ij}^a}^{x_{ij}}) < \ln \overline{Q_m} \tag{2.15}$$

$$\Rightarrow \sum_{i=1}^{N} \ln({c_{ij}^a}^{x_{ij}}) < \ln \overline{Q_m} \tag{2.16}$$

$$\Rightarrow \textstyle\sum_{i=1}^{N} c_{ij}^b x_{ij} < \ln \overline{Q_m}, \tag{2.17}$$

where $c_{ij}^b = \ln c_{ij}^a$.

Since $\frac{\sum_{i=1}^{N} c_{ij}^{b} x_{ij}}{\ln \overline{Q_m}} > 1$, and

$$\mathscr{I}_j(Q_m^j) < \frac{\sum_{i=1}^{N} c_{ij}^{b} x_{ij}}{\ln \overline{Q_m}}, \tag{2.18}$$

we can define $\mathscr{T}_j(x_{ij})$ as

$$\mathscr{T}_j(x_{ij}) = \frac{\sum_{i=1}^{N} c_{ij}^{b} x_{ij}}{\ln \overline{Q_m}}. \tag{2.19}$$

Therefore, the problem (2.9) can be bounded by the following optimization problem

$$\begin{aligned}
\max_{\mathscr{X}} \quad & \sum_{j=1}^{M} \mathscr{T}_j(x_{ij}) && (2.20)\\
s.t. \quad & \sum_{i=1}^{N} c_{ij}^{b} x_{ij} < \ln \overline{Q_m}, j \in \{1, \cdots, M\} && (2.21)\\
& \sum_{j=1}^{M} x_{ij} \leq K, i \in \{1, \cdots, N\} && (2.22)\\
& \sum_{i=1}^{N} x_{ij} \leq L, j \in \{1, \cdots, M\} && (2.23)\\
& x_{ij} \in \{0, 1\},
\end{aligned}$$

where constraint (2.21) results from (2.16).

Moreover, according to (2.9), one possible assignment for sensing channel j, i.e., $\{x_{ij}\}, i \in \{1, \cdots, N\}$, can contribute 1 at most to the objective function. Thus, the $\frac{c_{ij}^{b}}{\ln \overline{Q_m}}$ in (2.19) is unnecessary to be larger than 1. Defining

$$\upsilon_{ij} = \begin{cases} \frac{c_{ij}^{b}}{\ln \overline{Q_m}}, & \frac{c_{ij}^{b}}{\ln \overline{Q_m}} < 1 \\ 1, & \text{otherwise}, \end{cases} \tag{2.24}$$

and combining constraints (2.21) and (2.23), problem (2.20) can be further relaxed as

$$\max_{\mathscr{X}} \quad \sum_{j=1}^{M}\sum_{i=1}^{N} \upsilon_{ij}x_{ij} \tag{2.25}$$

$$s.t. \quad \sum_{i=1}^{N}(c_{ij}^{b}+1)x_{ij} < \ln\overline{Q_m} + L, j \in \{1, \cdots, M\} \tag{2.26}$$

$$\sum_{j=1}^{M} x_{ij} \leq K, i \in \{1, \cdots, N\} \tag{2.27}$$

$$x_{ij} \in \{0, 1\}.$$

Apparently, problem (2.25) defines a many-to-many assignment problem (MMAP) [59], where the "agent" and "task" can be regarded as CR and PC, respectively. Constraint (2.26) indicates that each "agent" can contribute its capacity of $c_{ij}^{b}+1$ to achieve the capacity limitation of the "task", i.e, $\ln\overline{Q_m}+L$, and constraint (2.27) denotes that each "agent" can be assigned K "tasks" at most. Thus, in the defined MMAP problem, both "agent" and "task" have capacity limitation so that each "task" can be assigned to a limited number of agents, and each "agent" can contribute partial capacity to execute one task. In this thesis, the method shown in [59] is adopted to solve the problem (2.25).

Since no algorithms with polynomial complexity exists for deriving the optimal solution of (2.9) due to its NP-hardness, new algorithms with low computational complexity should be proposed for the practical implementation.

2.3 Centralized cooperative multi-channel spectrum sensing

In this section, multi-channel spectrum sensing is considered in the centralized setup where a central fusion center, e.g., SBS, exists. A heuristic centralized scheme is proposed first based on full SNR information from CRs. Then, a greedy centralized scheme is developed to reduce the signaling overhead.

Algorithm 2.3.1 Heuristic centralized scheme

```
Initialization:
    • Each CR reports the SNRs on all PCs to SBS so that SBS can build a matrix
      $\Upsilon = (\gamma_{ij})_{N \times M}$ where each element $\gamma_{ij}$ denotes the SNR of CR$_i$ on PC$_j$;
    • Initialize the allocation matrix $\mathscr{X}$ with each element being $0$.
Main loop:
for $n = 1 : \mathcal{C}_{max}^{Cent}$ do
    RefTab=FormRefTab($\Upsilon$, $\mathscr{X}$, $n$, $\overline{Q_m}$, $\overline{Q_f}$)
    $\mathscr{X}$=AssgCH(RefTab, $\mathscr{X}$)
end for
Output: $\mathscr{X}$
```

2.3.1 Heuristic centralized scheme

The development of the heuristic scheme is based on the following observations.

1) Intuitively, since the number of CRs in the network and the number of channels sensed by each CR are limited, it is better to form coalitions with fewer CRs so that after each assignment, there are sufficient number of CRs left to sense other channels. That is, the channel assignment should be performed starting from the coalitions which contain smaller numbers of CRs.

2) Consider that we are going to allocate channels among coalitions with n ($n \in \{1, \cdots, \mathcal{C}_{max}^{Cent}\}$) CRs. Commonly, different channels may have a different number of candidate coalitions, and the more candidates the channel has, the greater chance it can be allocated. Therefore, we should consider the channels with fewer candidates first.

3) In addition to choosing a channel to be assigned, another issue is to find the best coalition among all candidates. Actually, each candidate coalition may be able to sense different number of channels. Among the candidate coalitions, the one that can sense more channels is more likely to be assigned a channel. Hence, the best approach is to assign the selected channel to the coalition which can sense the minimum number of channels.

Accordingly, we can design a heuristic centralized scheme as in **Algorithm 2.3.1**, which is performed at the SBS after each CR reports SNRs of all PCs.

In **Algorithm 2.3.1**, for facilitating implementation, we define a reference table, where each row consists of data:

- *CHSeq*: the sequence number of PC;
- *NumCoal*: the maximum number of candidate coalitions on the channel indicated by *CHSeq*;
- *CoalSeq*: the sequence number of the possible coalition to which such channel can be assigned;
- *NumCH*: the maximum number of possible channels sensed by the coalition indicated by *CoalSeq*;
- Q_m: the miss-detection probability of the coalition indicated by *CoalSeq* on the channel indicated by *CHSeq*.

For example, for a certain row with 3, 10, 2, 5, 0.03, it means that channel 3 has 10 candidate coalitions. Among these candidates, the coalition 2, which can sense 5 channels, has Q_m of 0.03 on channel 3. Note that, the constraint on Q_f has been considered by limiting the total number of CRs in the coalition according to (2.7).

The main loop of **Algorithm 2.3.1** starts the channel assignment from the coalitions with only one CR. During each loop, a reference table is formed through function FORMREFTAB, as shown in **Algorithm 2.3.2**, and then the channel assignment is implemented on the basis of the reference table in function ASSGCH by **Algorithm 2.3.3**. In function FORMREFTAB, there are three sorting processes. The first one sorts the channels based on the ascending order of the number of candidate coalitions. Since for each channel, there may be several candidate coalitions, the second sorting process is carried out among these candidate coalitions and those which can sense the least number of channels are located at the top. Note that, some rows of the reference table may have the same values for *NumCoal* and *NumCH*. The third sorting process guarantees that the row with the smallest Q_m can be sorted first. The purpose of the third sorting process is to improve Q_m of the system.

Algorithm 2.3.2 Heuristic centralized scheme-FORMREFTAB

```
function FORMREFTAB($\Upsilon$, $\mathscr{X}$, $n$, $\overline{Q_m}$, $\overline{Q_f}$)
    1. Find all possible coalitions including $n$ CRs which are not assigned $K$ PCs for
       sensing, and designate a sequence number to each of them.
    2. Calculate $Q_m$ and $Q_f$ according to (2.5) and (2.6), respectively, for each coalition
       on each unassigned channel, and then keep the coalitions satisfying the $\overline{Q_m}$ and
       $\overline{Q_f}$ constraints.
    3. Build a reference table RefTab including the following information on corre-
       sponding columns.

           | CHSeq | NumCoal | CoalSeq | NumCH | $Q_m$ |

       In this table, the size of the table is determined by all possible combinations of
       the channels, CHSeqs, and coalitions, CoalSeqs.
    4. Sort RefTab according to the following rules one-by-one
       • Sort rows according to the ascending order of NumCoal;
       • Sort the rows with the same NumCoal by the ascending order of NumCH;
       • Sort the rows with the same values of NumCoal and NumCH by the ascend-
         ing order of $Q_m$.
    return RefTab
end function
```

As shown in function ASSGCH, in each loop, only the channel in the first row of the reference table will be assigned. Thus, through the three sorting processes, the channel with the minimum number of candidate coalitions will be assigned to the coalition which can sense the minimum number of channels and has the smallest Q_m. To further understand this algorithm, an example is shown via Table 2.1, derived by applying **Algorithm 2.3.2** when the **Algorithm 2.3.1** runs in the loop of allocating channel to the coalitions with 3 CRs. In the table, three (i.e., coalitions 1, 2, and 3) and six candidate coalitions (i.e., coalitions 2, 3, 4, 5, 6, and 7) can be assigned to sense channels 3 and 5. Thus, according to the first sorting process, the rows for channel 3 are at the top of this table, i.e., channel 3 should be assigned to a coalition first. After that, since the candidate coalitions 1 and 2 have the smallest number of channels for sensing, the rows corresponding to these two coalitions for channel 3 are at the top of table, based on the second sorting process.

Algorithm 2.3.3 Heuristic centralized scheme-ASSGCH

```
function ASSGCH(RefTab, $\mathscr{X}$)
    $Tr$ (total number of rows in the reference table)
    while $Tr \neq 0$ do
        • Assign the channel in the first row to the corresponding coalition;
        • Update the corresponding element of $\mathscr{X}$ to 1;
        • Delete the selected row, and the rows including the assigned channel or the coalitions containing CRs with $K$ assigned channels. During this process, $Tr$ should be decreased by one as long as a row is deleted.
    end while
    return $\mathscr{X}$
end function
```

Finally, by applying the third sorting process via Q_m, the first row indicates that channel 3 should be assigned to coalition 2 since such assignment provides the smallest Q_m. Accordingly, after applying **Algorithm 2.3.3** to Table 2.1, channel 3 is allocated to coalition 2. Moreover, the rows related to channel 3, i.e., rows 1-3, and the row corresponding to coalition 2, i.e., row 4, are deleted. Following similar assignment principle, the channel assignment continues from row 5 and the implementation of **Algorithm 2.3.3** ends when no row is left.

Regarding to the complexity of **Algorithm 2.3.1**, we can deduce it from the size of the reference table. Notice that, after each main loop, some channels have been assigned, and some CRs may have been allocated K channels, so that the number of possible coalitions decreases. Thus, with the algorithm continuing, the size of the reference table decreases significantly and so is the complexity. After implementing **Algorithm 2.3.1**, SBS will broadcast the assignment results to all CRs.

2.3.2 Greedy centralized scheme

In the heuristic scheme, SNRs over all PCs from each CR have to be reported to SBS, which may introduce huge communication overhead. In fact, the constraint that each CR can sense K channels at most indicates that most of the reporting information may not be

Table 2.1: An example of heuristic centralized method

CHSeq	*NumCoal*	*CoalSeq*	*NumCH*	Q_m
3	3	2	2	0.01
3	3	1	2	0.02
3	3	3	3	0.01
5	6	2	2	0.02
5	6	3	3	0.01
5	6	5	4	0.02
5	6	4	4	0.01
5	6	7	4	0.02
5	6	6	5	0.01
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

used for the channel assignment. Thus, to reduce overhead, each CR should report SNRs of PCs selectively. According to (5.9), (5.13), (2.3), and (2.4), large SNR results in small Q_m and Q_f, and from the sensing performance standpoint, each CR prefers to sensing channels with large SNRs. By taking this observation into account, we propose a greedy centralized scheme with significantly reduced reporting overhead. Such scheme is performed round by round. At the beginning of round g, the set of channels which have not been assigned is denoted by $\mathcal{N}_{rd}$, and each CR$_i$ has been assigned k_i^{rd} channels for sensing. Then, by considering the constraint of K, CR$_i$ only reports SNRs of $K - k_i^{rd}$ channels which have the highest SNR values among channels in $\mathcal{N}_{rd}$. After receiving reports from all CRs, SBS has SNR information of some channels in $\mathcal{N}_{rd}$, each of which may include several reports from different CRs. Accordingly, the heuristic centralized algorithm in **Algorithm 2.3.1** can be implemented with fairly small size of the reference table. One round of the greedy scheme is summarized in **Algorithm 2.3.4**. Since some CRs may be assigned K channels at the end of each round, and should be excluded from further considerations, the communication overhead is reduced after each round. The implementation of **Algorithm 2.3.4** is ended when all channels have been assigned.

Algorithm 2.3.4 The greedy centralized scheme (one round)

```
1: Initialization:
     • $\mathcal{N}_{rd}$; $\{k_i^{rd}\}, i = \{1, \cdots, N\}$
2: One round operation:
     • CR$_i$ selects $K - k_i^{rd}$ channels over which it has the highest SNRs among channels
       in $\mathcal{N}_{rd}$ and reports their SNRs to SBS;
     • SBS implements the heuristic centralized scheme based on the possessed information;
     • SBS broadcasts the allocation results, and then each CR updates $\mathcal{N}_{rd}$ and $\{k_i^{rd}\}$
       to $\mathcal{N}_{rd+1}$ and $\{k_i^{rd+1}\}$, $i = \{1, \cdots, N\}$, respectively, for the next round .
```

2.4 Distributed cooperative multi-channel spectrum sensing

In this section, we consider the scenario where SBS is not available. Since there is no central controller, channel assignment process should be implemented distributively by jointly considering the coalition formation and channel assignment. A distributed cooperative spectrum sensing scheme is proposed for the multi-channel cognitive radio network via coalitional game theory.

2.4.1 Multi-channel coalitional game

A coalitional game is described by a pair $(\mathcal{N}, v)$, where $\mathcal{N}$ is the set of players and v denotes the coalition function, which designates a real number, called coalition value, to each coalition [60]. Obviously, in our system, each CR can be regarded as a player. Hence, the key issue is to find a suitable coalition function, i.e., $v_j(\mathcal{S})$ with $\mathcal{S}$ being a coalition over channel j. According to (2.9), our aim is to maximize the number of channels satisfying the constraints on $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$, or the number of *available channels*. Here, $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$ represent the achieved miss-detection probability and false alarm probability by CRs in coalition $\mathcal{S}$ over channel j, respectively. Therefore, $v_j(\mathcal{S})$ should be a decreasing function of $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$ and can be defined, for example, as

$$v_j(\mathcal{S}) = 1 - C_m(Q^j_{m,\mathcal{S}}) - C_f(Q^j_{f,\mathcal{S}}) \tag{2.28}$$

where $C_m(Q^j_{m,\mathcal{S}})$ and $C_f(Q^j_{f,\mathcal{S}})$ are cost functions, which are increasing functions of $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$, respectively. To define these two cost functions, two cases should be considered.
Case 1. $Q^j_{m,\mathcal{S}} < \overline{Q_m}$ and $Q^j_{f,\mathcal{S}} < \overline{Q_f}$

In this case, as long as $Q^j_{m,\mathcal{S}} < \overline{Q_m}$ and $Q^j_{f,\mathcal{S}} < \overline{Q_f}$ are held, the value of indicator function in (2.13) keeps unchanged (or the channel sensed by CRs in $\mathcal{S}$ keeps being available) even if $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$ increase. Nevertheless, different values of $Q^j_{m,\mathcal{S}}$ and $Q^j_{f,\mathcal{S}}$ may affect our objective function indirectly. Intuitively, to maximize the number of *available channels*, the number of CRs in each coalition should be as small as possible, so that more CRs are available to sense other channels. From (2.3) and (2.4), we can deduce that decreasing $Q^j_{m,\mathcal{S}}$ needs to increase the number of CRs in $\mathcal{S}$, while decreasing $Q^j_{f,\mathcal{S}}$ has an opposite requirement. Therefore, in this case, decreasing $Q^j_{f,\mathcal{S}}$ is more important than decreasing $Q^j_{m,\mathcal{S}}$, or when we define cost functions, $C_f(Q^j_{f,\mathcal{S}})$ should dominate $C_m(Q^j_{m,\mathcal{S}})$, i.e.,

$$\min_{Q^j_{f,\mathcal{S}} < \overline{Q_f}, \mathcal{S} \in \mathfrak{C}_j} C_f(Q^j_{f,\mathcal{S}}) > \max_{Q^j_{m,\mathcal{S}} < \overline{Q_m}, \mathcal{S} \in \mathfrak{C}_j} C_m(Q^j_{m,\mathcal{S}}) \tag{2.29}$$

where $\mathfrak{C}_j$ is the set of all possible coalitions over channel j. In other words, $Q^j_{m,\mathcal{S}}$ has an impact only on the coalitions with the same $Q^j_{f,\mathcal{S}}$.

Case 2. $Q^j_{m,\mathcal{S}} \geq \overline{Q_f}$ or $Q^j_{f,\mathcal{S}} \geq \overline{Q_m}$

In this case, the channel sensed by CRs in $\mathcal{S}$ is unavailable. Therefore, such case should be avoided, and both cost functions should tend to infinity.

By considering both cases together, in this thesis, a logarithmic barrier penalty function is selected to define $C_m(Q^j_{m,\mathcal{S}})$ and $C_f(Q^j_{f,\mathcal{S}})$ as

$$C_m(Q^j_{m,\mathcal{S}}) = \begin{cases} -(\overline{Q_m})^\beta \log(1 - (\frac{Q^j_{m,\mathcal{S}}}{\overline{Q_m}})^\beta), & Q^j_{m,\mathcal{S}} < \overline{Q_m} \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.30}$$

$$C_f(Q^j_{f,\mathcal{S}}) = \begin{cases} -\log(1 - \frac{Q^j_{f,\mathcal{S}}}{\overline{Q_f}}), & Q^j_{f,\mathcal{S}} < \overline{Q_f} \\ +\infty, & \text{otherwise.} \end{cases} \quad (2.31)$$

where β is a coefficient guaranteeing (2.29).

According to (2.4), $Q^j_{f,\mathcal{S}}$ increases with the number of CRs in $\mathcal{S}$. Hence, min$C_f(Q^j_{f,\mathcal{S}})$ is determined by the corresponding cost for a single CR, i.e., $C_f(p^{i,j}_f)$. Then, we have

$$\min_{Q^j_{f,\mathcal{S}} < \overline{Q_f}, \mathcal{S} \in \mathfrak{C}_j} C_f(Q^j_{f,\mathcal{S}}) = \min_{i \in \{1,\cdots,N\}, p^{i,j}_f < \overline{Q_f}} C_f(p^{i,j}_f). \quad (2.32)$$

However, for the right-hand side of (2.29), all possible coalitions over channel j should be obtained. Nevertheless, by setting $\beta \geq 2$, it can guarantee $\min\limits_{Q^j_{f,\mathcal{S}} < \overline{Q_f}, \mathcal{S} \in \mathfrak{C}_j} C_f(Q^j_{f,\mathcal{S}}) \gg C_m(Q^j_{m,\mathcal{S}})$ in most cases.

To compare two collections of coalitions, e.g., $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_s\}$ and $\mathcal{R} = \{\mathcal{R}_1, \cdots, \mathcal{R}_t\}$, a *comparison relation* $\rhd$ is defined. Note that $\mathcal{T}$ and $\mathcal{R}$ are formed by the same set of players, i.e., $\bigcup_{a=1}^{s} \mathcal{T}_a$=$\bigcup_{b=1}^{t} \mathcal{R}_b$. Here, the *Pareto order* is applied as a common comparison relation [60]. The *Pareto order* is defined as $\mathcal{T} \rhd \mathcal{R} \Leftrightarrow \{\mathscr{U}_n(\mathcal{T}) \geq \mathscr{U}_n(\mathcal{R}), \forall n \in \mathcal{T}, \mathcal{R}\}$ with at least one strict inequality ($>$) for a player. Here, $\mathscr{U}_n(\mathcal{T})$ and $\mathscr{U}_n(\mathcal{R})$ denote the utilities of the same player n in two different collections of coalitions, i.e., $\mathcal{T}$ and $\mathcal{R}$, respectively, and are determined by the coalition function.

Till now, a N-player coalitional game can be formulated over channel j by applying the *Pareto order*. For the multi-channel case, a same game can be played over each channel. However, simply repeating the same game over multiple channels, may result in a huge interaction overhead. In fact, it is unnecessary to involve all CRs over each channel since each CR can only sense K channels at most. As a result, each CR needs to select channels to play the game. An intuitive method is to choose the first K channels with the highest SNRs of PU signals. It is because according to (5.9), the CR can have the best sensing performance over these channels. Following this way, the number of CRs over each selected channel becomes less than N so that the interaction overhead can be reduced. After the channel selection, CRs can play the game using merge-and-split rule over each selected channel [55], and the coalition structure formed by this rule has the feature that each player has no incentive to leave its coalition, i.e., $\mathbb{D}_{hp}$-stable [54].

2.4.2 Coalition selection

After applying coalitional game, it is possible that there are multiple potential coalitions available for each channel. Since each channel needs to be sensed by one coalition only, intuitively, the best coalition among all possible coalitions is the one with the highest coalition value which results in the fewest CRs to meet the sensing performance requirement. To determine the best one among coalitions formed in the game, the coalition heads on the same channel should interact with each other after the game. We will describe this process in detail in the next subsection.

2.4.3 Distributed CMSS scheme

After coalition selection, several channels are assigned to some CRs for CMSS. However, there is no guarantee that each CR can get K channels for sensing by the aforementioned coalition selection process, since each channel is assigned to one coalition only. Hence, it is necessary to carry out multiple rounds of the proposed coalition selection process till K channels are assigned to each CR for sensing or no channel needs to be assigned. At the beginning of round rd, assume $\mathcal{G}_{rd}$ is the set of unassigned channels and CR$_i$ has been allocated k_i^{rd} channels for sensing. Then, CR$_i$ will select $K - k_i^{rd}(> 0)$ channels with the highest SNRs from $\mathcal{G}_{rd}$, and perform the multi-channel coalitional game. We summarized the one-round operation of the proposed distributed scheme in **Algorithm 2.4.1**. An example with 6 CRs and 8 channels is given in Table 2.2. In Table 2.2, for each CR, the corresponding column represents the channels sorted by the descending order of the SNRs. For example, CR 1 has the highest SNR on channel 3 and the lowest SNR on channel 5. Considering each CR can sense 3 channels at most, i.e., $K = 3$, in the first round, the channels selected by each CR are listed on rows 2 to 4. After that, in Table

Algorithm 2.4.1 Distributed scheme (one round)

```
Initialization:
    • $\mathcal{G}_{rd}$; $\{k_i^{rd}\}, i = \{1, \cdots, N\}$
One round operation:
    • Step 1: CR$_i$ selects $K - k_i^{rd}$ channels over which it has the highest SNRs among the channels in $\mathcal{G}_{rd}$;
    • Step 2: CRs selecting the same channel play multi-channel coalitional game based on merge-and-split rule;
    • Step 3: Each channel selected in this round is assigned to the coalition with the highest coalition value among the formed coalitions on it;
    • Step 4: Update $\mathcal{G}_{rd}$ and $\{k_i^{rd}\}$ to $\mathcal{G}_{g+1}$ and $\{k_i^{rd+1}\}, i = \{1, \cdots, N\}$, respectively, for the next round.
```

2.3, the CRs playing the coalition game on each channel and the coalitions formed through merge-and-split rule are shown in the second and third columns, respectively. For instance, on channel 1, 5 CRs play the coalitional game and two coalitions, i.e., (CR1, CR6) and (CR2, CR4, CR5), are formed with different coalition values. According to Step 3 in **Algorithm 2.4.1**, each channel is assigned to the coalition with the highest value, which is highlighted by boldface in the third column of Table 2.3, e.g., channel 1 is assigned to coalition (CR1, CR6) for sensing. Thus, after step 3, channels 7 and 8 are left for the assignment in the next round and a certain number of channels are assigned to each CR for sensing, e.g., CR1 is assigned channel 1 and 2 and it can be assigned one more channel for sensing in the next round.

Table 2.2: Channels sorted by SNRs over each CR

CR1	CR2	CR3	CR4	CR5	CR6
3	1	4	4	5	1
1	6	3	1	1	3
2	5	5	2	3	6
7	8	6	3	2	7
6	5	1	6	8	5
5	4	4	7	6	6

Table 2.3: Multi-channel coalitional game

Channel	CRs playing coalitional game	Coalitions formed by the game
1	CR1&CR2&CR4&CR5&CR6	**(CR1&CR6)**,(CR2&CR4&CR5)
2	CR1&CR4	**(CR1&CR4)**
3	CR1&CR3&CR5&CR6	(CR1&CR6), **(CR3&CR5)**
4	CR3&CR4	**(CR3&CR4)**
5	CR2&CR3&CR5	**(CR2&CR3&CR5)**
6	CR2&CR6	**(CR2&CR6)**

To implement distributed CMSS scheme, each CR should be aware of $\mathcal{G}_g$ and the end of each round. The latter means that all CRs have finished the coalitional game on the channels selected by themselves in the last round. Meanwhile, after finishing the multi-channel coalitional game in each round, the value of each coalition should be known by others over the same channel so that the one with the highest value on such channel can be chosen. Those requirements can be achieved via the communication over a multi-hop network formed by coalition heads through the common control channel. Here, we consider coalition head since it can gather all required information in the coalition including:

- the channels which are already assigned (it is for deriving $\mathcal{G}_{rd}$);
- whether CRs in the coalition finish the coalitional game on the selected channels or not (it is for being aware of the end of each round);
- the coalition value (it is for coalition selection).

Note that only the CRs within certain distance to the PU can sense the corresponding channel. Such distance can be indicated by an average SNR threshold, i.e., γ_{th}. In other words, each CR is only interested in the channels over which it has SNRs larger than γ_{th}. In addition, each coalition head does not have to interact with all other coalition heads in the network. Therefore, to reduce the overhead, the information on multiple channels sent by each coalition head is delivered to the CRs with acceptable SNRs on those channels. The detailed operation procedure is out of scope of this thesis, and an efficient method can be found from the previous studies, such as [61].

2.5 Simulation results

In this section, a multi-channel cognitive radio network is simulated, which consists of 100 PUs (each PU utilizes one PC) and 50 CRs in a 2Km×2Km square area. The parameters for SNR calculation are $P_{PU} = 0.05$W, N_0=-90dBm, and α=3. Other parameters are set as $\tau = 5$, $p_f = 0.01$, $p_e = 0.01$, $\overline{Q_m} = 0.05$, $\overline{Q_f} = 0.1$ and $\beta = 2$. The maximum number of channels sensed by each CR, i.e., K, is set to 6. In the following, the proposed heuristic centralized scheme in subsection 2.3.1 and the distributed scheme in subsection 2.4.3 are compared with the upper bound in (2.25) first. After that, the simulation results for centralized and distributed scenarios are presented separately.

2.5.1 Centralized schemes vs. distributed scheme

In this subsection, a unified scenario is used to fairly compare three schemes. A SBS, located at the center of the square area, has a coverage area with radius of 1Km. PUs and CRs are randomly distributed in the square area and SBS's coverage area, respectively. The upper bound indicated by (2.25) is derived by IBM CPLEX, using A Mathematical Programming Language (AMPL) codes [62].

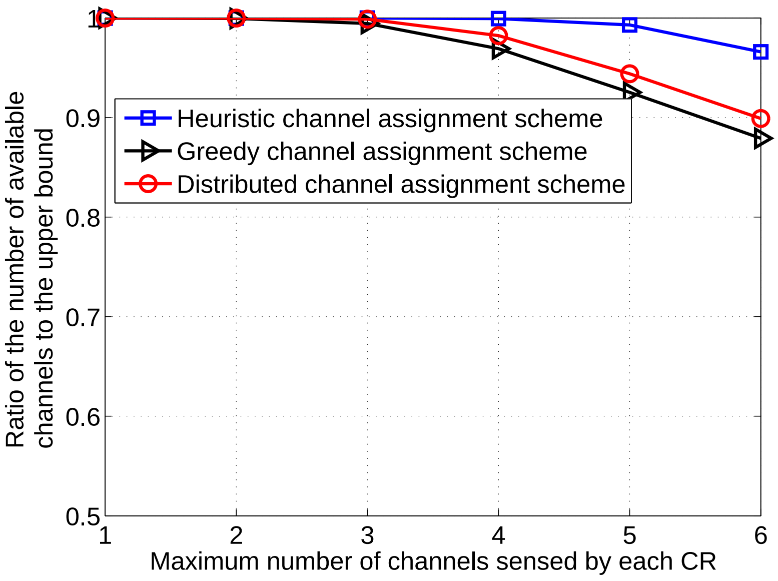


Figure 2.1: Heuristic centralized scheme vs. distributed scheme.

Fig. 2.1 illustrates the comparison results among two centralized schemes and the distributed scheme. The x-axis denotes K and y-axis means the ratio of the average number of the *available channels* to the upper bound. In the figure, three curves decrease with the increase of K. It is because the deviation from the upper bound to the optimal value of (2.9) increases with the increment of K. Specifically, in (2.25), the constraint (2.26) relaxes $Q_m^j < \overline{Q_m}$ (or constraint (2.21)) so that $\sum_{i=1}^{N} \upsilon_{ij} x_{ij}$ may be larger than 1, while $\mathscr{I}_j(Q_m^j)$ in the original problem (2.9) can be 1 at most. Moreover, the number of assignments resulting in such case increases when K increases. Nevertheless, Fig. 2.1 indicates that the heuristic scheme achieves near optimal results with the ratio larger than 0.95. In addition, it is observed that both greedy centralized scheme and distributed scheme are less optimal than the heuristic one because these two schemes cannot utilize all the information (i.e., SNRs for all PCs) to form coalitions. Moreover, the number of *available channels* derived by the distributed scheme is larger than that obtained by the greedy centralized one. It is because the former can achieve better sensing performance (i.e., smaller miss-detection probability and false alarm probability) than the latter one. Specifically, in distributed scheme, the sensing result reporting, impacted by p_e, is not applied to the coalition head. However, in greedy centralized one, each CR in the coalition has to report the sensing result to SBS. Thus, for the coalition with same set of CRs, equations (2.3) to (2.6) indicate that the distributed scheme can achieve smaller Q_m and Q_f than the greedy one.

2.5.2 Centralized scenario

In the simulation, the unified scenario defined in subsection 2.5.1 is reused. Although the greedy centralized scheme has worse performance than the heuristic one, Fig. 2.1 indicates that such degradation is small. Furthermore, by taking overhead into consideration, as shown in Fig. 2.2, the average number of SNRs reported by each CR in greedy scheme is much smaller than that in the heuristic centralized scheme (i.e., the number of PCs).

Thus, by integrating the communication overhead, the greedy centralized scheme outperforms the heuristic one. In this sense, the following discussions focus on the greedy

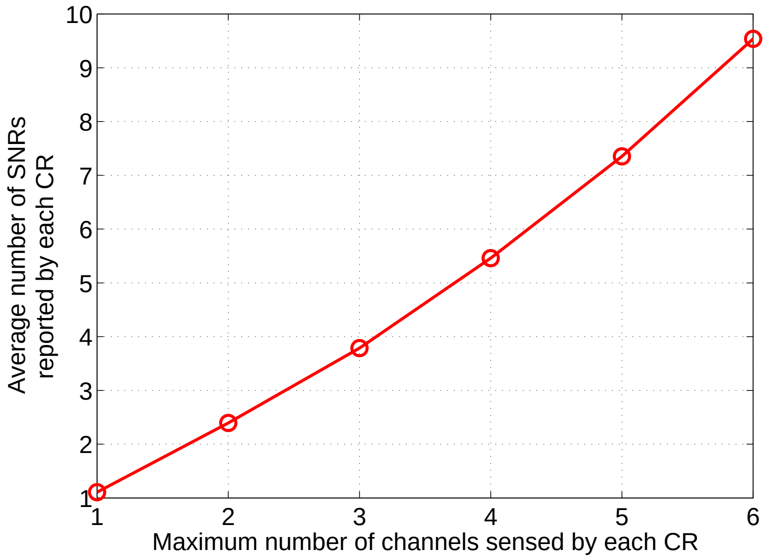


Figure 2.2: Average number of SNRs reported by each CR.

scheme.

Firstly, the performance of greedy centralized scheme is further studied by comparing with a traditional scheme [63], where each CR always selects the first K channels with the highest SNRs for CMSS. Fig. 2.3 presents the average number of *available channels* achieved by both schemes. In the figure, it can be seen that the greedy scheme derives larger number of *available channels* than the traditional one, and the achieved gain increases with the increment of K.

Secondly, the sensing performance of the greedy centralized scheme is studied in terms of the average $Q_f^{I,j}$ and $Q_m^{I,j}$, called average Q_f and Q_m, respectively. Fig. 2.4 demonstrates average Q_f vs. K. According to this figure, the greedy centralized scheme significantly decreases the average Q_f compared to that in the traditional scheme. The average Q_m vs. K is shown in Fig. 2.5. In this figure, the average Q_m in the greedy centralized scheme is larger than that in the traditional scheme. The reason is that in the proposed scheme, the coalition can be formed as long as the Q_m^j satisfies the corresponding constraint $\overline{Q_m}$. However, in the traditional scheme, each CR always selects the channels with the highest SNR for sensing which results in more CRs sensing one channel, i.e., smaller $Q_m^{I,j}$.

Thirdly, the performance of the greedy centralized scheme is studied with respect to α

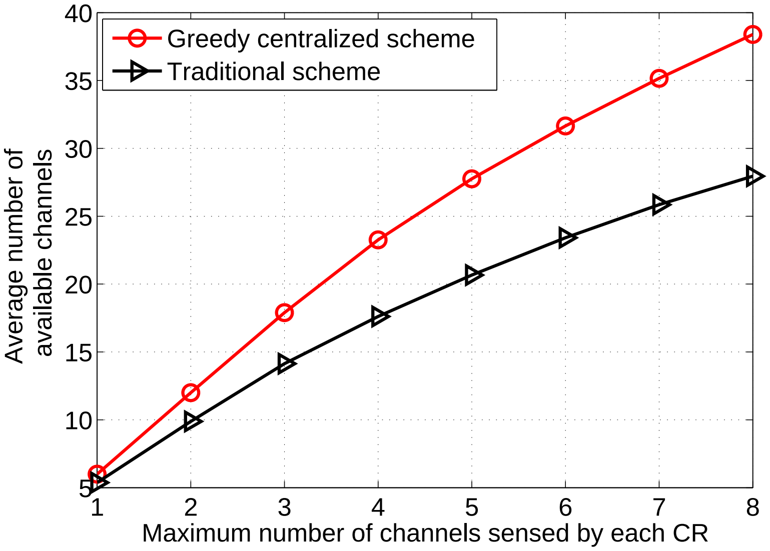


Figure 2.3: Average number of available channels for centralized scenario.

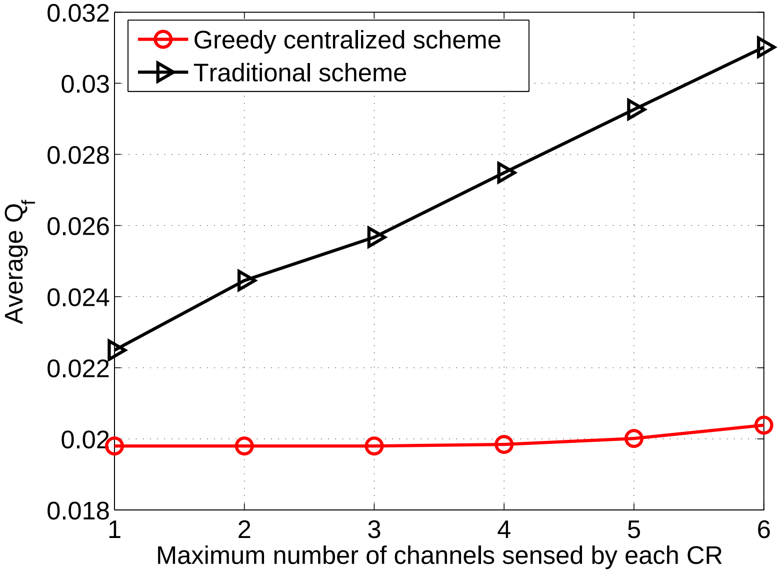


Figure 2.4: The average Q_f of the greedy centralized scheme.

in Fig. 2.6. In the figure, α is set to 2.5 and 3, respectively, with $P_e = 0.01$. Two curves indicate that the number of *available channels* decreases significantly with the increment of α. It is because a larger α causes more signal attenuation, which forces more CRs to sense one channel.

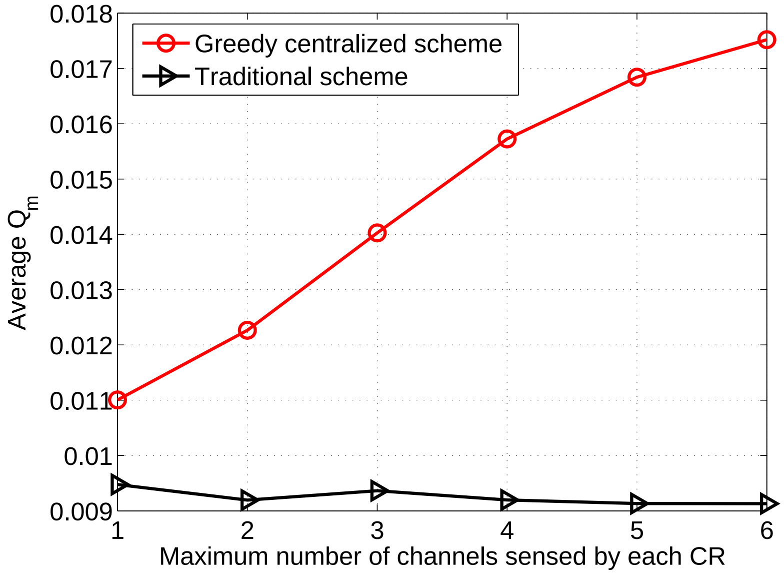


Figure 2.5: The average Q_m of the greedy centralized scheme.

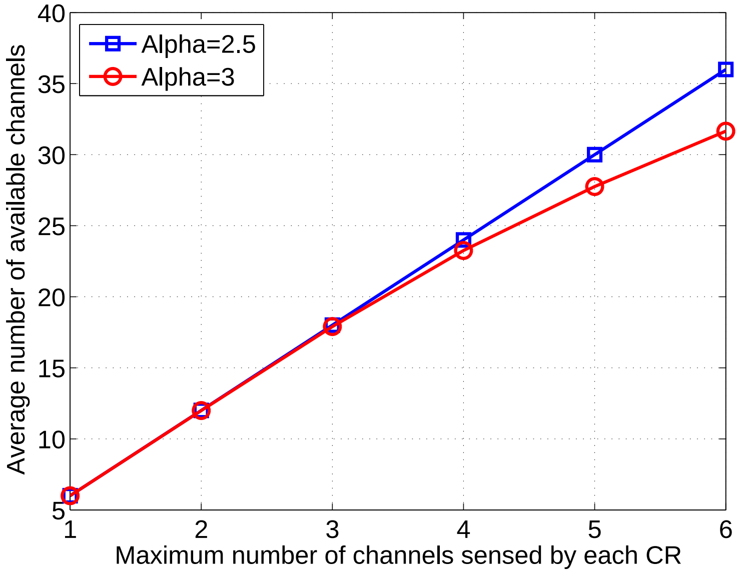


Figure 2.6: The performance of greedy centralized scheme via α.

2.5.3 Distributed scenario

In the distributed scenario, since SBS is not required, all PUs and CRs are deployed randomly in the square area. Similar to the centralized scenario, the traditional scheme, where each CR always selects the first K channels with the highest SNRs for CMSS is selected as the comparison benchmark. Moreover, the coalition head is selected from CRs which choose the same channel.

Fig. 2.7 presents the average number of *available channels* with respect to the sensing capability of each CR, i.e., K. From the figure, it can be observed that the proposed scheme can increase the number of *available channels* significantly. It can be explained by the fact that in the proposed scheme, the channel is assigned to the best coalition and the CRs which are not in this coalition will switch to other channels for sensing. However, in the traditional distributed scheme, no channel switch is carried out even a channel can be sensed by less number of CRs. In addition, Fig. 2.7 shows that the slope of the curve representing the proposed scheme decreases with the increment of K. It is because, when K is large, the remaining unselected channels have worse SNRs. Thus, the number of CRs selecting the same channel increases and the number of selected channels decreases.

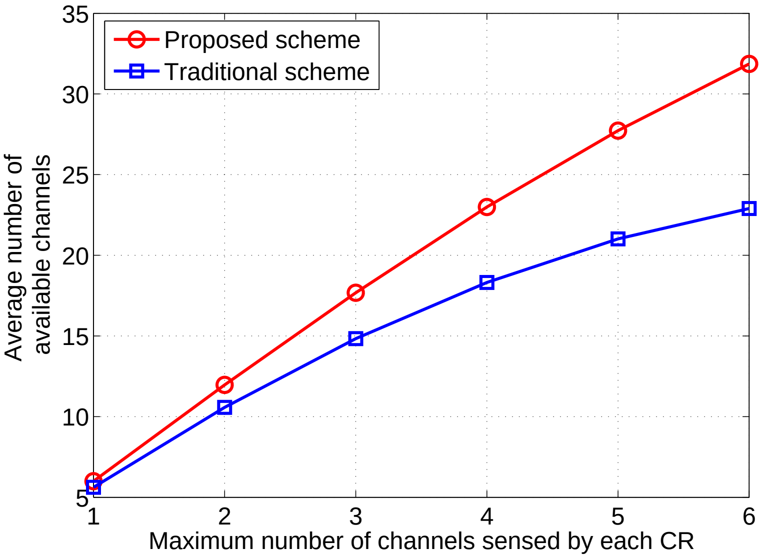


Figure 2.7: Average number of available channels for distributed scenario.

Chapter 3

Cooperative multi-channel spectrum sensing security against Byzantine attackers

In this chapter, we study the physical layer security threats, caused by Byzantine attackers, on the proposed cooperative multi-channel spectrum sensing. Two attacks are proposed along with their proper countermeasures to identify and alleviate the effects of Byzantine attacks.

3.1 Coalition head attack and counterattack

In this section, we introduce a new Byzantine attack called *coalition head attack*, where the ATT tries to become the coalition head so that a wrong final decision can be made deliberately to mislead all CRs in the coalition. The goal of the ATT is to force CRs in its coalition to vacate the PC when the PU is in fact absent so as to decrease the number of available PCs used by CRs. Fig. 3.1 shows a possible attack scenario. In the figure, there are three channels, i.e., the sensing channel between the PU and CRs, the reporting channel between CRs and the H_d, and the reverse reporting channel between the H_d and

CRs. Obviously, when H_d declares that PU is absent, the received final decision made by H_d (or ATT) at CR_1 and CR_2 may be different from it, due to the potential manipulation by H_d.

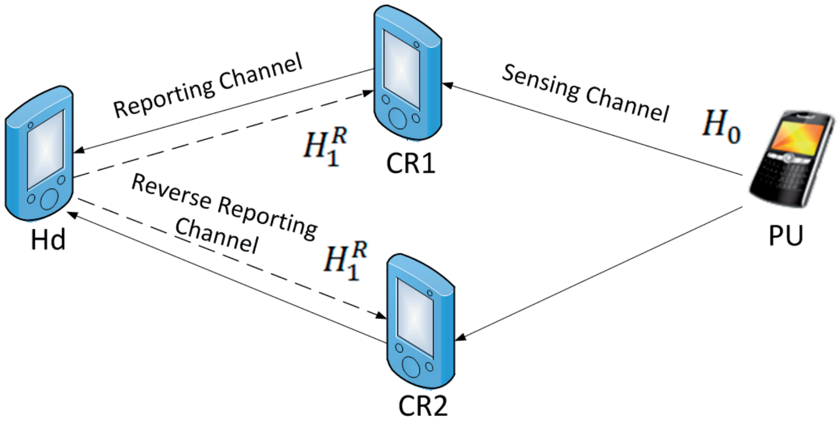


Figure 3.1: Possible attack scenario.

In order to combat this attack, the final decision received at the CRs in the coalition should be analyzed. Specifically, each CR needs to derive the probability, $p\{H_1^R|H_0\}$, that the received final decision shows the presence of PU, H_1^R, when PC is actually vacant, H_0, since the ATT attacks the coalition only when PU is absent.

Consider a single coalition with total $|\mathscr{C}|$ CRs. Let $H_0^{H_d}$ and $H_1^{H_d}$ be the decision events at the coalition head indicating the absence and presence of PU. In general, by considering OR-rule, the final decision made by the coalition head, H_d, is

$$\mathscr{F} \sim \begin{cases} H_0^{H_d}, & \{H_0^{H_d,1}, H_0^{H_d,2}, ..., H_0^{H_d,i}\} \\ H_1^{H_d}, & \text{otherwise.} \end{cases} \tag{3.1}$$

where $H_0^{H_d,i}$ is the local decision event from CR_i at H_d that there is no PU existing. Let $H_n^{R_i}$, $n = \{0, 1\}$, be the received decision event at CR_i from H_d. Since the final decision in the coalition is shared among all CRs, $H_n^{R_i}$ should be same for any i, i.e., $H_n^{R_i} = H_n^R$.

Then, following the similar procedure in [20], we derive

$$\begin{aligned}prob\{H_1^R|H_0\} = & \, prob\{H_1^R|H_0^{H_d}\}prob\{H_0^{H_d}|H_0\} \\ & +prob\{H_1^R|H_1^{H_d}\}prob\{H_1^{H_d}|H_0\}\end{aligned} \tag{3.2}$$

According to (3.1),

$$\begin{aligned}prob\{H_0^{H_d}|H_0\} & = prob\{H_0^{H_d,1}, ..., H_0^{H_d,k}|H_0\} \\ & = \prod_{i=1}^{k} prob\{H_0^{H_d,i}|H_0\} \\ & = (1 - Q_f^{H_d})\end{aligned} \tag{3.3}$$

where $Q_f^{H_d}$ is the false alarm probability at the H_d. In (3.3), the independence of channel sensing in each CR has been applied. Similarly,

$$prob\{H_1^{H_d}|H_0\} = Q_f^{H_d} \tag{3.4}$$

By considering the reverse reporting channel error and potential coalition head attack, $prob\{H_1^R|H_0^{H_d}\}$ and $prob\{H_1^R|H_1^{H_d}\}$ in (3.2) can be calculated as

$$prob\{H_1^R|H_0^{H_d}\} = (1 - p_e^r)p_{HA}^I + p_e^r(1 - p_{HA}^I) \tag{3.5}$$

$$prob\{H_1^R|H_1^{H_d}\} = (1 - p_e^r) \tag{3.6}$$

where p_e^r is the probability of error in the reverse reporting channel, and p_{HA}^I is the probability of coalition head attack. The derivation of p_{HA}^I will be provided later.

By substituting (3.3), (3.4), (3.5) and (3.6) in (3.2), we have

$$\begin{aligned}prob\{H_1^R|H_0\} = & \, (1 - Q_f^{H_d})[(1 - p_e^r)p_{HA}^n + p_e^r(1 - p_{HA}^n)] \\ & +Q_f^{H_d}(1 - p_e^r)\end{aligned} \tag{3.7}$$

In order to extend the aforementioned results to multi-channel scenario, we define $H_1^{j,I} = H_1^R$, where $H_1^{j,I}$ is the received decision from coalition head I on channel j. Let $\mathscr{O}^{j,i} = prob\{H_1^{j,i}|H_0\}$. We now introduce a new criterion for coalition head selection. A CR is selected as the coalition head if

$$I = \underset{i \in \mathscr{C}_j}{\operatorname{argmin}}(Q_m^{j,i} + \mathscr{O}^{j,i^{\zeta}}) \tag{3.8}$$

where ζ is a tuning factor and can be adjusted based on the system security level (reliable detection) and sensing performance. For example, if security is the major concern of the system, ζ should be chosen such that $\mathscr{O}^{j,i^{\zeta}}$ dominates $Q_m^{j,I}$. Note that in Chapter 2, coalition head selection is based on $Q_m^{j,i}$ only. As a result, the ATT can manipulate its $Q_m^{j,i}$ to be assigned as the coalition head.

In order to calculate $\mathscr{O}^{j,i}$, the probability of coalition head attack for CR_i on channel j, $p_{HA}^{j,i}$, should be derived properly. Since CRs in the coalition are in communication range of each other, it is expected that the received SNR for each CR, $\gamma_{i,j}$, should be in a certain range. Moreover, each CR has SNRs of all the other CRs in its coalition. Note that since each CR should have $p_m^{i,j}$ to calculate $Q_m^{j,I}$ and find if it can be the coalition head, $\gamma_{i,j}$ can be statistically analyzed and $p_{HA}^{j,i}$ can be assigned to the candidate coalition head. The basic idea is as follows.

Define $\Gamma^j = \{\gamma_{1,j}..., \gamma_{i,j}, ..., \gamma_{|\mathscr{C}_j|,j}\}$ as the set of SNRs of the CRs in the coalition j. Let $\mathscr{E}(.)$ be the mean value operator. Then, by comparing $\gamma_{i,j}$ with $\mathscr{E}(\Gamma^j)$, potential ATTs can be identified, since they intend to report high SNRs to become coalition head. Moreover, $p_{HA}^{j,i}$ should become larger when $\gamma_{i,j}$ exceeds $\mathscr{E}(\Gamma^j)$ more.

However, in some cases, setting $\mathscr{E}(\Gamma^j)$ as comparison threshold may not work well. For example, when the number of ATTs is more than that of honest CRs in the coalition, $p_{HA}^{j,i}$ may not be assigned fairly to CRs, since $\mathscr{E}(\Gamma^j)$ is closer to the ATTs' SNRs. Under these cases, ATTs may be hidden by assigning small $p_{HA}^{j,i}$. To solve this problem, we introduce a maximum possible SNRs in the coalition j, γ_{max}^j.

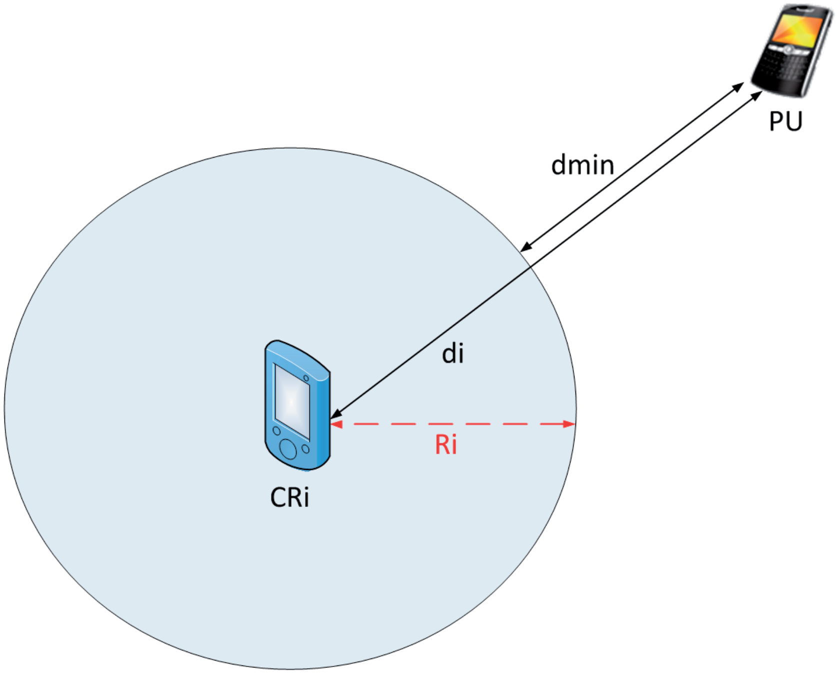


Figure 3.2: Minimum distance when CR_i has the minimum SNR in the coalition.

Consider CR_i has a minimum SNR in the coalition. Obviously, there is a high probability that CR_i is an honest CR, since it will never be assigned as coalition head. Then, the potential coalition head should be the closest CR to the PU (i.e., the largest SNR) in the communication range of CR_i. Fig. 3.2 shows how γ^j_{max} can be found. From Fig. 3.2, the minimum distance to PU is $(\theta^j_{max} - R_i)$, where R_i is the communication range of CR_i. Thus,

$$\gamma^j_{max} = \frac{P_{PU}(\theta^j_{max} - R_i)^{-\alpha}}{N_0}. \tag{3.9}$$

By jointly considering $\mathscr{E}(\Gamma^j)$ and γ^j_{max}, we determine $p^{j,i}_{HA}$ as

$$p^{j,i}_{HA} = \begin{cases} \frac{\gamma_{i,j} - min\{\mathscr{E}(\Gamma^j), \gamma^j_{max}\}}{\gamma_{i,j}} & \gamma_{i,j} > min\{\mathscr{E}(\Gamma^j), \gamma^j_{max}\} \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

Let $Q^{j,i}_f$ be $Q^{H_d}_f$ on channel j. According to (3.7), in order for ATTs to bypass the effect of $p^{j,i}_{HA}$ and become H_d, the following condition should be satisfied

$$(1 - Q^{j,i}_f)[(1 - p^r_e)p^{j,i}_{HA} + p^r_e(1 - p^{j,i}_{HA})] \ll Q^{j,i}_f(1 - p^r_e) \tag{3.11}$$

Since p_f and p_e^r are same for every CR, $Q_f^{j,i} \to 1$ which leads to

$$\prod_{m \in \mathscr{C}_j, m \neq i} [(1-p_f)(1-p_e) + p_f p_e] \to 0 \tag{3.12}$$

Therefore, the only way to achieve (3.11) is to increase the number of CRs in the coalition which is however limited according to (2.3). As a result, ATTs cannot decrease the effect of $p_{HA}^{j,i}$ when competing for H_d.

3.2 Byzantine counterattack in multi-channel cognitive radio networks

After making sure that the coalition head is not an ATT, the possibility that coalition members become ATTs should be considered. In this section, a new Byzantine counterattack is proposed for MBA. Note that, ATTs try to change their local sensing results on each channel to mislead H_d when ATTs know PU is absent.

In order to analyze the behavior of ATTs, we first calculate $prob\{H_0^{H_d,i}|H_0\}$ as

$$\begin{aligned} prob\{H_0^{H_d,i}|H_0\} = {} & prob\{H_0^{H_d,i}|H_0^i\}prob\{H_0^i|H_0\} \\ & + prob\{H_0^{H_d,i}|H_1^i\}prob\{H_1^i|H_0\} \end{aligned} \tag{3.13}$$

Obviously for ATTs, the second term in (3.13) does not change because they think that the channel is occupied. Therefore, only the first term is considered. Define p_A^i as the probability of attack for CR_i on a single channel and $H_n^{\prime i}$ as the manipulated H_n^i by CR_i. Then,

$$\begin{aligned} prob\{H_0^{H_d,i}|H_0^i\}prob\{H_0^i|H_0\} = {} & prob\{H_0^{H_d,i}|H_1^{\prime i}\}(p_A^i)prob\{H_0^i|H_0\} \\ & + prob\{H_0^{H_d,i}|H_0^{\prime i}\}(1-p_A^i) \\ & \times prob\{H_0^i|H_0\}, \end{aligned} \tag{3.14}$$

where p_A^i is defined as

$$p_A^i = prob\{H_1^{'i}|H_0^i\}. \tag{3.15}$$

Thus, $prob\{H_0^{H_d,i}|H_0\}$ can be rewritten as

$$prob\{H_0^{H_d,i}|H_0\} = p_e(p_A^i)(1-p_f) + (1-p_e)(1-p_A^i)(1-p_f) + p_f p_e \tag{3.16}$$

Let $p_A^{j,i}$ be p_A^i on channel j. Then, $Q_f^{j,I}$ can be calculated as

$$\begin{aligned} Q_f^{j,I} &= 1 - \textstyle\prod_{i\in\mathscr{C}_j, i\neq I} prob\{H_0^{H_d,i}|H_0\}, & (3.17)\\ &= 1 - [(1-p_A^{j,I})(1-p_f)] & (3.18)\\ &\quad \times \prod_{i\in\mathscr{C}_j, i\neq I} [p_e p_A^{j,i}(1-p_f) + (1-p_f)(1-p_A^{j,i})(1-p_e) + p_f p_e] \end{aligned}$$

In this thesis, we assume that all CRs in any coalition use a time stamp for sending their local sensing data to H_d. Hence, since all CRs use time stamps and have a same τ, ATT cannot eavesdrop others to see if they unintentionaly send “1” when PU is absent (alternative attack). Namely, ATTs have to send “1” when they want to attack the system (all-time attack).

In order to find $p_A^{j,i}$, we need to consider sensing and reporting channels for each CR. Assume at most one ATT exists in the coalition. We first calculate probability of attack by considering sensing channel error only, SL_A^i, which is defined as the probability that $(|\mathscr{C}_j|-1)$ CRs send correct local sensing results providing they sensed the channel correctly and is obtained as follows.

$$SL_A^i = \frac{|\mathscr{C}_j|-1}{|\mathscr{C}_j|}(1-p_f)^{|\mathscr{C}_j|} \tag{3.19}$$

If only the reporting channel error is considered, H_d may receive incorrect local sensing decisions from honest CRs even when ATT does not exist. Obviously, this should not be considered as an attack. Let RL_A^i be the probability of attack by considering reporting channel error only. Then, RL_A^i can be calculated as

$$RL_A^i = (1 - p_e)^{|\mathscr{C}_j|} \tag{3.20}$$

By combining (3.19) and (3.20), $p_A^{j,i}$ can be computed as

$$\begin{aligned} p_A^{j,i} &= SL_A^i RL_A^i \\ &= \frac{|\mathscr{C}_j| - 1}{|\mathscr{C}_j|}[(1 - p_e)(1 - p_f)]^{|\mathscr{C}_j|} \end{aligned} \tag{3.21}$$

Since, $p_A^{j,i}$ can change $C(Q_{f,\mathcal{S}}^j)$ in (2.7) and can eventually prevent the ATT from the corresponding coalition, it can be used as a reputation parameter in the coalition formation.

Based on the observation in [64], which states that it is better to cooperate on the first move and then reciprocate what the malicious users did on the previous move, we propose our algorithm as follows.

Define $p_A^{j,i}(q)$ as $p_A^{j,i}$ in round rd. Initially, all CRs (including ATTs) are considered to be honest, i.e., $p_A^i(1) = 0$. Then, the coalition formation algorithm is performed and coalitions are formed. After that, based on the received local decisions, H_d assigns $p_A^{j,i}$ to the CR which sent the different local decision different from the majority of other CRs. At the end of round rd, considering the behavior of the corresponding CR, $p_A^{j,i}(rd)$ is updated based on (3.21). If the incorrect decision is not recieved from the same CR again, $p_A^i(rd)$ is set to 0. In other words, when an ATT (because of the associated $p_A^{j,i}$) cannot be part of formed coalitions, it may be able to be in a coalition after the current round if it honestly cooperates. Otherwise, if it deviates again, it will be penalized again and its chance to be in any coalition decreases. Then, in the next round, new $Q_f^{j,I}$ is calculated and the coalition formation algorithm is re-performed.

3.3 Simulation results

In this section, we adopt the same network and simulation parameters as in Chapter 2, and set reverse reporting error probability $p_e^r = 0.09$. The coefficient ζ is obtained in each round of the algorithms, and the number of ATTs can increase to the same number of honest CRs at most.

We consider the CHA first. To evaluate the effectiveness of the proposed counterattack to the MCCRN, the average ratio of the number of invaded PCs with and without the counterattack, $\mathscr{N}_{inv}$, is introduced as the performance metric. We increase the number of ATTs from 0 (the system without attackers) to 25 (the maximum number of ATTs). The results are shown in Fig. 3.3. It can be seen that by applying the proposed counterattack, the number of invaded PCs is greatly decreased. For example, for the maximum number of ATTs, $\mathscr{N}_{inv} = 0.24$, i.e., 76% attacks are avoided.

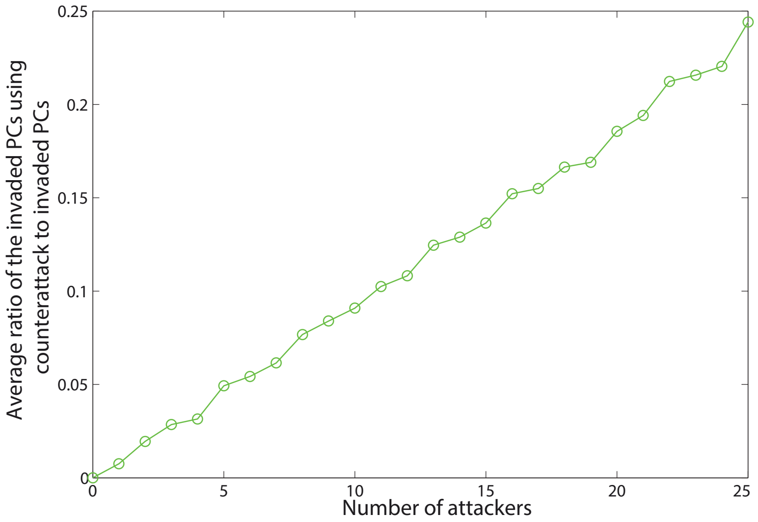


Figure 3.3: Average ratio of the number of invaded PCs using the proposed counterattack to the invaded PCs without using the proposed counterattack.

The evaluation of the proposed Byzantine counterattack is shown in Fig. 3.4. In this figure, the average ratio of the available PCs to all PCs, $\mathscr{N}_{av}$, is taken into consideration. Compared to the clean system (i.e., the system without attackers), MBA can considerably decrease $\mathscr{N}_{av}$ up to 32% when the number of ATTs increases to its maximum.

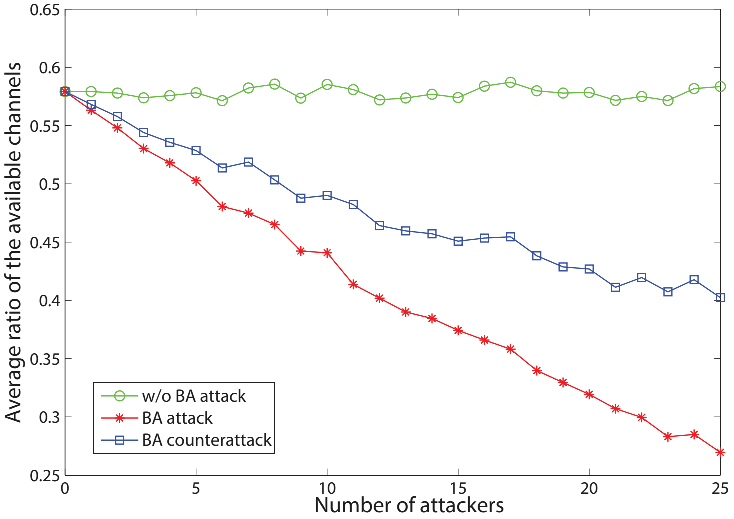


Figure 3.4: Average ratio of the number of available channels

Nonetheless, the proposed Byzantine counterattack can successfully increase $\mathscr{N}_{av}$ (up to 14% at the maximum number of ATTs) and compensate the MBA effects. The performance improvement results from the fact that in the proposed counterattack majority of ATTs are excluded from their target coalitions due to the assigned probability of attack. Note that $\mathscr{N}_{av}$ is expected to drop in the presence of the ATTs, since ATTs are part of MCCRN and by removing ATTs the number of active CRs which can sense PCs decreases.

Chapter 4

Energy-efficient key management in multi-channel cognitive radio networks

After considering physical layer security, we develop energy-efficient key management schemes to manage cryptographic keys in a distributed fashion, which helps provide MC-CRNs with higher layer cryptographic security.

4.1 Preliminaries

4.1.1 Bilinear Pairing

In general, parings are functions defined on elliptic curves and possess some unique features [65]. Mathematically, a paring can be defined as a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, where G_1 is a cyclic group generated by P with order of a prime p, and G_2 is a cyclic multiplicative group of the same order p. The discrete logarithm problems (DPLs) in both G_1 and G_2 are hard. A pairing should have the following properties:

1. **Bilinear:** $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q)\hat{e}(P_2, Q)$ and $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}, \forall P, Q \in G_1$ and $a, b \in \mathbb{Z}_q^*$

2. **Non-degenerate:** There is $P, Q \in G_1$, such that $\hat{e}(P, Q) \neq 1$

3. **Computability:** An efficient algorithm exists to compute $\hat{e}(P, Q), \forall P, Q \in G_1$.

4.1.2 Gap Diffie-Hellman Groups

The following groups are defined [66].

Definition 4.1. Let P be a generator of group $\mathcal{G}$ and a 3-tuple $(\mathcal{R}_a P, \mathcal{R}_b P, \mathcal{R}_c P)$, the *Decisional Diffie-Hellman (DDH) problem* is to decide whether $\mathcal{R}_c = \mathcal{R}_a \mathcal{R}_b$.

Definition 4.2. Let P be a generator of group $\mathcal{G}$ and a 3-tuple $(P, \mathcal{R}_a P, \mathcal{R}_b P)$, the *Computational Diffie-Hellman (CDH) problem* is to compute $\mathcal{R}_a \mathcal{R}_b P$.

Definition 4.3. For a given group $\mathcal{G}$, if CDH problem is hard, while DDH problem is easy, such group is called *Gap DiffieHellman (GDH) group*.

4.1.3 Boneh-Franklin's IBE

The first and most popular identity-based encryption scheme was proposed by Boneh and Franklin, called BF-IBE [49]. In this thesis, an extended version of the basic BF-IBE, called Fulldent, is considered which is able to be secure against enhanced indistinguishability under chosen ciphertext attacks (IND-CCA), called IND-ID-CCA [49]. In fact, IND-ID-CCA is a model of an adversary who has access to private keys of IDs of its choice and attacks an ID in an IBE scheme.

Definition 4.4. An ID-based encryption scheme is IND-ID-CCA secure, if no polynomially bounded adversary $\mathcal{A}$ has non-negligible advantage over the challenger $\mathcal{CH}$.

Similar to any IDE scheme, BF-IBE consists of four algorithms: *Setup*, *Extract*, *Encrypt* and *Decrypt*. In *Setup*, system parameters and a PKG master-key are generated. *Extract* takes system parameters and master key as input and generates a private key corresponding to an ID. Having system parameters and ID as inputs, *Encrypt* returns a ciphertext $\mathcal{C}$ as output. In *Decrypt*, system parameters and private key are given as inputs and massage $\mathcal{M}$ is returned as output.

4.2 System model

Consider an MCCRN with N CRs which are randomly distributed in a given coverage area. Here, threshold cryptography [50] is used to distribute the power of PKG to multiple DPKGs which are responsible for generating partial private keys for CRs. Applying DPKGs makes the system robust against PKG failure. Synchronous communication is assumed, i.e. the sender and receiver synchronize with each other before data are sent. Then, $N = 2\hat{t}+1$, where $\hat{t}$ is the number of allowed compromised CRs [67].

Since using all CRs as DPKG can cause superfluous communication overhead and can be energy consuming for CRs as well, we consider the case where only a group of CRs serve as DPKGs, i.e. $L < N$, where L is the number of DPKGs. Define $\Psi = \{\psi_1, ..., \psi_N\}$ as a vector of compromise probabilities for CR$_i$, $i = 1, 2, ..., N$. Further, we define ψ_{th} as a threshold to indicate the maximum acceptable probability of compromise to participate in DPKG formation and consider it as a system parameter.

Due to physical constraints of CRs, e.g. hardware design and energy consumption [24], each CR can only sense or utilize K out of M PCs. Each CR needs a channel key for each PC k in order to participate in spectrum sensing or have access for spectrum sharing. Note that a same channel key could be used for both spectrum sensing and sharing to greatly decrease the overhead and energy consumption caused by key management.

It is known that in transmit operation mode of the transceiver, the circuit power dissipation includes not only a static term, but a dynamic term which increases with clock frequency (scaled with data rate), [68]. Therefore, active-circuit power dissipation can be modeled as

$$P_i^{cir} = P_s + \kappa DR_i \tag{4.1}$$

where DR_i is the data rate of CR$_i$, and P_s and κ denote the static circuit power in the transmit mode and a constant based on dynamic power dissipation per unit of data rate, respectively. Therefore, the total dissipated power of CR$_i$ is

$$P_{di} = \mathcal{E}_D^{-1} P_{ti} + P_i^{cir} \tag{4.2}$$

where P_{ti} is the CR$_i$'s transmit power and $\mathcal{E}_D$ is the drain efficiency of power amplifier. Given P_{di}, battery life, in hours, can be obtained as [69]

$$L_{batt} = \frac{C_{batt} V_{batt}}{P_{di}} \tag{4.3}$$

where C_{batt} and V_{batt} are the battery capacity and voltage, respectively.

4.2.1 System initialization

The procedure of system initialization as shown in **Algorithm 4.2.1** is performed as follows. Let p be a prime number such that $p = 2mod3$ and $p = 6q-1$, $q > 3$ be a prime factor of $p+1$ and E be the elliptic curve defined by $y^2 = x^3+1$ over a finite field $\mathbb{F}_p$. We define the message space, $\mathcal{M}$, as $\{0,1\}^m$, where $m \in \mathbb{Z}^+$, and ciphertext space, $\mathcal{C}$, as $E/\mathbb{F}_p \times \{0,1\}^m$. Assume *GDH* assumption holds for G_1 and G_2. Then, four cryptographic hash functions are defined as $H_1 : \{0,1\}^* \to G_1$, $H_2 : G_2 \to \{0,1\}^m$, $H_3 : \{0,1\}^m \times \{0,1\}^m \to \mathbb{F}_q^*$ and $H_4 : \{0,1\}^m \to \{0,1\}^m$ [49]. We assume that all CRs and DPKGs are initialized following the aforementioned system parameters. Moreover, rather than adopting secure channels, we use a public channel to transmit control messages. Therefore, each CR$_i$ needs a pair of private and public keys for confidentiality (securely transmitting control messages related to key management) and authenticity. To achieve this, CR$_i$ generates its private key, d_i, with respects to its public key (ID_i). Then, it selects a random number $r_i \in \mathbb{F}_q$ and broadcasts $< ID_i, r_i >$. After receiving all r_i, each CR$_i$ calculates

$$r = \sum_{i=1}^{N} r_i \tag{4.4}$$

Algorithm 4.2.1 System Initialization

```
Message space $\mathcal{M} : \{0,1\}^m$
Ciphertext space $\mathcal{C} : E/\mathbb{F}_p \times \{0,1\}^m$
Given pair of primes $< p, q >$, Hash functions:
    • $H_1 : \{0,1\}^* \to G_1$
    • $H_2 : G_2 \to \{0,1\}^m$
    • $H_3 : \{0,1\}^m \times \{0,1\}^m \to \mathbb{F}_q^*$
    • $H_4 : \{0,1\}^m \to \{0,1\}^m$
Each CR$_i$ generates its private key, $d_i$, corresponding to its $ID_i$ and keeps it.
Each CR$_i$ selects a random number $r_i \in \mathbb{F}_q$ and broadcasts $< ID_i, r_i >$ to all CRs.
$r = \sum_{i=1}^{N} r_i$
$Q_r = H_1(r)$
$Q_u = \sum_{i=1}^{L} r_i Q_r$
```

which is same for all CRs. After that, CR$_i$ can obtain $Q_r = H_1(r)$, where $H_1(r)$ maps r to a point on E. Q_r will be used for share verification in the key management scheme. In addition to Q_r, DPKGs need to calculate $Q_u = \sum_{i=1}^{L} r_i Q_r$, where by knowing Q_r, the discrete logarithm problem is hard.

4.3 Energy-efficient DPKG selection

Obviously, DPKG selection becomes an important issue when no PKG exists in MCCRN. Moreover, as only L DPKGs are selected, a security threshold t (the number of possible compromised DPKGs) should be defined such that $L = 2t + 1$ to ensure the security of the selected DPKGs. In this section, we first propose a method to select t based on CRs' probability of compromise. Then, by considering the number of DPKGs (based on the obtained t), an algorithm will be proposed to optimally assign CRs who have more energy to become DPKGs.

4.3.1 Security threshold determination

In order to determine the security threshold ,i.e. t, the probability that a CR is compromised during the operation should be obtained. In practice, the probability of compromise can change by position of CRs, depending the CRs deployment. The fusion center, is the most reliable node in the network. Then, CRs closer to the fusion center should have lower risk of being compromised, while CRs far away from the fusion center are at a higher risk of being compromised. Therefore, we define the probability of compromise as $\psi_i = h(l_i)$, where l_i is the position of CR$_i$. Note that $h(.)$ is a function which maps the position to the probability of compromise. The probability of compromise should be an increasing function of the distance to the fusion center such that ψ starts from 0 at the fusion center. A suitable function for our purpose can be defined as

$$\psi_i = \hat{\alpha}(1 - e^{-\frac{\|l_i - l_c\|}{\theta_o}}), \tag{4.5}$$

where l_c is the position of the fusion center, θ_o is the maximum possible distance from the fusion center in the area of operation, and $\hat{\alpha} = 1/(1 - e^{-1})$ to ensure $0 \leq \psi_i \leq 1$.

Let ψ_{th} be the probability of compromise threshold and a system parameter. Note that ψ_{th} can be obtained by setting a threshold for distance from the fusion center, i.e., $\bar{l}_{th}$, which depends on the deployment of CRs and area of operation. Then, the area with the radius of $\theta_o - \bar{l}_{th}$ is still operable, but at higher risks. We categorize CRs into two sets, defined as

- $\mathcal{S}_d = \{CR_i, i \in N | \psi_i \geq \psi_{th}\}$
- $\mathcal{S}_s = \{CR_i, i \in N | \psi_i < \psi_{th}\}$

CRs in $\mathcal{S}_d$ have considerable chance of being compromised during the network operation. Thus, they will have minimal contributions to distributed private key generation scheme, if selected as DPKG, and can be neglected. Let $|\mathcal{S}_s|$ denote the number of CRs in $\mathcal{S}_s$. Therefore, $t = \frac{|\mathcal{S}_s|}{2} - 1$.

Algorithm 4.3.1 Security-based DPKG selection

```
Initialization:
    • Run Security threshold determination algorithm to obtain $t$
    • $\mathscr{X} = 0$
for $i = 1$ to $L$ do
    $i^* = \underset{i \in N}{\text{argmin}} \Xi$
    $x_i = 1$
end for
```

4.3.2 Security-based DPKG selection (SB-DPKG)

In order to minimize the number of compromised or faulty DPKGs during the operation, the probability of compromise should be taken into account as the most important parameter for DPKG selection. This is the case, especially in scenarios when spectrum management agility is of utmost importance, e.g. the sensing time is strictly limited, which forces the distributed private key generation and key management algorithms to be carried out without taking extra processing and communication time, imposed by compromised DPKGs.

Define the DPKG selection vector as $\mathscr{X} = \{x_i, ..., x_N\}$, where $x_i = 1$ means that CR$_i$ is selected as DPKG. Otherwise, $x_i = 0$. Then, the problem of DPKG selection can be formulated as the following assignment problem.

$$\min_{x} \quad \sum_{i=1}^{N} \psi_i x_i \tag{4.6}$$

$$s.t. \quad \sum_{i=1}^{N} x_i = L, \tag{4.7}$$

To assign DPKGs, CR_i with the minimum probability of compromise is selected. i.e. $x_i = 1$. Then, the next CR with the lowest probability of compromise is selected and the process continues until L CRs are selected as DPKG, as shown in **Algorithm 4.3.1**. Therefore, the initial set of qualified DPKGs, $S_{QL} = \{CR_i | x_i = 1\}$, is formed.

Algorithm 4.3.2 Energy-based DPKG selection

```
Initialization:
    • Run Security threshold determination algorithm to obtain $t$
    • $\mathscr{X} = 0$
for $i = 1$ to $L$ do
    $i^* = \underset{i \in \mathcal{S}_s}{\text{argmax}} \Xi$
    $\mathcal{S}_s = \mathcal{S}_{s \backslash \{i^*\}}$
    $x_i = 1$
end for
```

4.3.3 Energy-based DPKG selection (EB-DPKG)

Balancing the selected DPKGs' energy clearly plays a crucial role as they have to serve all CRs. Therefore, not only the probability of compromise, but the energy level of CRs should be jointly considered to select reliable DPKGs while maximizing the energy level of DPKGs. Let $\Xi = \{\xi_1, ..., \xi_N\}$ be the vector of energy level of CRs before DPKG formation. To balance the energy level of CRs, CRs with higher energy levels are given higher priorities as potential DPKGs, if they can meet the probability of compromise threshold. Then, the problem of security-based DPKG selection can be formulated as the following assignment problem.

$$\max_{x} \quad \sum_{i=1}^{N} \mathscr{I}(\psi_i)\xi_i x_i \tag{4.8}$$

$$s.t. \quad \sum_{i=1}^{N} x_i = L, \tag{4.9}$$

where $\mathscr{I}(\psi_i)$ is an indicator function which is defined as

$$\mathscr{I}(\psi_i) = \begin{cases} 1, & \psi_i < \psi_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

Therefore, the DPKG assignment is as follows. After determining t, only CRs with the probability of compromise that meet ψ_{th} are considered. Then, $CR_i, i \in \mathcal{S}_s$, with

Algorithm 4.4.1 Distributed Private Key Generation

```
Each DPKG$_i$ selects two random polynomials of degree $t$
    • $f_i(x) = \sum_{k=0}^{t} a_{ik}x^k$
    • $g_i(x) = \sum_{k=0}^{t} b_{ik}x^k$
where $a_{ik} \in \mathbb{F}_q$ and $b_{ik} \in \mathbb{F}_q$
Compute $s_{ij} = f_i(j)\text{mod}q$ and $s'_{ij} = g_i(j)\text{mod}q$, $j \in \mathcal{S}_{QL}$
Send $< Enc(s_{ij}), Sig_j >$ and $< Enc(s'_{ij}), Sig_j >$ to DPKG$_j$
Set commitment $\mathfrak{C}_{ik} = a_{ik}Q_r + b_{ik}Q_u$
Multicast $\mathfrak{C}_{ik}$ to the other DPKGs.
Each DPKG$_j$ runs **Master Key Share Verification** algorithm.
    • **Master key share generation:**
$s_j = \sum_{i \in \mathcal{S}_{QL}} s_{ji}$
```

maximum remaining energy is selected as part of S_{QL}, followed by the next $(L-1)$ CRs with the highest energy levels, which are sequentially selected. This procedure is called energy-based DPKG selection (EB-DPKG) and shown in **Algorithm 4.3.2**.

4.4 Distributed private key generation

After DPKG selection, distributed private key generation scheme is proposed in this section where DPKGs cooperatively compute the master key (private key) and the corresponding public key.

In the proposed scheme, each $CR_i, i \in S_{QL}$, denoted as DPKG$_i$, first randomly selects two polynomials of degree t as follows.

$$f_i(x) = \sum_{k=0}^{t} a_{ik}x^k \tag{4.11}$$

$$g_i(x) = \sum_{k=0}^{t} b_{ik}x^k \tag{4.12}$$

where a_{ik} and b_{ik} are defined over finite field $\mathbb{F}_q$. Then, DPKG$_i$ calculates $s_{ij} = f_i(j)\text{mod}q$ and $s'_{ij} = g_i(j)\text{mod}q$, for any other $j \in \mathcal{S}_{QL}$. After that, DPKG$_i$ encrypts, signs and sends s_{ij} and s'_{ij} to the corresponding DPKG$_j$. In order for each DPKG$_j$ to verify the received s_{ij}

Algorithm 4.4.2 Master Key Share Verification

```
Initialization:
    • Define the complaint matrix: $\mathscr{C} = (c_{ij})_{L\times L}$, $c_{ij} \in \{0, 1\}$
    • $\mathscr{C} = 0$
for $j \in \mathcal{S}_{QL}$ do
    for $i \in \mathcal{S}_{QL}, i \neq j$ do
        if $s_{ij}Q_r + s'_{ij}Q_u \neq \sum_{k=0}^{t} j^k \mathfrak{C}_{ik}$ then
            broadcast complaint against CR$_i$, i.e., $c_{ji} = 1$
            if DPKG$_i$ receives $c_{ji} = 1$ from DPKG$_j$ then
                DPKG$_i$ computes $s_{ij}^{corr}$ and $s_{ij}'^{corr}$
                DPKG$_i$ sends $< Enc(s_{ij}^{corr}), Sig_j >$ and
                $< Enc(s_{ij}'^{corr}), Sig_j >$ to DPKG$_j$
                if $s_{ij}^{corr}Q_r + s_{ij}'^{corr}Q_u \neq \sum_{k=0}^{t} j^k \mathfrak{C}_{ik}$ then
                    update $\mathcal{S}_{QL} = \mathcal{S}_{QL\setminus\{i\}}$
                end if
            end if
        end if
    end for
end for
    • Final $\mathcal{S}_{QL}$ update:
        – Each DPKG$_i$ checks $\mathscr{C}$
for $i \in \mathcal{S}_L$ do
    for $j \in \mathcal{S}_L$ do
        $c_i = \sum_{j\in\mathcal{S}_L} c_{ji}$
        if $c_i > t$ then
            update $\mathcal{S}_{QL} = \mathcal{S}_{QL\setminus\{i\}}$
        end if
    end for
end for
```

and s'_{ij}, DPKG$_i$ needs to set the commitment using Q_r and Q_u, derived in the initialization stage, as

$$\mathfrak{C}_{ik} = a_{ik}Q_r + b_{ik}Q_u \tag{4.13}$$

and multicast it to all DPKGs. After receiving $\mathfrak{C}_{ik}$, each DPKG$_j$ verifies its collected shares and constructs its master key share as follows.

$$s_j = \sum_{i\in\mathcal{S}_{QL}} s_{ji} \tag{4.14}$$

The algorithm for distributed private key generation is summarized in **Algorithm 4.4.1**. In the following two subsections, the procedures for master key share verification and public key generation in one DPKG will be discussed in details.

4.4.1 Master key share verification

Let $\mathscr{C}\mathscr{P} = (c_{ij})_{L\times L}$, $c_{ij} \in \{0,1\}$ be the complaint matrix. Matrix $\mathscr{C}$ is initialized to **0**, since there is no complaint against any DPKG at the beginning. Each DPKG$_j$ checks s_{ij} and s'_{ij} by using DPKG$_i$'s commitment $\mathfrak{C}_{ik}$. If check fails, i.e. $s_{ij}Q_r+s'_{ij}Q_u \neq \sum_{k=0}^{t} j^k\mathfrak{C}_{ik}$, DPKG$_j$ signs and broadcasts $c_{ji} = 1$ as the complaint against DPKG$_i$. Then, DPKG$_i$ should compute the correct values for s_{ij} and s'_{ij}, denoted as s_{ij}^{corr} and $s_{ij}^{\prime corr}$, and securely sends them to DPKG$_j$. If check fails again, i.e. $s_{ij}^{corr}Q_r + s_{ij}^{\prime corr}Q_u \neq \sum_{k=0}^{t} j^k\mathfrak{C}_{ik}$, DPKG$_i$ is disqualified and excluded from $\mathcal{S}_{QL}$.

- Final $\mathcal{S}_{QL}$ update

DPKG$_i$ may still be disqualified, if more than t DPKG$_j$ send complaints against it. Let c_i be the total number of complaints against DPKG$_i$, i.e. $c_i = \sum_{j\in\mathcal{S}_L} c_{ji}$. If $c_i > t$, other DPKGs exclude DPKG$_i$ from $\mathcal{S}_{QL}$.

4.4.2 Public key generation

Now, DPKGs can jointly generate PKG public key, denoted as Pub. First, each DPKG$_i$, $i \in \mathcal{S}_{QL}$, broadcasts $A_{ik} = a_{ik}Q_r$, $k = \{0,...,t\}$. Then, each DPKG$_j$ checks

$$s_{ij}Q_r = \sum_{k=0}^{t} j^k A_{ik}. \tag{4.15}$$

If check fails, DPKG$_j$ broadcasts $c'_{ji} = 1$, otherwise, it broadcasts $c'_{ji} = 0$. If $c'_{ji} = 0$ is received, A_{ik} is approved. In the case of $c'_{ji} = 1$, DPKG$_j$, $j \in \mathcal{S}_{QL}, j \neq i$, calculates and

Algorithm 4.5.1 Identity-Based Channel Key Management (IBCKM)

```
Initialization:
    • Run Energy-based DPKG selection algorithm.
    • Run Distributed Private Key Generation algorithm to obtain the master key $s_i$.
    • DPKG$_j$ computes $P_j = s_j Q_r$ and broadcasts it to CRs
    • Channel Key Request:
for $e = 1$ to $K$ do
    CR$_i$ multicasts its channel key request, CKR$^e_i$, along with its $ID_i$, i.e. $I^e_i =< ID_i, CKR^e_i >$, to $t+1$ qualified DPKG$_j$s, $j \in \mathcal{S}^i_{QL}$ and $\mathcal{S}^i_{QL} \subset \mathcal{S}_{QL}$.
    Each selected DPKG$_j$, $j \in \mathcal{S}^i_{QL}$:
    computes $Q^e_i = H_1(I^e_i)$.
    computes $Q^{eij}_p = s_j Q^e_i$.
    sends $< Enc(Q^{eij}_p), Sig_j >$ to CR$_i$.
    CR$_i$ runs Share verification algorithm for $< Q^e_i, P_j, Q_r, Q^{eij}_p >$ to verify $Q^{eij}_p$.
    CR$_i$ computes $Q^e_{di} = \sum_{j=1}^{t+1} \prod_{j=1, j\neq i}^{t+1} \frac{j}{j-i} Q^{eij}_p$.
end for
```

broadcasts $\lambda_{j,\mathcal{S}_{QL}} = \prod_{n \in \mathcal{S}_{QL} \setminus \{j\}} \frac{n}{n-j}$. Then, the reconstructed A_{i0} can be calculated as

$$y_i = \sum_{j \in \mathcal{S}_{QL}} s_{ij} \lambda_{j,\mathcal{S}_{QL}} mod(q)$$

$$A_{i0} = y_i Q_r. \tag{4.16}$$

By setting $Pub_i = A_{i0}$, $i \in \mathcal{S}_{QL}$, the PKG public key is calculated as

$$PUB = \sum_{i \in \mathcal{S}_{QL}} Pub_i. \tag{4.17}$$

4.5 Identity-based key management in MCCRNs

In this section, a new identity-based channel key management (IBCKM) is proposed based on the elliptic curve discrete logarithm (ECDL) problem, as shown in **Algorithm 4.5.1**. In secure MCCRNs, CRs need to request keys for up to K PCs based on their channel conditions.

Let $\mathcal{S}^i_{QL}$ be the set of $t+1$ qualified DPKGs picked by CR$_i$ from $\mathcal{S}_{QL}$. Each DPKG$_j$,

Algorithm 4.5.2 Share verification

```
Initialization:
    • set $round = 0$
while $|\mathcal{S}^i_{QL}| \neq t + 1$ after at least one round do
    if $round \neq 0$ then
        CR$_i$ selects $|\mathcal{S}^i_{DQ}|$ new DPKG$_j$s,
        $j \in \mathcal{S}_{QL}$ and $j \notin \mathcal{S}^i_{QL} \cup \mathcal{S}^i_{DQ}$.
        CR$_i$, according to the main algorithm:
        sends $I^e_i$ to new DPKG$_j$s and receives new $Q^{eij}_p$.
    end if
    for $j \in \mathcal{S}^i_{QL}$ do
        while $\nu_j \leq \nu_{th}$ for CR$_i$ do
            if $\hat{e}(Q^e_i, P_j) \neq \hat{e}(Q^{eij}_p, Q_r)$ then
                CR$_i$ sends $< F_i, Sig_i >$ to DPKG$_j$
                DPKG$_j$ regenerates $Q^{eij}_p$
                DPKG$_j$ sends $< Enc(Q^{eij}_p), Sig_j >$ to CR$_i$
            else
                break
            end if
            $\nu_j = \nu_j + 1$
        end while
        if $\nu_j > \nu_{th}$ then
            $\mathcal{S}^i_{QL} = \mathcal{S}^i_{QL\setminus\{j\}}$
            $\mathcal{S}^i_{DQ} \leftarrow j.$
        end if
    end for
    update $round.$
end while
```

$j \in \mathcal{S}^i_{QL}$, computes $P_j = s_j Q_r$ and broadcasts it to all CRs as the public parameter which will be used in share verification. After that, CR$_i$ selects a favorable PC, e.g., e, and sends its *channel key request*, CKR^e_i, along with its ID$_i$, i.e., $I^e_i =< ID_i, CKR^e_i >$, to $t + 1$ DPKGs. This makes I^e_i a unique ID for CR$_i$ on PC e so that the private key generated is specific for CR$_i$ on PC e. Then, each DPKG$_j$ obtains the point Q^e_i corresponding to I^e_i and then computes $Q^{eij}_p = s_j Q^e_i$ as its share of the private key for CR$_i$. After that, Q^{eij}_p is signed and sent to CR$_i$. Upon receiving Q^{eij}_p, CR$_i$ can verify it by running share verification algorithm, as shown in **Algorithm 4.5.2**.

In **Algorithm 4.5.2**, for each DPKG$_j$, $j \in \mathcal{S}^i_{QL}$, CR$_i$ first checks the validity of Q^{eij}_p by testing

$$\hat{e}(Q_i^e, P_j) = \hat{e}(Q_p^{eij}, Q_r) \tag{4.18}$$

If (4.18) does not hold, it means that either DPKG$_j$ is compromised or has some malfunctions. Thus, CR$_i$ sends a signed false message, F_i, to DPKG$_j$. After receiving F_i, DPKG$_j$ regenerates Q_p^{eij}, and sends it back to CR$_i$. Since an attacker can continue such retransmission procedure for as much time as it wants, we define a threshold, ν_{th}^i, on the number of retransmissions by DPKG$_j$, ν_j. For simplicity and without loss of generality, we assume that all CRs have the same ν_{th}. As soon as ν_j exceeds ν_{th}, retransmission is terminated by CR$_i$ and CR$_i$ adds DPKG$_j$ to its list of disqualified DPKGs, i.e. S_{DQ}^i. Let $|\mathcal{S}_{DQ}^i|$ be the number of DPKGs in $\mathcal{S}_{DQ}^i$. In order to guarantee the number of DPKGs, i.e., $t+1$, CR$_i$ need to replace the disqualified DPKGs in $\mathcal{S}_{DQ}^i$ by selecting $|\mathcal{S}_{DQ}^i|$ new DPKG$_n$ with $n \in \mathcal{S}_{QL}$ and $n \notin \mathcal{S}_{QL}^i \cup \mathcal{S}_{DQ}^i$. CR$_i$ repeats the procedure until $t+1$ qualified DPKGs are found.

After share verification, CR$_i$ can construct its private key on PC e, Q_{di}^e, by calculating [50]

$$Q_{di}^e = \sum_{j=1}^{t+1} \prod_{j=1, j\neq i}^{t+1} \frac{j}{j-i} Q_p^{eij} \tag{4.19}$$

After key management, the IND-ID-CCA secure encryption-decryption can be achieved by the Fulldent BF-IBE in [49].

4.6 Certificate-based key management in MCCRNs

In this section, a new certificate-based key management scheme, called *certificate-based channel key management* (CBCKM), is proposed for the scenarios requiring certificates. CBCKM is also considered as the main counterpart of IBCKM proposed in the previous section for performance comparison. Similar to IBCKM, the proposed scheme adopts distributed CAs, i.e. DPKGs in IBCKM, for secret sharing. CBCKM, as shown in **Algorithm 4.6.1**, starts with broadcasting of $V_j = s_j Q_r$ by each CA$_j$ and is followed by two phases:

Algorithm 4.6.1 Certificate-Based Channel Key Management (CBCKM)

```
Initialization:
    • Given the master key $s_i$, CA$_j$ computes $V_j = s_jQ_r$ and broadcasts it to CRs
    • General Certificate Request:
CR$_i$ multicasts its public key, $p_i$, to $t+1$ qualified CA$_j$s, $j \in \mathcal{S}^i_{QL}$ and $\mathcal{S}^i_{QL} \subset \mathcal{S}_{QL}$.
Each selected CA$_j$, $j \in \mathcal{S}^i_{QL}$:
computes $P_i = H_1(p_i)$
computes $Cert_p^{ij} = s_jP_i$
sends $< Cert_p^{ij}, Sig_j^{CA} >$ to CR$_i$
CR$_i$ runs Share verification algorithm for $< P_i, V_j, Q_r, Cert_p^{ij} >$ to verify $Cert_p^{ij}$.
CR$_i$ computes $Cert_i = \sum_{j=1}^{t+1} \prod_{j=1,j\neq i}^{t+1} \frac{j}{j-i} Cert_p^{ij}$
    • Channel Certificate Request:
for $e = 1$ to $K$ do
    CR$_i$ generates a public key $p_{ci}^e =< r_{ci}, CCR_i^e >$ for its requested channel, where $r_{ci}$ is a random number and CCR$_i^e$ is the channel certificate request.
    CR$_i$ generates $d_{ci}^e$ as the private key corresponding to $p_{ci}^e$
    CR$_i$ multicasts $Req_i^e =< Cert_i, p_{ci}^e, Sig_i >$ to $t+1$ qualified CA$_j$s, $j \in \mathcal{S}^{i'}_{QL}$ and $\mathcal{S}^{i'}_{QL} \subset \mathcal{S}_{QL}$.
    Each selected CA$_j$, $j \in \mathcal{S}^{i'}_{QL}$:
    computes $P_{ci}^e = H_1(p_{ci}^e)$
    computes $Cert_c^{eij} = s_jP_{ci}^e$
    sends $< Cert_c^{eij}, Sig_j^{CA} >$ to CR$_i$
    CR$_i$ runs Share verification algorithm for $< P_{ci}^e, V_j, Q_r, Cert_c^{eij} >$ to verify $Cert_c^{eij}$.
    CR$_i$ computes $Cert_c^{ei} = \sum_{j=1}^{t+1} \prod_{j=1,j\neq i}^{t+1} \frac{j}{j-i} Cert_c^{eij}$
end for
```

general certificate request and *channel certificate request*.

- General certificate request

In this phase, CR$_i$ selects its general public key, p_i, and sends it to $t+1$ CAs to obtain its certificate. Each selected CA$_j$ uses its secret share, s_j, and the received public key, p_i, to compute its share of the certificate, $Cert_p^{ij}$, and sends it to CR$_i$. Upon receiving $Cert_p^{ij}$, CR$_i$ performs a similar share verification algorithm as shown in **Algorithm 4.5.2** by replacing DPKG$_j$, Q_i, P_j, Q_r and Q_p^{ij} with CA$_j$, P_i, V_j, Q_r and $Cert_p^{ij}$. Then, CR$_i$ can extract its certificate by $Cert_i = \sum_{j=1}^{t+1} \prod_{j=1,j\neq i}^{t+1} \frac{j}{j-i} Cert_p^{ij}$.

- Channel certificate request

In this phase, CR$_i$, submits its request to $t+1$ CA$_j$s to acquire a channel certificate on the demanded PC. Let r_{ci} be a random number generated by CR$_i$ and CCR$_i^e$ be the CR$_i$'s channel certificate request on PC e. Then, CR$_i$ first generates a key pair for PC e, $< p_{ci}^e, d_{ci}^k >$, where $p_{ci}^e =< r_{ci}, CCR_i^e >$ and d_{ci}^k are its public and private keys on PC e, respectively. After that, CR$_i$ multicasts p_{ci}^e to $t+1$ CAs. Each selected CA$_j$ computes $P_{ci}^e = H_1(p_{ci}^e)$, and generates $Cert_c^{eij} = s_j P_{ci}^e$, which is signed and sent to CR$_i$. With parameters $< P_{ci}^e, V_j, Q_r, Cert_c^{eij} >$, CR$_i$ can then perform share algorithm and construct its PC e certificate by $Cert_c^{ei} = \sum_{j=1}^{t+1} \prod_{j=1, j\neq i}^{t+1} \frac{j}{j-i} Cert_c^{eij}$.

4.7 Simulation results

In this section, we consider an MCCRN consisting of 50 CRs. The parameters are $P_s = 0.05\mu$W, $\mathcal{E}_D = 0.40$, $\kappa = 0.02$W/Mbits, $\psi_{th} = 0.5$, $\nu_{th} = 2$, $N_0 = -50$dBm, and $B = 0.5$MHz. The channel gain is defined as $cg_{i,j} = A_0 |\sigma_{fi,j}|^2 \theta_{i,j}^{-\alpha} 10^{\sigma_{si,j}/10}$, where $A_0 = 80$dB is a constant parameter including antenna gain, $\sigma_{fi,j}$ is a random variable with Rayleigh distribution, $\theta_{i,j}$ is the distance between CR$_i$ and CR$_j$, $\alpha = 3$ is the path-loss exponent, and $\sigma_{si,j}$ is a zero-mean log-normal random variable with standard deviation $sd_{\sigma_s} = 6$dB. We define energy efficiency as $\eta = \frac{DR_{ov}}{P_{ov}}$, where DR_{ov} and P_{ov} are the overall system data rate and power consumption, respectively. At each CR, a standard Nokia Nickel Metal Hydride (NiMH) battery is used with $C_{batt} = 650$mAh, and $V_{batt} = 6$V. According to NIST recommendations [70] for time period 2011-2030, ECC and RSA key sizes are 224 and 2048 bits, respectively. ECDSA-based and RSA-based certificate sizes are 673 and 4096 bits excluding overhead of the ID$_i$ which is assumed 50 bits. The identity-based signature scheme in [71] is applied with 320-bit signatures.

First, we compare our proposed EB-DPKG and SB-DPKG with the other DPKG selection scheme, called all-in DPKG (AI-DPKG) scheme. In AI-DPKG, all CRs perform DPKG task.

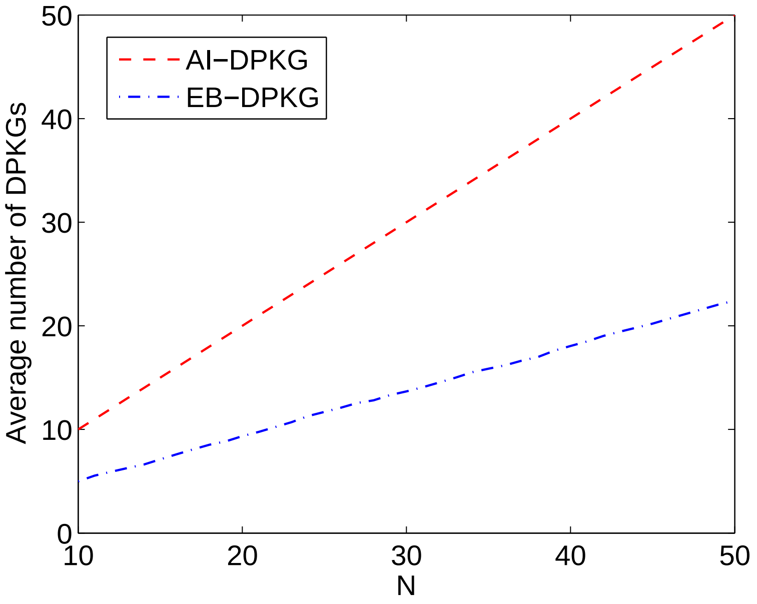


Figure 4.1: Average number of DPKG for AI-DPKG and EB-DPKG.

The average number of DPKGs vs. the number of CRs for AI-DPKG and EB-DPKG is depicted in Fig. 4.1. It is shown that by applying the proposed EB-DPKG method, L can be decreased up to 60% of that of AI-DPKG. In addition, by increasing the number of CRs, the difference between the two selection schemes increases which may benefit in avoiding congestion when the network gets crowded.

In order to show that the bound on the number of allowed compromised DPKGs is met, the number of compromised DPKGs after DPKG formation vs. the number of CRs is illustrated in Fig. 4.2. According to the figure, the number of compromised DPKGs for EB-DPKG is more than that of SB-DPKG as expected, since EB-DPKG selects CR with highest energy levels as DPKGs if they have acceptable probability of compromise. However, the curve of EB-DPKG is still far from the number of allowed compromised DPKGs, which ensures its security against Byzantine attackers.

Fig. 4.3 shows the communication overhead, measured by the number of messages transferred, when single PKG, AI-DPKG and EB-DPKG are applied. As shown in the figure, EB-DPKG significantly decreases overhead compared to AI-DPKG, especially when the number of CRs increases due to the reduced needs on the number of DPKGs. Moreover,

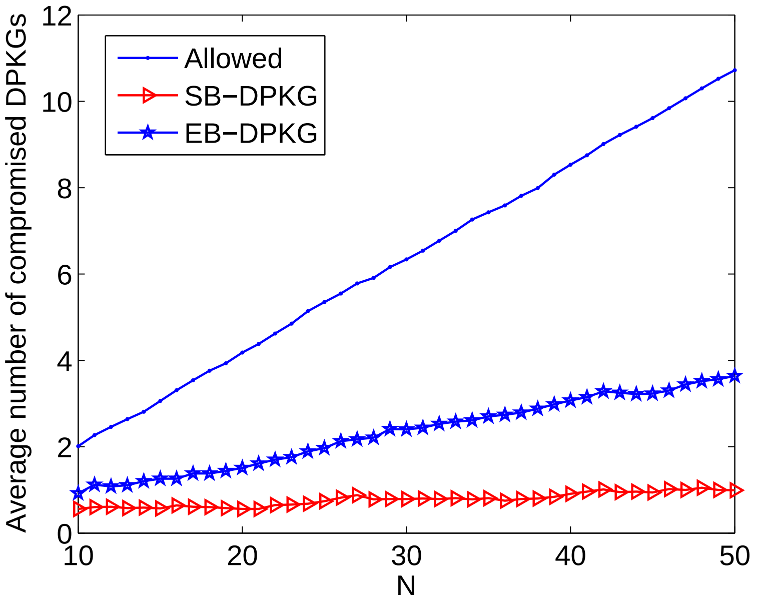


Figure 4.2: Average number of compromised DPKGs vs. the number of CRs.

the overhead of the EB-DPKG is only slightly larger than the single PKG. By considering the severe single point failure problem in the single PKG, EB-DPKG is more suitable for practical applications. The overhead for SB-DPKG and EB-DPKG are similar, since the number of DPKGs used is same in both schemes.

Fig. 4.4 shows the energy levels of 30 CRs (including DPKGs) before and after DPKG formation. As shown in the figure, compared to AI-DPKG and SB-DPKG, EB-DPKG can better balance the energy consumption amongst CRs. It is because in EB-DPKG, CRs with low energy levels are not selected as DPKGs. In addition, it is shown in Fig. 4.4a that applying all CRs in DPKG formation can substantially decrease the energy levels of some CRs and make those CRs die fast.

To compare the performance between IBCKM and CBCKM, we consider the following three certificate types.

1) *ECC-based certificates*: ECC-based certificates are made up of ID_i, ECC-based public key and CA's signature.

2) *RSA-based certificates*: In this case, ID_i, RSA-based public key and CA's signature compose the certificate.

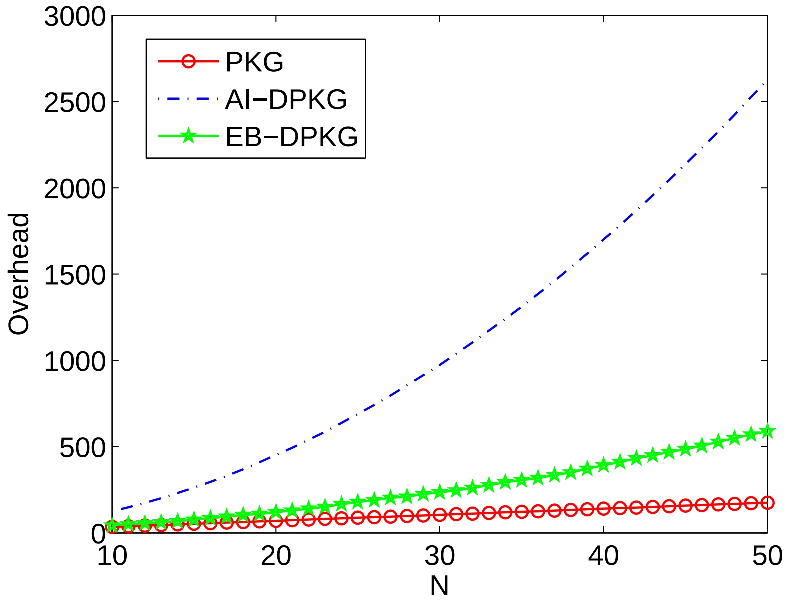
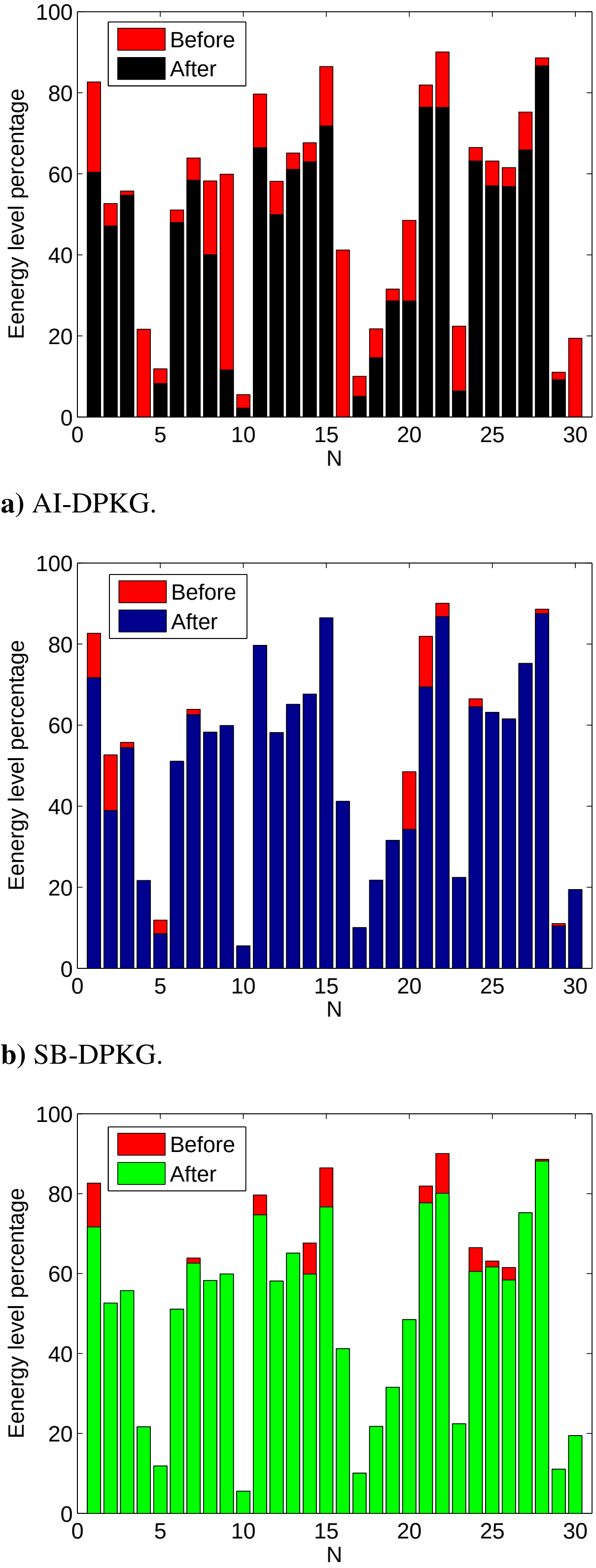


Figure 4.3: Overhead for PKG, AI-DPKG and EB-DPKG schemes.

3) *Hybrid certificates*: Hybrid certificates consist of ID$_i$, ECC-based public key and RSA-based CA's signature.

Fig. 4.5 shows the overhead with respect to the increase of the number of CRs from 10 to 50 and $K = \{2, 4, 6\}$ for IBCKM and ECC-CBCKM. Note that since in CBCKM, all three certificate types follow the same algorithm, they are similar in the number of messages transferred. As shown in the figure, IBCKM has a significant advantage in overhead compared to CBCKM specially when the number of CRs is fairly large. It is because CRs need to get general certificates in addition to channel certificates from CAs, while they only need to obtain the channel keys in IBCKM. Moreover, it is shown that overhead increases with the increase of the number of CRs for both schemes. However for both schemes, it dramatically increases when K increases.

The average power consumed by a CR is depicted in Fig. 4.6. It can be observed that the proposed IBCKM is more efficient in terms of power consumption compared to Hyb-CBCKM, RSA-CBCKM, and ECC-CBCKM. Such advantage in power consumption becomes more significant when K increases from 2 to 4 which results in 70% and 30% increase in the average consumed power for ECC-CBCKM and IBCKM, respectively.



(a) AI-DPKG.

(b) SB-DPKG.

(c) EB-DPKG.

Figure 4.4: Energy level before and after DPKG formation

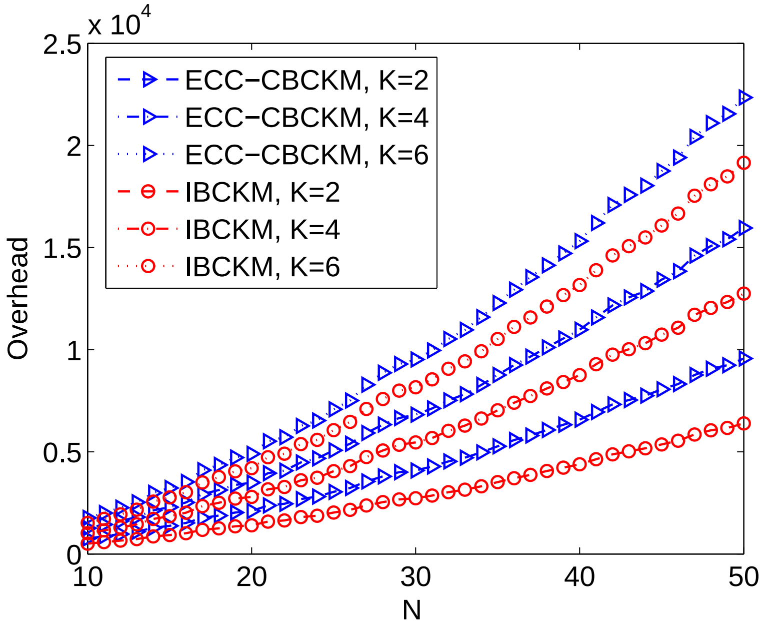


Figure 4.5: Overhead vs. the number of CRs for $K = \{2, 4, 6\}$.

Fig. 4.7 shows energy-efficiency v.s. the number of CRs. As expected, when the number of CRs increases, η decreases. According to the figure, IBCKM outperforms CBCKM for all the numbers of CRs considered. The reason is that the total power consumed by applying IBCKM is much less than that in CBCKM for each bit. In addition, the figure also shows that for a large number of CRs, the curves for IBCKM and CBCKM almost converge, because of the significant increase of power consumption.

Fig. 4.8 and Fig. 4.9 show the average battery life, in hours, for each CR with the variation of the number of CRs and K, respectively. As shown in both figures, the average CR battery life decreases with the increase of the number of CRs and K. As for IBCKM, it can improve the average CR battery life over ECC-CBCKM scheme by up to 35%. CRs applying Hyb-CBCKM or RSA-CBCKM schemes have similar but worst average battery life because of large certificate sizes.

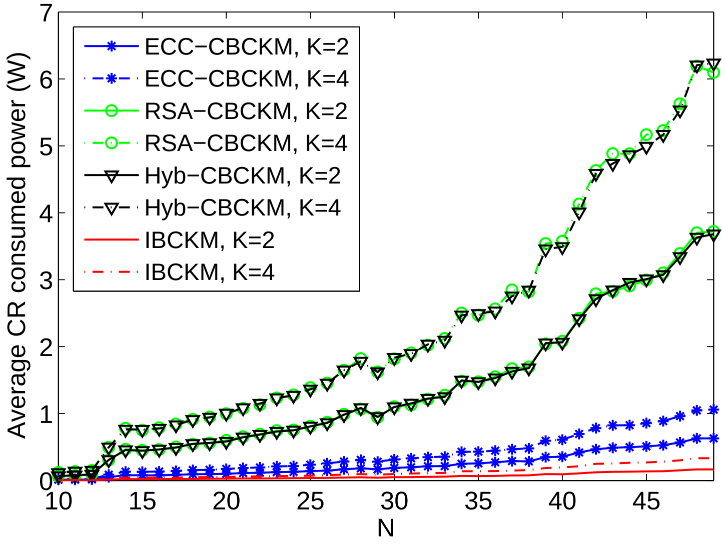


Figure 4.6: Average CR consumed power vs. the number of CRs for K=2 and K=4.

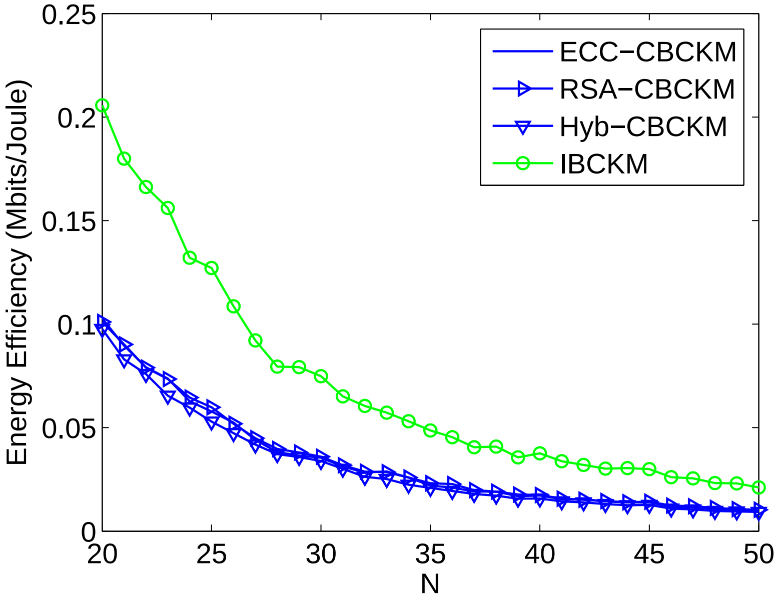


Figure 4.7: Energy efficiency vs. the number of CRs.

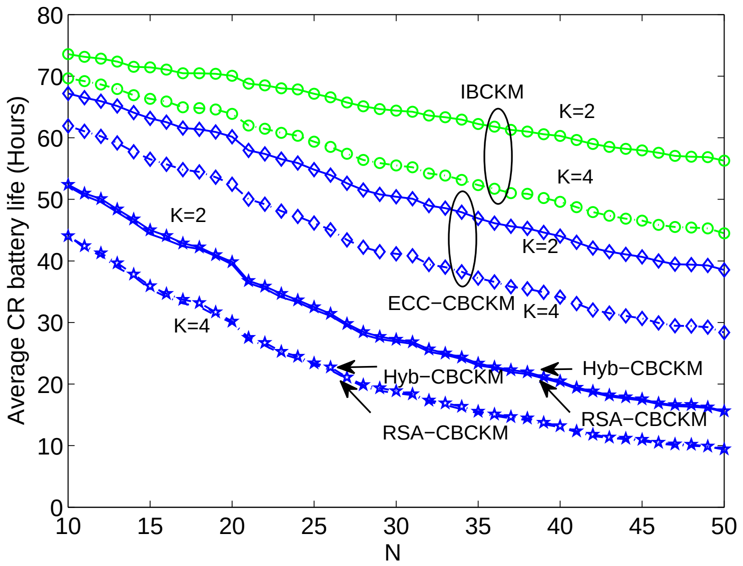


Figure 4.8: Average CR battery life vs. the number of CRs for K=2 and K=4.

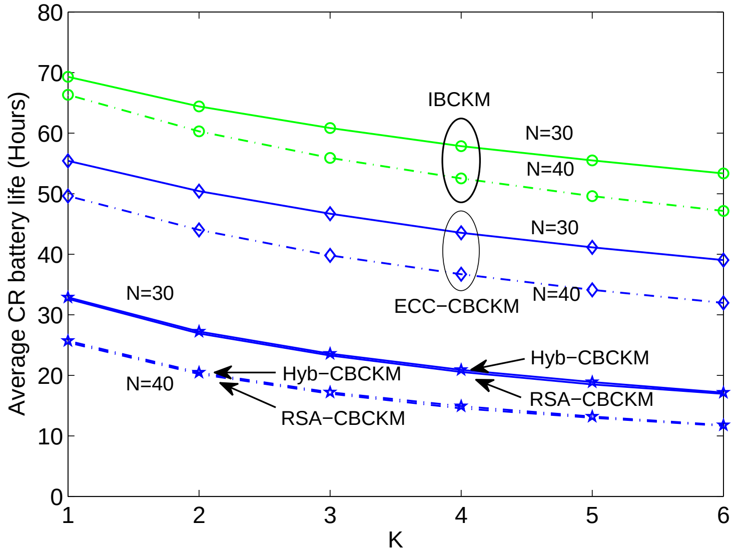


Figure 4.9: Average CR battery life vs. K for N=30 and N=40.

Chapter 5

Distributed private key generator assignment

DPKG assignment in IBC systems is vital for MCCRNs, since utilization of the vacant spectrum is highly time-limited, as a licensed user may appear at any time. It means that CRs may need to obtain a new private key for the next round of sensing-sharing which is energy consuming. In addition, SBS can preserve the energy of CRs and extend the lifetime of CRNs, by carefully assigning CRs as DPKGs. Therefore, in this chapter, we design a centralized DPKG assignment scheme for infrastructure-base MCCRNs, which can preserve energy-efficiency and fairness. Note that the proposed scheme can be applied in ad hoc MCCRNs. In tactical MCCRNs, and other hierarchical MCCRNs, the leader is a higher level authority, which usually has more resources, and in other ad hoc MCCRNs, a CR with good power resources can perform DPKG assignment, as leader, for other users. There are efficient and secure leader election methods available in literature which can be used in this case, e.g. [72], [73].

5.1 System model and problem formulation

Consider a network with N randomly distributed CRs and a SBS, located at the center. We assume synchronous communication, where the sender and receiver synchronize with each other before communication. Let t be the number of allowed compromised CRs. Then, $N = 2t + 1$. Threshold cryptography [50] is adopted to distribute the authority to DPKGs so that they can generate partial private keys for CRs and no t CRs can recover the private key. Therefore, $L = t + 1$, where L is the number of DPKGs. Define p_{ij}^{t} as the minimum transmit power of CR$_i$ to CR$_j$, which can be obtained as $p_{ij}^{t} = \overline{\gamma}_{ij} N_0 / (\theta_{ij})^{-\alpha}$. Note that $\overline{\gamma}_{ij}$ is the minimum acceptable average SNR of the received signal from CR$_j$ to CR$_i$, θ_{ij} is the distance between CR$_i$ and CR$_j$, α denote the path loss exponent and N_0 indicates the noise power.

Each CR has two roles. One role is to get service from L DPKGs and the other is to serve others by generating partial private keys as a DPKG. Define a matrix $\mathscr{X} = (x_{ij})_{N \times N}$, where $x_{ij} = 1$ denotes the DPKG$_j$ is assigned to CR$_i$, otherwise $x_{ij} = 0$.

The main objective is to minimize the consumed power of CRs for distributed private key generation process to attain energy-efficiency. Then, we define the first objective function as

$$\mathcal{F}_1 = \sum_{j=1}^{N} \sum_{i=1}^{N} \theta_{ij} x_{ij} \tag{5.1}$$

In addition, DPKG assignment should be fair to all DPKGs so that they desire to cooperate and the life-time of the network increases. We define a new fairness criterion as follows.

Definition 5.1. DPKG Fairness. All CRs, as DPKGs in the network should serve as close number of CRs as possible to the number they get service from.

In order to achieve DPKG fairness, we define our second objective as

$$\mathcal{F}_2 = \sum_{j=1}^{N} (\sum_{i=1}^{N} x_{ij} - \sum_{i=1}^{N} x_{ji})^2 \tag{5.2}$$

Let $\mathcal{F}$ be the function of two objectives as

$$\mathcal{F} = \{\mathcal{F}_i | i \in \{1, 2\}\} \tag{5.3}$$

Therefore, the multi-objective optimization problem is defined as

$$
\begin{aligned}
(\mathscr{P}_{MO}) \qquad & \min_{\mathscr{X}} \mathcal{F} && (5.4)\\
s.t. \quad & \sum_{j=1}^{N} x_{ij} = L, i \in \{1, \cdots, N\} && (5.5)\\
& \sum_{i=1}^{N} p_{ij}^{t} x_{ij} \leq P_{DPKGj}, j \in \{1, \cdots, N\} && (5.6)\\
& x_{ij} \in \{0, 1\}^N && (5.7)
\end{aligned}
$$

where P_{DPKGj} is the power resource of DPKG$_j$ to serve other CRs. The constraint (5.5) indicates that each CR needs to be assigned to L DPKGs to collect enough shares of the partial private keys. Note that the objective of $\mathcal{F}_1$ is to minimize the total energy consumption for partial private key handling in the network and $\mathcal{F}_2$ minimization aims to ensure DPKG fairness in the network. As a result, both searching and decision making should be considered to solve ($\mathscr{P}_{MO}$) [74]. To solve multi-objective optimization, several main approaches exist in the literature such as *priori* and *posteriori* methods [75]. In *priori* methods, decision maker employs a preference before solving the optimization problem. Conversely in *posteriori* methods, a set of all possible solutions is found and decision maker selects the best solution based on the preference. While both of these methods have weaknesses, a class of methods, called *interactive* methods, exist which can mitigate such issues by letting the decision maker interact with the program [75]. In interactive methods, preference is applied in the objective function, making the problem single objective. Because of its advantages, we consider interactive approach. First, by applying a weight vector $\mathbf{w} = \{w_i | i \in \{1, 2\}\}$, all objective functions, i.e. $\mathcal{F}_1$ and $\mathcal{F}_2$, are combined into a single objective function as

$$\mathcal{F} = \sum_{i=1}^{2} w_i \mathcal{F}_i \tag{5.8}$$

In this paper, the main goal is to reach an energy-efficient DPKG assignment, while maintaining an acceptable level of DPKG fairness. Thus, we define $\mathbf{w} = \{1, \vartheta\}$, where ϑ is the weight of $\mathcal{F}_2$ and considered as the preference. As a result, problem $\mathscr{P}_{MO}$ is transferred to

$$(\mathscr{P}_Z) \qquad Z = \min_{\mathscr{X},\vartheta} \sum_{j=1}^{N}\sum_{i=1}^{N} \theta_{ij} x_{ij} + \vartheta \sum_{j=1}^{N} (\sum_{i=1}^{N} x_{ij} - \sum_{i=1}^{N} x_{ji})^2 \tag{5.9}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} = L, i \in \{1, \cdots, N\} \tag{5.10}$$

$$\sum_{i=1}^{N} p_{ij}^{t} x_{ij} \leq P_{DPKGj}, j \in \{1, \cdots, N\} \tag{5.11}$$

$$x_{ij} \in \{0, 1\}^N, \tag{5.12}$$

which is a nonlinear integer programming problem.

5.2 Interactive solution for the optimization problem

In this section, a new interactive algorithm is proposed for the problem $(\mathscr{P}_Z)$. The proposed interactive algorithm is the core of the system, where decision maker, i.e., SBS, interacts with the problem.

The proposed interactive algorithm for DPKG assignment is shown in **Algorithm 5.2.1** and can be described as follows. In the first step, each CR$_i$ reports P_i and p_{ij}^t to SBS. Note that SBS sets $\{p_{ij}^t = \infty | j = i\}$ to avoid assignment of CR$_i$ to itself. Then, SBS has the information of the vector $\mathbf{P}_{DPKG} = \{P_i | i \in \{1, ..., N\}\}$ and the matrix $\mathscr{P} = (p_{ij}^t)_{N \times N}$. Afterwards, ϑ is initialized as ϑ_0 by SBS. Note that any value that captures scaling factor between $\mathcal{F}_1$ and $\mathcal{F}_2$ can be selected for ϑ_0, since the objective functions $\mathcal{F}_i$ have different amplitudes. Let $\iota = (1/NL^2)\mathcal{F}_2$, where $1/NL^2$ is the normalization coefficient guaran-

Algorithm 5.2.1 Interactive DPKG assignment

```
Initialization:
    • Each CR$_i$ reports $P_i$ and $\{p^t_{ij}|j \in \mathcal{S}_{\gamma_i}\}$ to SBS.
    • SBS builds $\mathbf{P}_{DPKG} = \{P_i|i \in \{1, ..., N\}\}$ and $\mathscr{P} = (p^t_{ij})_{N\times N}$, by setting $\{p^t_{ij} = \infty|j = i\}$.
    • Set $\iota = (1/NL^2)\sum_{j=1}^{N}(\sum_{i=1}^{N} x_{ij} - \sum_{i=1}^{N} x_{ji})^2$
    • Initialize $\vartheta$ by $\vartheta_0$ and $\iota_{th}$ according to SBS acceptable level of DPKG fairness, and set $\chi = \varsigma_a\vartheta_0$.
    • Run optimal DPKG assignment to obtain initial DPKG assignment $\mathscr{X}$
Main loop:
while $\iota > \iota_{th}$ do
    $\vartheta = \vartheta + \chi$
    Run optimal DPKG assignment to obtain $\mathscr{X}$
end while
while $\iota_{th} - \iota > \epsilon_\iota$ do
    $\vartheta = \vartheta - \varsigma_b\chi$
    Run optimal DPKG assignment to obtain $\mathscr{X}$
end while
Output: $\mathscr{X}$
```

teeing $\iota < 1$. Define ι_{th} as the reference point, acceptable level of fairness, which is specified by SBS. We introduce χ to adjust ϑ in each iteration which is assumed to be $\varsigma_a\vartheta_0$, $0 < \varsigma_a < 1$. Then, the initial DPKG assignment, $\mathscr{X}$, is obtained by the optimal DPKG assignment which will be discussed in the next section. Second step is the interaction with SBS or decision maker, where SBS checks whether ι satisfies its reference point, i.e. $\iota < \iota_{th}$. Until this has not been met, SBS increases the weight of objective $\mathcal{F}_2$ with χ and perform optimal DPKG assignment. The obtained solution guarantees the acceptable fairness level, but there is a chance that $\mathcal{F}_1$ can be further decreased, while ι_{th} is met. Let ϵ_ι be a small real number. While $\iota_{th} - \iota > \epsilon_\iota$, ϑ can be adjusted as $\vartheta = \vartheta - \varsigma_b\chi$, where ς_b is a positive real number such that $\varsigma_b < 1$, and incorporated to optimal solution to obtain DPKG assignment.

5.3 Optimal DPKG assignment

In this section, we analyze the optimal solution of the problem $(\mathscr{P}_Z)$ as the program to which decision maker communicates with. Since $\sum_{i=1}^{N} x_{ji} = L, j \in \{1, \cdots, N\}$, we can rewrite (5.9) as

$$Z = \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} \theta_{ij}x_{ij} + \vartheta \sum_{j=1}^{N}(\sum_{i=1}^{N} x_{ij} - L)^2 \tag{5.13}$$

$$= \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} \theta_{ij}x_{ij} + \vartheta \sum_{j=1}^{N}(\sum_{i=1}^{N} x_{ij})^2 - \sum_{j=1}^{N}\sum_{i=1}^{N} 2\vartheta L x_{ij} + \sum_{j=1}^{N} \vartheta L^2 \tag{5.14}$$

$$= \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} (\theta_{ij} - 2\vartheta L)x_{ij} + \vartheta \sum_{j=1}^{N}(\sum_{i=1}^{N} x_{ij})^2 + \vartheta N L^2 \tag{5.15}$$

$$= \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} (\theta_{ij} - 2\vartheta L)x_{ij} + \vartheta \sum_{j=1}^{N}\sum_{i=1}^{N} x_{ij}^2 + 2\vartheta \sum_{j=1}^{N}\sum_{\substack{i=1 \\ i \neq i'}}^{N} x_{ij}x_{i'j} + \vartheta N L^2 \tag{5.16}$$

Because of the constraint $x_{ij} \in \{0, 1\}^N$, $\sum_{j=1}^{N}\sum_{i=1}^{N} x_{ij}^2$ can be replaced by $\sum_{j=1}^{N}\sum_{i=1}^{N} x_{ij}$ in (5.16), i.e.,

$$Z_1 = \vartheta N L^2 + \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} (\theta_{ij} + \vartheta - 2\vartheta L)x_{ij} + 2\vartheta \sum_{j=1}^{N}\sum_{\substack{i=1 \\ i \neq i'}}^{N} x_{ij}x_{i'j} \tag{5.17}$$

Therefore, by considering constraints (5.5), (5.6) and (5.7), Problem $\mathscr{P}_Z$ is transferred to the following optimization problem

$$(\mathscr{P}_{Z_1}) \qquad Z_1 = \min_{\mathscr{X}} \sum_{j=1}^{N}\sum_{i=1}^{N} (\theta_{ij} + \vartheta - 2\vartheta L)x_{ij} + 2\vartheta \sum_{j=1}^{N}\sum_{\substack{i=1 \\ i \neq i'}}^{N} x_{ij}x_{i'j} \tag{5.18}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} = L, i \in \{1, \cdots, N\} \tag{5.19}$$

$$\sum_{i=1}^{N} p_{ij}^t x_{ij} \leq P_{DPKGj}, j \in \{1, \cdots, N\} \tag{5.20}$$

$$x_{ij} \in \{0, 1\}^N \tag{5.21}$$

Problem $\mathscr{P}_{Z_1}$ is a nonlinear integer programming. To study the optimality of $\mathscr{P}_{Z_1}$, upper and lower bounds will be obtained in the following.

5.3.1 Upper bounds

An upper bound of the problem $\mathscr{P}_{Z_1}$ is associated with a feasible solution of $\mathscr{P}_{Z_1}$. Define Z^* as the best known feasible solution to problem $\mathscr{P}_{l_{Z_1}}$. Z^* can be obtained as follows. First, CRs with smaller $(\theta_{ij} + \vartheta - 2\vartheta L)$ are assigned to DPKG$_j$ while constraint (5.20) is satisfied. Then, extra assignments for each CR$_i$ are released to guarantee the constraint (5.19).

5.3.2 Lower bounds via Lagrangian relaxation

Lagrangian relaxation provides tight bounds for the original problem. We dualize the constraints (5.19) and (5.20) to produce a Lagrangian problem which can be solved easily and whose optimal value is a lower bound on the optimal value of the Problem $\mathscr{P}_{Z_1}$. Hence, the Lagrangian problem $\mathscr{P}_{LR_u}$ is defined as

$$\begin{aligned}(\mathscr{P}_{LR_u})\ Z_D(u) &= \min_{\mathscr{X}\in\{0,1\}} \sum_{j=1}^{N}\sum_{i=1}^{N}(\theta_{ij} + \vartheta - 2\vartheta L)x_{ij} + 2\vartheta \sum_{j=1}^{N}\sum_{\substack{i=1\\ i\neq i'}}^{N} x_{ij}x_{i'j} \\ &\quad + \sum_{j=1}^{N} u_{pj}(\sum_{i=1}^{N} p_{ij}^{t} x_{ij} - P_{DPKGj}) + \sum_{i=1}^{N} u_{li}(\sum_{j=1}^{N} x_{ij} - L) \qquad (5.22) \\ &= \min_{\mathscr{X}\in\{0,1\}} \sum_{j=1}^{N}\sum_{i=1}^{N}(\theta_{ij} + \vartheta - 2\vartheta L + p_{ij}^{t} u_{pj} + u_{li})x_{ij} + 2\vartheta \sum_{j=1}^{N}\sum_{\substack{i=1\\ i\neq i'}}^{N} x_{ij}x_{i'j} \\ &\quad - \sum_{j=1}^{N} P_{DPKGj} u_{pj} - \sum_{i=1}^{N} L u_{li} \qquad (5.23)\end{aligned}$$

where u_{pi} and u_{li} are Lagrangian multipliers, and $u = \{u_{pi}, u_{li}\}, i \in \{1, ..., N\}$. The Problem $\mathscr{P}_{l_{Z_1}}$ reduces to an unconstrained quadratic binary programming problem, which can be solved by methods in [76–78]. In fact, u should be determined by the optimal solution to the dual problem

$$Z_D = \min_{u} Z_D(u) \tag{5.24}$$

Since $Z_D(u)$ is non-differentiable, the subgradient method can be applied, where gradients are substituted with subgradients. Let u^0 be the initial value. The outcome of subgradient method is a sequence of u which is obtained by [79]

$$u^{k+1} = \max\{u^k + S_k^{SG}\Delta u, 0\} \tag{5.25}$$

where $\Delta u = \{\Delta u_l, \Delta u_p\}$, $\Delta u_l = \sum_{j=1}^{N} x_{ij}^k - L$, $\Delta u_p = \sum_{i=1}^{N} p_{ij}^t x_{ij}^k - P_{DPKGj}$, x_{ij}^k is an optimal solution to problem $\mathscr{P}_{LR_{u^k}}$ and S_k^{SG} is a scalar step size. A common practical choice for the step size is given by [79]

$$S_k^{SG} = \frac{\varsigma_k^{SG}(Z^* - Z_D(u^k))}{\parallel \Delta u \parallel^2} \tag{5.26}$$

where $\{0 < \varsigma_k^{SG} \leq 2 | \varsigma_k^{SG} \in \mathbb{N}\}$. The sequence ς_k^{SG} is determined by setting $\varsigma_k^{SG} = 2$ at the beginning and halving ς_k^{SG} whenever $Z_D(u^k)$ has failed to increase in a specific number of iterations.

After obtaining upper and lower bounds, a branch-and-bound algorithm adopted to achieve optimal solution. Let $\tilde{\mathscr{X}}$ be the feasible solution and UB be the corresponding upper bound of $\mathscr{P}_{Z_1}$, Z^*. For each CR$_i$, compute a lower bound LB$_i$ by lagrangian relaxation with $\tilde{x}_{ij}$ fixed at $1 - x_{ij}$. If LB$_i \geq$ UB, set $x_{ij} = \tilde{x}_{ij}$. Update UB if a better feasible solution is found during the procedure. Then, at each node, a LB of the corresponding subproblem is computed. If LB $\geq$ UB, then the node is fathomed. Otherwise, the node is branched into two nodes by $x_{ij} = 0$ and $x_{ij} = 1$.

5.4 Low-complexity DPKG assignment

In order to implement a low-complexity solution to problem $\mathscr{P}_{l_{Z_1}}$, we propose a new DPKG assignment algorithm, as shown in **Algorithm 5.4.1**.

The development of the proposed algorithm is as follows. There are multiple demands for service of each DPKG based on the received p^t_{ij}. **Algorithm 5.4.1** is run for each DPKG and CRs are assigned to DPKG$_j$ till P_{DPKGj} has been reached. CRs only need to get service from L DPKGs, but chances are they are assigned more than L DPKGs, called over-achieved CRs, or less than L DPKGs, called under-achieved CRs. Therefore, a procedure should be performed such that over-achieved CRs are released from extra DPKGs of need which opens up room for under-achieved CRs to get service from those DPKGs. To assure DPKG fairness, $\iota \leq \iota_{th}$, we can have $(1/NL^2)(\sum_{i=1}^{N} x_{ij} - \sum_{i=1}^{N} x_{ji})^2 \leq \iota_{th}/N$ for each $j \in \{1, ..., N\}$. As a result, $\sum_{i=1}^{N} x_{ij} \leq (1 + \sqrt{\iota_{th}})L$ and $\sum_{i=1}^{N} x_{ij} \geq (1 - \sqrt{\iota_{th}})L$ should hold.

We now describe the proposed DPKG assignment in details. **Algorithm 5.4.1** starts with an initialization procedure. First, SBS forms a matrix $\mathscr{P} = (p^t_{ij})_{N\times N}$. In addition, SBS initializes the assignment matrix $\mathscr{X}$ with $\{x_{ij} = 0 | i, j \in \{1, ..., N\}\}$ and defines two sets for over-achieved, $\mathcal{S}_{O\mathcal{A}}$, and under-achieved CRs, $\mathcal{S}_{\mathcal{U}\mathcal{A}}$, respectively. Note that in the first round, all CRs are included in both $\mathcal{S}_j$ and $\mathcal{S}_i$; however, after that, the algorithm is run for under-achieved CRs over DPKGs of over-achieved CRs only. After initialization, CRs are sorted based on the incremental order of $p_{i,j} \in \mathscr{P}$ and are assigned one by one while $\sum_{i=1}^{N} p^t_{ij} x_{ij} \leq P_{DPKGj}$ and $\sum_{i=1}^{N} x_{ij} \leq (1 + \sqrt{\iota_{th}})L$. It means that CRs with the lower transmit power are assigned to DPKG$_j$. In the next phase, the sets $\mathcal{S}_{O\mathcal{A}}$ and $\mathcal{S}_{\mathcal{U}\mathcal{A}}$ should be updated and unnecessary assignments should be released to ensure energy efficiency.

If $\sum_{j=1}^{N} x_{ij} > L$ for CR$_i$ in $\mathcal{S}_j$, the set $\mathcal{S}_{RE} = \{p_{i,j} | x_{ij} = 1, j \in \{1, ..., N\}\}$ is defined and SBS sorts $\mathcal{S}_{RE}$ based on the decremental order. Let $L_{OA} = \sum_{j=1}^{N} x_{ij} - L$ be the number of extra assignments of CR$_i$. Then, first L_{OA} assignments of CR$_i$ are vacated, i.e. $x_{ij} = 0$.

In other words, the most costly assignments with high $p_{i,j}$ are released to provide more DPKG capacity for accommodating more under-achieved CRs. If CR$_j$ has enough capacity, $P_{DPKGj} - \sum_{i=1}^{N} p_{ij}^t x_{ij}$, to accommodate any CR$_i$, $i \in S_{UA}$, with minimum requirement, i.e. $i = \text{argmin}(p_{ij}^t | i \in S_{UA})$, CR$_j$ is included in $\mathcal{S}_{\mathcal{OA}}$. Otherwise, if $\sum_{j=1}^{N} x_{ij} < L$, CR$_i$ should be included in $\mathcal{S}_{\mathcal{UA}}$. Both sets $\mathcal{S}_{\mathcal{OA}}$ and $\mathcal{S}_{\mathcal{UA}}$ will be used in the next round as $\mathcal{S}_i$ and $\mathcal{S}_j$. It means that DPKGs with released assignments can serve under-achieved CRs up to their remaining capacity, i.e. $P_{DPKGj} - \sum_{i=1}^{N} p_{ij}^t x_{ij}$. A set $\mathcal{S}_{UT}$ is defined such that $\sum_{i=1}^{N} x_{ij} < (1 - \sqrt{\iota_{th}})L$ for each member CR$_j$, who fails to satisfy fairness. Therefore, each CR$_j$ should serve more CRs. Then, for each CR$_j$, $j \in \mathcal{S}_{UT}$, a set $\mathcal{S}_{UT_i}$ is defined by sorting CRs according to the incremental order of $p_{ij}^t \in \mathscr{P}$. For $i \in \mathcal{S}_{UT_i}$, if $x_{ij} = 0$, the assignment should change to $x_{ij} = 1$, while both $\sum\limits_{m \in \mathcal{S}_{UT_i}} p_{mj} x_{mj} < P_{DPKGj}$ and $\sum\limits_{m \in \mathcal{S}_{UT_i}} x_{mj} \leq (1 - \sqrt{\iota_{th}})L$ are met. Since the extra assignment is added to CR$_i$, $i \in \mathcal{S}_{UT_i}$, one of its assignments should be released to fix the number of assignments at L. Then, the assignment with the highest cost, p_{ik}, is released as $x_{in} = 0$, if $\sum_{i=1}^{N} x_{in} - 1 \geq (1 - \sqrt{\iota_{th}})L$, to maintain fairness for DPKG$_n$.

5.5 Simulation results

In this section, we consider a network consisting of a SBS at the center and CRs randomly deployed. The simulation parameters are $\alpha = 3$, $\overline{\gamma}_{ij} = 2.5$, $\varsigma_a = 0.5$, $\epsilon_\iota = 0.1$, $\varsigma_b = 0.5$, $\iota_{th} = 0.3$. Furthermore, $\vartheta_0 = \overline{\theta}/L^2$, where $\overline{\theta}$ is the average of $\theta_{ij}, i \neq j$. The simulations were performed on a PC with Intel Core i7-2670QM CPU and 6GB of RAM, and for the increase of the number of CRs from 5 to 30.

First, in Fig. 5.1, we compare the power consumption of the network for the proposed DPKG assignment algorithm, optimal DPKG assignment with and without DPKG fairness consideration and random selection, where each CR selects its DPKGs randomly. It is shown in the figure that total power consumption increases when the number of CRs increases. This increase is highly considerable for random selection, since each CR selects

DPKGs regardless of the corresponding channel gain and required transmit power. Thus, CRs consume a huge amount of power for key distribution. However, by assigning DPKGs to CRs with better channel gains and less transmit power, the proposed algorithm not only improves the power consumption remarkably, but its performance is close to the optimal DPKG assignment. Moreover, optimal DPKG assignment without DPKG fairness consideration has a similar, slightly better, power consumption compared to both proposed algorithm and optimal one which shows fairness can be taken into account without a considerable compromise on power consumption.

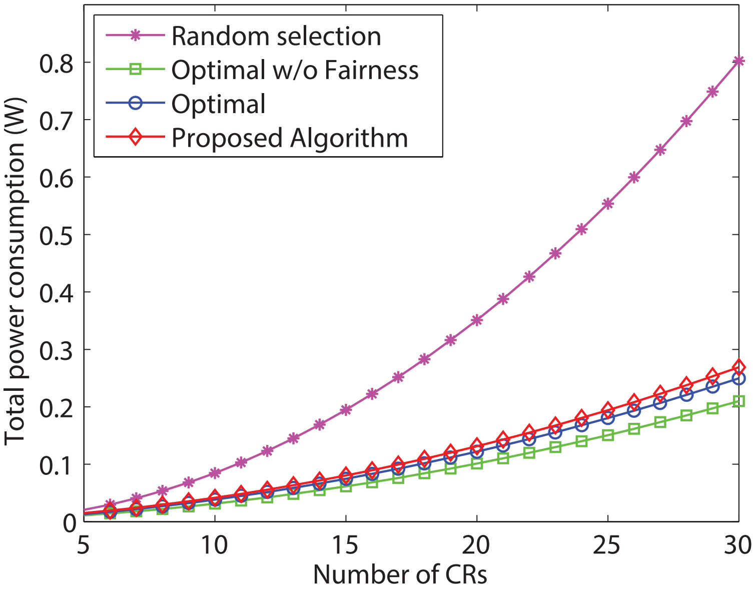


Figure 5.1: Total power consumption vs. Number of CRs

Fig. 5.2 shows the average DPKG fairness, defined as $\mathcal{F}_1/N$ for the proposed DPKG assignment algorithm, optimal DPKG assignment with and without DPKG fairness. When the number of CRs increases, the average fairness of optimal assignment without fairness increases drastically, since each CR may be assigned to serve more CRs as DPKG compared to the number of CRs it receives service from. On the other hand, the average DPKG fairness increases at considerably lower pace for the proposed algorithm and close to the optimal assignment, since each CR only serves a number of CRs that satisfy condition on ι_{th}.

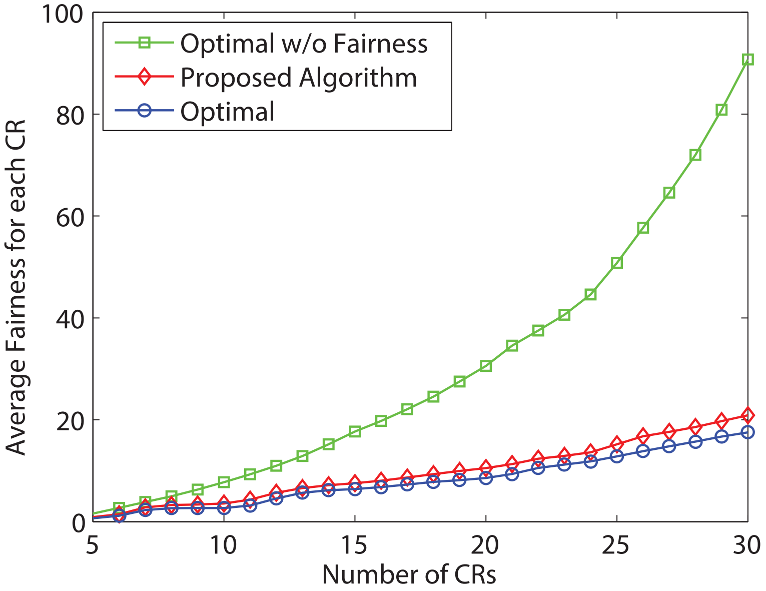


Figure 5.2: Average fairness for each CR vs. Number of CRs

The processing time of the proposed DPKG assignment algorithm and optimal DPKG assignment is shown in Fig. 5.3. It is shown that the optimal assignment takes a considerable time, while the processing time for the proposed algorithm constantly maintains at a low level and is negligible compared to the optimal one. Therefore, the proposed algorithm can clearly outperform the optimal solution in terms of computational complexity. By further considering the fact that the proposed algorithm can achieve near optimal performance in power consumption, it is a better fit for applications in agile MCCRNs. For example, the processing time of DPKG assignment for key distribution in CMSS can be crucial as the sensing time is limited.

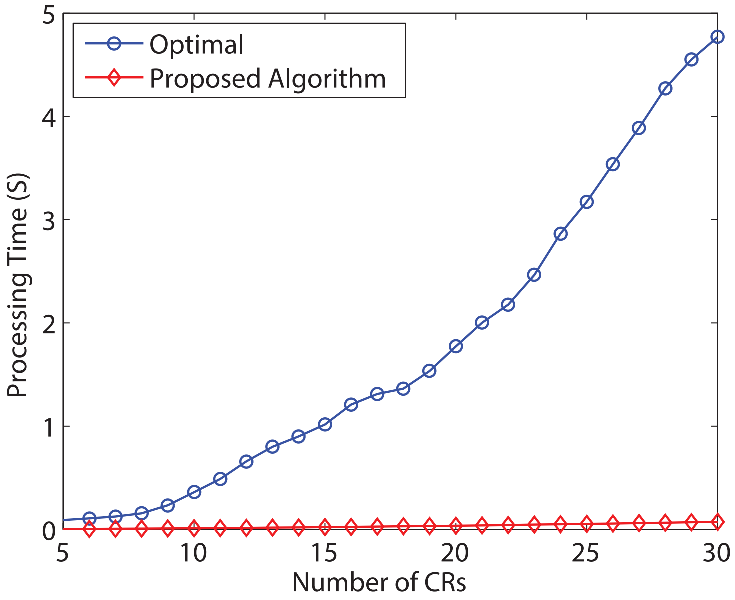


Figure 5.3: Processing Time vs. Number of CRs

Algorithm 5.4.1 DPKG assignment

```
**Initialization:**
    • SBS forms a matrix $\mathscr{P} = (p_{ij}^t)_{N\times N}$.
    • Initialize $\mathscr{X}$ with $\{x_{ij} = 0|i, j \in \{1, ..., N\}\}$, $\mathcal{S}_{\mathcal{OA}} = \varnothing$, $\mathcal{S}_{\mathcal{UA}} = \varnothing$, $round = 1$.
while $\mathcal{S}_{\mathcal{UA}} \neq \varnothing \;||\; round = 1$ do
    $\mathcal{S}_j = \{i|i \in \{1, ..., N\}\}$ in $round = 1$, $\mathcal{S}_j = \{i|i \in \mathcal{S}_{\mathcal{UA}}\}$, otherwise.
    $\mathcal{S}_i = \{j|j \in \{1, ..., N\}\}$ in $round = 1$, $\mathcal{S}_i = \{j|j \in \mathcal{S}_{\mathcal{OA}}\}$, otherwise.
    for $j \in \mathcal{S}_i$ do Sort $\mathcal{S}_j$ according to the incremental order of $p_{i,j} \in \mathscr{P}$
        while $\sum_{i=1}^{N} p_{ij}^t x_{ij} \leq P_{DPKGj}$ and $\sum_{i=1}^{N} x_{ij} \leq (1 + \sqrt{\iota_{th}})L$ do
            for $i \in \mathcal{S}_j$ do $x_{ij} = 1$ end for
        end while
    end for
    for $i \in \mathcal{S}_j$ do
        if $\sum_{j=1}^{N} x_{ij} < L$ then $\mathcal{S}_{\mathcal{UA}} \longleftarrow \{i\}$
        else if $\sum_{j=1}^{N} x_{ij} > L$ then
            Set $L_{OA} = \sum_{j=1}^{N} x_{ij} - L$
            Set $\mathcal{S}_{RE} = \{p_{i,j}|x_{ij} = 1, j \in \{1, ..., N\}\}$ and sort $\mathcal{S}_{RE}$ based on
            a decremental order
            for $j = \mathcal{S}_{RE}(1)$ to $\mathcal{S}_{RE}(L_{OA})$ do $x_{ij} = 0$ end for
            for $j = 1$ to $N$ do
                if $P_{DPKGj} - \sum_{i=1}^{N} p_{ij}^t x_{ij} \geq \min(p_{ij}^t|i \in S_{UA})$ then $\mathcal{S}_{\mathcal{OA}} \longleftarrow \{j\}$
                end if
            end for
        end if
    end for
    Update $round$
end while
for $j = 1$ to $N$ do
    if $\sum_{i=1}^{N} x_{ij} < (1 - \sqrt{\iota_{th}})L$ then
        $\mathcal{S}_{UT} \longleftarrow \{j\}$
    end if
end for
for $j \in \mathcal{S}_{UT}$ do $\mathcal{S}_{UT_i} = \{$sort $i$ according to incremental order of $p_{ij}^t \in \mathscr{P}\}$
    for $i \in \mathcal{S}_{UT_i}$ do
        if $x_{ij} = 0$ then
            while $\sum_{m\in\mathcal{S}_{UT_i}} p_{mj}x_{mj} < P_{DPKGj}$ and $\sum_{m\in\mathcal{S}_{UT_i}} x_{mj} \leq (1 - \sqrt{\iota_{th}})L$ do $x_{ij} = 1$
                while $\{x_{in} = 1|n \in$ decrementally sorted $p_{ik} \in \mathscr{P}, i \in \mathcal{S}_{UT_i}$,
                $k \in \{1, ..., N\}\}$ and $\sum_{i=1}^{N} x_{in} - 1 \geq (1 - \sqrt{\iota_{th}})L$ do $x_{in} = 0$
                end while
            end while
        end if
    end for
end for
```

Chapter 6

Securing coalitional game against cooperative Byzantine attacks on distributed cooperative spectrum sensing

In this chapter, we propose a cooperative Byzantine attack on the proposed distributed cooperative spectrum sensing. This type of attack is more intelligent than regular Byzantine attacks, as Byzantine attackers collude to spread themselves over as many coalitions as possible, and perform Byzantine attack, which significantly threatens the availability in MCCRNs. Then, we propose a hierarchical ID-based key management to secure the proposed coalitional game against such attacks.

6.1 Distributed cooperative attack on the multi-channel spectrum sensing

In this section, the proposed attack scenario is discussed first. Then, a multi-channel game (MCG) is proposed for ATTs.

6.1.1 Attack Scenario

Consider Byzantine attack and assume that each CR can change its own local decision prior to sending it to the coalition head. The ATTs' objective is to maximize the total number of available PCs they can attack. In addition, since OR-rule is chosen at the coalition head, if the sensing constraint permits, one ATT is enough to change the final decision in a coalition (PC). Then, the number of ATTs in each coalition should be minimized so that they can be distributed to as many PCs as possible. Note that although OR-rule is used in our discussion, the proposed method can be applied to other fusion rules.

The proposed attack scenario consists of two phases.

Phase 1

Since ATTs should follow regular CRs' objective (or act like ordinary CRs) to prevent from being detected, ATTs can only maximize the number of invaded PCs under the condition that the actual number of *available channels* is maximized. Moreover, since ATTs should adjust their game with the CRs' coalitional game, there is a possibility that more than one potential ATT is assigned to the same PC. Hence, in phase 1, after gathering information (i.e., SNR) of their neighbors (including CRs), ATTs play a coalitional game among themselves to determine which ATT should play on which PC so that as few as possible ATTs are assigned to each coalition.

Define PC allocation matrix as $\mathscr{X} = (x_{ij})_{N \times M}$, where $x_{ij} = 1$ means the allocation of PC j to CR i ($x_{ij} = 0$ otherwise). The optimization problem for ATTs then can be formulated as

$$\max_{\mathscr{X},\mathscr{A}} \quad \sum_{j=1}^{M} \mathscr{I}_j^A(Q_m^j, Q_f^j, A_{\mathcal{C}}^j) \tag{6.1}$$

$$s.t. \quad \sum_{j=1}^{M} x_{ij} \leq K, \; i \in \{1, \cdots, N\} \tag{6.2}$$

$$0 \leq \sum_{i=1}^{N} x_{ij} < \mathcal{C}_{max}^{Dist}, \; j \in \{1, \cdots, M\} \tag{6.3}$$

where $A_{\mathcal{C}}^{j} = \sum_{i \in \mathscr{A}_j} a_{ij}$ is the number of attackers in the coalition j, $x_{i,j}^{a}$ is the channel allocation to ATTs, $A_{max,\mathcal{C}}$ denotes the maximum number of ATTs on each PC and $\mathscr{I}_j^A(Q_m^j, Q_f^j, A_{\mathcal{C}}^j)$ is defined as

$$\mathscr{I}_j^A(Q_m^j, Q_f^j, A_{\mathcal{C}}^j) = \begin{cases} 1, & Q_m^j < \overline{Q_m}, Q_f^j < \overline{Q_f}, A_{\mathcal{C}}^j < A_{max,\mathcal{C}} \\ 0, & \text{otherwise.} \end{cases} \tag{6.4}$$

The constraint (6.2) means the maximum number of PCs assigned to CR i for sensing is K, and the constraint (6.3) limits the maximum number of sensing CRs for each PC j.

Phase 2

In the second phase, based on the results of the previous phase, ATTs try to emulate the behavior of the honest CRs and play with them on the PCs. Therefore, the objective in this phase is to maximize the number of *available channels*, as discussed in Chapter 2. Since a minimum number of ATTs are determined on each PC, unnecessary ATTs which are already on the same PC (coalition) should anonymously leave the PC according to the results of Phase 1, and play in another favorable coalition. Therefore, an ATT can change the information so that it will be forced to be excluded by CRs.

After Phase 2, stable coalitions on different PCs will be formed with maximum number of PCs assigned to ATTs. Then, each ATT can attack the coalition it locates.

6.1.2 Multi-channel coalitional game attack

Since the coalition formation in phase 2 has been discussed in Chapter 2, in this section, we focus on the coalitional game formulation in phase 1 only. Define $v_j(\mathcal{C})$ as the coalition function of the coalition $\mathcal{C}$ on PC j. To apply coalitional game theory, the coalition function should be appropriately defined. According to (6.4), $v_j(\mathcal{C})$ should be a decreasing function of the number of attackers in the coalition ($A_{\mathcal{C}}^j$), false alarm probability ($Q_{f,\mathcal{C}}^j$), and miss-detection probability ($Q_{m,\mathcal{C}}^j$). Since ATTs should follow the game played in the CRN by honest CRs, according to Chapter 2, the coalition function for ATTs can be written as

$$v_j(\mathcal{C}) = 1 - C_f(Q^j_{f,\mathcal{C}}) - C_m(Q^j_{m,\mathcal{C}}) - C_A(A^j_{\mathcal{C}}) \tag{6.5}$$

where the cost functions $C_f(Q^j_{f,\mathcal{C}})$, $C_m(Q^j_{m,\mathcal{C}})$ and $C_A(A^j_{\mathcal{C}})$ are increasing functions of $Q^j_{f,\mathcal{C}}$, $Q^j_{m,\mathcal{C}}$ and $A^j_{\mathcal{C}}$, respectively.

There are two possibilities for selecting suitable cost functions which are described as follows.

- Satisfied constraints:

This case implies that $Q^j_{f,\mathcal{C}} < \overline{Q_f}$, $Q^j_{m,\mathcal{C}} < \overline{Q_m}$ and $A^j_{\mathcal{C}} < A_{max,\mathcal{C}}$. Note that as long as the constraints are met, any change of the system variables such as $Q^j_{m,\mathcal{C}}$, $Q^j_{f,\mathcal{C}}$ and $A^j_{\mathcal{C}}$ has no impact on (6.4). In other words, the corresponding PC remains available. Thus, for ATTs, in order to achieve the objective of assigning ATTs to more PCs, the number of ATTs assigned to each coalition should be minimized. By considering the opposite effects of the coalition size on $Q^j_{f,\mathcal{C}}$ and $Q^j_{m,\mathcal{C}}$, as shown in (2.3) and (2.4), the defined cost function $C_f(Q^j_{f,\mathcal{C}})$ should dominate $C_m(Q^j_{m,\mathcal{C}})$ so as to minimize the number of CRs in each coalition. Similarly, $C_A(A^j_{\mathcal{C}})$ should dominate $C_m(Q^j_{m,\mathcal{C}})$ as well. Furthermore, $C_f(Q^j_{f,\mathcal{C}})$ should also dominate $C_A(A^j_{\mathcal{C}})$. We consider an opposite case for explanation. For example, assume one possible coalition consists of two ATTs, i.e. $\mathcal{C}_1 = \{ATT_1, ATT_2\}$, and the other one consists of one ATT and two CRs, i.e. $\mathcal{C}_2 = \{ATT_1, CR_1, CR_2\}$. If $C_A(A^j_{\mathcal{C}})$ is more important than $C_f(Q^j_{f,\mathcal{C}})$, $\mathcal{C}_2$ is preferred, since the number of ATTs is smaller. However, as ATTs play game with regular CRs according to Phase 2, $\mathcal{C}_1$ is preferred because of the smaller number of players ($|\mathcal{C}_1| < |\mathcal{C}_2|$) and $\mathcal{C}_2$ is not eventually selected. Therefore, $C_f(Q^j_{f,\mathcal{C}})$ should dominate $C_A(A^j_{\mathcal{C}})$.

In summary, we have the following relation among three cost functions.

$$\max_{A^j_{\mathcal{C}} < A_{max,\mathcal{C}}, \mathcal{C} \in \mathfrak{C}_j} C_A(A^j_{\mathcal{C}}) < \min_{Q^j_{f,\mathcal{C}} < \overline{Q_f}, \mathcal{C} \in \mathfrak{C}_j} C_f(Q^j_{f,\mathcal{C}}). \tag{6.6}$$

$$\max_{Q^j_{m,\mathcal{C}}<\overline{Q_m},\mathcal{C}\in\mathfrak{C}_j} C_m(Q^j_{m,\mathcal{C}}) < \min_{A^j_{\mathcal{C}}<A_{max,\mathcal{C}},\mathcal{C}\in\mathfrak{C}_j} C_A(A^j_{\mathcal{C}}). \tag{6.7}$$

where $\mathfrak{C}_j$ is the set of all possible coalitions over PC j.

Note that $C_m(Q^j_{m,\mathcal{C}})$ is effective only when the number of CRs and ATTs are same for coalitions and $C_A(A^j_{\mathcal{C}})$ is only effective when a same number of CRs exists in the coalition.

- Un-satisfied constraints:

This situation happens when $Q^j_{f,\mathcal{C}} \geq \overline{Q_f}$ or $Q^j_{m,\mathcal{C}} \geq \overline{Q_m}$ or $A^j_{\mathcal{C}} \geq A_{max,\mathcal{C}}$. In such situation, either the PC is not available or the attack cost in terms of the number of ATTs exceeds a pre-defined threshold. Hence, forming coalitions or being in the coalitions is not worth for ATTs. As a result, coalition value should tend to infinity to ensure that such coalition will never form.

Based on the aforementioned features, the logarithmic barrier function can nicely suit our requirements. Hence, $C_f(Q^j_{f,\mathcal{C}})$, $C_m(Q^j_{m,\mathcal{C}})$ and $C_A(A^j_{\mathcal{C}})$ can be defined as

$$C_f(Q^j_{f,\mathcal{C}}) = \begin{cases} -\log(1-\frac{Q^j_{f,\mathcal{C}}}{\overline{Q_f}}), & Q^j_{f,\mathcal{C}} < \overline{Q_f} \\ +\infty, & \text{otherwise.} \end{cases} \tag{6.8}$$

$$C_m(Q^j_{m,\mathcal{C}}) = \begin{cases} -(\overline{Q_m})^{\beta}\log(1-(\frac{Q^j_{m,\mathcal{C}}}{\overline{Q_m}})^{\beta}), & Q^j_{m,\mathcal{C}} < \overline{Q_m} \\ +\infty, & \text{otherwise.} \end{cases} \tag{6.9}$$

$$C_A(A,\mathcal{C}^j) = \begin{cases} -(\frac{1}{A_{max,\mathcal{C}}})^{\hat{\beta}}\log(1-(\frac{A^j_{\mathcal{C}}}{A_{max,\mathcal{C}}})^{\hat{\beta}}), & A^j_{\mathcal{C}} < A_{max,\mathcal{C}} \\ +\infty, & \text{otherwise.} \end{cases} \tag{6.10}$$

where $\hat{\beta}$ and β are the coefficients that guarantee (6.6) and (6.7), and can be easily found.

Given coalition function, an MCG can be formed. CRs and ATTs over each selected PC play the game through the merge-and-split rule. After the termination of merge-and-split algorithm, a number of $\mathbb{D}_{hp}$-stable coalitions are formed, per Chapter 2. Since players prefer to get more *available channels*, only one coalition (with maximum coalition value) among all stable coalitions on each PC is selected.

6.2 Hierarchical ID-based key management as a lightweight solution

The major flaw in the regular multi-channel game is that attackers can collude and play a coalitional game among themselves to wisely participate in the regular game with other honest CRs. In addition, in a multi-channel system, single-channel key management cannot deal with attackers, as they can obtain one key and use it for as many channels as they will. In order to secure CRN against the proposed attack, access for participation in coalitional game should be limited to at most K PCs for each CR and no unauthorized CR should be able to eavesdrop the exchanged messages in a coalition. In addition, access for sensing a PC should be granted to those who have privilege among all coalitions on that PC. Thus, security management requires a different key for a CR on each PC and consists of two levels of access permissions (one level for game participation permission and the other for sensing permission).

IBC is a suitable tool for this purpose. The notion of IBC was proposed by Shamir [44] to solve certificate management issues in public key infrastructure and make public key cryptography easier. After that, Boneh and Franklin proposed the first secure and practical identity-based encryption (BF-IBE) scheme [49]. In IBC systems, an arbitrary string is chosen by the user as public key which enables other users to encrypt messages by knowing its identity which omits the need for certificate storage and exchange. However, the user should obtain the corresponding private key from a trusted party, called private key generator (PKG). PKG knows the private keys of all users; therefore, it can decrypt all exchanged messages in the network. This property, called key escrow, is not favorable in many practical scenarios, while in some others a central entity should have access to exchanged data, if necessary. In practice, IBC systems seem to be attractive especially in CRNs, where CRs have limited capabilities and resources, since they decrease the communication overhead, power and required memory.

In this section, a two-level ID-based key management scheme is proposed. In the first level, CRs are offered up to K access keys which means ATTs are forced to make their final decisions on requested PCs before being able to participate in the regular games, while the channel access is granted to the selected coalition on each PC, which avoids future attacks on sensing process, in the second level.

6.2.1 Game access key management

In traditional coalitional games, CRs can freely participate in any game on any PC. This opens back-doors for ATTs to participate in more than K PCs. Thus, in order to limit the effects from ATTs, PC access has to be limited. Specifically, authorized CRs are given private keys in order to access a PC, and a number of CRs are selected as DPKGs, which are responsible for CRs' private key distributions. CRs should use *game access keys* for message confidentiality and authentication throughout the game process until the final coalition is selected for a certain PC.

The proposed game access key management is shown in **Algorithm 6.2.1**. First, a number of DPKGs are selected from CRs in the same coverage area. DPKG selection can be based on different criteria such as the location of CRs and the remaining energy of CRs. Once DPKGs are selected, each CR$_i$ measures its SNR on PC$_j$, γ_i^j. If $\gamma_i^j > \gamma_{th}$, where γ_{th} is the SNR threshold, CR$_i$ sends game access request for PC$_j$ to $t+1$ DPKGs, where t denotes the security threshold and set to be $\frac{N}{2}$. This shows that CR$_i$ is going to participate in a game on PC$_j$. Since the sensing ability of each CR is limited to K PCs, CR$_i$ should be allowed to play in up to K different games. Thus, CR$_i$ sends its ID$_i$ along with $\Delta_i = \{\delta_i^j, j \in \{1, ..., M\}\}$ to $t+1$ DPKGs of its choice, where δ_i^j is the CR$_i$ game access request on PC$_j$ and Δ_i is the vector of CR$_i$ game access requests on PCs. In addition, CR$_i$ generates and sends a random number $\mathscr{N}_i$ as its nonce to avoid replay attacks. Upon receiving information from CR$_i$, each DPKG extracts ID$_i$ and checks whether $|\Delta_i| < K$, where $|\Delta_i|$ is the number of δ_i^j. If check fails, DPKG$_n$ broadcasts DisQ$_n^i$ to all CRs in the area to announce CR$_i$ disqualification. If $t+1$ DisQ$_n^i$ are received, all keys issued for CR$_i$ are revoked and ID$_i$ is added to the set of banned CRs, $\mathcal{S}_B$, so that until the next sensing

Algorithm 6.2.1 Game access key management

```
Initialization:
    • $2t + 1$ DPKGs are selected in the area which are responsible for partial game access key distribution of available PCs.
CR$_i$ sorts the available PCs based on $\gamma_i^j$ and forms the vector of game access requests, i.e. $\Delta_i$, containing its requests on sorted PCs, e.g. $\delta_i^j$ is the game access request on PC$_j$.
CR$_i$ generates a nonce $\mathscr{N}_i$.
CR$_i$ multicasts $< ID_i, \Delta_i >$ along with $\mathscr{N}_i$ signed by Sig$_i$ to $t{+}1$ DPKGs of its choice, i.e. $\mathcal{S}_{DPKG}^i$.
for $n \in \mathcal{S}_{DPKG}^i$ do
    if $|\Delta_i| < K$ then
        for $j \in \Delta_i$ do
            DPKG$_n$ computes $Q_{ID_i^j} = H_1(ID_i^j)$
            DPKG$_n$ computes $Q_d^{nij} = s_n Q_{ID_i^j}$
            DPKG$_n$ computes generates a nonce $\mathscr{N}_n$
            and lifetime of the partial game access key,
            i.e. $LifeTime_{Q_d^{nij}}$.
            DPKG$_n$$\longrightarrow$CR$_i$:
            $Enc_i(Q_d^{nij})$, $LifeTime_{Q_d^{nij}}$, $\mathscr{N}_i$, $\mathscr{N}_n$, Sig$_n$
            CR$_i$$\longrightarrow$DPKG$_n$: $\mathscr{N}_n$, Sig$_i$
        end for
    else
        DPKG$_n$ broadcasts DisQ$_n^i$, Sig$_n$ to all CRs in the
        area to announce CR$_i$ disqualification.
    end if
end for
if $|DisQ_n^i| = t + 1, n \in \mathcal{S}_{DPKG}^i$ then
    All $Q_d^{nij}$ issued for CR$_i$ are revoked and ID$_i$ $\rightarrow \mathcal{S}_B$
    so that until the next sensing cycle, no other keys
    will be issued for CR$_i$.
end if
```

cycle, no other keys will be issued for CR$_i$. If check passes, DPKG$_n$ uses its master key share, s_n, and generates a set of *partial game access keys*, defined as DPKG$_n$ share of *game access key*, for ID$_i$ on each requested PC. In this regard, DPKG$_n$ uses the combination of $ID_i^j =< ID_i, \delta_i^j >$ as a unique ID of CR$_i$ on PC$_j$ and computes $Q_{ID_i^j} = H_1(ID_i^j)$, where H_1 is a hash function. Then, the *partial game access key* is computed as $Q_d^{nij} = s_n Q_{ID_i^j}$. After that, DPKG$_n$ generates its nonce $\mathscr{N}_n$ and sends encrypted Q_d^{nij}, i.e. $Enc_i(Q_d^{nij})$,

$LifeTime_{Q_d^{nij}}$, $\mathscr{N}_i$ and $\mathscr{N}_n$ to CR$_i$, where $LifeTime_{Q_d^{nij}}$ is the life time of Q_d^{nij} equal to sensing cycle. $LifeTime_{Q_d^{nij}}$ indicates that Q_d^{nij} cannot be used in coalitional games played in the next sensing cycle.

Proposition 6.1. No ATT is able to collect more than K *game access keys*.

Proof. According to **Algorithm 6.2.1**, in order to get more than K *game access keys* ATTs need to figure out at least 2 sets of DPKGs. Let $\mathcal{S}_{DPKG}$ be the set of all DPKGs. Assume $\mathcal{S}_1 \subset \mathcal{S}_{DPKG}$ and $\mathcal{S}_2 \subset \mathcal{S}_{DPKG}$ are two sets of DPKGs. ATT$_i$ may send its request $|\Delta_i^1|$ and $|\Delta_i^2|$ for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Since more than K *game access keys* are needed, we have $|\Delta_i^1| + |\Delta_i^2| > K$. However, each DPKG set consists of at least $\frac{N}{2} + 1$ DPKGs, i.e $|\mathcal{S}_1| = |\mathcal{S}_2| \geq \frac{N}{2}+1$. Thus, if $\mathcal{S}_1 \bigcap \mathcal{S}_2 = \varnothing$, $|\mathcal{S}_1|+|\mathcal{S}_2| \geq N+2$, which exceeds the total number of CRs, i.e. N, which is contradiction. It implies that $\mathcal{S}_1 \bigcap \mathcal{S}_2 \neq \varnothing$ and at least one DPKG$_n$ exists such that $n \in \mathcal{S}_1 \bigcap \mathcal{S}_2$ which can inform all DPKGs in $\mathcal{S}_1 \bigcup \mathcal{S}_2$. Then, $t+1$ DPKGs in each set confirm their corresponding $|\Delta_i^1|$ and $|\Delta_i^2|$, which shows that CR$_i$ requested more than K PCs, leading to CR$_i$ disqualification. Therefore, ATT cannot submit different requests, Δ_i, to two individual sets and obtain more than K *game access keys*. ■

At last, CR$_i$ sends back $\mathscr{N}_n$ to DPKG$_n$ to ensure that it received data from it. Note that the sender in each step of the algorithm has to sign the message prior to transmission as part of authentication.

After collecting all *partial game access keys*, CR$_i$ combines them to extract the *game access key* on PCj, i.e. Q_d^{ij}. Here, we use threshold secret sharing where Q_d^{ij} can be constructed only when enough number of shares ($t + 1$) are combined together so that no individual DPKG can construct Q_d^{ij} on its own [50]. The desired *game access key* is $Q_d^{ij} = \sum_{j=1}^{t+1} \prod_{j=1, j\neq i}^{t+1} \frac{j}{j-i} Q_d^{nij}$.

6.2.2 Sensing key management

Once stable coalitions are formed, members of the best coalition on each PC are granted privilege to sense the corresponding PC. Therefore, the access of other coalitions should

Algorithm 6.2.2 Sensing key management

```
Initialization:
    • All CRs of the best coalition on PC$_j$, $\mathcal{C}^*$, are selected as higher level authorities, HLAs.
    • HLAs use the same system parameters, $G_1$, $G_2$ and $P$.
Each HLA$_n$ generates $s_{HLAn} \in \mathbb{F}_q^*$ and $P_{HLAn} \in G_1$.
HLA$_n \Longrightarrow$ CR$_i$, $i \in \mathcal{C}^*$: $P_{HLAn} \in G_1$, Sig$_n$
CR$_i$ generates a nonce $\mathscr{M}_i$
CR$_i \Longrightarrow$ HLA$_l$ $l \in \mathcal{C}_j^{*-i}$: $\mathscr{M}_i$, Sig$_i$
for $l \in \mathcal{C}_j^{*-i}$ do
    HLA$_l$ computes $Q_{ID_i^j} = H_1(ID_i^j)$
    HLA$_l$ computes $Q_c^{lij} = s_{HLAl} Q_{ID_i^j}$
    DPKG$_n \longrightarrow$CR$_i$:
    $Enc_i(Q_c^{lij})$, $LifeTime_{Q_c^{lij}}$, $\mathscr{M}_i$, $\mathscr{M}_l$, Sig$_l$
    CR$_i \longrightarrow$HLA$_l$: $\mathscr{M}_l$, Sig$_i$
end for
CR$_i$ computes $Q_c^{ij} = \sum_{l \in \mathcal{C}_j^{*-i}} Q_c^{lij}$
CR$_i$ computes $Q_s^{ij} = Q_d^{ij} + Q_c^{ij}$
```

be canceled and their *game access keys* should be revoked. As a result, only one coalition for each PC is granted the keys for sensing the PC, called *sensing keys*. In this section, we introduce the second phase of the ID-based key management, called sensing key management, as shown in **Algorithm 6.2.2**, which does not impose any key revocation. In sensing key management, members of the best coalition, called $\mathcal{C}^*$, are selected as the higher level authorities (HLAs), each of which is responsible for distributed key generation for other members. Note that $\mathcal{C}_j^*$ is the coalition with the maximum coalition value among the formed coalitions on PC j. Because of its importance, the concept of bilinear paring [65] is defined first.

In this thesis, HLAs use the same system parameters G_1, G_2 and P. Each HLA$_i$ generates a key pair (P_{HLAi}, s_{HLAi}), where $P_{HLAi} \in G_1$ and $s_{HLAi} \in \mathbb{F}_q^*$ are the public and private keys of HLA$_i$, respectively.

Since the coalitional game is played over one PC$_j$, each CR$_i \in \mathcal{C}_j^*$ has only one public key, ID$_i^j$. CR$_i$ needs to obtain its private keys, called *partial coalition keys*, from each

CR$_l$, $l \in \mathcal{C}_j^{*-i}$, where $\mathcal{C}_j^{*-i}$ defines a coalition of CRs on PC$_j$ excluding CR$_i$. Each HLA computes $Q_{ID_i^j} = H_1(ID_i^j)$. Then, the *partial coalition key* is computed as

$$Q_c^{lij} = s_{HLAl} Q_{ID_i^j} \tag{6.11}$$

After receiving all *partial coalition keys*, CR$_i$ can calculate its *coalition key* as

$$Q_c^{ij} = \sum_{l \in \mathcal{C}_j^{*-i}} Q_c^{lij} \tag{6.12}$$

Note that the *coalition key*, Q_c^{ij}, is unique for each CR$_i \in \mathcal{C}_j^*$. Thus, if one CR is compromised, *sensing keys* of other CRs in the coalition are safe from reconstruction.

The final *sensing key*, Q_s^{ij}, equals

$$Q_s^{ij} = Q_d^{ij} + Q_c^{ij} \tag{6.13}$$

Therefore, only CRs which have Q_s^{ij} can take part in sensing PC$_j$ and there is no need to revoke other CRs' *game access key* for sensing rights.

Proposition 6.2. The corresponding public key of the multi-level trust authority (combination of DPKGs and HLAs) used in encryption is $P_s^{i,j} = P_{DPKG} + P_{HLA}^i$, where P_{HLA}^{ij} is the public key of HLAs selected by CR$_i$ on PC$_j$, i.e. $P_{HLA}^{ij} = \sum_{l \in \mathcal{C}_j^{*-i}} P_{HLAl}$.

Proof. As an important part of decryption phase, CR$_i$ should compute $\hat{e}(P, Q_s^{ij})$. We can write

$$
\begin{aligned}
\hat{e}(P, Q_s^{ij}) &= \hat{e}(P, Q_d^{ij} + Q_c^{ij}) \\
&= \hat{e}(P, Q_d^{ij})\hat{e}(P, Q_c^{ij}), \\
&= \hat{e}(P, Q_d^{ij}) \prod_{l \in \mathcal{C}_j^{*-i}} \hat{e}(P, s_{HLAl} Q_{ID_i^j}), \\
&= \hat{e}(P, Q_{ID_i^j})^{s_{DPKG}} \prod_{l \in \mathcal{C}_j^{*-i}} \hat{e}(P, Q_{ID_i^j})^{s_{HLAl}}, \\
&= \hat{e}(s_{DPKG} P, Q_{ID_i^j}) \prod_{l \in \mathcal{C}_j^{*-i}} \hat{e}(s_{HLAl} P, Q_{ID_i^j}), \\
&= \hat{e}(P_{DPKG}, Q_{ID_i^j}) \prod_{l \in \mathcal{C}_j^{*-i}} \hat{e}(P_{HLAl}, Q_{ID_i^j}), \\
&= \hat{e}(P_{DPKG}, Q_{ID_i^j})\hat{e}(\sum_{l \in \mathcal{C}_j^{*-i}} P_{HLAl}, Q_{ID_i^j}), \\
&= \hat{e}(P_{DPKG} + \sum_{l \in \mathcal{C}_j^{*-i}} P_{HLAl}, Q_{ID_i^j}), \\
&= \hat{e}(P_{DPKG} + P_{HLA}^{ij}, Q_{ID_i^j}), \\
&= \hat{e}(P_s^{i,j}, Q_{ID_i^j}), \qquad (6.14)
\end{aligned}
$$

where $P_s^{i,j}$ is the public key of the multi-level trust authority. ■

Note that $\hat{e}(P_s^{i,j}, Q_{ID_i^j})$ should be applied in encryption phase. An outline of the proposed hierarchical ID-based key management is shown in Fig. 6.1. As figure depicts, there are two levels of security. In the first level, i.e. game access key management, DPKGs distribute private keys for CRs to let them play coalitional game on their requested PCs. The arrow shows the authority transition to the higher level, sensing key management, where CRs of the selected coalition on each PC serve as HLAs for other members of the same coalition. As CRs on this level have gained permission in the first and second level, they are the only CRs which can participate in sensing PCs.

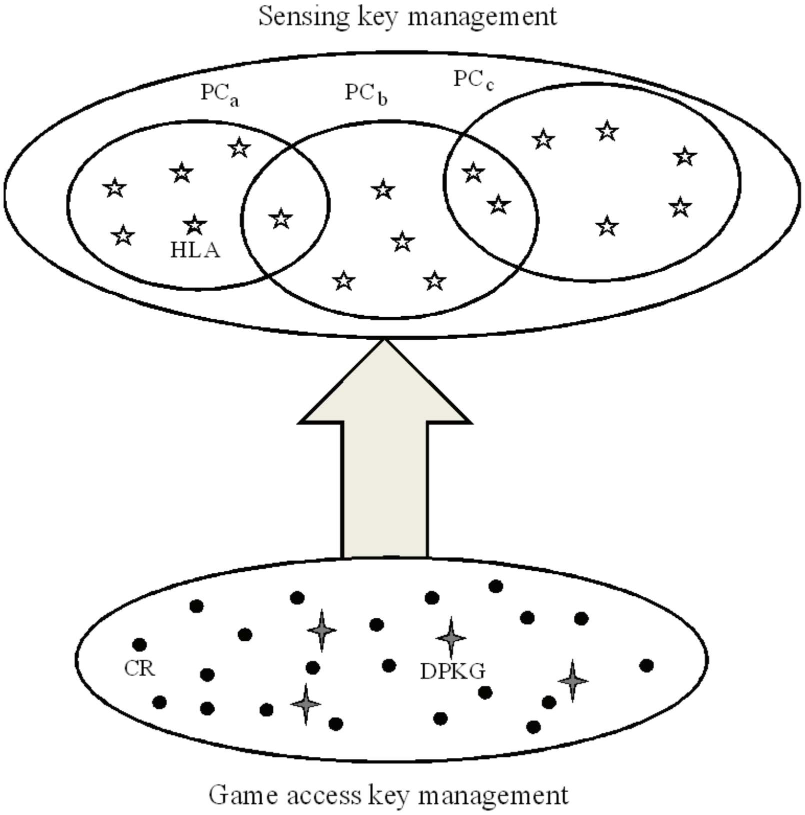


Figure 6.1: Outline of the proposed hierarchical ID-based key management.

6.3 Performance evaluation

In this section, simulation results are presented first. Then, some important properties of security protocols are defined and a formal verification of the proposed algorithms is provided by using Scyther tool [51].

6.3.1 Simulation results

Consider a multi-channel cognitive radio network with randomly deployed 100 PCs, i.e. 100 PUs, and total of 30 CRs including ATTs and honest CRs in a 2Km×2Km area. Each CR is equipped with a standard Nokia Nickel Metal Hydride (NiMH) battery with $C_{batt} = 650$mAh and $V_{batt} = 6$V. Other parameters are $P_s = 0.05\mu$W, $\zeta = 0.40$ and $\kappa = 0.02$W/Mbits. The obtained tuning coefficients for cost functions in (6.9) and (6.10), i.e. β and $\hat{\beta}$, are 2 and 1.7, respectively. In the simulation, the maximum number of ATTs on each PC, $A_{max,\mathcal{C}}$, and in the MCCRN, A_{max}, are set to the half of all CRs in each coalition, i.e., $\lceil \mathcal{C}_{max}^{Dist}/2 \rceil$, and the MCCRN, i.e. 15, respectively, so that the maximum number of ATTs in the network can become the same as the honest CRs in the MCCRN. The other simulation parameters are the same as in Chapter 2. For the purpose of comparison, our purposed MCG scheme in Chapter 2 is also simulated.

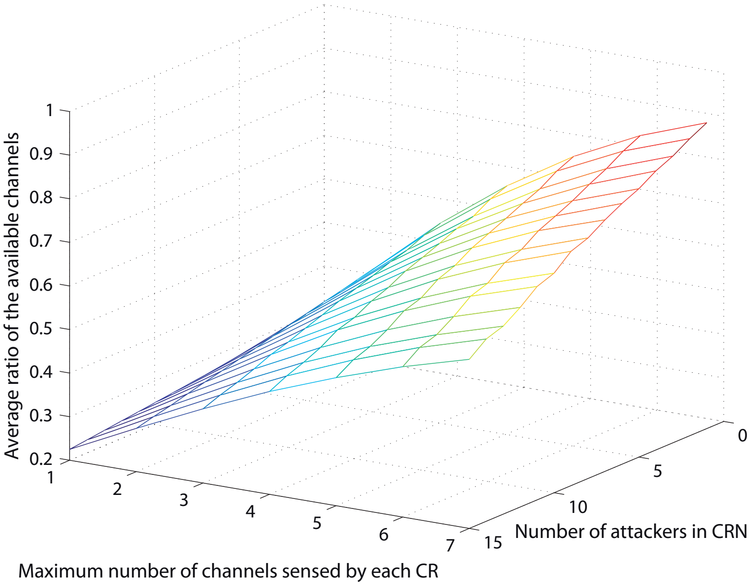


Figure 6.2: Average ratio of available channels against maximum number of attackers and sensible channels by a CR.

Fig.6.2 illustrates the effects of the proposed attack on the performance of the cooperative spectrum sensing in terms of the average ratio of the *available channels*, ϖ. Here, ϖ is defined as the ratio between the number of *available channels* with and without attack. Note that the PC becomes unavailable, if there is at least an ATT exists even though the false alarm and the miss detection probabilities are satisfied. The sensing capability of each CR, K, and the number of ATTs are set as simulation variables. From the figure, we can observe that the proposed attack can reduce ϖ up to 34%. Moreover, the slope of the ϖ decreases with the increase of K. It is because with the increase of sensing capacity of each CR, the distribution of ATTs over potential coalitions becomes wider compared to honest CRs.

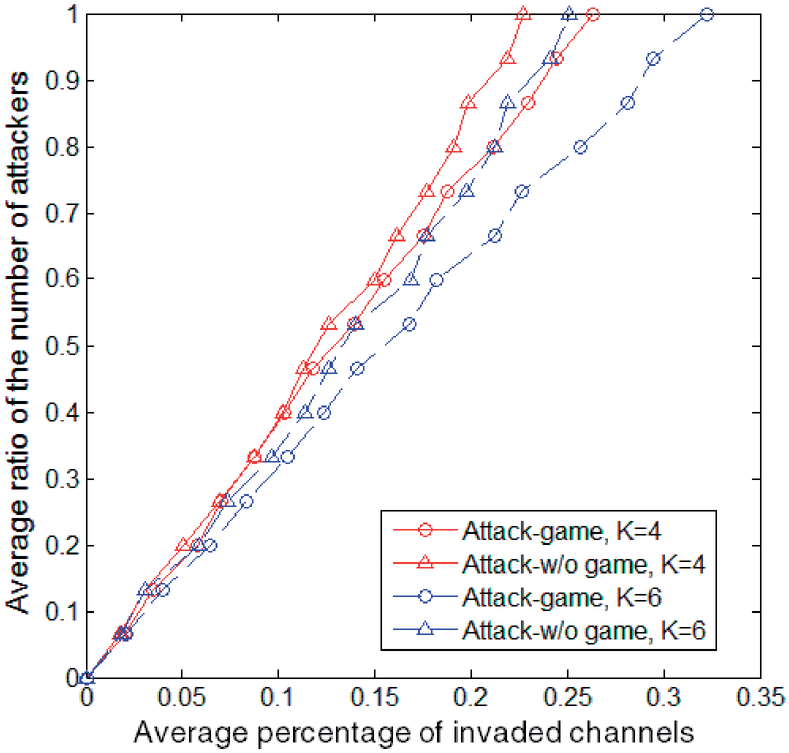


Figure 6.3: Ratio of the number of attackers against the average percentage of invaded channels for the multi-channel coalitional game attack.

To evaluate the attack cost, in this thesis, we introduce a performance metric, ϱ, which is defined as the ratio of the number of ATTs used in attack over A_{max} under the condition that a certain percentage of PCs is invaded. For comparison, two scenarios are considered. In the first scenario, ATTs use the proposed MCG-based attack while in the second scenario, ATTs follow the regular Byzantine attack. In the latter, attackers do not collude and play the first coalitional game (*Phase 1*). Thus, they play the second coalitional game with honest CRs as the regular game for channel assignment in MCCRN. Therefore, assigning channels to attackers follows the same procedure of a regular coalitional game which treats all CRs to be honest. The results are shown in Fig.6.3. It can be seen that the attack cost (ϱ) decreases when K increases. It is because ATTs' resources increase and more PCs can be invaded by one ATT. Moreover, when the MCG-based attack is applied, ϱ further decreases due to the wide distribution of ATTs over PCs. For example, to conquer 23% of the average number of available channels applying MCG-based attack results in 14% and 18% decrease

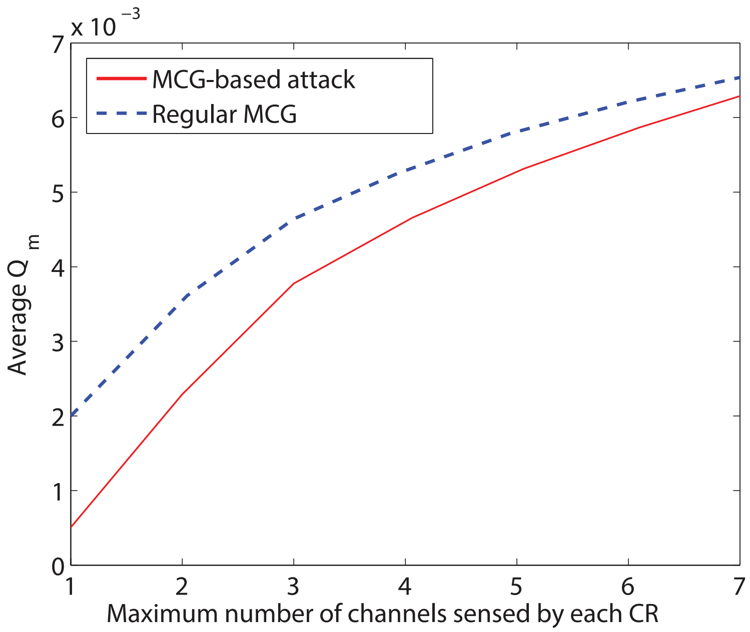


Figure 6.4: Average Q_m against maximum number of channels sensed by each CR.

on ϱ for $K = 4$ and 6, respectively, compared to the regular Byzantine attack.

We further investigate the effects of the proposed MCG-based attack on sensing performance compared to regular MCG (when all CRs are honest) in terms of the average $Q_m^{I,j}$ and $Q_f^{I,j}$ over all formed coalitions, called average Q_m and Q_f. The average Q_m vs. the maximum number of channels sensed by each CR is shown for 30 CRs in Fig. 6.4. In this figure, the average Q_m of the MCG-based attack is smaller than that of the regular MCG, and by increasing K, the curves of average Q_m from both schemes merge. It is because the coalitions with minimum number of CRs are preferred in the regular MCG, and by applying the *Phase 1* of the attack, more CRs are required to form the coalition on some channels in the absence of an attacker with better local sensing performance to ensure sensing constraints are satisfied, leading to coalitions with more CRs, i.e. smaller $Q_m^{I,j}$. By increasing K, there is a better chance that more CRs with better sensing performance are available and less number of CRs are needed to form coalitions, which result in the merge of two curves. Fig. 6.5 depicts the corresponding average Q_f vs. the maximum number of channels sensed by each CR. According to this figure, the proposed attack increases the average Q_f compared to that of the regular MCG. Similarly, both curves merge when K

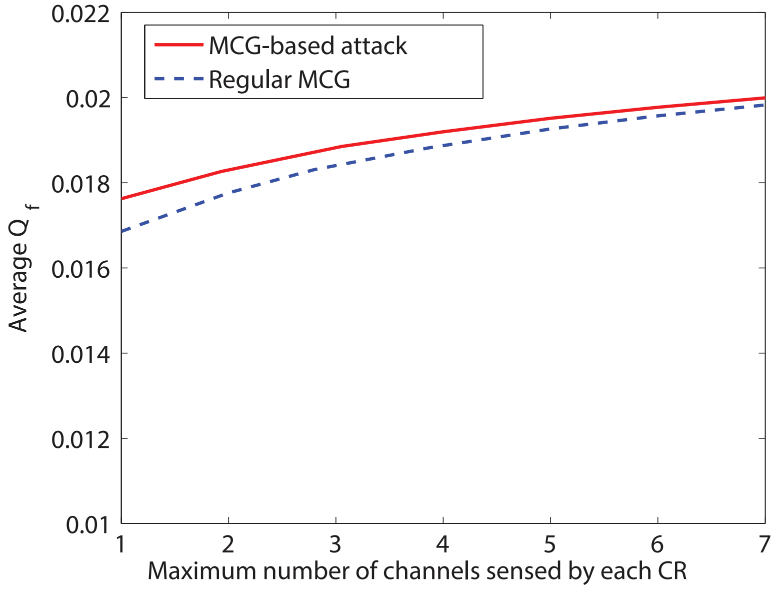


Figure 6.5: Average Q_f against maximum number of channels sensed by each CR.

increases. Moreover, average Q_f slightly increases with the increase of K, since coalitions formed with minimum number of CRs, i.e. minimum $Q_f^{I,j}$, while $Q_m^{I,j}$ only needs to satisfy $\overline{Q_m}$. Nevertheless, the difference between the MCG-based attack and the regular MCG is small for both average Q_m and Q_f. Note that in order for attackers to paly with honest CRs in *Phase 2*, they have to consider sensing constraints, i.e. $\overline{Q_m}$ and $\overline{Q_f}$ in *Phase 1*. Therefore, $\overline{Q_m}$ and $\overline{Q_f}$ are met by applying the MCG-based attack.

In order to study the energy efficiency of the proposed Hierarchical ID-based key management, called HIDKM, we compare it to a multi-channel certificate-based key management, called MCKM. In MCKM, CRs obtain certificates for their public keys from distributed certificate authorities, i.e. DPKG counterparts. To do that, CR i first multicasts public key to $t+1$ certificate authorities to obtain partial certificates and reconstruct the general certificate, $Cert_i$, based on threshold secret shairing. Then, CR i generates public key for each PC that it wants to play on, denoted as $P_{ci}^j =< r_{ci}, \delta_i^j >$, where r_{ci} is a real number. After that, CR i multicasts $P_{ci} = \{P_{ci}^j\}$ to certificate authorities. If $|P_{ci}| < K$, each certificate authority n issues partial channel certificate on PC j, $Cert_{ci}^{jn}$; otherwise, if $t+1$ certificate authorities broadcasts $DisQ_n^i$, all partial channel certificates are revoked.

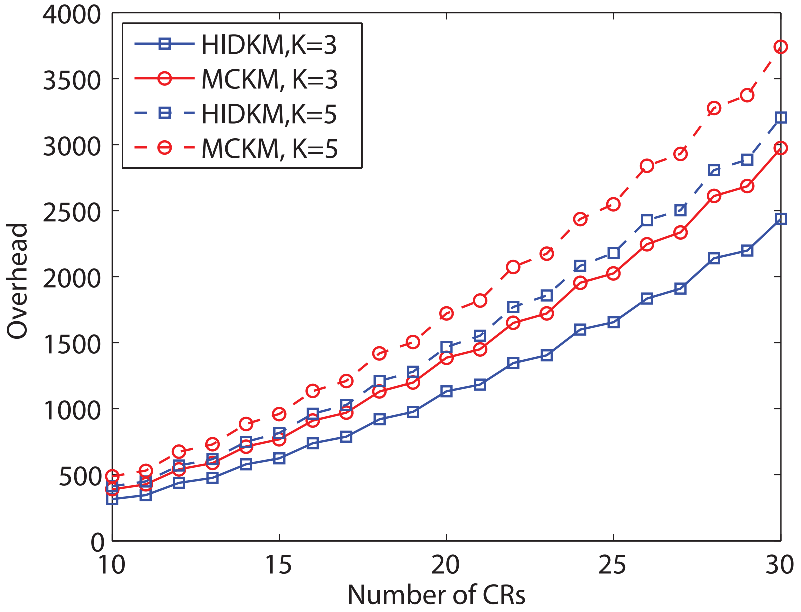


Figure 6.6: Overhead against the number of CRs.

The simulation parameters are as follows. Nonces and LifeTime sizes are 10 bits. ECC key size is 224 bits, per NIST recommendations [70] for 2011-2030. ECDSA-based certificate size is 673 bits excluding overhead of the ID$_i$ (assumed to be 50 bits). We adopt the identity-based signature scheme in [71] with 320-bit signatures.

Fig. 6.6 shows the communication overhead in terms of the number of transferred messages, vs. N and $K = \{3, 5\}$, for HIDKM and MCKM. According to the figure, HIDKM outperforms MCKM, and difference increases with the increase of N. The reason is that, in MCKM, CRs have to obtain general and channel certificates from certificate authorities. However, in HIDKM, CRs only need to get the channel keys. Furthermore, increasing K leads to the increase of overhead which is more considerable in MCKM. The average power consumption for each CR is illustrated in Fig. 6.7. As shown in the figure, HIDKM is considerably more energy-efficient than MCKM. In addition, by increasing K, the CR power consumption increases at a faster pace in MCKM up to the point that the difference between two curves associated with $K = 3$ and $K = 5$ for 30 CRs in MCKM is 44$\%$ more than that of HIDKM. It is due to the fact that, although it adopts ECDSA-based certificates (which are much lighter than RSA-based one), MCKM still suffers from certificate size and

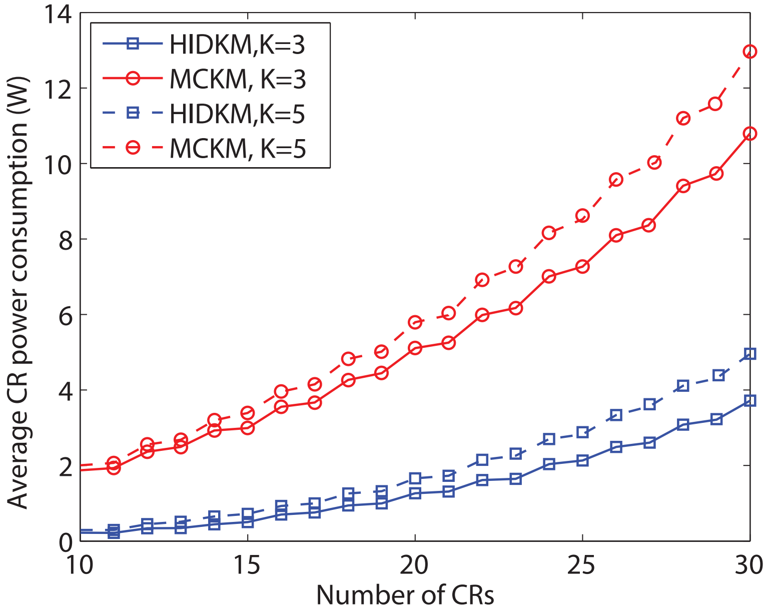


Figure 6.7: Average CR power consumption against the number of CRs.

excessive certificate exchange burden, which is critical for multi-channel key management. Finally, Fig. 6.8 shows the average CR battery life, in hours, vs. N. It can be observed that HIDKM can significantly save the battery life over the span of N, even for the extreme number of CRs, i.e. 30 CRs. For example, CR battery life in HIDKM is 67% and 62% more than that of MCKM for $K = 3$ and 5, respectively.

6.3.2 Formal security analysis

Now, we analyze the formal verification of the proposed key management protocols by using Scyther tool [51] to investigate all possible interactions between an attacker and the proposed protocols and extract any hole or threat that may exist. Scyther is an effective automated tool for verification falsification and analysis of security protocols. While guaranteeing termination, Scyther is able to verify security protocols with unbounded number of runs by generating counter examples to claimed properties. The performance efficiency of Scyther is a cutting-edge tool for security protocol analysis as its run time is almost constant compared to other tools where their run time increases exponentially with the increase

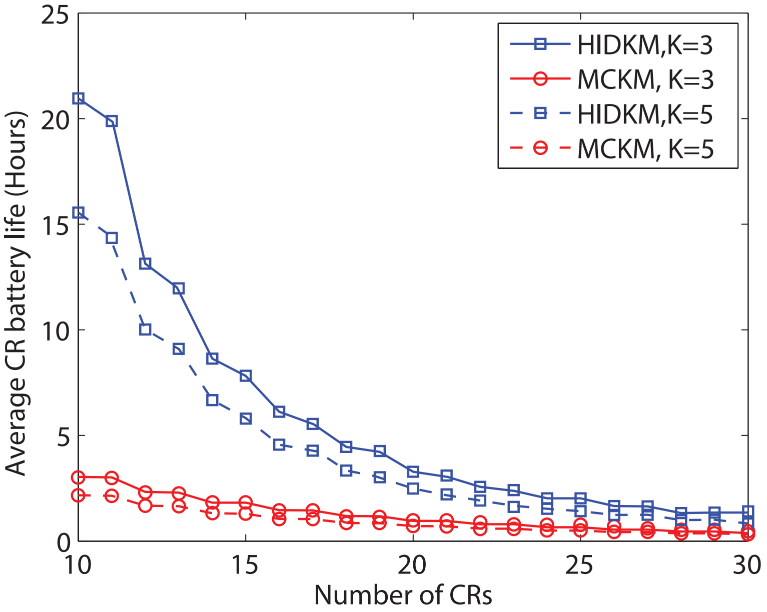


Figure 6.8: Average CR battery life against the number of CRs.

of the number of runs [80]. Scyther applies backward symbolic state search technique for analysis of the security protocols, where attacks can be found by backward search from the claim that is broken. By using it, infinite state spaces can be explored. In addition, Scyther can verify authentication properties including synchronization. Note that the description of the security protocol is written in SPDL, i.e. security protocol description language.

The following fundamental principles are considered for formal analysis of the proposed protocols. Note that in all categories unbounded number of protocol runs are taken into account to ensure the effectiveness of the proposed protocol for any given number of sessions.

Pseudonymity

It means that no third party, by eavesdropping the transferred messages, can relate traffic to a specific user, i.e. CR$_i$. It is an important property especially for **Algorithm 6.2.1**, as the ID$_i$ and the requested PCs of CR$_i$ should be kept secret so that ATTs cannot obtain any information about other CRs' favorable PCs. This claim is satisfied when $< ID_i, \Delta_i >$ is secret. Scyther verifies no attack is possible. It is because $< ID_i, \Delta_i >$ and ID_i^j are encrypted so that only selected DPKGs and HLAs can decrypt them in **Algorithm 6.2.1** and **Algorithm 6.2.2**, respectively.

Confidentiality

The disclosure of data sent by CR$_i$ to unauthorized CRs should be prevented. The confidentiality is proven by Scyther as the exchanged data is encrypted by CR$_i$ and no eavesdropper can get it disclosed.

Integrity

Integrity ensures that data, specifically generated keys, cannot be modified without being detected. Therefore, both sides, CRs and DKPGs or HLAs should make sure that the generated keys for CR$_i$ are untempered. Scyther validates the integrity of the generated keys (Q_d^{nij} and Q_c^{lij} in **Algorithm 6.2.1** and **Algorithm 6.2.2**, respectively) at both sides.

Key freshness

The key is called fresh, if it is guaranteed to be new, instead of being an old one either because of the actions of ATTs or DPKGs or HLAs. It is proven by Scyther that the generated Q_d^{nij} and Q_c^{lij} are fresh from the CR$_i$'s viewpoint.

Authorized access

DPKGs and HLAs should make sure that only authorized CRs gain access to services that they offer. In addition, no unauthenticated CRs can impersonate another one to use offered services. This claim is satisfied, if the service is bound to the authorized CRs, i.e. DPKGs and HLAs ensures that the sent data from them remains secret. Scyther confirms that ATT cannot get knowledge of data sent by either DPKGs or HLAs.

Authentication

The focus of authentication is that the existence of a communication partner should be guaranteed when a protocol role is executed. In formal verification, three properties can be considered for authentication which are aliveness, agreement and synchronization. Aliveness is the property when CR$_i$ plays a role in the protocol and assumes that it is communicating with a trusted party (e.g. DPKG or HLA) that has executed an event. Agreement states that after a successful completion of the protocol, the two sides agree on the values of variables. Synchronization guarantees that the message transmission is exactly occurred as stated in the protocol. It is to say that if CR$_i$ and DPKG$_n$ (or HLA$_n$) complete a run of the protocol, all messages should be received in the same order that the protocol describes. Synchronization is a strictly stronger property than agreement [81], since in synchronization, corresponding send and receive messages must be executed in the described order; however in agreement, a message may be received before it is sent which results in preplay attack where an ATT injects a message while it is yet to be sent. Scyther verifies aliveness and synchronization (as a result agreement) from both sides' perspectives. It is because nonces are generated and used by both sides, and are kept secret so that no ATT can obtain them, initiate messages by replacing its nonce and make the synchronization fail.

Chapter 7

Conclusions and Future Work

In this research, cooperative spectrum sensing and its related security issues in multi-channel cognitive radio networks have been studied.

First, cooperative multi-channel spectrum sensing in cognitive radio networks has been discussed. The optimal channel assignment for sensing has been formulated as a nonlinear integer programming problem and its performance upper bound has been derived by approximation. New centralized and distributed cooperative multi-channel spectrum sensing schemes have been proposed. The proposed centralized schemes consist of two implementation options with different signalling overheads, while the proposed distributed scheme solves the multiple channel sensing problem based on coalitional game theory. The simulation results have demonstrated that the proposed schemes improve the number of *available channels* significantly compared to the counterparts, and can reach near optimal performance.

The physical layer security threats of the proposed schemes have been studied by introducing two new Byzantine attacks, i.e., coalition head and multi-channel Byzantine attacks. By considering statistical properties of the exchanged data in each coalition, the probability of attack for the coalition head and cognitive radios were defined and two new counterattacks have been proposed for both attacks, respectively. Simulation results have shown that the proposed counterattacks can successfully block attackers from coalitions and can increase the number of available channels in the presence of attackers.

In addition to physical layer security, management of cryptographic keys were considered as important parts of the system design to bring authenticity, confidentiality and data integrity to cognitive radio networks. Then, an energy-efficient identity-based and a certificate-based key management schemes have been proposed for multi-channel cognitive radio networks. A new stochastic security threshold has been introduced, and new security-based and energy-based DPKG selection algorithms have been proposed to prioritize security and balance the energy amongst CRs, respectively. In the proposed key management scheme, CRs may request channel key from the DPKGs and construct it after collecting the shares. For certificate-based applications, a distributed certificate-based key management scheme has been proposed, where certificate authorities are distributed based on threshold cryptography. Simulation results have shown that i) the proposed energy-based DPKG selection can significantly reduce the overhead, balance the energy level of CRs and increase the lifetime of MCCRN, ii) the proposed identity-based scheme is more energy efficient than the certificate-based schemes and significantly decreases the traffic overhead, iii) among certificate-based schemes, our ECC-CBCKM outperforms the other ones.

Because of its importance, we have proposed a new energy-efficient DPKG assignment which can capture DPKG fairness. The DPKG assignment was incorporated into a new interactive algorithm to solve the multi-objective problem. The optimal solution and upper bounds have been studied for DPKG assignment and new DPKG assignment algorithm has been proposed for low-complexity implementation. Simulation results have shown that the proposed DPKG assignment algorithm can achieve a near optimal performance in power consumption and improve computational performance compared to the optimal solution.

Finally, a more intelligent Byzantine attack, where attackers collude to achieve their goal, was considered, and a new distributed cooperative attack on multi-channel cooperative spectrum sensing has been proposed. The proposed attack consists of two phases. First, attackers play a game with themselves following the similar rules applied to other

CRs. Then, they play with other CRs only on the channels that can maximize their own objective. To make coalitional game and spectrum sensing secure against these attacks, a hierarchical ID-based key management scheme has been proposed. Simulation results have shown that the proposed attack can greatly decrease the number of available channels with low attack cost, and the proposed key management scheme can significantly improve energy efficiency. Furthermore, formal analysis verified that the proposed key management scheme is immune against possible attacks.

In future, the following issues and extensions of this work will be considered.

- Traditional cryptographic techniques are important to bring security for wireless networks. However, such mechanisms do not directly consider the unique properties of the wireless medium to alleviate some security threats, e.g. Byzantine attacks. The physical characteristics of the wireless medium can provide some good domain-specific information to help improve security mechanisms. The unique threats that CRNs face in spectrum sensing and sharing, e.g. spectrum sensing data falsification, can also be mitigated by considering specific physical layer properties such as correlation of the wireless links. Therefore, new physical layer security mechanisms can be crucial for cross-layer security design in CRNs.

- Estimation of probability of compromise should be studied for more precise attack models. For example, not only position of the attackers, but duration of attack and attackers' resource constraints should be taken into consideration for practical applications. In addition, mobility and node heterogeneity can be added to other important variables, especially in tactical cognitive radio networks. The use of accurate mobility models that can capture the behavior of different nodes with different mobility patterns, e.g. individual and grouped move of human-mounted node, ground and aerial vehicles, is crucial. Therefore, mobility and topological features (node distribution on different surface planes) should be considered for the estimation of the probability of compromise.

- By considering different critical parameters, such as probability of compromise and energy consumption, distributed multi-objective optimization of DPKG selection can be taken into account for MCCRNs. Coalitional game theory can be considered as a good solution, where CRs can form coalitions to achieve the common objectives, such as energy efficiency, in the network. Another solution can be distributed evolutionary algorithm, e.g. genetic-based algorithm, which is one of the most influential tools for solving multi-objective programming based on natural evolution.

- Designing more robust, energy efficient and lightweight access control and authentication mechanisms can be considered in MCCRNs. An interesting choice is Zero-Knowledge Protocols (ZKPs). ZKP can provide services such as authentication, without unveiling users' IDs and with less computational overhead than other public key protocols, which make it suitable for agile and complex networks such as MCCRNs. In ZKP, the verifier cannot obtain any knowledge from the protocol, and prover and verifier cannot deceive each other. By employing parings and ECC in ZKP, the security strength of the system can be improved. In addition, pairing-based ZKP can be nicely applied to the paring-based schemes proposed in this thesis, leading to a good integrated and more efficient system. Therefore, a lightweight paring-based ZKP can be designed for MCCRNs.

- The security in IEEE 802.22, as a prime application of CRNs, can be considered. Most of the security concerns in IEEE 802.22 result from the absence of enough protection for inter-cell beacons against forgery and illegal modifications in the security sub-layer of the standard. Thus, integrity and authenticity of beacons should be taken care of, which requires a proper key management scheme. A centralized key management scheme can be employed by use of backhaul infrastructure. However, since different cells may belong to different service providers, a common backhaul among all service providers in the area may not exist, which makes this approach less feasible. A distributed key management mechanism may be a better solution.

The proposed paring-based key management schemes can be nicely adopted in this scenario, which will help design a distributed lightweight key management scheme.

References

[1] S. Haykin, “Cognitive radio: brain-empowered wireless communications”, *IEEE Journal on Select. Areas Commun.*, vol. 23, no. 2, pp. 201220, Feb. 2005.

[2] Federal Communications Commission, “Spectrum policy task force report, FCC 02–155”, Nov. 2002.

[3] J. Mitola, G.Q. Maguire, “Cognitive radio: making software radios more personal”, *IEEE Personal Commun.*, vol. 6, no. 4, pp. 13-18, 1999.

[4] K. Baclawski, D. Brady, M. Kokar, “Achieving dynamic interoperability of communication at the data link layer through ontology based reasoning”, *SDR Forum Technical Conf.*, 2005.

[5] IEEE 1900.1 Draft Document, Standard Definitions and Concepts for Spectrum Management and Advanced Radio System Technologies, 2006.

[6] J. Neel, J. Reed, A. MacKenzie, “Cognitive radio network performance analysis”, Cognitive Radio Technology, pp. 501-580, 2006.

[7] T.W. Rondeau, C.J. Rieser, B. Le, C.W. Bostian, “Cognitive radios with genetic algorithms: intelligent control of software defined radios”, *SDR Forum Technical Conf.*, Phoenix, AZ, pp. C-3C-8, 2004.

[8] L. Berlemann, S. Mangold, B. Walke, “Policy-based reasoning for spectrum sharing in cognitive radio networks”, *IEEE int’l symposium on new frontiers in dynamic spectrum access* (DySPAN’05), 2005.

[9] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, “Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey”, *Computer Networks*, vol. 13, pp. 2127-2159, 2006.

[10] M.T. Zhou, H. Harada, “Cognitive maritime wireless mesh/ad hoc networks”, *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 518-26, 2012.

[11] I.F. Akyildiz, W.Y. Lee, K.R. Chowdhury, “CRAHNs: cognitive radio ad hoc networksAd Hoc Networks”, Journal Ad Hoc Networks, vol. 7, no. 5, pp. 810936, 2009.

[12] J. Mitola, Cognitive radio: an integrated agent architecture for software defined radio, Ph.D. Thesis, KTH, Stockholm, 2000.

[13] R.V. Prasad, P. Pawelczak, J.A. Hoffmeyer, H.S. Berger, “Cognitive functionality in next generation wireless networks: standardization efforts”, *IEEE Commun. Mag.*, vol. 46, no. 4, Apr. 2008.

[14] J. Guenin, “IEEE standards coordinating committee 41 on dynamic spectrum access networks: activities, technical issues, and results”, *ITU-R WP5A SDR/CR Seminar*, Feb. 2008.

[15] FCC Notice of Proposed Rule Making FCC 04-113, May 25, 2004.

[16] http://www.ieee802.org/22/

[17] D. Grandblaise, W. Hu, “Inter base stations adaptive on demand channel contention for IEEE 802.22 WRAN self coexistence”, IEEE documents, IEEE 802.22-07/0024r0, 2007.

[18] H. Arslan, S. Ahmed, “Applications of cognitive radio”, Cognitive radio, software defined radio, and adaptive wireless systems, Springer, 2007.

[19] D. Maldonado, B. Le, A. Hugine, T.W. Rondeau, CW. Bostian, “Cognitive radio applications to dynamic spectrum allocation”, *IEEE int’l symposium on new frontiers in dynamic spectrum access* (DySPAN’05), Baltimore, MD, USA, pp. 597-600, Nov. 2005.

[20] W. Zhang and K. Letaief, “Cooperative spectrum sensing with transmit and relay diversity in cognitive networks”, *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4761–4766, Dec. 2008.

[21] A. Ghasemi and E. S. Sousa, “Collaborative spectrum sensing for opportunistic access in fading environments”, *IEEE int’l symposium on new frontiers in dynamic spectrum access* (DySPAN’05), Baltimore, Maryland, USA, pp. 131-136, Nov. 2005.

[22] E. Visotsky, S. Kuffner, and R. Peterson, “On collaborative detection of TV transmissions in support of dynamic spectrum sensing”, *IEEE int’l symposium on new frontiers in dynamic spectrum access* (DySPAN’05), Baltimore, MD, USA, pp. 338-345, Nov. 2005.

[23] J. Shen, S. Liu, L. Zeng, G. Xie, J. Gao and Y. Liu, “Optimisation of cooperative spectrum sensing in cognitive radio network”, *IET Commun.*, vol. 3, no. 7, pp. 1170–1178, 2009.

[24] Q. Zhao, L, Tong, A. Swami and Y. Chen, “Decentralized cognitive MAC for opportunistic spectrum access in Ad Hoc networks: a POMDP framework”, *IEEE Journal on Select. Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.

[25] W. Wang, J. Cai, and A. Alfa, “Receiver-aided spectrum sensing scheme with spatial differentiation in OFDM based cognitive radio networks”, *IEEE INFOCOM’10 Workshop on Cognitive Wireless Communications and Networking*, San Diego, CA, USA, Mar. 2010.

[26] R. Fan and H. Jiang, “Optimal multi-channel cooperative sensing in cognitive radio networks, *IEEE Trans. on Wireless Communun.*, vol. 9, no. 3, pp. 1128–1138, Mar. 2010.

[27] P. Kaligineedi and V. K. Bhargava, “Sensor allocation and quantization schemes for multiband cognitive radio cooperative sensing system”, *IEEE Trans. on Wireless Communun.*, vol. 10, no. 1, pp. 284–293, Jan. 2011.

[28] H. Sun, A. Nallanathan, C. Wang and Y. Chen, “Wideband spectrum sensing for cognitive radio networks: a survey”, *IEEE Wireless Commun.*, vol. 20, no. 2, 2013.

[29] Z. Quan, S. Cui, H. Poor, and A. Sayed, “Collaborative wideband sensing for cognitive radios”, *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 60–73, 2008.

[30] H. Sun, A. Nallanathan, J. Jiang, D. Laureson, C. Wang, and H. Poor, “A novel wideband spectrum sensing system for distributed cognitive radio networks”, *IEEE Global Telecommunications Conf.* (Globecom’11), Houston, TX, USA, pp.1–6, Dec. 2011.

[31] Z. Tian and G. Giannakis, “Compressive sensing for wideband cognitive radios”, *IEEE Int’l Conf. on Acoustics, Speech, and Signal Processing* (ICASSP’07), Honolulu, HI, USA, pp. 1357–1360, Apr. 2007.

[32] J.L. Burbank, “Security in cognitive radio networks: the required evaluation in approaches to wireless network security”, *Int’l Conf. on CrownCom.*, Singapore, pp. 1-7, 2008.

[33] R. Chen, J.M. Park, Y.T. Hou, J.H. Reed, “Toward secure distributed spectrum sensing in cognitive radio networks”, IEEE Commun. Magazine, vol. 50, no. 5, 2008.

[34] X. Zhang, C. Li, “The security in cognitive radio networks: a survey”, *Int’l Conf. on wireless commun and mobile computing* (WCMC’09), Leipzig, Germany, pp. 309-13, 2009.

[35] T. Qin, H. Yu, C. Leung, Z. Shen, C. Miao, “Towards a trust aware cognitive radio architecture”, Newsletter ACM SIGMOBILE Mobile Computing and Commun. Review, vol. 13, no. 2, pp. 86-95, 2009.

[36] R. Chen, J. M. Park and K. Bian, “Robust distributed spectrum sensing in cognitive radio networks”, *IEEE Int’l Conf. on Computer Commun.* (INFOCOM’08), Phoenix, AZ, pp. 1876–1884, April 2008.

[37] A.S. Rawat, P. Anand, C. Hao, P.K. Varshney, ”Countering byzantine attacks in cognitive radio networks,” *2010 IEEE Int’l Conf. on Acoustics Speech and Signal Processing* (ICASSP’10), pp.3098-3101, Mar. 2010.

[38] A.S. Rawat, P. Anand, C. Hao, P.K. Varshney, ”Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks,” *IEEE Trans on Signal Processing*, vol.59, no.2, pp.774–786, Feb. 2011.

[39] X. He, H. Dai, P. Ning, “A byzantine attack defender: the conditional frequency check”, *IEEE Int’l Symposium on Information Theory* (ISIT), pp.975–979, Boston, MA, 1–6 Jul. 2012.

[40] A.M. Hegland, E. Winjum, S.F. Mjolsnes, C. Rong, O. Kure, P. Spilling, “A survey of key management in ad hoc networks,” *IEEE Commun. Surveys & Tutorials*, vol.8, no.3, pp.48–66, 3rd Qtr. 2006.

[41] S.A. Vanstone, “Next generation security for wireless: elliptic curve cryptography,” *Elsevier Computers and Security*, Vol. 22, No. 5, pp. 412–415, Jul. 2003, .

[42] A. Menezes, P.C. van Oorschot; S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Oct. 1996.

[43] V. Katiyar, K. Dutta, S. Gupta,“A survey on elliptic curve cryptography for pervasive computing environment,” *Int’l Journal of Computer Applications*, vol.11, no. 10, pp. 41–46, Dec. 2010.

[44] A. Shamir, “Identity-based cryptosystems and signature schemes,” *Advances in Cryptology, Springer Verlag Lecture Notes in Computer Science*, (CRYPTO’85), vol. 196, pp. 47–53, 1985.

[45] C. Cocks, “An identity-based encryption scheme based on quadratic residues,” *Cryptography and Coding, Springer Verlag Lecture Notes in Computer Science*, vol. 2260, pp. 360-363, 2001.

[46] V. Goyal, “Reducing trust in the PKG in identity based cryptosystems,” *Springer Verlog Advances in Cryptology*, pp. 430-447, 2007.

[47] L. Zhou and Z. Haas, “Securing ad hoc networks,” *IEEE Network Magazine*, vol. 13, no. 6, pp. 24-30, Dec. 1999.

[48] R. Gangishetti, M. Choudary Gorantla, M. D. Das, and A. Saxena, “Threshold key issuing in identity-based cryptosystems,” *Elsevier Computer Standards & Interfaces*, vol. 29, no. 2, pp. 260-264, 2007.

[49] D. Boneh, M. Franklin, “Identity-based encryption from the weil pairing,” *Advances in Cryptology, Springer Verlag Lecture Notes in Computer Science*, (CRYPTO’01), vol. 2139, pp. 213229, Aug. 2001.

[50] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, Nov. 1979.

[51] C. Cremers, “The Scyther tool: Automatic verification of security protocols”, http://people.inf.ethz.ch/cremersc/scyther/index.html, 2009.

[52] A. Ghasemi and E. Sousa, Opportunistic spectrum access in fading channels through collaborative sensing, *Journal of Commun.*, vol. 2, no. 2, pp. 71–82, 2007.

[53] S. Ahmed, M. S. Hossain, M. Abdullah, and M. A. Hossain, “Cooperative spectrum sensing over Rayleigh fading channel in cognitive radio”, *Int’l Journal of Electronics and Computer Science Eng.*, vol. 1, no. 4, pp. 2583–2592, 2012.

[54] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, “Coalitional games for distributed collaborative spectrum sensing in cognitive radio networks”, *IEEE Int’l Conf. on Computer Commun.* (INFOCOM09), Rio de Janeiro, Brazil, Jun. 2009.

[55] K. Apt and A. Witzel, “A generic approach to coalition formation” , *Int’l Workshop on Computational Social Choice* (COMSOC’06), Amsterdam, the Netherlands, Dec. 2006.

[56] R. Karp, ”Reducibility among combinatorial problems”, *50 Years of Integer Programming 1958-2008: From the Early Years and State-of-the-Art*, Springer-Verlag, 2010.

[57] M. Garey and D. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W. H. Freeman, 1979.

[58] R. Hemmecke, M. Koppe, J. Lee, R. Weismantel, “Nonlinear integer programming”, *50 Years of Integer Programming 1958-2008: From the Early Years and State-of-the-Art*, Springer-Verlag, 2010.

[59] I. Litvinchev, S. Rangel, and J. Saucedo, “A Lagrangian bound for many-to-many assignment problems”, *Journal of Comb. Optim.*, vol. 19, no. 3, pp. 241-257, 2010.

[60] R. Myerson, *Game theory, analysis and conflict*, Harvard University Press, Cambridge, MA, USA, Sep. 1991.

[61] J. Luo, D. Ye, L. Xue, and M. Fan, “A survey of multicast rouitng protocols for mobile ad-hoc networks”, *IEEE Commun. Survey & Tutorials*, vol. 11, no. 1, pp. 78–91, First Quarter, 2009.

[62] http://www.ampl.com/downloads/index.html.

[63] E. Peh and Y. Liang, "Optimization for cooperative sensing in cognitive radio networks", *IEEE Wireless Commun. and Networking Conf.* (WCNC'07), Hongkong, China, Mar. 11–15, 2007.

[64] R.M. Axelrod, " The evolution of cooperation", Basic books, Inc., Publishers, New York, 1984.

[65] S.D. Galbraith, K.G. Paterson, N.P. Smart, "Pairings for Cryptographers," IACR Cryptology ePrint Archive, May 2006.

[66] A. Joux and K. Nguyen, "Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups," *Journal of Cryptology*, vol. 16, no. 4, pp. 23947, 2003.

[67] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure distributed key generation for discrete-log based cryptosystems", *Advances in Cryptology*, EUROCRYPT99, pp. 295310, 1999.

[68] C. Isheden and G. P. Fettweis, Energy-efficient multi-carrier link adaptation with sum rate-dependent circuit power, *IEEE Global Telecomm. Conf.* (GLOBECOM'10), pp.1-6, Miami, FL, USA, Dec. 2010.

[69] M. Bloem, T. Alpcan, and T. Basar, "A stackelberg game for power control and channel allocation in cognitive radio networks, I*nt'l ICST Workshop on Game Theory in Commun. Networks* (GameComm'07), France, Oct. 2007.

[70] Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 3, NIST, 05/2011.

[71] J. C. Cha and J. H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," *Springer-Verlag Lecture Notes in Computer Science*, vol.2567, pp.1830, 2003.

[72] S. Vasudevan, B. DeCleene, N. Immerman, J. Kurose, D. Towsley, "Leader election algorithms for wireless ad hoc networks," *DARPA Info. Survivability Conf. and Exposition*, pp.261–272, Washington D.C., Apr. 2003.

[73] N. Mohammed, H. Otrok, W. Lingyu, M. Debbabi, P. Bhattacharya, "Mechanism design-based secure leader election model for intrusion detection in MANET," *IEEE Trans on Dependable and Secure Computing*, vol.8, no.1, pp.89-103, Jan.-Feb. 2011.

[74] J. Horn, "Multicriteria decision making", Handbook of evulutionary computation, Institute of Physics Publishing, 1997.

[75] K. Miettinen, Nonlinear multi-objective optimization, Springer, 1999.

[76] Y. Crama, P. Hansen, B. Jaumard, " The basic algorithm for pseudo-Boolean programming revisited", *Discr. Appl. Math.*, no. 29, pp. 171185, 1990.

[77] P.L. Hammer, S. Rudeanu, Boolean methods in operations research and related areas, Springer, Berlin, 1968.

[78] F. Glover, R.E.D. Woolsey, "Further reduction of zero-one polynomial programs to zero-one linear programming problems", Oper. Res. no. 21, pp. 156161, 1973.

[79] M. Held, P. Wolfe, H.D. Crowder, “ Validation of subgradient optimization”, Mathematical Programming, vol. 6, pp. 62-68, 1974.

[80] C. Cremers, P. Lafourcade, P. Nadeau, “Comparing state spaces in automatic protocol analysis”, *Formal to Practical Security, Springer Lecture Notes in Computer Science*, Springer-Verlag, pp. 70–94, 2009.

[81] C. Cremers, Scyther-semantics and verification of security protocols, Eindhoven University of Technology Press. 2006.

Publications

[1] B. Kasiri, J. Cai, A.S. Alfa, "Securing coalitional game for distributed cooperative spectrum sensing in multi-channel cognitive radio networks", Advances in Security and Privacy for Future Mobile Communications, *Electronic Commerce Research*, Springer, Accepted, 2013.

[2] W. Wang, B. Kasiri, J. Cai, A.S. Alfa, "Channel assignment schemes for cooperative spectrum sensing in multi-channel cognitive radio networks", *Wireless Communications and Mobile Computing*, Wiley, doi: 10.1002/wcm.2442, 2013.

[3] B. Kasiri, J. Cai, A.S. Alfa, "Energy-efficient lightweight key management in multi-channel cognitive radio networks", Submitted, 2013.

[4] B. Kasiri, J. Cai, A.S. Alfa, "Energy-efficient distributed private key generator assignment for next generation wireless networks", Submitted, 2013.

[5] B. Kasiri, J. Cai, A.S. Alfa, "Lightweight key management in distributed multi-channel cognitive radio networks," *Military Communications Conference* (MILCOM'12), pp. 1-6, Orlando, FL, USA, Oct. 2012.

[6] B. Kasiri, J. Cai, A.S. Alfa, W. Wang, "A distributed cooperative attack on the multi-channel spectrum sensing: a coalitional game study," *IEEE Global Telecommunications Conference* (GLOBECOM'11), pp. 1-5, Houston, TX, USA, Dec. 2011.

[7] B. Kasiri, J. Cai, A.S. Alfa, "Secure cooperative multi-channel spectrum sensing in cognitive radio networks," *Military Communications Conference* (MILCOM'11), pp. 272-276, Baltimore, MD, USA, Nov. 2011.

[8] W. Wang, B. Kasiri, J. Cai, and A. S. Alfa, "Channel assignment of cooperative spectrum sensing in multi-channel cognitive radio networks", *IEEE Int'l Conference on Communications* (ICC'11), Kyoto, Japan, June 2011.

[9] W. Wang, B. Kasiri, J. Cai, A.S. Alfa, "Distributed cooperative multi-channel spectrum sensing based on dynamic coalitional game", *IEEE Global Communication Conference* (GLOBECOM'10), Miami, FL, USA, Dec. 2010.

[10] B. Kasiri, J. Cai, "Effects of correlated shadowing on soft decision fusion in cooperative spectrum sensing", *IEEE Conference on Computer Communications* (INFOCOM'10) *Cognitive Wireless Communications and Networking Workshop*, San Diego, CA, USA, Mar. 2010.