NOTE TO USERS

This reproduction is the best copy available.

Hierarchical Key Management and Distributed Multimodal Biometric Authentication in Mobile Ad Hoc Networks

by

Fei Wang

A thesis submitted to the

Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario

September, 2009

Your file Votre référence
ISBN: 978-0-494-63823-1
Our file Notre référence
ISBN: 978-0-494-63823-1

Abstract

Hierarchical key management is suitable for military applications where the organization of the network is already hierarchical in nature. Existing key management schemes concentrate only on network structure and key allocation algorithms, and the dynamic nature of MANETs is largely ignored. In this thesis, we propose a distributed hierarchical key management scheme that can select the best nodes to work as private key generator (PKG) from all available nodes based on their security conditions and energy states. The proposed scheme can improve the network security and increase the network lifetime. The node selection is formulated as a stochastic restless bandit problem. A primal dual index heuristic is used to solve the problem, which decreases the computational complexity.

Another security challenge addressed in this thesis is multimodal biometric-based authentication. We propose a distributed multimodal biometric authentication scheme that can dynamically decide which biometric device should be used for authentication. The proposed scheme is distributed as each biometric device works independently to decide if an authentication is required. Simulation results show the effectiveness of the proposed schemes.

Acknowledgments

I would like to thank the invaluable supervision and support of my supervisors, Professor F. Richard Yu and Dr. Helen Tang, during the development of this work. They have offered great guidance since I came to school and has helped me in every way possible to achieve success.

To my family, their constant support and love through the course of my studies.

To my friend Jamie Liu, Pengbo Si and Zhiqiang Li, their consistent support and encouragement helped me to overcome difficulties.

Table of Contents

List of Figures

List of Abbreviations

CO-MDP	Completely Observable Markov Decision Process
DoS	Deny of Service
GSM	Global System for Mobile communications
HIDE	Hierarchical ID-based Encryption
HIDES	Hierarchical Intrusion Detection Systems
HMM	Hidden Markov Model
IBE	ID-based encryption
IDS	Intrusion Detection System
ID-PKI	ID-based Public Key Infrastructure
LP	Linear Programming
MANET	Mobile Ad hoc Network
MDC	Markov Decision Chain
MDP	Markov Decision Process
MANETs	Mobile Ad Hoc Networks
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKG	Private Key Generator
PWLC	Piecewise Linear and Convex
POMDP	Partially Observable Markov Decision Process
SIM	Subscriber Identity Module
TA	Trusted Authority

List of Symbols

a_n^t	Action adopted by node n at time t
b_{ij}	Observation probability when state is i and observed as state j
p_{ij}	Transition probability from state i to state j
q	System accumulated reward
r_{ij}^a	System reward when state is changed from i to j and action a is adopted
t_i	Threshold value at level i
B	State observation probability
D_{th}	Network death threshold
G_i	Cyclic group i
H	Hash function
K	Shared key in IBE
N_{th}	Threshold in cryptography
O^n	Observation matrix of the device state
O_s^n	Observation matrix of the device security state in POMDP model
O_e^n	Observation matrix of the device energy state in POMDP model
S_n^t	State of node n at time t
V	Value function
$\mathscr{A}$	Action space
$\mathscr{D}$	State-action space
$\mathcal{L}$	Network lifetime
$\mathscr{S}$	State space
$\mathscr{U}$	Policy space

Θ	State observation state set
δ	System policy
γ	Balance factor for system cost
β	Discount factor for System cost
δ_i	Project i is selected
π	Information state of POMDP
$\overline{\lambda}_{i_n}$	Dual solution to restless bandit problem

Chapter 1

Introduction

1.1 Research Overview

Identity (ID)-based cryptography or ID-based encryption (IBE) and the associated ID-based public key infrastructure (ID-PKI) have a number of properties that make them attractive in providing security services for mobile ad hoc networks (MANETs) [1], which are gaining importance with the increasing number of potential applications, such as military battlefield communications.

ID-PKI typically involves a global trusted authority (TA) or central authority (CA) who has a master secret key and is responsible for generating private keys for other nodes, based on their IDs, and distributing these keys to the nodes over a secure channel [1]. In ID-based cryptography, user identity is usually composed of a unique ID, such as an email address or a telephone number, and a preset expiration date indicating the lifetime of the key. Users should contact the TA to get a new private key before the current ones expire, which is called key update or key refresh. The security of the TA becomes the major part of the whole network security. Compared to wired networks, MANETs are inherently less secure because of the shared wireless medium and lack of any central authority. The unique characteristics of MANETs

present new challenges to security design in MANETs.

Threshold cryptography [2, 3] is proposed to allow some or all network nodes share a network master key and collaboratively issue private keys for other nodes. In a MANET with n nodes, any k nodes in the group are capable of generating private keys using their shares of the master key, which is called (k, n) threshold cryptography. Using threshold cryptography, the security of the network can be maintained except when a threshold number of node secrets are compromised.

In large MANETs, a hierarchical key management structure would serve well for military applications where the organization of the network is already hierarchical in nature. In hierarchical key management, a root TA needs only distribute keys to a small number of organizations, each of which can work as private key generator (PKG) and distribute keys to smaller and smaller units, until finally all the end-nodes get their secret keys. Several hierarchical key management schemes have been proposed. In [4] the authors give a hierarchical and ID-based key management scheme with low memory size and high resistance to collusion attacks. In [5] the authors give a hierarchical key management scheme based on randomized subset and nodes will distribute a subset of its keys to its children. In [1] the authors give a lightweight and secure framework enabling key refreshing in MANETs which can be extended to support hierarchical scheme. Authors in [6] give a non-interactive hierarchical key management which combines all the advantages of schemes proposed in [4, 7]. The scheme proposed in [6] is an ID-based threshold system which is fully resilient against compromise of any numbers of leaves in the hierarchy and a threshold of nodes in each of the upper levels of the hierarchy. Although these works have been done for hierarchical key management in MANETs, most of them concentrate on the network structures and key management algorithms. None of the existing proposals considers the dynamic behavior of nodes in key management, specifically, how to select the best

nodes to work as PKG is largely ignored.

In MANETs with hierarchical ID-based key management, new users can be added to the hierarchy by their parents acting as the PKG or a threshold of siblings acting as the PKG. Due to the distributed nature of tactical MANETs, nodes security conditions change dynamically, some nodes are in safe state while some nodes are attacked or even compromised by adversaries. To select a compromised node as a member of the PKG will definitely pose great danger to the security of the whole network. Another issue confronting with MANETs is the node energy, because most wireless mobile devices are powered by batteries with limited energy, the battery power should be consumed wisely to maximize the network lifetime. Finally, how to select the best nodes to construct the PKG while taking into account the node states should be carefully investigated. In this thesis, we propose an optimal and distributed hierarchical key management scheme to select the best nodes to work as PKG considering the node security and energy states. Therefore, the network security is improved and network lifetime can be increased with the proposed scheme.

Another security challenge addressed in this thesis is the multimodal biometric authentication in MANETs. User authentication, as the front line of the MANETs security, is the core requirement for system integrity, confidentiality and non-repudiation [8]. Authentication is the process of confirming the identity claimed by a user and ensure the resources accessed by an authentic user. Traditional user authentication schemes like password, tokens etc., has no direct connection with the user itself and any uses having the key can get access to the network.

Biometric technologies, such as fingerprints, hand geometry, iris, face, retina, facial thermogram etc., verify an individual by his or her physiological or behavioral characteristics [9]. Only the users having the biometric information can use the system. Therefore, biometrics can be used in high security MANETs environments. Another

advantage of biometric technology is it can provide continuous authentication during the system operation, since it has direct connection with user identity and needs little user interruption. For tactical MANETs in hostile environments, continuous user authentication is necessary during the lifetime of MANETs, where chances of node capture are high. Due to the above mentioned features, biometric authentication has been implemented in many security related systems, including MANETs [9].

However, each biometric technology has some drawbacks. For example, biometrics is not 100% accurate since biometric matching is a pattern-recognition problem and not a simple bit-by-bit comparison, different biometrics may have different accuracy. In addition to that, some biometrics may be too expensive for low-power, computation limited mobile devices. Multimodal biometrics can alleviate some drawbacks of one mode of biometrics by providing multiple verifications of the same identity. Multimodal biometric authentication verify a user by multiple biometric traits and also, the authentication can be continuously performed without user interaction. Consequently, user can be authenticated and monitored even without realization of the authentication process. Multimodal biometrics may hold the key to more accurate and practical biometric authentication.

IDS is detection based security facility that inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. Information from IDS can be combined with multimodal biometric technology to protect the network security. Some authors in [10–13] have addressed multimodal authentication systems, but none of them considered the combination of multimodal authentication and IDS. Authors in [14] have proposed a combined IDS and multimodal biometric continuous authentication scheme in MANETs. In the scheme [14], IDS is modeled as a sensor and is used to detect system security state. The information of the IDS

is shared with other biometric sensors. Through information exchange, IDS works jointly with other biometric sensors to provide an optimal authentication. However, the scheme in [14] is a framework, it provides a centralized model and does not give concrete implementation in a real network.

In this thesis, we propose an optimal and distributed multimodal biometric authentication scheme. The proposed scheme combines IDS and multimodal biometrics and considers both security conditions and energy states of the device to select the best sensor for authentication. Therefore, the proposed scheme can improve the security of the node as well as the lifetime of the node. We also give the multiple biometric sensors online scheduling process which can be used in a real network.

1.2 Thesis Motivations

The motivations behind the proposed hierarchical key management scheme are as follows.

- In MANETs with public-private key structure, key management is the core of the system security. Although threshold key management enables multiple nodes work as central authority which can improve the system security, the security of the whole network is breached when a threshold number of node secrets are compromised. On the other hand, in dynamic MANETs environments each node is in uncertain and different conditions, and some nodes may be secure while others may be attacked or even compromised by adversaries. Thus how to optimally select the best nodes to work as PKG should be carefully investigated.

- There is no centralized control point in MANETs, therefore, the node selection scheme should be distributed. New nodes can be added to the network freely.

- Most mobile devices in MANETs are powered by batteries with limited energy and low cost central processing unit, so nodes have constraints in energy and computational capacity. The proposed scheme should be simple and easy to implement.

For the distributed multimodal biometric authentication scheme, our motivations are as follows:

- Multimodal biometric authentication makes up some drawbacks of unimodal biometrics. However, continuous multimodal biometric authentication can be expensive for energy limited mobile devices. Multiple biosensors should be used wisely with consideration of both security requirements and energy conditions.

- Detection based IDS and prevention based multimodal biometrics can work jointly in system security. The information from IDS can be used in multimodal biometric authentication. Through information sharing, a joint scheme can provide higher level security while keeping the energy consumption at low level.

- Due to the dynamic and distributed nature of MANETs, the proposed scheme should be distributed, there is no centralized arbitrator to coordinate the authentication process.

We propose two schemes in this thesis to cope with the two important security problems in MANETs.

1.3 Thesis Contributions

In this thesis, we first propose an optimal and distributed hierarchical key management scheme to select the best nodes to work as PKG while taking account into the nodes security conditions and energy states. The proposed scheme not only minimizes

the compromising probability of the network, but also increases the network lifetime. Some distinct features of the proposed scheme include:

- The proposed scheme always selects the best nodes (either a parent node or a threshold of siblings) for constructing the PKG while taking into account both the security conditions and energy states of all possible nodes. Therefore the proposed scheme can improve the security of the network as well as increase the network lifetime.

- The node selection problem in tactical MANETs is formulated as a restless bandit problem, which is a well-studied generalization of the stochastic multi-armed bandit problem and has been successfully used in solving stochastic scheduling problem [15], such as clinical trials, project management and aircraft surveillance [16]. The restless bandit approach is based on Markov decision chain and has a "indexable" property [16–19], which means the optimal policy is just selecting the nodes with the smallest indices.

- Since the restless bandit problem is known to be PSPACE Hard, a primal dual index heuristic [16] is used to solve the problem. The priority indices can be computed off-line and kept as an index table. In the online part of our scheme, the priority indices table can be easily accessed and used for node selection. The computation and implementation complexity are reduced dramatically.

- The scheme is fully distributed and nodes can join and leave the network freely, thus the scheme is very suitable in military environment involving collaboration between forces from different countries and different agencies. The scheme can also be easily extended to support more properties of nodes such as bandwidth, channel states, etc.

The second scheme we propose in this thesis is a distributed multimodal biometric authentication scheme in MANETs. The scheme combines IDS and multimodal biometrics and considers both security and energy states of the device to select the best sensors for user authentication. The sensor selection process is formulated as a Partially Observable Markov Decision Process (POMDP) since the observation of the system states is not accurate. The proposed scheme is designed to improve the network security as well as the lifetime of the node. The main features of the proposed scheme include:

- Based on current device states, each biometric sensor works independently to decide if an authentication is required, there is no interaction among the sensors. Therefore the proposed scheme is fully distributed and there is no coordinators in the sensor scheduling process, thus the scheme is more reliable than a centralized scheme.

- Since the observation of the device state is not accurate, the biosensors selection decision is a POMDP.

- We consider both the security states and energy states in the proposed scheme, and the proposed scheme can maintain the system security and improve the device lifetime.

The two proposed schemes have some similarities. They both use information from IDS for decision making. The hierarchical key management scheme is designed for large networks with many nodes at different levels, therefore, network based IDS is used for node selection. In the proposed multimodal biometric authentication, IDS is used to monitor the mobile device states, therefore, host based IDS is used. More detailed information for the two kinds of IDSs will be described in Chapter II.

Due to different system properties, the hierarchical key management is an MDP and the multimodal biometric authentication is a POMDP. MDP, POMDP and the corresponding solutions will be described in Chapter III.

1.3.1 Published and Accepted Papers

Several papers have been published and accepted based on this work. One journal papge is in preparation.

- F.R. Yu, H. Tang, F. Wang and V.C.M. Leung, "Distributed Node Selection for Threshold Key Management with Intrusion Detection in Mobile Ad Hoc Network", in *Proc. IEEE/IFIP TrustCom'09*, Vancouver, BC, Aug. 2009. (Best paper award).

- F. Wang, H. Tang, F.R. Yu and P.C. Mason, "A Hierarchical Identity Based Key Management Scheme in Tactical Mobile Ad Hoc Networks", to be presented at IEEE Milcom'09, Boston, MA, USA, Oct. 2009.

- F. Wang, F.R. Yu and Anand Srinivasan, "Identity Based Key Management Scheme in Tactical Mobile Ad Hoc Networks", to be presented at IEEE Globecom'09, Honolulu, Hawaii, USA, Dec. 2009.

1.4 Thesis Organization

The rest of the thesis is organized as follows. In Chapter II, we describe the research background of this thesis, which includes ID-Based cryptography, threshold technology, hierarchical key management and multimodal biometric authentications, etc. In Chapter III, we present system models and formulations of Markov Decision Process and Partially Observable Markov Decision Process. Our system formulations

are based on the two models. We also investigate the solutions to MDP and POMDP. In Chapter IV, we propose the hierarchical key management scheme. We give the system model, restless bandit formulation and the primal dual index heuristic solution. The key update process of our proposed scheme and related discussions are also described in Chapter IV. The distributed multimodal authentication scheme and the corresponding solution are described in Chapter V. In Chapter VI, we provide the simulation results and discussions for both schemes. Some simulation examples are used to show the effectiveness of the proposed schemes.

Finally, we conclude this thesis in Section VII. Moreover, we highlight a number of research areas for future work.

Chapter 2

Research Background

2.1 Mobile Ad hoc Networks

In recent years, MANETs have become a popular subject because of their self-configuration and self-organization capabilities. Each device in MANETs is free to move independently in any direction, and will therefore change its links to other devices frequently. Wireless nodes can establish a dynamic network without the need of a fixed infrastructure. A node can function both as a network router for routing packets from the other nodes and as a network host for transmitting and receiving data. MANETs are particularly useful when a reliable fixed or mobile infrastructure is not available. Instant conferences between notebook PC users, military applications, emergency operations, and other secure-sensitive operations are important applications of MANETs due to their quick and easy deployment.

2.1.1 Self-Organization of MANETs

Due to the complete lack of centralized control, MANETs nodes cooperate with each other to achieve a common goal. The major activities involved in self-organization are

neighbor discovery, topology organization, and topology reorganization. Through periodically transmitting beacon packets, or promiscuous snooping on the channels, the activities of neighbors can be acquired. Each node in a MANET maintains the topology of the network by gathering the local or entire network information. MANETs need to update the topology information whenever the networks change such as participation of new node, failure of node and links, etc. Therefore, self-organization is a continuous process that has to adapt to a variety of changes or failures.

2.1.2 Security and Constraints in MANETs

The security in MANETs is very important especially in tactical military environments. Unlike the wired networks, MANETs are inherently not secure because of the lack of any central authority and shared wireless medium. The major security threats that exist in MANETs are as follows: denial of service (DoS), resource consumption, host impersonation, information disclosure, and interference. The unique characteristics of MANETs present some new challenges to security design [20].

- Shared wireless broadcast radio: A node can receive and transmit data from and to all the nodes within its direct transmission range, so adversaries in the same range can monitor the data.

- Insecure operation environment: MANETs may operate in hostile environments, especially for the tactical MANETs. Nodes frequently move in and out of hostile enemy territory. The chances of node capture are high in such environments, which require re-authentication.

- Lack of central coordination: There is no centralized network management functionality in MANETs; centralized coordination violate MANETs basic structure

and also can be too expensive for mobile nodes. The existing security solutions for wired networks cannot be applied directly to the MANETs domain.

- Lack of association: Because of the dynamic characteristic of MANETs, it is difficult to find a proper authentication mechanism to use for associating nodes with a network.

- Limited resource availability: Bandwidth, battery power, and computational power are scarce in MANETs.

In tactical MANETs, there are some extra requirements for security design. Since MANETs need to transmit some critical information, security is paramount important.

2.1.3 User Authentication in MANETs

In MANETs, a complete security scheme should include three security components: prevention, detection, and reaction [21]. Among those facilities, user authentication (prevention based) is the most crucial to protect the network, because it is the first step for use to get access for a device. User authentication is a process of confirming user identity and ensure the resources be accessed by an authentic user.

Traditional authentication scheme may use a password, a personal identification number (PIN), or something like a secret question, etc. The above authentications are simple, easy to use. However, the security can be easily breached because there is no direct connection between a user and a password. A malicious user can get access to the system only if he has the password or PIN, etc.

Another authentication is an electrical token, a private key, or for example a subscriber identity module (SIM) card. For this scheme, like a password, it still

does not have connection between user and a token, it also subject to loss or being counterfeited.

Biometric technology can make up the drawbacks of those schemes. It will not be lost and also, users do not need to remember something like a password. Biometrics can also be monitored without user interaction, therefore, it can be used for continuous authentication.

2.2 ID-Based Cryptography

ID-based cryptography was introduced by Shamir [22] In 1984. In the paper he asked for a public key encryption scheme in which the public key can be an arbitrary string. Shamir's original motivation for identity-based encryption was to simplify certificate management in email systems. When Alice sends an email to Bob at "bob@company.com" she simply encrypts her message using the public key string "bob@company.com". There is no need for Alice to obtain Bob's public key certificate. When Bob receives the encrypted mail he contacts a third party, i.e. PKG. Bob authenticates himself to the PKG in the same way he would authenticate himself to a CA and obtains his private key from the PKG. Bob can then read his email. Note that unlike the existing secure email infrastructure, Alice can send encrypted mail to Bob even if Bob has not yet setup his public key certificate. Also note that key escrow is inherent in ID-based email systems: the PKG knows Bob's private key.

ID-based encryption (IBE) did not have practical solutions until the break-through paper of Boneh and Franklin [23] in which the first efficient and provably secure ID-based encryption scheme was presented, which is based on bilinear maps. We will give some preliminary knowledge of bilinear maps. An IBE scheme will be presented to follow that.

2.2.1 Bilinear Maps and the BDDH Assumption

Bilinear map can be described simply as:

Let G_1 and G_2 be two cyclic groups (for example additive group) of order q for some large prime q. Let e be a mapping $e : G_1 \times G_1 \rightarrow G_2$. The mapping e is:

1. Bilinear if $e(P^a, Q^b) = e(P, Q)^{ab}$ for any $P, Q \in G_1, a, b \in Z_q$.
2. Non-degenerate if e does not send all pairs to the identity in G_2.
3. Computable if there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Bilinear mappings that can be computed efficiently are known based on Weil and Tate pairings in Abelian varieties. G_1 and G_2 will be the group of points on an elliptic curve.

Sakai et al. [24] propose the following non-interactive (but not hierarchical) key-agreement scheme. The central authority sets up the parameters for an identity based public key system, by fixing two cyclic groups G_1, G_2 and the bilinear map $e : G_1 \times G_1 \rightarrow G_2$. Furthermore, it chooses a cryptographic hash function $H : \{0,1\}^* \rightarrow G_1$. It then chooses a secret key $s \in Z_q$ and provides a node with identity ID with the secret key $S_{ID} = H(ID)^s \in G_1$.

The shared key between two nodes with identities ID_1 and ID_2 is:
$K = e(H(ID_1), H(ID_2))^s \in G_2$. When party ID_1 want to communicate with ID_2, it will compute $K = e(S_{ID_1}, H(ID_2))$, and ID_2 will compute $K = e(H(ID_1), S_{ID_2})$. The security of this scheme can be reduced to the BDDH assumption in the random-oracle model, as was shown in [25]. The central hardness assumption this scheme based on is the Bilinear Decisional Diffie-Hellman Problem (BDDH) assumption introduced by Boneh and Franklin [3].

BDDH problem can be described simply as:

Let G_1, G_2 and e be as above. Given a random $P \in G_1$, $P^a, P^b, P^c \in G_1$ for

random $a, b, c \in Z_q$, and given $h \in G_2$, it is hard to distinguish the case where $h = e(P,P)^{abc}$ from the case where $h = e(P,P)^r$ for a random and independent $r \in Z_q$.

Formally, an algorithm A has advantage in solving the BDDH in $\langle G1, G2, e\rangle$, if

$$Pr[A(P, P^a, P^b, P^c, e(P,P)^{abc}) = 1] - Pr[A(P, P^a, P^b, P^c, e(P,P)^r) = 1] \geq \epsilon, \quad (2.1)$$

where the probability is over the random choice of $P \in G_1, a, b, c, r \in Z_q$, and the internal randomness of A. The BDDH assumption (with respect to $\langle G_1, G_2, e\rangle$) states that feasible adversaries can have only an insignificant advantage.

2.2.2 IBE Key Allocation Scheme

In ID-based systems, any two nodes can compute a unique shared secret key without interaction; shared key between two parties in communication is computed based on one party's private key and another party's identity, such as an email address or a telephone number, etc.

Boneh and Franklin's IBE scheme [23] is given by the following four stages:

- *Setup*: PKG specifies two groups G_1 and G_2, and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ between them. It also specifies three hash functions:
  - $H_1 : \{0,1\}^* \rightarrow G_1$, this hash function is used to extract point from ID.
  - $H_2 : G_2 \rightarrow \{0,1\}^l$, where l is the length of a plaintext message (hash to the message space).
  - $H_3 : G_2 \rightarrow Z_q^*$, this function is hash to the finite field, which will be used in the proposed key issuing protocol.

PKG picks a master key $s_0 \in Z_q^*$ at random and computes his public key

$P_0 = s_0P$. PKG then publishes description of the groups G_1, G_2, the bilinear map e, the hash functions H_1, H_2, H_3, and his public key P_0.

- *Extract*: Let Alice be a sender and Bob be a receiver. Bob requires a private key for his $ID \in \{0,1\}^*$ to PKG. For given Bob's ID, the PKG computes Bob's public key as $Q_{ID} = H_1(ID)$ and the corresponding private key as $D_{ID} = s_0Q_{ID}$. Note that D_{ID} is a short signature [26] of the PKG on the message ID. Then he sends D_{ID} to Bob through a secure channel. Bob can check the validity of his private key by if: $e(D_{ID}, P) = e(Q_{ID}, P_0)$.

- *Encrypt*: To encrypt a message $m \in \{0,1\}^l$ with the public key of the receiver Bob, Alice first computes Bob's public key by $Q_{ID} = H_1(ID)$. Then she picks a random number $r \in Z_q^*$ and computes $U = rP$ and $V = m \oplus H_2(e(Q_{ID}, P_0)^r)$. Then the ciphertext $C = (U, V)$ is sent to Bob.

- *Decrypt*: The receiver Bob can decrypt the ciphertext C $= (U, V)$ using his private key D_{ID} by $V \oplus H_2(e(D_{ID}, U)) = m$. The decryption works because of the bilinear property of the map e,

$$e(D_{ID}, U) = e(s_0Q_{ID}, rP) = e(Q_{ID}, P_0^r). \tag{2.2}$$

The major advantage of ID-based cryptography is it require less interaction and lower bandwidth than traditional PKI but offer greater flexibility, which make it well-suited for MANETs [27–29].

2.3 Threshold Secret Sharing

Boneh and Franklin considered the distributed key generation in their original proposal [23] of ID-based encryption to protect the secrecy of the master key, not to

protect the privacy of the private keys of users. The principal of threshold secret sharing [2, 3] is a group of N nodes called shareholders sharing the secret instead of a single node. The secret can only be reconstructed by a threshold N_{th} of nodes. Each of them holds a unique secret share of the master secret key, and no one is able to reconstruct the master private key based on its own information. Any $k(k \geq N_{th})$ nodes can reconstruct the master private key jointly, where it is infeasible for less than k nodes to do so, thus the scheme is resilient against compromise of N_{th} of nodes.

Threshold schemes generally involve key generation, encryption, share generation, share verification, and share combining algorithms. Threshold models can be broadly divided into single secret sharing threshold, e.g. Shamir's t-out-of-n scheme based on Lagrange's interpolation and threshold sharing functions, e.g. geometric based threshold [3].

2.4 Intrusion Detection Systems

Intrusion Detection Systems are becoming an exciting and important technology in MANETs security in recent years because the intrusion prevention techniques cannot satisfy the security requirements in mission critical systems. An IDS can automatically monitor and analyze the events, once the attacks are detected, alarms will be generated or protective measures will be activated to protect systems.

Intrusion detection presents a strong mechanism of defense and is commonly used in high-survivability network. There are three types of IDS categorized by the type of data IDSs use [30]:

1. Network-based intrusion detection: It runs at the gateway of a network and examines all incoming packets, it detects all packets to find possible attacks.

2. Host-based intrusion detection: It receives the necessary audit data from the

operating system and analyzes the generated events.

3. Router-based intrusion detection: It is installed on routers to prevent intruders from entering into the network.

For MANETs, the most commonly used IDS is host-based since there may be no centralized gateway or router exists. Also, host-based IDS is designed to protect the node local security so can be used in our multimodal biometric authentication scheme. In the proposed multimodal biometric authentication scheme, host-based IDS will be used to monitor the device to find possible attacks. Intrusion detection and response systems: multimodal biometric authentication work collaboratively to meet the needs of MANETs. In our proposed distributed multimodal biometric authentication scheme, the node will take appropriate actions according to the observation from IDS. IDS is also model as a sensor with cost less than biometric sensors, and used only for detection of system state. The information observation from IDS is used to decide which biosensor to work for user authentication.

In large scale MANETs, especially in military applications, network-based IDSs are necessary because security is of high priority. Therefore, in the proposed hierarchical key management scheme, a network-based IDS is used, the IDS continuously or periodically monitors the network, and maintains the nodes states. The IDS also have storage of a priority index table for all available nodes, which is used for node selection. Whenever a key update request is received, IDS will select the best nodes to work as PKG-based on current nodes states.

2.5 Hierarchical ID-based Key Management in MANETs

In single PKG-based (not hierarchical) networks, all nodes get their private keys from the exclusive PKG. The key allocation process is simple because there is no online lookup of the PKG. However, it is undesirable for a large network because the PKG has a burdensome job. Not only is a private key generation computationally expensive, but also the PKG must verify proofs of identity and must establish secure channels to transmit private keys.

Hierarchical ID-based encryption (HIDE) allows a root PKG to distribute the workload by delegating private key generation and identity authentication to lower-level PKGs, thus the computation burden is alleviated and key generation becomes distributed.

In a hierarchical key management scheme, a root PKG needs only generate private keys for domain-level PKGs, who in turn generate private keys for users in their domains in the next level. Authentication and private key transmission can be done locally.

Another advantage of HIDE schemes is damage control: disclosure of a domain PKGs secret does not compromise the secrets of higher-level PKGs.

Using ID-based and threshold cryptography, hierarchical key management can offer a scheme that has all of the advantages of these schemes [6]. A typical implementation that is based on multiple variables polynomial [6] is introduced as follows.

In this scheme, each node has a secret polynomial (in the role of a secret key), and the shared key between two leaf nodes is computed by evaluating the polynomial held by one node at a point that corresponds to the identity of the other node.

Fig. 2.1 shows the network structure of the hierarchical key management scheme.

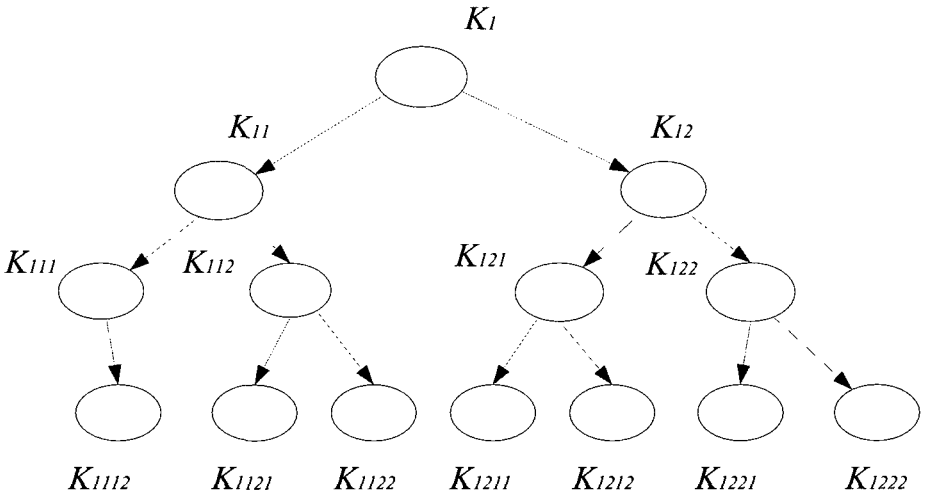

Figure 2.1: Hierarchical key management structure.

In this tree-like structure, let L be the depth of the hierarchy, i.e., the nodes are arranged in a tree with L levels. Each node identity corresponds to the path from the root to the node (thus a node at level i will have as identity a vector with i components $(I_1, ..., I_l)$ where each I_i is an integer).

For desired threshold parameters t_i, the root authority chooses a random polynomial (over Z_q for a large enough prime q) $F(x_1, y_1, \ldots, x_L, y_L)$, where the degree of x_i, y_i is t_i. F is chosen such that:

$F(x_1, y_1, \ldots, x_L, y_L) \equiv F(y_1, x_1, \ldots, y_L, x_L)$, i.e. F is symmetric between the x's and y's. A simple implementation to choose such polynomial is to choose a random polynomial f on the same variables, and then set $F(x_1, y_1, \ldots, x_L, y_L) = f(x_1, y_1, \ldots, x_L, y_L) + f(y_1, x_1, \ldots, y_L, x_L)$. The size of the description of F (ie. number of coefficients) is $\prod_{i=1}^{L} \frac{(t_i+1)(t_i+2)}{2}$ (the half is due to the symmetry of the polynomial). Note that when t_i is big, the coefficients of the polynomial will be large.

The master secret key of the system is the polynomial F itself. The secret key of node with identity I in the first level of the hierarchy is the polynomial $F_I = F(I, y_1, x_2, y_2, \ldots)$ that has $2L - 1$ variables. Similarly, the secret key of a node at level i with identity $\vec{I} = \langle I_1, \ldots, I_i \rangle$ is the polynomial: $F_{\vec{I}} = F(I_1, y_1, \ldots, I_i, y_i, x_{i+1}, y_{i+1}, \ldots)$ that has $2L - i$ variables, and the secret key of the leaf with identity $\langle I_1, \ldots, I_L \rangle$ is the polynomial in L variables $F(I_1, y_1, \ldots, I_L, y_L)$.

The shared key between the two leaf nodes $\langle I_1, \ldots, I_L \rangle$ and $\langle J_1, \ldots, J_L \rangle$ is the value of the polynomial $F(I_1, J_1, \ldots, I_L, J_L) = F(J_1, I_1, \ldots, J_L, I_L)$, that each node can compute by evaluating its secret polynomial on the points that correspond to its peer's identity.

We call a node compromised if the attacker has learned all of the nodes secrets (i.e., all the coefficients of the polynomial the node holds, and hence all of its descendants shared keys), otherwise we call it uncompromised. This scheme guarantees that the key shared between any two uncompromised nodes is information theoretically secure, namely, all values of the key are equally possibly given the attackers view. Some characteristics of this scheme includes:

- Non-interactive: Any two nodes can compute a unique shared secret key without interaction.
- ID-based: To compute the shared secret key, each node only needs its own secret key and the identity of its peer.
- Hierarchical: The scheme is decentralized through a hierarchy where intermediate nodes in the hierarchy can derive the secret keys for each of its children without any limitations or prior knowledge on the number of such children or their identities.
- Resilient: The scheme is fully resilient against compromise of any number of

leaves in the hierarchy, and of a threshold number of nodes in each of the upper levels of the hierarchy.

Note that a node N in the hierarchy can be compromised (i.e., all its secrets learned) by directly breaking into N and finding its secrets or by breaking into other nodes from which the information in N can be reconstructed. For example, one can learn all of N's secrets by breaking into an ancestor of N or by breaking into $t + 1$ of its children (where t is the node's threshold). Here, the word secrets can refer to the coefficients of the polynomial held by a node N or, equivalently, to the set of pairwise shared-keys known to N and its descendants (i.e., the set of keys shared by these nodes with every other node in the hierarchy).

An alternative approach to building a hierarchical scheme is to use subset-based key pre-distribution schemes as in Eschenauer and Gligor [31], and extend it to a hierarchical scheme as in Ramkumar et al. [5]. Roughly, in this protocol the root authority chooses a large number of secret keys for its key-ring, the key-ring of every node contains a random subset of these keys, and the shared key for two nodes is computed from the intersection of the keys in their respective key-rings.

Most of the existing schemes concentrate on the hierarchical network structure and key management algorithms while our proposed scheme gives more consideration to the dynamic behavior in the key management, specifically, we consider selecting the best nodes to work as the PKG while taking account into security conditions and energy states of all nodes. Our proposed scheme can be easily combined with existing schemes, which will be shown in Chapter IV.

2.6 Key Update in Tactical Hierarchical MANETs

In MANETs, keys need to be updated or refreshed at intervals [32] to ensure system security. There are several situations where it is necessary to update private keys of

nodes [1].

- The identifier in ID-based system may be a short-term one, for example, In ID-based cryptography, node ID usually comes with an expiration date encoded as part of the identifier. The expiry date of the ID may be imminent, but the field operation on-going. Thus in ID-based systems, node keys need to be updated in some intervals.

- The node holding the private key may need to change its TA. For example, it may become temporarily assigned in the field to a coalition force and require and appropriate private key from the TA for that force in order to maintain communications with its new group of peers. This can happen frequently in military operation.

- TA may decide to update its public parameters, necessitating an update to all private keys. One might expect this to be a relatively rare event in situations where the hierarchy is static and well defined. But in the dynamic coalition-forming environments it may become more common. In particular, and as described above, two TAs from different coalition forces may wish to (temporarily or permanently) generate a common set of public parameters and a common master secret, and to issue new private keys to all entities under their joint command.

Key update scheme may be trivial in single TA based MANETs where all nodes get their keys from the unique TA who works as the PKG. The security of the network depends solely on the single TA, if the TA is compromised then the whole network secrets are compromised. Another problem with the single TA system is the high volume of traffic when the TA starts key updating for all other nodes in the network.

Since the single TA scheme is vulnerable and centralized it may works well only in small networks.

In large MANETs where there are many functional units, especially, in military environment where organizations are already hierarchical in nature, a hierarchical key management is preferred. Key update in hierarchical network can be processed at different hierarchies since multiple PKGs exist at different levels. It is possible for nodes to get private keys from either its parent or a threshold of sibling nodes. The security risks of the hierarchical network thus distributed at different levels. Compromising of a subtree will not affect the security of another subtree, only if they does not belong to the same ancestor.

The aim of our work is to dynamically decide which node/nodes (can be a parent node or k nodes among the n nodes with secret key shares) to work as PKG based on these nodes states. The system model and the key update process of the proposed scheme is described in Chapter IV.

2.7 Multimodal Biometric-Based Authentication

Another scheme proposed in this thesis is distributed multimodal biometric authentication. Biometrics is a technique commonly known as the automatic identification or verification of an individual by his or her physiological or behavioral characteristics [9]. Biometrics provides some possible solutions to authentication used in MANETs, since it has direct connection with user identity and needs little user interruption [9, 11].

Biometrics is not an 100% accurate authentication scheme, because there are always normal variations in measured features and measurement environments. The authentication of biometrics is a closeness of match, or a probability. Thus the access control portion of a biometric authentication system must be set to accept the identity of users at some threshold.

Multimodal biometric system provides more reliable authentication because it involves multiple statistically independent biometric traits [33]. This system can make up the shortcomings of unimodal biometric system by selecting different biometric traits based on the security requirement.

However, the authentication processes of biometrics generally require more computation than password or token verification. This concern is more substantial in continuous authentication, especially for mobile devices with limited energy. Hence, the energy costs must be addressed for continuous authentication with biometrics in MANETs.

Multimodal biometric systems help to achieve an increase in performance over that only using a single biometric. However, finding an effective fusion system is necessary to combine the information presented by multiple biosensors.

A general biometric system has four important components:

- Sensor module which acquires the biometric data of a user.
- Feature extraction module which processes the data to extract the feature values.
- Matching module which compares the feature values with the template to generate a matching score.
- Decision-making module in which the result of verification or authentication is acquired: accept or reject.

2.8 Related Works

2.8.1 Related Works in Hierarchical Key Management

Several hierarchical key management schemes have been proposed. In [4] the authors give a hierarchical and ID-based key management scheme with low memory size and high resistance to collusion attacks. In [5] the authors give a hierarchical key management scheme based on randomized subset and nodes will distribute a subset of its keys to its children.

A hierarchical ID-based key sharing scheme with partial collusion-resistance is given in [34]. Horwitz and Lynn [35] introduce hierarchical ID-based encryption, and proposed a 2-level HIDE scheme with total collusion-resistance at the first level and with partial collusion resistance at the second level, i.e., (a threshold number of) users can collude to obtain the secret of their domain PKG (and thereafter masquerade as the domain PKG). This scheme may be practical for applications where collusion below the first level is not a concern.

Authors in [6] give a non-interactive hierarchical key management which combines all the advantages of schemes proposed in [4, 7]. The scheme proposed in [6] is an ID-based threshold system which is fully resilient against compromise of any numbers of leaves in the hierarchy and a threshold of nodes in each of the upper levels of the hierarchy.

Although the aforementioned works have been done for hierarchical key management in MANETs, most of them concentrate on the network structures and key management algorithms. None of the existing proposals consider the node dynamic behavior in tactical MANETs, specifically, how to select the best nodes to work as PKG should be carefully investigated.

2.8.2 Related Works in Multimodal Biometric Authentication

Some authors have proposed some schemes in continuous multimodal biometric authentication. In [12] the authors use Dynamic Bayesian network (DBN) to solve the uncertainty and encode the system's dynamic model. By modeling more hidden variables, DBNs were cable of modeling important contextual information. Authors in [10] proposed a continuous multimodal biometrics system using a hidden Markov model (HMM). The authors proposed several new metrics which took the time factor into consideration for multimodal biometrics used in continuous verification. However, how to optimally schedule different biometrics based on the system security is not mentioned.

In [13], Altinok and Turk applied the face, voice and fingerprint biometrics for continuous authentication. They argued that continuous authentication need to integrate the time and modality. The multimodal system weighted each modality at the score level, and the weighting factor was decreased monotonically with the time since the last measurement.

The authors in [11] developed a multimodal authentication system for the secure phone project by using the combination of voice, face, and signature biometrics. The storage and processing of the client's biometric prole all were done on the SIM-card in the PDA and only the service provider can access to the SIM-card. The benchmark databases BANCA (audio-visual) and BIOMET (signature) are used. This paper proved that simple multimodal biometrics authentication scheme can be implemented on a small personal digital assistant (PDA) device.

Authors in [14] have proposed a combined IDS and multimodal biometric continuous authentication scheme in MANETs. In the proposed scheme [14], IDS is modeled as a sensor to detect system security state, and works jointly with other biometric

sensors to provide optimal authentication. The scheme in [14] is a framework, it provides a centralized model and does not give who will coordinate the biosensors scheduling and how the system operate in a real network. In this thesis, we propose a distributed scheme with consideration for both security and energy of the nodes. In the proposed scheme, each biometric sensor works independently (with only a little information from IDS) to decide if an authentications is needed.

2.9 Summary

In this chapter we presented the background knowledge of this thesis, which includes ID-Based cryptography, threshold technology, hierarchical key management scheme, MANETs, multimodal biometrics, and IDS etc.

Because of no fixed infrastructure available and the dynamic nature of ad hoc networks, the network security of MANETs remains an open question. Hierarchical key management schemes can provide high secure services but the existing schemes only focus on the key allocation and revocation algorithms, and the dynamic environment is not considered in those schemes. That is the main motivation of this thesis, and we also take into account the node energy, since node energy are essential resources in MANETs.

We also address the multimodal biometric authentication, which is also popular in high security MANETs. Multimodal biometric authentication can provide continuous authentication services in high security environment. When combined with IDS, the authentication scheme can be energy efficient, which is also the main motivation of our scheme.

Chapter 3

MDP and POMDP

MDP and POMDP are powerful models in solving stochastic control problems. We will give some background knowledge of MDP and POMDP in this chapter. Our proposed hierarchical key management scheme is based on MDP, and the distributed multimodal biometric authentication scheme is based on POMDP.

3.1 Markov Decision Process

3.1.1 MDP Model

Markov decision processes (MDPs), named after Andrey Markov, provides a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. Decision making is an important part in everyday life such as changing the components or not in machine-maintenance problem. Normally decision making is based on the system environment, the interrelationship between the system components, the previous experience. However, with the system becomes more complex and the degree of uncertainty increases, it is more difficult to make an appropriate decision by human. Hence, the automated decision making is desirable in many sophisticated applications in which the human

decision making is not feasible or impossible to formalize.

MDPs are useful for studying a wide range of optimization problems solved via dynamic programming and reinforcement learning. MDPs were known at least as early as the 1950s. Much research in the area was spawned due to Ronald A. Howard's book, Dynamic Programming and Markov Processes, in 1960. Today they are used in a variety of areas, including robotics, automated control, economics and in manufacturing.

MDP is defined by a quadruple $\langle S, A, P, R \rangle$ model, which consists of a finite set of states, a finite set of actions, the system states transition probabilities and the action-reward pairs. They are defined as:

- States S: The system states denote the way the system currently stay in. Although in most cases, the system stays in a continuous states, it is convenient to discretize these continuous spaces. S stands for the system state space, and s_i denotes the state of project i.

- Actions A: The actions are a set of decisions that can be taken at each time point. A denotes the action space.

- Transitions P: Different actions have different effects on the system states. The transition P defines transition probabilities each action changes the system states. Probability $p_{ij} = Pr(S^{k+1} = j | S^k = i, A^k = a)$ is used in MDP to specify the transition possibility from the current system state i to another system state j, here k denotes the time instant.

- Rewards R: Since different actions changes the system state in different way, we use immediate reward to measure an action's value which will be used to compare the effects of actions. r_{ij}^a represents the immediate reward when action a is performed while system stays in state s_i and moves to state s_j.

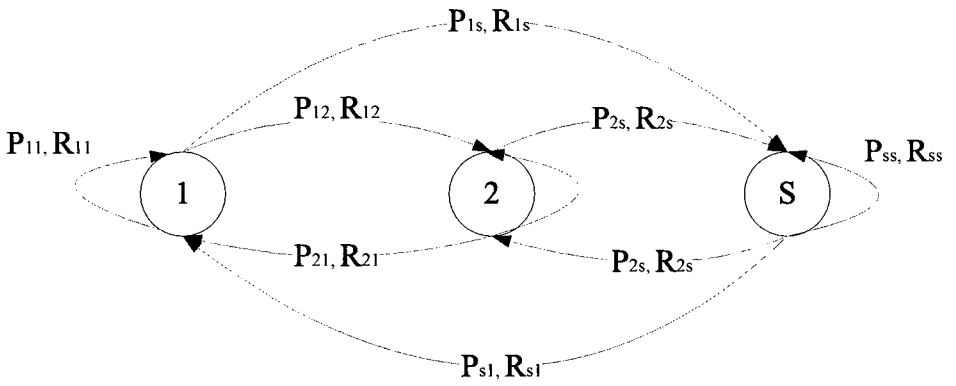

Figure 3.1: Markov Decision Process.

Fig. 3.1 shows the transition and reward model of MDP.

It is important to mention that in MDP model, the next state of the system is solely determined by the current state and the current action. In practise, there are some forms of uncertainty for MDP: the results of actions may not always have the same effects; our perceptions of the system being controlled are not always very accurate. Therefore, we let q_i^a represent the expected rewards, what we would expect to receive on average in the long term:

$$q_i^a = \sum_j p_{ij}^a r_{ij}^a \tag{3.1}$$

where r_{ij}^a are the actual rewards that would be received. The value of q_i^a only depends on the current system state and action chosen.

The system will evolve as follows: at each time instant, the system is in a particular state s_i, an action a is taken and the system will move to another system state s_j with the probability p_{ij}, an immediate reward r_{ij}^a is associated with this specific process. A common optimization objective is to maximize or minimize the reward.

3.1.2 Policy

A policy refers to the solution to a MDP and it specifies the best actions to take for each of system states. A policy completely specifies the actions which are appropriate for all system states that possibly occur during the system process.

Policy can be classified due to different criteria:

- *Deterministic or stochastic*: A policy could be deterministic or stochastic. A deterministic policy specifies only one action to execute for a system state. A stochastic policy specifies a number of actions to execute for a system state, and each action is associated with a probability distribution over the set of actions. A deterministic policy is a special case of stochastic policy since its action to execute has the probability 1 for a specific system state.

- *Stationary or non-stationary*: A policy can be classified into stationary and non-stationary. A stationary policy is irrelevant with time and same policy will be applied to the system when it is used. We use $\delta = \{\delta, \delta, \ldots, \delta\}$ denote the stationary policy. In other words, the policy only depends on the system state. A non-stationary policy depends on both time and system state. $\delta = \{\delta_1, \delta_2, \ldots, \delta_k\}$ is used to denote the non-stationary policy, here k is the time instant.

- *Finite or infinite*: The time axis is divided into slots of equal duration which correspond to the time interval between two continuous decisions, and this time interval is also known as horizon in this paper. The finite-horizon problem is the one in which decisions need to be made only for finite time steps. The policy for finite horizons consists of a sequence of deterministic policies, one for each time instant, denoted as $\delta = \{\delta_1, \delta_2, \ldots, \delta_k\}$. Another problem is infinite horizon problem in which we do not know the number of time horizons over

which the decision need to be made. Under this situation, a stationary policy is needed. At any time instant, the policy δ is same because there are unbounded amount of time left for both time instant k and $k+1$.

3.1.3 System Reward and Value Function

The immediate reward function q_i^a helps to guide the decision, but maximum immediate reward does not mean the maximum long term reward. There are many ways to balance the trade-off between the immediate rewards and long term accumulated rewards. Expected future discounted reward [36] is considered in this paper:

$$\left[\sum_{k=0}^{K-1} \beta^k q_{S^k}^{A^k}\right], 0 \leq \beta \leq 1, \tag{3.2}$$

where S^k and A^k are the random variables for the state and action chosen at time instant k, β is a discount factor that means rewards receives later in time will have less value than an equivalent reward received closer to the present.

For the infinite horizon problem, the accumulated reward is:

$$\left[\sum_{k=0}^{\infty} \beta^k q_{S^k}^{A^k}\right], 0 \leq \beta \leq 1, \tag{3.3}$$

With the above definition, the aim in solving the MDP is to find a control policy which maximizes expected future discounted reward, and this policy is called optimal policy.

The optimal policy for MDP can acquire a maximum expected future discounted reward. We use value function to compute a mapping from states to actions, which represents the best actions to take for each state.

Policy Over Finite Horizon

For a finite MDP with non-stationary process, we can calculate the value function with the recursive equation. We define $V_k^{\delta^k}(s)$ as the value of starting in state s and executing the policy δ^k for $K-k$ time steps. The reward is the immediate reward received for the current state, current action and next state r_{ij}^a, plus the value of state s_j with one less step remaining $V_{k+1}^{\delta^{k+1}} s(j)$:

$$V_k^{\delta^k}(s_i) = q_i^{\delta^k(s_i)} + \beta \sum_j p_{ij}^{\delta^k(s_i)} V_{k+1}^{\delta^{k+1}} s(j), \tag{3.4}$$

In order to calculate the equation effectively, we will use dynamic programming to compute the optimal policy. In other words, compute the equation (3.4) backward from time K down to time 0.

Policy Over Infinite Horizon

For the problem with stationary policy over infinite horizon, the policy is always same at any time instant. The value function can be written as:

$$V^{\delta}(s_i) = q_i^{\delta^k(s_i)} + \beta \sum_j p_{ij}^{\delta(s_i)} V^{\delta} s(j), \tag{3.5}$$

These value functions help us to compute a metric for the policies and compare the policies to get an optimal policy, the optimal mapping from a state to an action.

3.2 Partially Observable Markov Decision Process

In MDP, the states of the system are completely known at all times and the decisions are made based on those states. That is MDP can also be called completed observed Markov decision process (CO-MDP). However, in some cases, the system states are

not directly observed and only some observations that give a hint of the current system state. This is called a partially observable Markov decision process, which is a generalization of a Markov Decision Process. A POMDP models an agent decision process in which it is assumed that the system dynamics are determined by an MDP, but the underlying state can only be observed inaccurately, or with some probabilities. Therefore the system state is a probabilistic distribution over the state based on some local observations.

The POMDP framework is general enough to model a variety of real-world sequential decision processes. Applications include robot navigation problems, machine maintenance, and planning under uncertainty in general.

3.2.1 POMDP Model

POMDP can be defined by a hex-tuple $\langle S, A, P, \Theta, B, R \rangle$, The definitions of S, A, P are the same as those in MDP. S stands for a finite set of states with state i denoted by s_i. A stands for a finite set of actions with action i denoted by a_i. P stands for transition probabilities for each action in each state, and p_{ij}^a denotes the probability that system moves from state s_i to state s_j when action a is performed. Θ stands for a finite set of observations where θ_i denotes the observation of project i. B is the observation model in which $b_{j\theta}^a$ denotes the probability that Θ was observed when the system state is s_j and last action taken at time $k-1$ is a. Individual observation probability $b_{j\theta}^a$ is defined as $b_{j\theta}^a = Pr(\theta^k = u | S^k = j, A^{k-1} = a)$, and the observation is dependent on the resulting state in the state transition. In POMDP, R is defined and it stands for the immediate reward. $r_{ij\theta}^a$ denotes the immediate reward received at time k for performing action a at time $k-1$, and the system state moves from s_i at time $k-1$ to state s_j at time k, and the observation is θ at time k. Like the

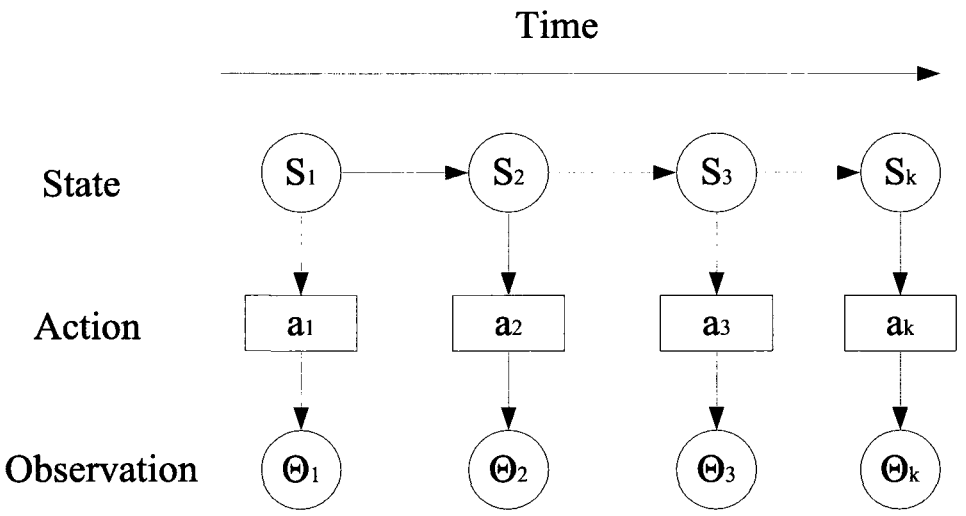

Figure 3.2: Partially Observable Markov Decision Process.

definition in MDP, we still use q_i^a to denote the expected immediate reward:

$$q_i^a = \sum_{j\theta} p_{ij}^a b_{j\theta}^a r_{ij\theta}^a. \tag{3.6}$$

The POMDP model is shown in Figure 3.3. At each system state, the observation is made after an action is taken. The system states cannot be observed directly at each time instant, and only observations which give hints of the system states can be acquired, k stands for time instant at which a decision needs to be made.

3.2.2 Information State

In MDP, the optimal policy is defined based on the history of actions and system states. The properties of MDP provide the solution to the problem of making decision by finding a simple mapping from states to actions. However, in POMDP, the states cannot be observed directly and making decision only based on the observations is

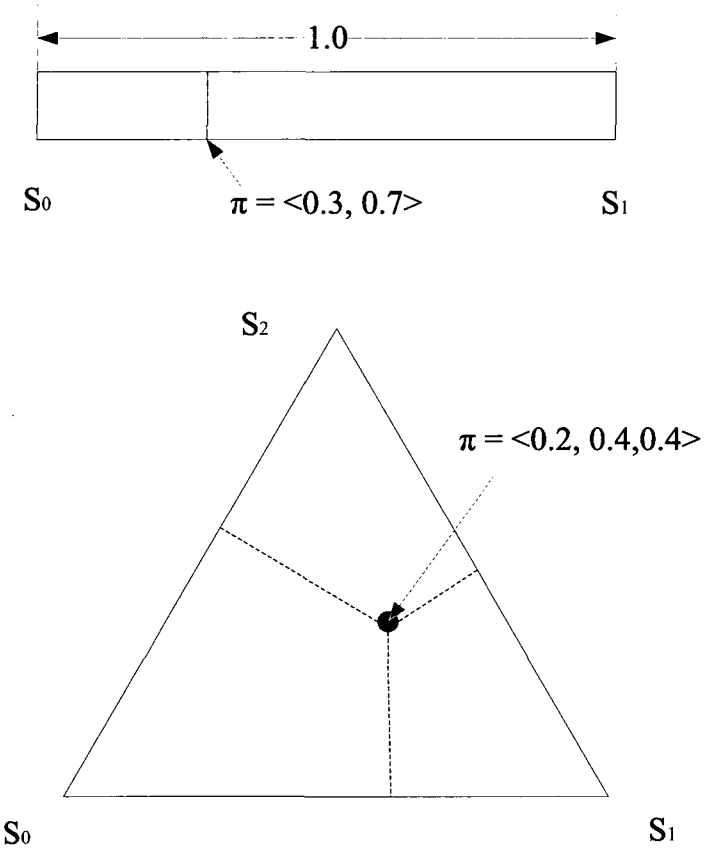

Figure 3.3: Information state of two and three states.

not possible. It is fortunate that we can derive a summary statistic for the entire history of a process with the definition of information state, which can be used for decision making.

We will refer to a probability distribution over states as information state and the entire probability space (the set of all possible probability distributions) as the information space. The information spaces with 2 and and 3 states are shown in Figure 3.3. The corners are the Markov chain states s_0, s_1, and s_2. For a system with 2 states, its information state is a one-dimension line, the thickness of the line is only used for clarifying the explanations. The distance from the left axis is the first component s_0 and the distance from the right is the second component s_1. In the

two state information state figure, the information state is $\langle 0.3, 0.7 \rangle$, which means the current system state is at s_0 with 30% probability, and with 70% probability at state s_1.

For the system with 3 states, the information state is a two-dimension triangle, the value of a point in the information state can be obtained by the perpendicular distance to the sides of the triangle. For example in the figure, current information state is $\pi = \langle 0.2, 0.4, 0.4 \rangle$, in which each element stands for the probability of the system stay at state i. Note the the sum of all the elements in π is 1.

An information state is a sufficient statistic for the history, which means that decision making can be based on the information state, denoted by a vector $\pi^k = (\pi_1^k, \pi_2^k, \ldots, \pi_s^k)'$, and $1_S' \pi^k = 1$, where k stands for the time instant, denotes the number of states, i denotes the probability that system is currently in state s_i, $1_S'$ represents an S-dimensional vector of ones. One important property of the information state is that it can be easily updated with *Bayes Rule* by incorporating one additional observation into the history:

$$\pi_j = \frac{\sum_i \pi_i' p_{ij}^a b_{j\theta}^a}{\sum_{ij} \pi_i' p_{ij}^a b_{j\theta}^a}, \tag{3.7}$$

where $b_{j\theta}^a$ stands for the observation probability when the system state changes from j to θ when action a is adopted. The new information state will be a vector of probabilities computed according to the above formula. The information states capture all the history information which is represented as $\{a^0, a^1, \ldots, a^{k-1}, \theta^1, \theta^2, \ldots, \theta^k, \}$, at time k. Therefore, we can save all the past actions and observations by constantly updating the information state, and it is reasonable to make decisions according to the information state.

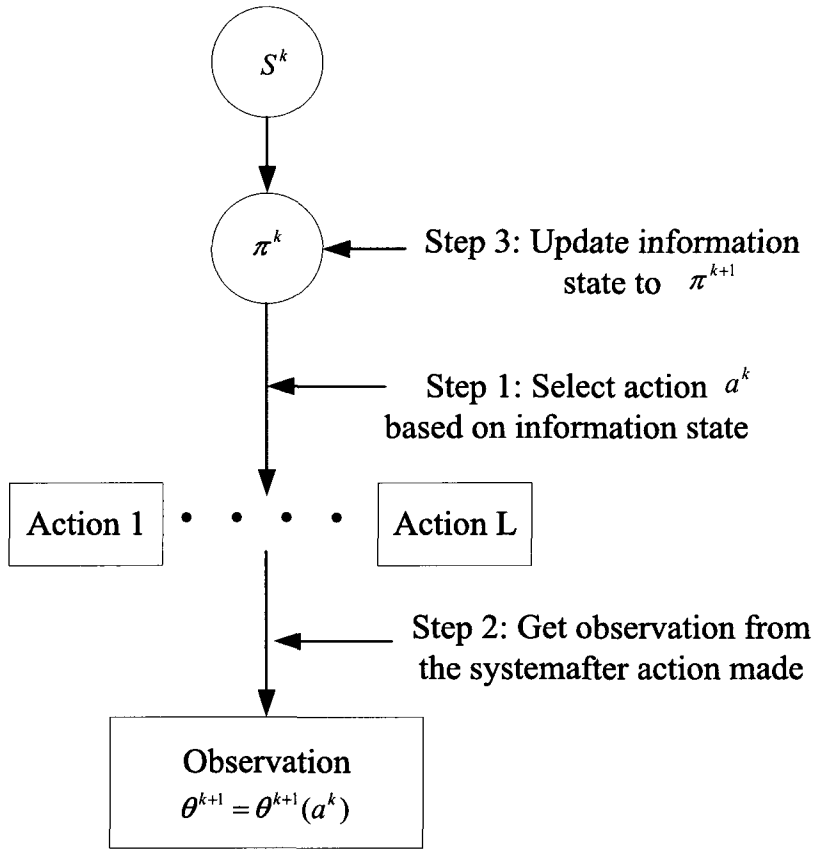

Figure 3.4: Information state update of POMDP.

In fact, we can convert a discrete POMDP problem into a continuous space CO-MDP problem. The action is generated as a function of information state and the system state is observed after the action is taken. The information state will be updated with the latest observation.

3.2.3 POMDP Policies and Value Function

In CO-MDP model, a policy is a mapping from states to actions. Since the state space and action space are finite sets, and the value function and policy are easy to calculate. But in POMDP, the underlying states cannot be observed directly, the continuous information state, i.e., the likelihood of being in each state is used instead to make decision. Our task is to compute a policy that obtains, based on the information state, the maximum expected reward for a single action. The POMDP policy can be derived from a value function which is defined over the entire information space.

For the k-horizon value function of POMDP, the value function can be defined as follows:

$$V_n^*(\pi) = \max_{a \in A} \left[\sum_i \pi_i \sum_j p_{ij}^a \sum_\theta b_{j\theta}^a (r_{ij}^a + \beta V_{n-1}^*(\pi_\theta^a)) \right], 0 \leq \beta \leq 1, \tag{3.8}$$

using the formulation 3.6 the equation can be simplified as:

$$V_n^*(\pi) = \max_{a \in A} \sum_i \pi_i q_i^a + \beta \left[\sum_i \pi_i p_{ij}^a b_{j\theta}^a V_{n-1}^*(\pi_\theta^a) \right], 0 \leq \beta \leq 1, \tag{3.9}$$

The POMDP value function is more complicated than value function of MDP, because the observation θ is a probabilistic distribution over the possible states.

To solve the POMDP is not easy using 3.9, fortunately, Sondik and Anthony have showed that the optimal finite horizon value function is *piecewise linear and convex* (PWLC) for any horizon K [36, 37]. This means for each horizon k, the value function

can be represented with a set of linear segments $\eta^k = \{\eta_k^1, \eta_k^2, \ldots, \eta_k^R\}$, and

$$V_k^*(\pi) = \max_{r \in R} \sum_i \pi_i \eta_{i,k}^r, \tag{3.10}$$

here $i \in \{1, 2, \ldots, S\}$ for S-state POMDP problem.

Theoretically, for the piecewise and linear POMDP value function, $V^*(\cdot)$ can be approximated with large enough horizon. In practise, non-linear value function can also be approximated with a piecewise linear function as closely as possible.

Despite the uncertainty about the value function of infinite POMDP, the $V^*(\cdot)$ is always convex [36].

There are several linear programming based algorithms in the POMDP literature [36–38]. They all use dynamic programming to solve the problem. The only difference among these algorithms is the ways to compute a single dynamic programming step. The detail information and correspondent programming codes of these algorithms can be found in [36, 39].

3.3 Summary

In this chapter we illustrated Markov decision process and partially observable Markov decision process. MDP can be defined by a simple quadruple $\langle S, A, P, R \rangle$, which is an action-reward system. The immediate system reward is only related to system states. The observation of system states is accurate in MDP. A value function is defined to solve the MDP problem. The objective is to find optimal policy which maximize long term system reward.

We also presented the POMDP definition and solution, which can be defined as a hex-tuple $\langle S, A, P, \Theta, B, R \rangle$ model. POMDP is different to MDP in the observation of system states. In POMDP, the system state can only be observed with some

probabilities; thus system states become a continuous distribution over states. By adopting the definition of information space, the continuous state POMDP problem can be converted into a discreet state MDP problem. Therefore, the theories and results used for MDP can be used for POMDP problems. The solution to POMDP can also be found by value function. The value function of POMDP is piecewise linear and convex (PWLC), and it is very useful because it allows the value function to be represented using finite resources. Finally, the common algorithms used to solve POMDP problems were introduced in this chapter.

Chapter 4

Hierarchical ID-Based Key Management Scheme

4.1 The Proposed Scheme

Although the aforementioned works [4–6, 34, 35] have been done for hierarchical key management in MANETs, most of them concentrate only on the network structures and key allocation. The system dynamics of the tactical MANETs is largely ignored. We propose a hierarchical key management scheme in this chapter. The proposed scheme focuses on the dynamic key update process, and nodes private keys can be updated by their parents acting as the PKG or a threshold of siblings acting as the PKG. Whenever a PKG is needed, our proposed scheme can select the best nodes to work as PKG while considering the security conditions and energy states of all available nodes, thus improve the network security and prolong the network lifetime. Most importantly, the scheme can be combined with any existing scheme without affecting their algorithms.

The system time in the scheme is divided into equal slots that correspond to the time intervals [32]. The length of time slot depends on the security requirements and system environment. In tactical environment where security is of the first priority,

the time slot can be very short. At each time interval, node states may change or stay at its original states with some probabilities.

4.1.1 Security Model

Assume each node $n(n \in 1, \ldots, N)$ in the network has a finite number of I_n states standing for the security conditions. The security state space $\mathcal{S}$ can be defined as: $\{safe, attacked, compromised\}$. The security state of the potential node n at the time instant $t(t \in 1, \ldots, T)$ is defined as d_n^t, and its state evolves according to an I_n -state Markov chain with one-step transition probability matrix:

$$A_n^a = (\phi_{ij})_{i,j \in I_n} = Pr(d_n^{t+1} = j | d_n^t = i), \tag{4.1}$$

where a stands for an action.

In our system there are two actions $\{0, 1\}$; action 1 means the node is selected or active and 0 means the node is not selected or passive. So A_n^1 is the transition probability matrix when the node is active and A_n^0 is the transition probability matrix when the node is passive.

The security condition d_n^t can be observed by intrusion detection systems (IDSs), which are popular in MANETs as the second wall of protection [40]. An IDS continuously or periodically monitors the node activities, compares them with stored profiles and maintain nodes states information. We assume the state observation by IDS is accurate, the system security states evolve as a Markov chain.

4.1.2 Energy Model

Most mobile devices are powered by batteries with limited energy, the energy should be consumed carefully and efficiently to maximize the network lifetime. The nodes

with high residual energy should be used to avoid over-utilizing of certain nodes.

The residual battery energy can be detected locally as a random variable e_n^t, which means the residual energy level of node n at time t. For simplification, the continuous battery residual energy can be divided into discrete levels, denoted by: $\mathcal{E} = (e_1, e_2, \ldots, e_h)$, where h is the number of available energy state levels. Inspired by [41], we model the transition of energy levels of nodes in MANETs as Markov chains with one-step transition probability matrix:

$$B_n^a = (\psi_{ij})_{i,j \in \mathcal{E}} = Pr(e_n^{t+1} = j | e_n^t = i). \tag{4.2}$$

4.1.3 Network Lifetime

The definition of lifetime $\mathcal{L}$ depends on the underlying network application, and one of the commonly used lifetime definitions is the number of dead nodes reaches a threshold D_{th} that the network can no longer achieve the targeted estimation performance [42, 43]. In our scheme, we also assume that the network lifetime terminates when there are k nodes are compromised.

4.1.4 Cost Model

The costs associated with node selection are defined as information leaking $c_l(d_n^t, a_n^t)$ and energy cost $c_e(d_n^t, a_n^t)$.

The information leakage $c_l(d_n^t, a_n^t)$ is due to that when a node is selected to work as PKG, its information emission can be monitored by adversaries. By using crypt-analysis, the node's private key can be compromised. The $a_n^t \in \{0, 1\}$ stands for the action adopted by node n at time t where 1 means the node is selected and 0 means the node is not selected. Note that when a node is passive, the information leakage is low or even 0.

At time t, the instantaneous cost incurred due to the selected node n is:

$$c_n^t = (1-\gamma)c_l(d_n^t, a_n^t) + \gamma c_e(e_n^t, a_n^t), \tag{4.3}$$

where $\gamma \in (0,1)$ is the weight factor of the two kinds of costs. Since there are M active nodes at time t, the cost of all the nodes for key update at time t is:

$$q(t) = \sum_{n=1}^{M} c_n^t, \tag{4.4}$$

where $n \in [1, \ldots, M]$ means all active nodes at time t. The total expected discounted cost of over infinite time horizon is given by:

$$Z(u) = Exp\left[\sum_{t=0}^{\infty} \beta^t q(t)\right], \tag{4.5}$$

where u denotes policy that is the history of all actions, Exp denotes mathematical expectation; $\beta \in (0,1)$ is the discount factor to ensure the expectation is bounded.

The optimization objective is to find the optimal policy u to minimize the information leaking and maximize the network lifetime, i.e., to determine the optimal policy u that minimizes the cost in (4.5).

4.2 Restless Bandit Formulation and Solution

In this section, we formulate the node selection problem as restless bandit problem. Restless bandit problem is a well studied framework where a decision-maker must dynamically schedule multiple projects to get the maximum reward [19, 44, 45]. We first introduce the system formulation, then we discuss the solutions to the restless bandit problem.

4.2.1 The Restless Bandit Problem

The restless bandit problem can be simply described as: There are N projects, of which M can be worked on at any time period. Project n is characterized at (discrete) time t by its state s_n^t, which belongs to a finite state space. If project n is worked on at time t, one receives a rewards $r(s_n^t)$. The state s_n^t then evolves to a new state according to given transition probabilities. The states of all idle projects are also evolved, possibly using different transition probabilities. (that is why it is called restless). The goal is to find a policy which decides at each time period which projects to work on in order to maximize the expected sum of the discounted rewards over an infinite horizon.

In our scheme, we use cost instead of reward, the optimization objective is to minimize the cost. The N projects in our scheme are all available nodes that can work as PKG. The number M is threshold in our scheme. The system state in our scheme includes both security state and energy conditions. The system formulation is as follows.

4.2.2 System Formulation

Node States

The state of nodes $n \in \{1, 2, \ldots, N\}$ in time slot $t \in \{0, 1, \ldots, T-1\}$ is modeled as:

$$s_n^t = [d_n^t, e_n^t], \tag{4.6}$$

where d_n^t is the security state and e_n^t is the energy state which are defined in system model. The state space of s_n^t is represented as $\mathscr{S}_n$ and $s_n^t \in \mathscr{S}_n$. The state s_n^t evolves

with one-step transition probability matrix:

$$P_n^a = [A_n^a \otimes B_n^a], \tag{4.7}$$

where A_n^a is security state transition matrix and B_n^a is energy state transition matrix. $\otimes$ denotes Kronecker product which is used here to expand the transition matrices. Note that system security state is independent of the energy conditions, that is why we can use $\otimes$ to expand the states.

Policies

We denote by $\mathscr{U}$ the class of all admissible policies. The admissible policy $u \in \mathscr{U}$ is a $T \times N$ matrix, whose element of the tth row and the nth column is a_n^t, which representing the action taken by node n in time slot t. u satisfies:

$$u \times \underbrace{(1, 1, \ldots, 1)}_{N}{}' = M \times \underbrace{(1, 1, \ldots, 1)}_{T}{}', \tag{4.8}$$

which means in each time slot, the number of active nodes is equal to M. u can be considered as the "active policy", which determines the "passive policy" that is defined as

$$\widetilde{u} = -(u - U). \tag{4.9}$$

where U is a $T \times N$ matrix with all elements equal to 1.

Costs

An instantaneous cost $c(s_n^t, a_n^t)$ is accrued in each time slot t for the node n in state s_n^t and takes action (become active). Define the cost vector as

$$\vec{C}(t) = (c(s_1^t, a_1^t), c(s_2^t, a_2^t), \ldots, c(s_M^t, a_M^t)'. \tag{4.10}$$

When a node is in safe state, the cost of taking action is lower; while a node is in compromised state, the cost of taking action is higher.

The total expected discounted cost over the time horizon is defined in (4.5), written in vector form. The total expected discounted reward over the time horizon is:

$$Z(u) = \frac{(\beta^0, \beta^1, \ldots, \beta^{T-1})\vec{C}(t)}{T}, \tag{4.11}$$

The optimization objective is:

$$Z^* = \min_{u \in \mathscr{U}} Z(u). \tag{4.12}$$

The optimal policy u^* is the policy that achieves the optimization objective, i.e., minimal cost. According to (4.12) the optimal policy is

$$u^* = \arg\min_{u \in \mathscr{U}} Z(u). \tag{4.13}$$

Priority Index

The Priority Index for potential node n with state s_n^t at time t is represented as δ_{k_n}. The optimal policy has an index rule: The M nodes with the smallest indices in a given time slot t act as the active nodes. That is, assuming $\{\delta_{k_1}, \delta_{k_2}, \ldots, \delta_{k_M}\}$ to be the set of indices arranged from the smallest value to the largest value in time slot t,

the node n's action should be

$$a_n^t = \begin{cases} 1, \text{ if } n \in \{k_1, k_2, \ldots, k_M\}, \\ 0, \text{ otherwise.} \end{cases} \tag{4.14}$$

Thus, to solve the node selection problem, computing the priority indices is the key step, which is described later.

4.2.3 Solving the Restless Bandit Problem by LP Relaxation

In this subsection, to solve the restless bandit problem, a hierarchy of increasingly stronger LP relaxations [16] is developed based on the result of LP formulations of Markov decision chains (MDCs).

LP Formulation

To formulate the problem, we first introduce

$$I_j^a(t) = \begin{cases} 1, \text{ if action } a \text{ is taken at time } t \text{ in state } j, \\ 0, \text{ otherwise.} \end{cases} \tag{4.15}$$

With $I_j^a(t)$, let

$$x_j^a(u) = E_u \left[\sum_{t=0}^{T-1} I_j^a(t) \beta^t \right] \tag{4.16}$$

represent the total discounted time that action a is taken in state j under policy u. The state-action space is denote by $\mathscr{D} = \{(i, a) : i \in \mathscr{S}, a \in \mathscr{A}\}$. Consequently,

(4.12) can be translated into:

$$Z^* = \min_{u \in \mathscr{U}} \sum_{(i,a) \in \mathscr{D}} c_i^a x_i^a(u). \tag{4.17}$$

Let's introduce the performance vector $\boldsymbol{x}(u) = (x_j^a(u))_{j \in \mathscr{S}, u \in \mathscr{U}}$ under all $u \in \mathscr{U}$. We can rewrite (4.17) as:

$$Z^* = \min_{\boldsymbol{x} \in X} \sum_{(i,a) \in \mathscr{D}} c_i^a x_i^a, \tag{4.18}$$

where $X = \{\boldsymbol{x}(u), u \in \mathscr{U}\}$.

Let α_i represent the probability that the initial state is $i \in \mathscr{S}$, thus the initial state probability vector $\boldsymbol{\alpha} = (\alpha_i)_{i \in \mathscr{S}}$ is given. We denote by $\mathscr{P}$ the polyhedron:

$$\mathscr{P} = \left\{ \boldsymbol{x} \in \mathfrak{R}_+^{|\mathscr{D}|} : \sum_{a \in \mathscr{A}} x_j^a = \alpha_j + \beta \sum_{(i,a) \in \mathscr{D}} p_{ij}^a x_i^a, j \in \mathscr{S} \right\}, \tag{4.19}$$

which is a bounded polyhedron, or polytope [16]. It is proved that $X \subseteq \mathscr{P}$, and if $\boldsymbol{\alpha} > 0$, $\mathscr{P}$ coincides precisely with performance region X [46]. Authors of [16] further strengthened that, in this problem, polytope $\mathscr{P}$ always coincides with X regardless whether $\boldsymbol{\alpha} > 0$, and the vertices of $\mathscr{P}$ are achievable by stationary deterministic policies.

We decompose (4.16) for two admissible actions:

$$x_{i_n}^1(u) = E_u \left[\sum_{t=0}^{T-1} I_{i_n}^1(t) \beta^t \right], \tag{4.20}$$

and

$$x_{i_n}^0(u) = E_u \left[\sum_{t=0}^{T-1} I_{i_n}^0(t) \beta^t \right], \tag{4.21}$$

where

$$I_{i_n}^1(t) = \begin{cases} 1, \text{ if node } n \text{ is active at time } t \text{ in state } i_n, \\ 0, \text{ otherwise,} \end{cases} \tag{4.22}$$

and

$$I_{i_n}^0(t) = \begin{cases} 1, \text{ if node } n \text{ is passive at time } t \text{ in state } i_n, \\ 0, \text{ otherwise.} \end{cases} \tag{4.23}$$

Thus the restless bandit problem can be formulated as the following linear program (LP):

$$Z^* = \min_{\boldsymbol{x} \in X} \sum_{n \in \mathcal{N}} \sum_{i_n \in \mathcal{S}_n} \sum_{a_n \in \{0,1\}} c_{i_n}^{a_n} x_{i_n}^{a_n}, \tag{4.24}$$

where $X = \{\boldsymbol{x} = (x_{i_n}^{a_n}(u))_{i_n \in \mathcal{S}_n, a_n \in \{0,1\}, n \in \mathcal{N}} \,|u \in \mathcal{U}\}$.

The approach to solve this problem is to construct relaxations of polytope X that yield polynomial-size relaxations of linear program. Denote by $\widehat{X} \supseteq X$ the relaxations, not on the space of the original variables x_i^a, but in a higher-dimensional space that includes new auxiliary variables [16].

First-Order LP Relaxation

Define the polytope $\mathcal{Q}_n^1 = \{\boldsymbol{x}_n = (x_{i_n}^{a_n}(u))_{i_n \in \mathcal{S}_n, a_n \in \mathcal{A}^1} \,|u \in \mathcal{U}\}$, where $\mathcal{A}^1 = \{0, 1\}$ is the action space of the first-order MDC which can be induced by the restless bandit problem over each potential active node n. $\mathcal{Q}_n^1$ is precisely the projection of restless bandit polytope $\mathcal{P}$ over the space of the variable $x_{i_n}^{a_n}$ for node n. A complete formulation of $\mathcal{Q}_n^1$ is given by [16]:

$$\mathcal{Q}_n^1 = \Bigg\{ \boldsymbol{x} \in \Re_+^{|\mathcal{S}_n \times \{0,1\}|} | x_{j_n}^0 + x_{j_n}^1 = \alpha_{j_n} + \beta \sum_{i_n \in \mathcal{S}_n} \sum_{a_n \in \{0,1\}} p_{i_n j_n}^{a_n} x_{i_n}^{a_n}, j_n \in \mathcal{S}_n. \Bigg\}. \tag{4.25}$$

Now the first-order relaxation can be formulated as the linear program:

$$\begin{aligned} & Z^1 = \min \sum_{n \in \mathcal{N}} \sum_{i_n \in \mathcal{S}_n} \sum_{a_n \in \{0,1\}} c_{i_n}^{a_n} x_{i_n}^{a_n} \\ & \text{subject to} \\ & \boldsymbol{x}_n \in Q_n^1, n \in \mathcal{N}, \\ & \sum_{n \in \mathcal{N}} \sum_{i_n \in \mathcal{S}_n} x_{i_n}^1 = \frac{M}{1-\beta}. \end{aligned} \tag{4.26}$$

There are $O(N\,|\mathcal{S}_{\max}|)$ variables and constraints of this linear program, where $|\mathcal{S}_{\max}| = \max_{n \in \mathcal{N}} |\mathcal{S}_n|$, with the size polynomial in the problem dimensions.

4.2.4 Primal-Dual Priority-Index Heuristic

In this subsection, a heuristic for the restless bandit problem that uses the information contained in optimal primal and dual solutions to the first-order relaxation (4.26) is presented. The primal-dual heuristic is interpreted as a priority-index heuristic as well. The dual of (4.26) is

$$\begin{aligned} & D^1 = \min \sum_{n \in \mathcal{N}} \sum_{j_n \in \mathcal{S}_n} \alpha_{j_n} \lambda_{j_n} + \frac{M}{1-\beta} \lambda, \\ & \text{subject to} \\ & \lambda_{i_n} - \beta \sum_{j_n \in \mathcal{S}_n} p_{i_n j_n}^0 \lambda_{j_n} \geq c_{i_n}^0, i_n \in \mathcal{S}_n, n \in \mathcal{N}, \\ & \lambda_{i_n} - \beta \sum_{j_n \in \mathcal{S}_n} p_{i_n j_n}^1 \lambda_{j_n} \geq c_{i_n}^1, i_n \in \mathcal{S}_n, n \in \mathcal{N}, \end{aligned}$$

$$\lambda \geq 0. \tag{4.27}$$

We denote by $\{\overline{x}_{i_n}^{a_n}\}$ and $\{\overline{\lambda}_{i_n}, \overline{\lambda}\}$ the optimal primal and dual solution pair to the first-order relaxation (4.26) and its dual (4.27). Let $\{\overline{\gamma}_{i_n}^{a_n}\}$ represent the corresponding optimal reduced cost coefficients:

$$\begin{aligned}
\overline{\gamma}_{i_n}^0 &= \overline{\lambda}_{i_n} - \beta \sum_{j_n \in \mathscr{S}_n} p_{i_n j_n}^0 \overline{\lambda}_{j_n} - c_{i_n}^0, \\
\overline{\gamma}_{i_n}^1 &= \overline{\lambda}_{i_n} - \beta \sum_{j_n \in \mathscr{S}_n} p_{i_n j_n}^1 \overline{\lambda}_{j_n} - c_{i_n}^1,
\end{aligned} \tag{4.28}$$

which must be non-negative. Furthermore, $\overline{\gamma}_{i_n}^0$ and $\overline{\gamma}_{i_n}^1$ can be interpreted as the rates of decrease in the objective-value of linear program (4.26) per unit increase in the value of the variable $x_{i_n}^0$ and $x_{i_n}^1$, respectively.

We define a directed graph from the transition probabilities for each potential authentication node $n \in \mathscr{N}$: $G_n = (\mathscr{S}_n, A_n)$, where $A_n = \{(i_n, j_n) | p_{i_n j_n}^0 > 0$, and $p_{i_n j_n}^0 > 0, i_n j_n \in \mathscr{S}_n\}$. Thus under the mixing assumption that G_n is connected for every n, every extreme point $\overline{x}$ of polytope $\mathscr{P}^1$ has the following properties [16]:

1. There are at most one node l and one state $i_l \in \mathscr{S}_l$ for which $\overline{x}_{i_l}^1 > 0$ and $\overline{x}_{i_l}^0 > 0$.

2. For all other nodes n and all other states either $\overline{x}_{i_n}^1 > 0$ or $\overline{x}_{i_n}^0 > 0$.

Based on the cost coefficients computed in (4.28), the index of the sender n in state i_n is defined as:

$$\delta_{i_n} = \overline{\gamma}_{i_n}^1 - \overline{\gamma}_{i_n}^0. \tag{4.29}$$

The priority-index rule is to select the M nodes that have the smallest indices to be active. In case of ties, set active node with $\overline{x}_{i_n}^1 > 0$.

4.3 Key Update Process of Proposed Scheme

In this section, we describe the key update process of proposed scheme. To further decrease the computational complexity, the key update process can be divided into off-line part and on-line part.

4.3.1 Off-line Priority Index Computation

During the off-line part, priority indices are computed from (4.29). The inputs are nodes states, transition matrix and corresponding cost matrix of all available nodes. The priority indices are computed and saved as an index table. In online part of our scheme, the priority index table will be used to select the best nodes based on nodes' instantaneous states.

4.3.2 Online Key Update Process

Fig 4.1 illustrates the key update process when a node wants to join an existing MANET:

1. When node $\mu_1\nu_1$ in network 2 at level L want to join network 1, it sends a message to a node $\alpha_2\beta_3$ in network 1.

2. Node $\alpha_2\beta_3$ in network 2 relays the message to the IDS.

3. The IDS performs a priority index table lookup to find the best node/nodes based on current states of all available nodes. In figure 4.1 we assume node $\alpha_2\beta_1$ and $\alpha_2\beta_2$ are selected. IDS then sends messages to the selected nodes to request for construction of the PKG.

4. The selected nodes $\alpha_2\beta_3$ and $\alpha_2\beta_2$ construct a temporary PKG and generate a private key for node $\mu_1\nu_1$.

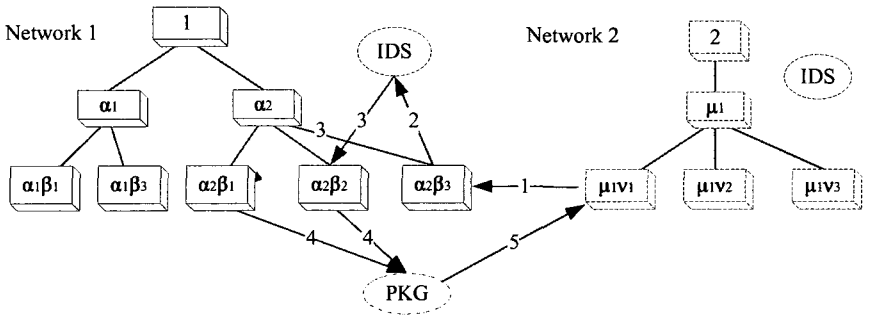

Figure 4.1: Key update process of the proposed scheme.

5. The private key is transmitted to node $\mu_1\nu_1$, and node $\mu_1\nu_1$ joins the network successfully.

Note that node states are monitored and maintained by IDS at fixed time interval. The length of time interval depends on the system security requirement.

4.3.3 Remarks on the Proposed Scheme

The proposed scheme is a supplement to existing schemes. It does not affect existing network structure and communication protocals. Because the priority index of a node depends only on node properties such as transition probability, cost matrix and current states, the proposed node selection scheme is fully distributed. Nodes can join and leave from MANETs freely in the proposed scheme.

In MANETs with heterogeneous nodes, some nodes are more secure, computationally more powerful, or more trustworthy. In our formulation, the node dynamics allow for them to have different security states, state transition probabilities, which allow our proposed scheme to be suited for MANETs with heterogeneous nodes.

In online part of the proposed scheme, the IDS is responsible for keeping the priority indices and making node selection decisions, which may involve significant

computation and traffic to the IDS. We envisage that a Hierarchical Intrusion Detection Systems (HIDE) [47–49] could be used in large hierarchical MANETs, in which multiple IDSs cooperate to monitor and maintain the network security. In large network with HIDE, node selection will be processed by the IDS who is responsible for the domain. Computational and traffic loads can be alleviated.

One might argue that the IDS, since it is performs a monitoring function for the network, presents risk as a single-point of failure similar to a centralized trust authority. However, a significant discriminator between the two is that the IDS does not hold any keys.

On the other hand, one may also consider a network without an IDS. Under that circumstances, nodes make decisions locally without the interferences of the IDS. However that scheme brings much traffic ($O(N^2)$) to the network since nodes need to communicate with each others to learn their states, moreover, compromised nodes may send bogus information to other nodes trying to get the right to work as PKG. Therefore, leaving the node selection be proceeded in IDS is more secure and reliable.

Another non-trivial question for the proposed scheme is the setup of the nodes transition matrices and cost matrices. We suppose that most node types have predefined properties, that is realistic since in most cases we already know the nodes properties when deploying the network. However it may not be realistic to know all the nodes properties when the proposed scheme is used in dynamic environment where heterogeneous nodes may join the network. Under that circumstances, we can use the IDS to predict the node properties from the history of actions and observations. The prediction of IDS may not be accurate, the system becomes a partially observable Markov decision process, which is left for future research. Clearly, the further the system evolves in real-time, the greater the likelihood that the information contained within the on-line accessible index table has become dated and the less optimal the

node selection process becomes. A mitigating strategy could be to feed updated system information back to an off-line system for recalculation and redistribution. It is worth noting that a completely outdated index table means the decision-making process is no better than random, which is no worse than existing schemes.

4.4 Summary

In this chapter we illustrated the system model of the proposed hierarchical key management scheme. The proposed scheme is based on restless bandit formulation, which is a powerful frame work in modeling stochastic decision systems.

The proposed scheme is designed to improve the security in key management and the network lifetime. The system states is a combination of security states and energy conditions. System cost is based on the information leakage and energy states. To mitigate the computational complexity, a primal dual heuristics is used to solve the restless bandit problem. Also, the optimal node selection policy can be divided to off-line and online parts. In the off-line part, the priority indices are computed and saved; which will be used for online node selection process. The computation complexity is decreased dramatically. We also discussed some issues of the proposed scheme.

Chapter 5

Distributed Multimodal Biometric Authentication

In this chapter, we formulate the whole system of the multimodal biometric authentication as a partially observable Markov decision process [20].

In this scheme, each device is equipped with several biosensors which are used for continuous user re-authentication. The frequency of applying re-authentication depends on the severity of the environment, system security requirements and system energy conditions. Our scheme can automatically decide which biosensor will be used at each time instant. The optimal policy can be acquired by solving POMDP with dynamic programming-based hidden Markov model scheduling algorithms.

5.1 System Model

In tactical MANETs, if the user-to-device authentication fails, it means that the device is not in right hand and appropriate measures can be taken accordingly. Under that consumption, the time axis is divided into slots of equal duration $[0, 1, \ldots, T]$ which corresponds to the time interval between two continuous authentications.

Our biosensor selecting problem is assumed to evolve as a S-state POMDP. Under

this model, the state of the device evolves according to a discrete time, S-state first order Markov chain $\{X^k\}$, where k denotes the authentication time instant. Assuming the node or device works in military environment and have N biosensors, for example, iris, fingerprint biosensors are equipped. The state of the node at time t is $x(t)$, which evolves with time as discrete Markov chain. The state $x(t)$ includes the device security and energy state, which will be described later. We also assume that during the field operation of the node, a host-based IDS continuously or periodically monitor the node security states. The observation from IDS is not accurate, thus the system is a POMDP.

5.1.1 Security Model

Like the node security state in the hierarchical key management scheme, the security state $d(t)$ represents current security conditions of the device. In this scheme, the security state space can have I states such as $\{safe, attacked, compromised\}$. The security state is observed by host-based IDS and every time when a biosensor n is used, the security state $d(t)$ evolves according to an I state Markov chain with one-step transition probability matrix:

$$A_n^a = (\phi_{ij})_{i,j \in I} = Pr(d(t+1) = j | d(t) = i), \tag{5.1}$$

where a stands for an action. In our system there are two actions $\{0, 1\}$; action 1 means the biosensor is selected or active, and 0 means the biosensor is not selected or passive. The security condition $d(t)$ is observed by IDS and we assume the state observation by IDS is not accurate, the system security state evolves as a Markov chain.

5.1.2 Energy Model

Considering that most mobile devices are powered by batteries with limited energy, the energy should be used carefully. The biosensor with lower energy consumption should be used when the system is secure; high cost biosensor such as iris sensor should be used when the system is in negative state.

Like the hierarchical key scheme, the residual battery energy of the device can be detected locally as $e(t)$. The battery residual energy can be divided into discrete levels, denoted by $\mathcal{E} = (e_1, e_2, \ldots, e_h)$, where e_i is the residual energy and there are totally h levels in energy. Note that there is only one device in this model, the energy is the level of the single device (in hierarchical key management scheme, each node has its energy level). As defined in [41], we model the transition of energy levels of nodes in MANETs as a Markov chain with one-step transition probability matrix:

$$B_n^a = (\psi_{ij})_{i,j\in\mathcal{E}} = Pr(e(t+1) = j | e(t) = i). \tag{5.2}$$

System State

The system state $x(t)$ includes security condition and energy state, which is defined as $[d(t), e(t)]$. Thus the system state transit with probability matrix:

$$P_n^a = [A_n^a \otimes B_n^a], \tag{5.3}$$

where $\otimes$ denotes the Kronecker product.

Observation Model

The state of the device is partially observed by the IDS. Since we can not observe the server state accurately, the observation is a probabilistic distribution over the server

states.

Define the observation of security state as O_s^l and the observation of energy state as O_e^l, the device observation matrix is $O^n = [O_s^l \otimes O_e^l]$. Assume that there is a finite M_l observation set indexed by $m(l) = 1, 2, \ldots, M_l$. Denote $Y^k = (y^1(a^0), \ldots, y^k(a^{k-1}))$ as the observation history for time instant k. Let $O(l) = (b_{im}(l))_{i \in N_l, m \in M_l}$ denote the observation probability matrix of the HMM, where each element $O(l) = Pr(y^{k+1}(l) = m | x^{k+1}(l) = i, a^t = l)$, in which $a^t \in \{1, 2, \ldots, L\}$ denotes that biosensor l is the active sender at time instant t. A finite M_l observation set indexed by $m(l) = 1, 2, \ldots, M_l$ is assumed.

5.1.3 Cost Model

When a biosensor is selected for authentication, two kinds of cost are considered in our scheme. The first is security related, since the biometric information can be detected by adversaries when it is used, the information leakage is defined as $c_l(d(t), a_n^t)$, which is a function of the node states and the action adopted. The second cost is energy related, we define energy cost $c_e(e(t), a_n^t)$. The energy cost stands for the energy consumption of the biosensor usage. At time t, the instantaneous cost incurred due to the usage of biosensor n is:

$$C_n^t = (1 - \gamma) c_l(d(t), a_n^t) + \gamma c_e(e(t), a_n^t), \tag{5.4}$$

where $\gamma \in (0, 1)$ is the weight factor for the two kinds of costs. The total expected discounted cost of over infinite time horizon is given by:

$$Z(u) = Exp\left[\sum_{t=0}^{\infty} \beta^t C(t)\right], \tag{5.5}$$

where u is the policy history and Exp denotes mathematical expectation; $\beta \in (0,1)$ is the discount factor to ensure the expectation is bounded. Note that when no biosensor is not selected, the cost is 0 since we assume their is no information leakage and energy consumption if no biosensor used. The optimization objective is to find the optimal policy to minimize the cost in (5.5).

5.2 Solution to the Proposed Scheme

In this section, we provide solution to the proposed scheme. The system formulated as a POMDP with multi-armed bandit structure, we further introduce the Gittins index solution to the problem.

A POMDP is defined by: The system has a set S of states; a set A of actions; a set Z of observations; a transition function or matrix T, where $T(s,a,s')$ denotes the probability $Pr(s'|s,a)$ of transitioning to state s' when action a is taken at state s; an observation function Z, where $Z(s,a,z)$ denotes the probability $Pr(z|s,a)$ of making observation z in s after performing a; and a reward function R, where $R(s,a)$ denotes the immediate reward associated with state s and action a. The objective of POMDP is to maximize the reward R on infinite horizon POMDP with a discount factor.

5.2.1 System Formulation

Information State

Since the observation is not accurate, information state is used to describe the node state, which is a probabilistic distribution over states. Information state is derived from the observation history and is also a sufficient statistic for the history. Defined

the information state $x(t)$ to be:

$$x_i^t(l) = Pr(s^t(n) = i|Y^t, a^{t-1} = l), \tag{5.6}$$

in which Y^t is the observation of the system state and a^{t-1} is the action at $t-1$, which means biosensor l works on time instant $t-1$. The state space of information states $x_i(l)$ is defined by:

$$\chi(l) = \{x(l) \in \Re^{I_l} : 1_{I_l}^{'} x(l) = 1,\ 0 \leq x_i(l) \leq 1, \\ \text{for all } i \in 1, \ldots, I_l\}. \tag{5.7}$$

$x_i(l)$ is a $l-1$ dimension simplex. The information state $x^t(l)$ can be recursively updated by the HMM state filter that is known as forward algorithm with the new observation $y^{t+1}(l)$:

$$x^{t+1}(l) = \frac{O(l, y^{t+1}(l))A^{'}(l)x^t(l)}{1_{I_l}^{'} O(l, y^{t+1}(l))A^{'}(l)x^t(l)}, \tag{5.8}$$

Gittins Index

For each biosensor, there is a value $\gamma^k(l, x^t(l))$ called Gittins index, which is the function of biosensor l and its information state $x^t(l)$. That is, the policy has an index rule: The biosensor with the biggest Gittins indices at time instant t should be selected. Thus, the problem can be transformed to compute Gittins index, which decrease the computational complexity significantly.

5.2.2 Value Iteration Algorithm for Computing Gittins Index

The Gittins index can be solved with dynamic programming formulation. For convenience, we will make the object function (5.5) a reward function (which is simply the negative of a cost function). So maximizing the reward is equivalent to minimizing the cost.

For each node l, let a positive real number $M(l)$ for each potential node l denote a positive real number:

$$0 \leq M(l) \leq \bar{M}(l), \ \bar{M}(l) = \max_{i \in I_l} \frac{R(s^t(l) = i, a^t = l)}{1 - \beta}. \tag{5.9}$$

For simplification, we omit the l in $M(l)$ and $\bar{M}(l)$ and the superscript t in $x^t(l)$. Define the Gittins index of node l with information state $x(l)$ to be $\gamma(l, x(l)) = \min\{M : V_l(x(l), M) = M\}$, where $V_l(x(l), M)$ is the value function for node l and satisfies:

$$V_l(x(l), M) = \max \Bigg\{ M, R^{'}(l)x(l) + \beta \sum_{m=1}^{M_l} V_l \left(\frac{O(l,m)A^{'}(l)x(l)}{1_{I_l}^{'} O(l,m)A^{'}(l)x(l)}, M \right) 1_{I_l}^{'} O(l,m)A^{'}(l)x(l) \Bigg\}, \tag{5.10}$$

where M denotes the parameterized retirement reward.

For the finite time horizon T, the value iteration algorithm $t = 0, \ldots, T-1$:

$$V_l^{t+1}(x(l), M) = \max \Bigg\{ M, R^{'}(l)x(l)\beta \sum_{m=1}^{M_l} V_l^t + \left(\frac{O(l,m)A^{'}(l)x(l)}{1_{I_l}^{'} O(l,m)A^{'}(l)x(l)}, M \right) 1_{I_l}^{'} O(l,m)A^{'}(l)x(l) \Bigg\}, \tag{5.11}$$

where $V^T(x(l), M)$ is the value function of an $T-$horizon dynamic programming

recursion. We denote $\gamma^T(l, x(l))$ to be the approximate Gittins index computed via (5.11), i.e.,

$$\gamma^T(l, x(l)) = \min\{M : V_l^T(x(l), M) = M\}.$$

The finite horizon Gittins index can be arbitrarily accurate by choosing the horizon T large enough [50].

The value iteration recursion (5.11) does not translate into practical solution methodologies. The problem with (5.11) is that at each iteration t, one needs to compute $V_l^T(x(l), M)$ over an uncountably infinite set. However, under a different coordinate basis, $V_l^T(x(l), M)$ can be expressed as a standard POMDP, whose value function is known to be piecewise linear and convex, thus the Gittins index can be computed with these piecewise linear segments. A fictitious retirement information state can be added in order to find a solution of computing Gittins index. Once the information state reaches this value, it remains there for all time and accruing no reward. The $(T_l + 1)$ dimensional augmented information state is defined as $\bar{x} \in \{[x', 0]', [0'_{T_l}, 1]'\}$, where $x \in \chi(l)$, and $\bar{x}^t = [0'_{T_l}, 1]'$ denotes the retirement information state. Define the augmented observation process as $y^t \in \{1, \ldots, M_l + 1\}$, and the observation $M_l + 1$ corresponds to a fictitious observation that causes the information state jumps into the retirement state. Coordinate transformation is used to construct a standard POMDP. The value function in the new POMDP, $\bar{V}^t(l, \pi(l))$, has several characteristics as shown in [50]:

1. The value function in the new POMDP is equal to the value function defined in (5.11).

2. The value function $\bar{V}^t(l, \pi(l))$ is piecewise linear and convex and has the finite

dimensional representation $\bar{V}^t(l,\pi(l)) = \max_{\lambda_i^t \in \Lambda^t(l)} (\lambda_i^t)'\pi(l)$, where each $2(T_l+1)$-dimension vector λ_i^t is of the form

$$\lambda_i^t = \begin{pmatrix} (\lambda_i^t(1))' & 0 & (\lambda_i^t(3))' & 0 \end{pmatrix}', \tag{5.12}$$

where $\lambda_i^t(1), \lambda_i^t(3) \in \Re^{T_l}$. There always exists a unique vector in $\Lambda^t(l)$ which is denoted by $\lambda_1^t = \begin{pmatrix} \bar{M}1'_{T_l} & 0'_{T_l+2} \end{pmatrix}'$ with optimal control $v^t = 2$, if all the elements of $R(l)$ are not equal, else if all the elements of $R(l)$ are equal, then $\Lambda^t(l)$ compromises of a single vector $\lambda_1^t = \begin{pmatrix} \bar{M}1'_{T_l} & 0 & \bar{M}1'_{T_l} & 0 \end{pmatrix}'$.

3. The Gittins index $\gamma^T(l,x(l))$ for the information state $x(l) \in \chi(l)$ of node l is given by the finite dimensional representation

$$\gamma^T(l,x(l)) = \max_{\lambda_i^T \in \Lambda^T} \frac{\bar{M}(\lambda_i^T(3))'x(l)}{(\lambda_i^T(3) - \lambda_i^T(1))'x(l) + \bar{M}}. \tag{5.13}$$

Thus, using the value function of the standard POMDP, we can calculate the Gittins index of biosensor l.

5.2.3 Optimal Algorithm

As described in Chapter 3, there are several algorithms to solve finite horizon POMDP such as Sondik's algorithm [37], incremental pruning, Cheng's linear support algorithm, the witness algorithm and etc. The detail explanation and correspondent programming codes of these algorithms are presented in [39]. Each algorithm has the same basic framework and the only difference is the ways to compute a single dynamic programming step. The code of incremental pruning algorithm from [39] will be used in our simulation examples. The desired solutions to POMDP are represented by a

set of vectors, together with the optimal actions.

5.2.4 Distributed Multimodal Biometric Sensor Scheduling Process

Like the hierarchical key management scheme, t reduce the computational complexity of the scheme, the distributed multimodal biometrics authentication process can also be divided into off-line part and online part.

a) Off-line Computation of Policy Vector

For each biosensor $l = 1, 2, \ldots, n$, input: $T(l)$ {Transition probability matrix}, $O(l)$ {Observation probability matrix}, $R(l)$ {Reward vector}, $x^0(l)$ {A initial state estimate at time 0}, T {Horizon length}, and β {Discount factor}, then off-line compute finite set of vectors $\Lambda^T(l)$. At time $t = 0$, compute $\gamma^T(l, x^0(l))$ according to equation (5.13).

b) Real-time Sensor Selection over Horizon T

At any time instant t, each potential biosensor stores the n-dimensional vector γ), which is the vector of Gittins indices of the n nodes, arranged in descending order, i.e., $\gamma = (\gamma(1, x^t(1)), \gamma(2, x^t(2)), \ldots, \gamma(n, x^t(n)))$. Then the real-time steps for biosensor selection are:

1. Select the biosensor l with the highest Gittins indices.

2. The selected biosensor will be used for authentication, the sensor state $y^{t+1}(l)$ is observed from the IDS.

3. Update the information state of the selected biosensor using the HMM filter (5.8).

4. Compute Gittins index $\gamma^T(l, x^{t+1}(l))$ of biosensor l according to equation (5.13).

5. Keep the Gittins index unchanged $\gamma^T(q, x^{t+1}(q)) = \gamma^T(q, x^t(q))$ for the other biosensors.

6. Node l will broadcast $\gamma^T(l, x^{t+1}(l))$ to other potential biosensors.

7. On receiving the message, the other biosensors get their Gittins indices updated.

5.3 Remarks on the Proposed Scheme

In the multi-armed bandit formulation, the states of passive nodes are assumed not changed, which is different to restless bandit problem. In restless bandit formulation, passive projects also change their states.

Another difference of the two schemes is that the observation of the multi-armed bandit formulation is not 100% accurate. Therefore the observation is a continuous probabilistic distribution over states. The computation of reward over a continuous state space is more complicated than that over discrete state space.

5.4 Summary

User authentication can be used to lock and unlock devices. Multimodal biometric authentication provides a good solution for user authentication but it usually have high cost in energy consumption. We proposed a distributed scheme to optimally schedule multiple biometric sensors. A highlight of the scheme is that IDS is combined with multimodal biometric sensors and jointly works for continuous authentication.

There is information leakage connected to each biometric device since the authentication process will emit some information. The threat level will decrease if the node takes some counter measures to protect itself. The IDS in MANETs provides the measurement of the threat level for the device. Since the observation of system states

is not accurate, the system is formulated as a partially observed Markov decision process.

In order to minimize the long term cost including information leakage and energy, we used multi-armed bandits solution to derive an optimal policy and decide which biosensor will be used for authentication at a certain time. The biosensor with the highest Gittins index will be used. The off-line and online algorithms are also described in this chapter. Some numerical examples will be given in Chapter VI.

Chapter 6

Simulation Results and Discussions

6.1 Simulation Results and Discussions about the Hierarchical Key Management Scheme

In this section, we illustrate the performance of the proposed scheme by simulation examples.

To get a feel for the time and feasibility for key update in a tactical network, we set up a simulation scenario with one parent node and five heterogeneous child nodes, each with different transition probabilities, states, and cost matrices. $(2, 5)$-threshold secret sharing is used in this simulations.

We compare the cost of the proposed scheme with existing scheme [28], in which nodes are selected randomly without considering the security situation of the MANETs. The cost of the proposed scheme without a parent node is also compared with existing scheme. We further consider system performance when nodes have different transition probabilities. We also test performance with more nodes and different thresholds.

For simplicity, we use two security states: $safe$ and *compromised* at first, and three energy states: $high(b1), middle(b2), low(b3)$, so totally there are six states:

$sb1, sb2, sb3, cb1, cb2, cb3$. The security state transition probability matrices of these nodes when they are active are set as follows:

$$A_1^1 = \begin{pmatrix} 0.94 & 0.06 \\ 0.05 & 0.95 \end{pmatrix}, A_2^1 = \begin{pmatrix} 0.97 & 0.03 \\ 0.05 & 0.95 \end{pmatrix},$$

$$A_3^1 = \begin{pmatrix} 0.92 & 0.08 \\ 0.03 & 0.97 \end{pmatrix}, A_4^1 = \begin{pmatrix} 0.91 & 0.09 \\ 0.03 & 0.97 \end{pmatrix},$$

$$A_5^1 = \begin{pmatrix} 0.94 & 0.06 \\ 0.02 & 0.98 \end{pmatrix}, A_6^1 = \begin{pmatrix} 0.999 & 0.001 \\ 0.001 & 0.999 \end{pmatrix}, \tag{6.1}$$

The passive transition probability matrices are defined as:

$$A_i^0 = \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix}$$

, for $i = (1, \ldots, 5)$, and

$$A_6^0 = \begin{pmatrix} 0.999 & 0.001 \\ 0.001 & 0.999 \end{pmatrix},$$

for node 6.

Transition probability matrix stands for the probability changes from one state to another state. For example node 1 could be compromised with probability 0.06, and it could be snatched back from the compromised state to the safe state with probability 0.05. Node 6 is a parent node and has high transition probability 0.999 that means it is more stable than lower level nodes. We also assume that when a node is not selected, the transition probability is lower than when the node is selected.

The energy transition probability matrices of nodes B_i^1 when the node is active is set as:

$$B_i^1 = \begin{pmatrix} 0.98 & 0.02 & 0 \\ 0 & 0.97 & 0.03 \\ 0 & 0 & 1 \end{pmatrix}, \tag{6.2}$$

for $i = (1, \ldots, 5)$ and

$$B_6^1 = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.3}$$

When the node is passive, set

$$B_i^0 = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0 & 0.99 & 0.01 \\ 0 & 0 & 1 \end{pmatrix}, \tag{6.4}$$

for $i = (1, \ldots, 6)$. We assume that when nodes are in passive states, the energy

changes less than when the nodes are active, also when the battery residual energy is at low, it can not transit to high level.

From 5.3, we can compute the transition probability matrix. For example:

$$T_1^1 = A_1^1 \otimes B_1^1 = \begin{pmatrix} 0.9212 & 0.0188 & 0 & 0.0588 & 0.0012 & 0 \\ 0 & 0.9118 & 0.0282 & 0 & 0.0582 & 0.0018 \\ 0 & 0 & 0.9400 & 0 & 0 & 0.0600 \\ 0.0490 & 0.0010 & 0 & 0.9310 & 0.0190 & 0 \\ 0 & 0.0485 & 0.0015 & 0 & 0.9215 & 0.0285 \\ 0 & 0 & 0.0500 & 0 & 0 & 0.9500 \end{pmatrix}, \quad (6.5)$$

other transition probability matrices can be computed accordingly.

Since there is more information leakage when a node is compromised than when a node is in safe state, the cost of selecting a safe node is lower than that of selecting a compromised node. The cost matrices for the simulation are defined as follows:

$$c(1) = \begin{pmatrix} 5.5 \\ 7.5 \\ 15 \\ 60 \\ 62 \\ 100 \end{pmatrix}, c(2) = \begin{pmatrix} 8 \\ 10 \\ 15 \\ 55 \\ 58 \\ 95 \end{pmatrix}, c(3) = \begin{pmatrix} 4.5 \\ 6.5 \\ 15 \\ 47 \\ 50 \\ 90 \end{pmatrix}, \quad (6.6)$$

$$c(4) = \begin{pmatrix} 6 \\ 8 \\ 15 \\ 50 \\ 55 \\ 110 \end{pmatrix}, c(5) = \begin{pmatrix} 9 \\ 10 \\ 15 \\ 55 \\ 57 \\ 110 \end{pmatrix}, c(6) = \begin{pmatrix} 15 \\ 21 \\ 25 \\ 140 \\ 150 \\ 200 \end{pmatrix}, \tag{6.7}$$

which corresponds to the system state matrix $(sb1, sb2, sb3, cb1, cb2, cb3)'$.

The data is chosen based on the assumption that nodes in the upper level usually have higher security property than lower level nodes; which is realistic in military environment. Therefore the cost of selecting the upper level nodes is higher than selecting the lower level nodes because there are more information leaking in the upper level, while the information in the lower level nodes are less because they only have a part of the secret shares. The information leakage of the lower level nodes is more tolerable than that of upper level nodes as threshold cryptography is used in the scheme.

All simulations are run on Windows XP, Core Duo T5400 CPU (1.5G), 1G memory, and the policy vector is computed on Red Hat Linux with Kernel version is 2.4.20 - 31.9.

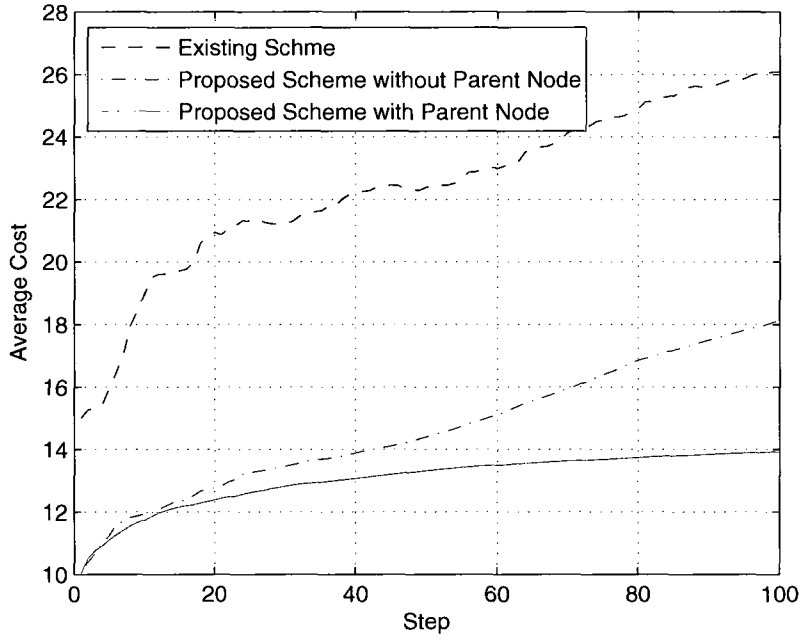

Figure 6.1: Cost Cmparison on different steps.

6.1.1 Performance Improvement over the Existing Scheme

Cost Comparison of Proposed Scheme and Existing Scheme

We first compare the cost of the proposed scheme and existing scheme, the performance of the proposed scheme without selecting the parent node is also considered. We perform simulations with 400 steps for 20 times and calculate the average cost of each time slot. The step can be regarded as a fixed time interval such as one hour or one day etc. in reality.

The proposed scheme shows distinct cost reduction over existing scheme as shown in Fig. 6.1. At the beginning, the proposed scheme with the parent node has the similar performance as the scheme without the parent node. With the time going on, some child nodes battery become exhausted, because the parent node has better energy than child nodes, so the proposed scheme shows better performance than the scheme without parent node.

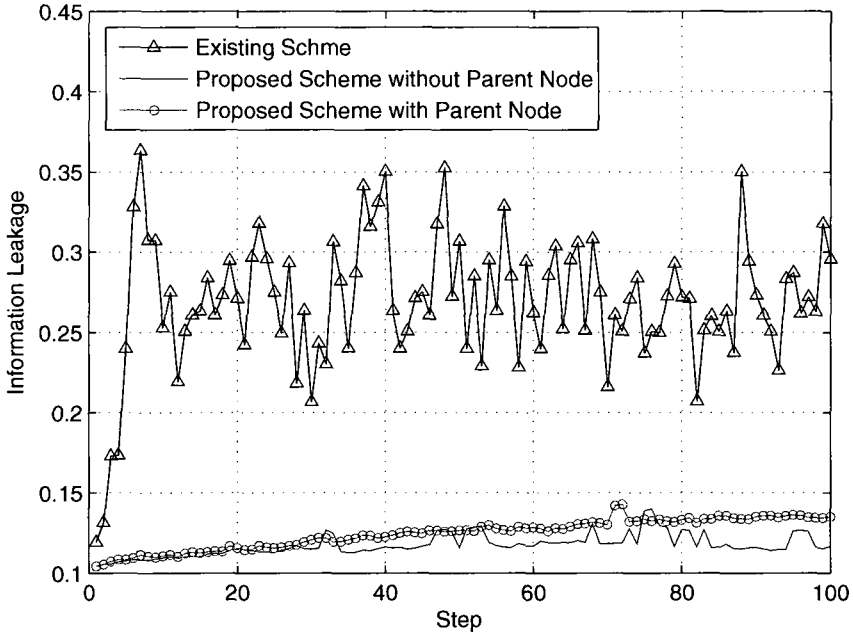

Figure 6.2: Information leakage on different steps.

The cost definition in (5.4) has two components, the cost is kept low thus the information leakage is also kept low. To show the reduction of the information leakage, we use a a relative measurement to represent the information leakage which is shown in Fig. 6.2. The relative measurement is defined as the information leakage of the node selection divided by the information leakage when the selected nodes are in worst states. From Fig. 6.2 we can see that the information leakage is kept low and stable while the information leakage of the existing scheme is high and fluctuating. The proposed scheme without a parent node also have better performance than existing scheme. Thus through optimal node selection, the information leakage is kept low and the system can be more secure.

Cost Comparison of Different Transition Probabilities

To verify the dynamic stability of the proposed scheme, we consider different transition probabilities for the nodes in our scheme. Fig. 6.3 shows the cost comparison

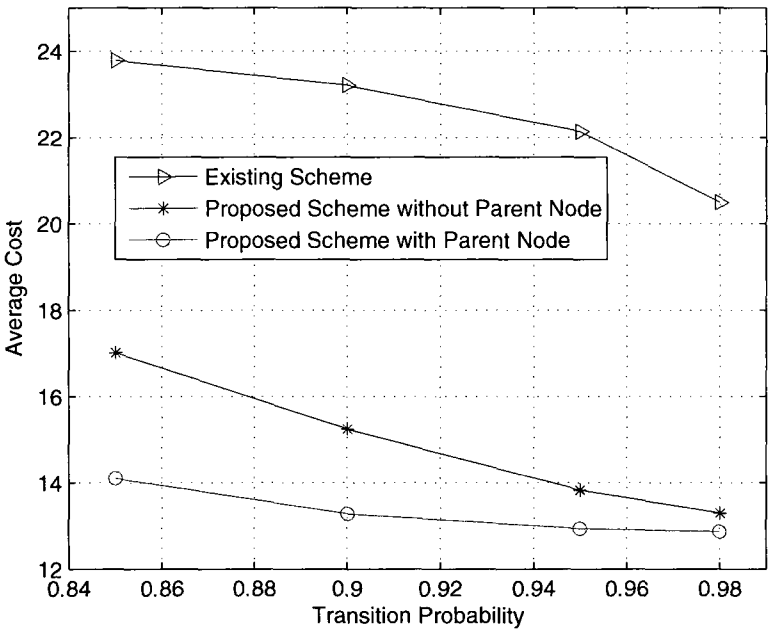

Figure 6.3: Cost under different security transition probabilities.

over existing scheme when the first component in the state transition probability matrix changes from 0.85 to 0.98. With the increase of the transition probabilities, which means nodes have less probabilities of being compromised; the system becomes more secure and the cost is decreased while the proposed scheme always have lower cost than the existing scheme.

Fig. 6.4 shows the comparison of information leakage under different transition probabilities. All the schemes have stable information leakage, while the proposed scheme always have less information leakage than existing scheme.

Cost Comparison of Different Nodes

The proposed scheme is easily scalable. Fig. 6.5 shows the cost when there are more nodes in the network. With the number of available nodes in the network increases from 5 to 30, the cost of the proposed scheme is kept low while the cost of the

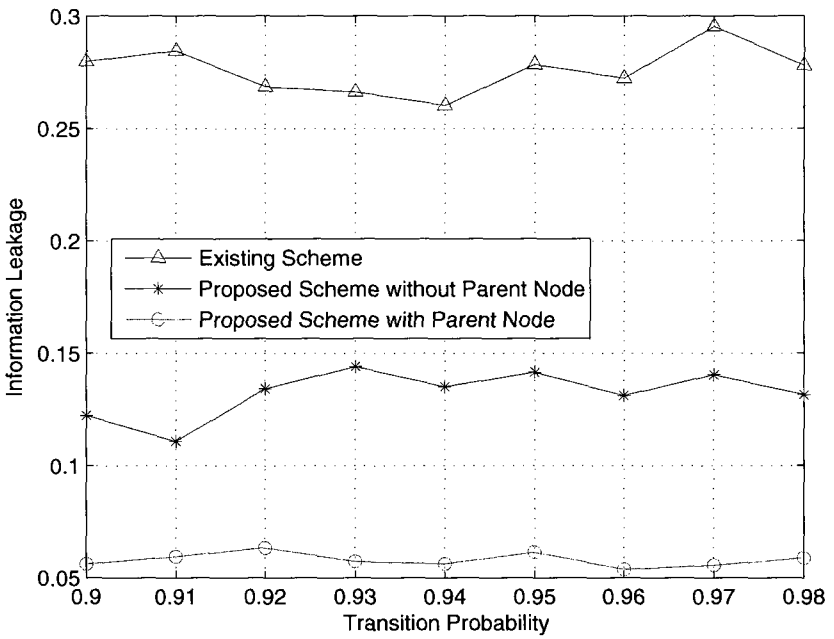

Figure 6.4: Comparison of information leakage with different security transition probabilities.

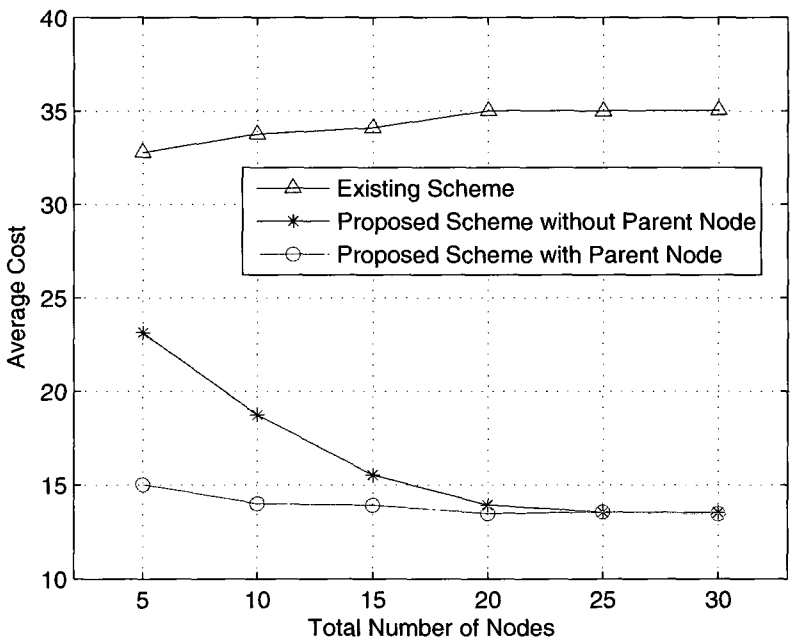

Figure 6.5: Cost under different numbers of nodes.

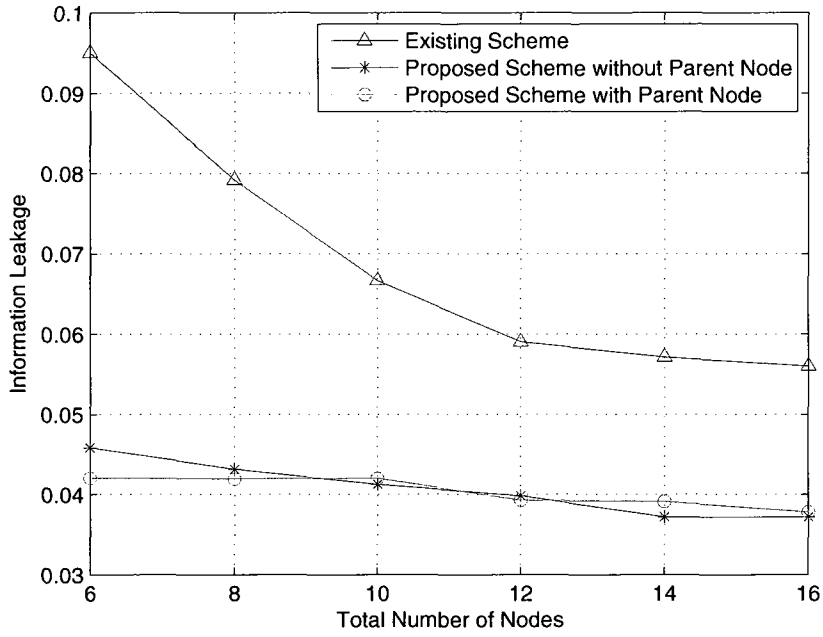

Figure 6.6: Comparison of information leakage with different nodes.

proposed scheme decreases with the number of total nodes increase because there are more nodes can be selected to work as PKG.

Fig. 6.6 shows the information leakage comparison of proposed scheme and existing scheme when there are different nodes exist in the network. With the increase of nodes in the network, information leakage decreases because there are more choices when there are more nodes.

Cost Comparison of Different Thresholds

Fig. 6.7 illustrates system performance under different thresholds. We simulate with 10, 20 and 30 nodes with threshold N_{th} is set to 2 to 6. The cost of the existing scheme increase fast because the number of selected nodes increases, more nodes means higher cost. The cost of the proposed scheme is kept low because it can select the parent node and the cost is stable.

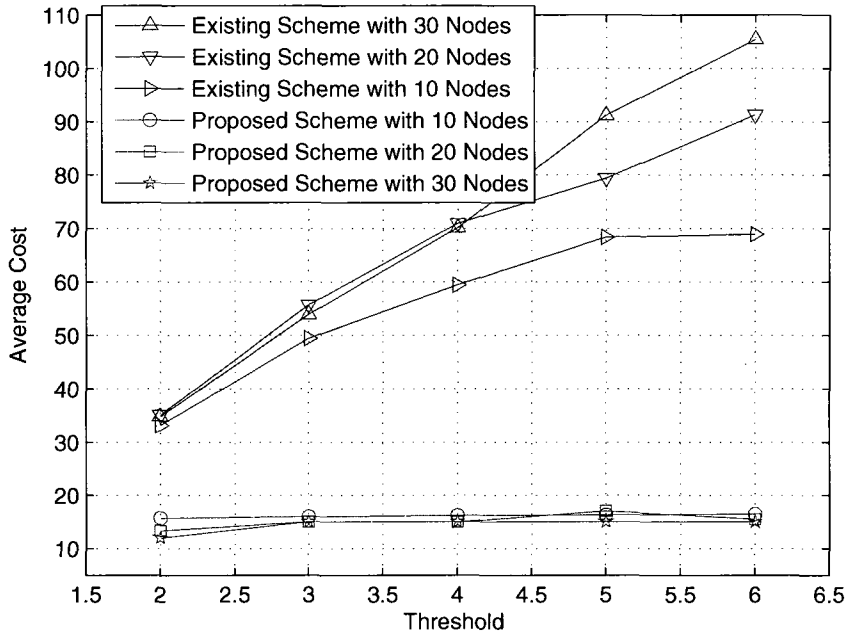

Figure 6.7: Cost under different threshold N_{th}.

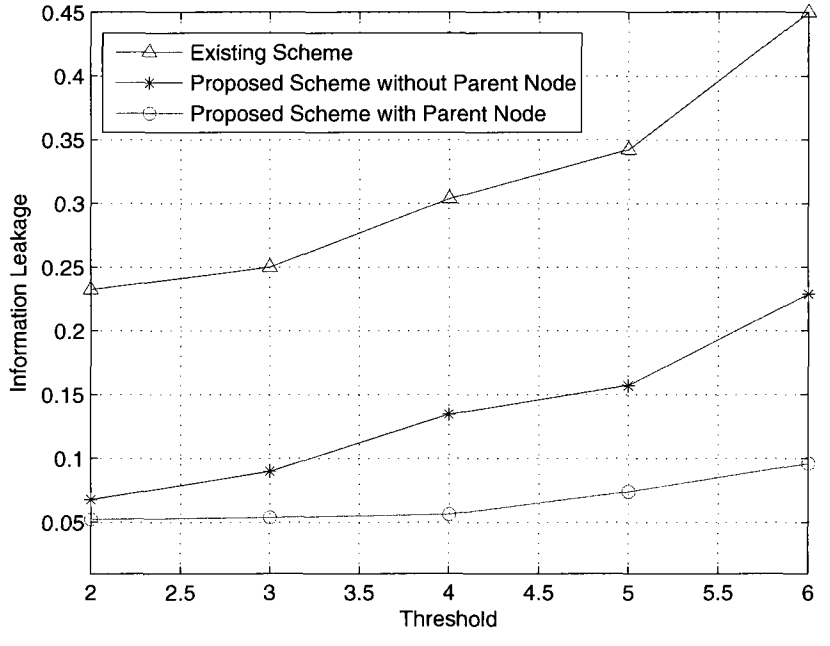

Figure 6.8: Comparison of information leakage with different thresholds.

We then show the corresponding information leakage when different thresholds are used. Fig. 6.8 shows the comparison result. We can see that information leakage increases a little when the threshold increases. That is because when more nodes are used, there are more information leakage.

6.1.2 Network Compromising Probability Improvement

In this simulation we investigate the network compromising probability of the proposed scheme. We assume the attacker has already known all public parameters of the system and try to obtain the secret keys of nodes in order to compromise the network. The network is compromised when the root node is compromised or a threshold N_{th} of children are compromised.

To make the simulation more realistic, we use three security states: $safe$, *attacked* and *compromised*, and two energy states: $high(b1), low(b2)$, so totally there are six states: $sb1, sb2, ab1, ab2, cb1, cb2$.

The security transition probability is defined as:

$$A_6^i = \begin{pmatrix} 0.98 & 0.02 & 0 \\ 0.02 & 0.97 & 0.01 \\ 0 & 0 & 1 \end{pmatrix},$$

for $i = 1, \ldots 5$, and

$$A_6^1 = \begin{pmatrix} 0.999 & 0.01 & 0 \\ 0.05 & 0.94 & 0.01 \\ 0 & 0 & 1 \end{pmatrix}.$$

The parent node is more secure and reliable, which is realistic because in most hierarchical system, the higher level has higher security.

The energy transition probability is defined as:

$$B_6^i = \begin{pmatrix} 0.99 & 0.1 \\ 0 & 1 \end{pmatrix},$$

for $i = 1, \ldots 5$, and

$$B_6^1 = \begin{pmatrix} 0.999 & 0.001 \\ 0 & 1 \end{pmatrix},$$

for node 6. Therefore according to 4.7, the system state transition probability matrix is:

$$T_1^i = A_1^i \otimes B_1^i = \begin{pmatrix} 0.9702 & 0.0098 & 0.0198 & 0.0002 & 0 & 0 \\ 0 & 0.9800 & 0 & 0.0200 & 0 & 0 \\ 0.0198 & 0.0002 & 0.9603 & 0.0097 & 0.0099 & 0.0001 \\ 0 & 0.0200 & 0 & 0.9700 & 0 & 0.0100 \\ 0 & 0 & 0 & 0 & 0.9900 & 0.0100 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 \end{pmatrix}, \quad (6.8)$$

We first compare the network compromising probability of the proposed scheme with existing scheme when security transition probabilities are set from 0.90 to 1.0 and $N_{th} = 2$. The result shown in Fig. 6.9 indicate the proposed scheme always has lower network compromising probability than existing scheme. When the transition

probabilities are closer to 1, all of the schemes are asymptotically closer to 0. So if the nodes security states do not change, the system will remain safe; while the security states changes more, the proposed scheme shows better improvement than existing scheme. Overall, the proposed scheme can decrease the network compromising probability consistently.

In Fig. 6.10 we compare the network compromising probability when there are more nodes in the network. With the increase of the total number of nodes in the network, all the schemes show downward trend in compromising probabilities, that is because more nodes means more choices for node selection. The network compromising probability of the proposed scheme is always kept lower than existing scheme. To further decrease the compromising probability, we increase the threshold N_{th} from 2 to 6. As shown in Fig. 6.11, the proposed scheme always has lower network compromising probability than existing scheme. Thus the proposed scheme has the potential to improve the network security.

6.1.3 Network Lifetime Improvement

In this simulation we investigate the network lifetime improvement of the proposed scheme over existing scheme. For easy comparison of the network lifetime we set the node security transition probability to be 0.99, so the network will die from energy exhaustion in most cases, instead of network compromise. We first check the performance when different transition probabilities are used. There are 6 nodes in the network, and D_{th} is set to 2, so if there two nodes in the 6 nodes run out of power, the network is dead. The energy transition probability is set from 0.88 to 0.98. As shown in Fig. 6.12, the proposed scheme always has longer network lifetime than existing scheme because the proposed scheme select nodes considering their energy properties.

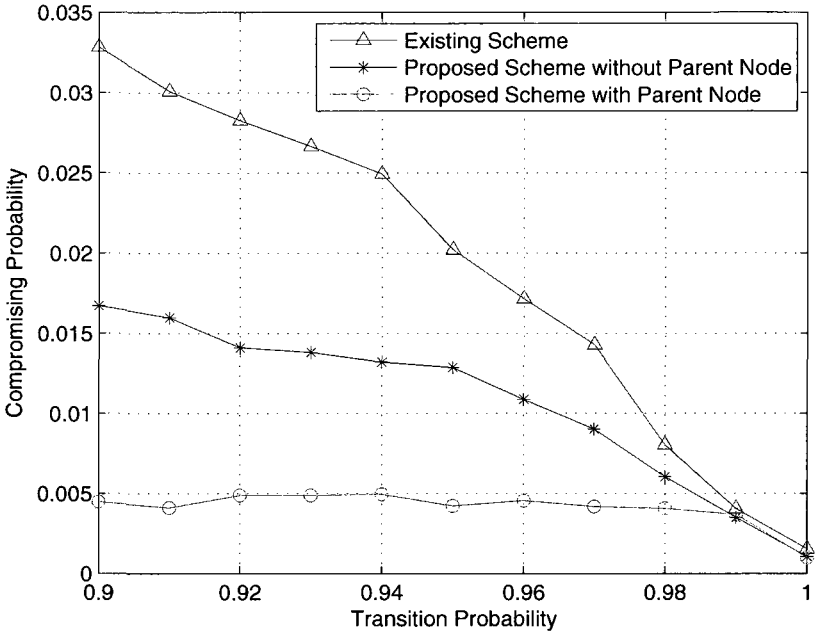

Figure 6.9: Network compromising probabilities in different transition probabilities.

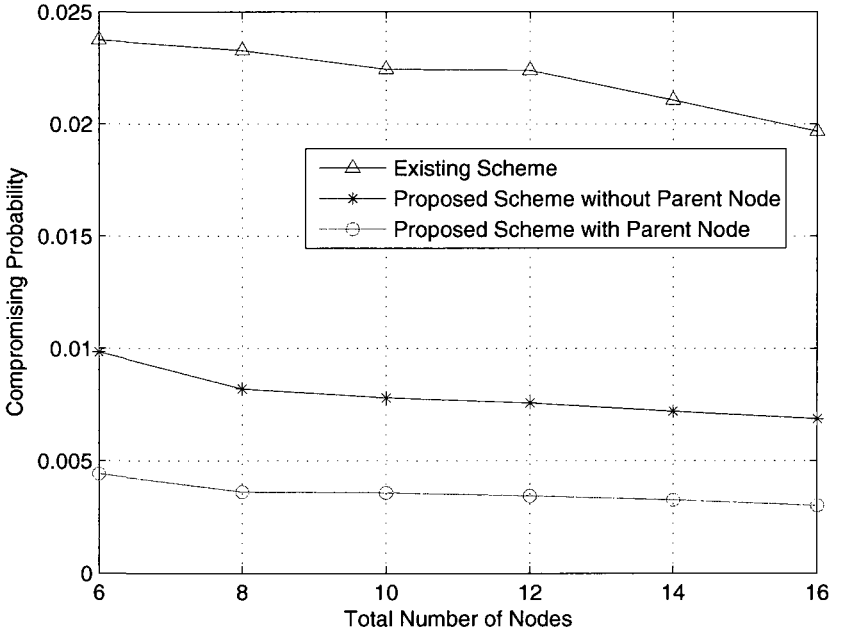

Figure 6.10: Network compromising probabilities under different nodes.

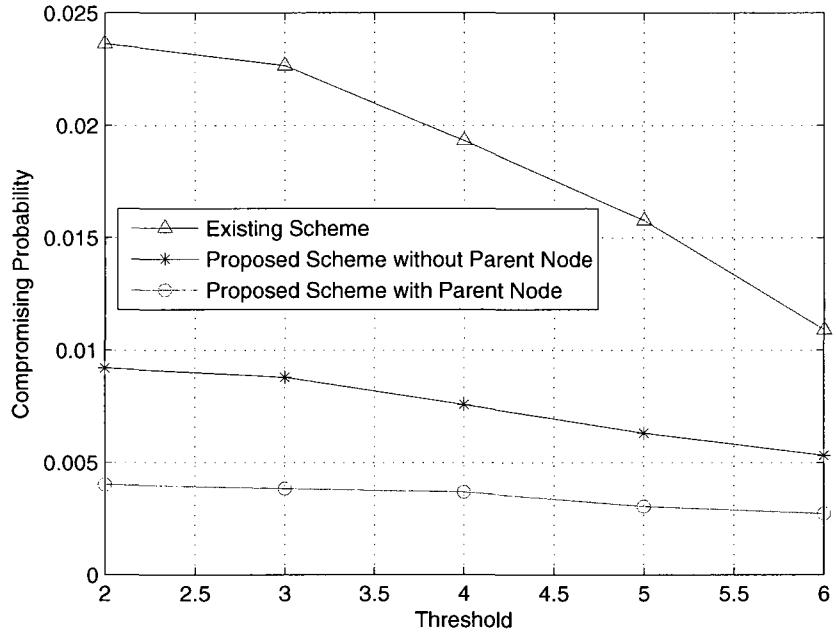

Figure 6.11: Network compromising probabilities under different thresholds.

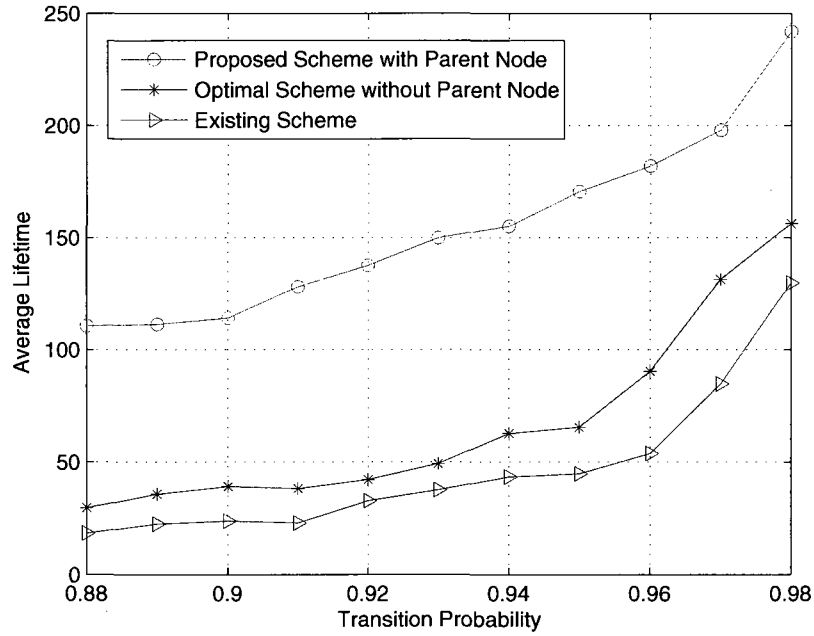

Figure 6.12: Network lifetime under different energy transition probabilities.

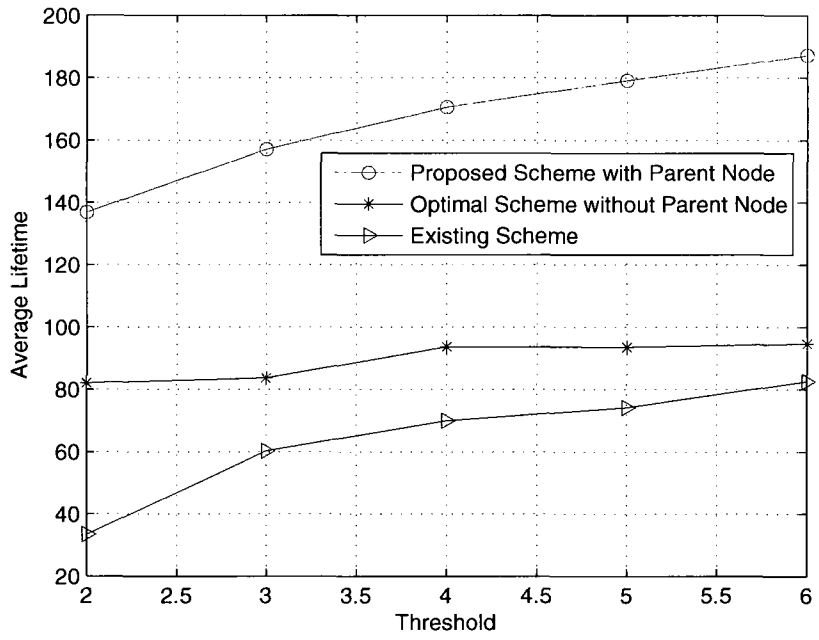

Figure 6.13: Network lifetime under different threshold D_{th}.

Fig. 6.13 compares the lifetime improvement when threshold D_{th} is set from 2 to 6 in a network with 10 nodes. It is shown that the network lifetime increases with the increase of the D_{th}, while the proposed scheme always has longer lifetime than existing scheme.

Finally we compare the network lifetime in Fig. 6.14 when there are more nodes available in the network. The network lifetime increases with the total number of nodes and the proposed scheme shows distinct improvement over existing scheme. Thus the proposed scheme provides longer network lifetime through optimal node selection.

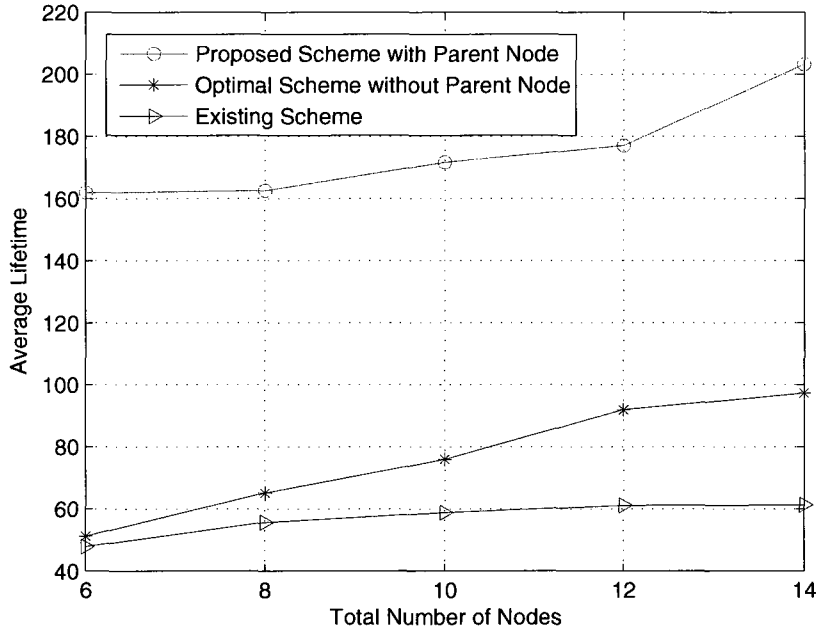

Figure 6.14: Comparison of network lifetime with different nodes.

6.2 Simulation Results and Discussions about the Distributed Biometric Authentication

In this section, we illustrate the proposed scheme using simulation examples. In the simulation we assume a device with two biosensors: iris sensor and fingerprint sensor. In addition to the IDS, there are totally three sensors in the device. We compare the performance of the proposed scheme with an existing scheme, in which sensors are selected randomly without considering the device states and IDS detection.

The security transition matrices are defined as:

$$A_1^1 = \begin{pmatrix} 0.95 & 0.05 \\ 0.30 & 0.7 \end{pmatrix}, A_2^1 = \begin{pmatrix} 0.92 & 0.08 \\ 0.1 & 0.9 \end{pmatrix}, A_3^1 = \begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix},$$

The energy transition matrices are defined as:

$$B_1^1 = \begin{pmatrix} 0.96 & 0.04 \\ 0 & 1 \end{pmatrix}, B_2^1 = \begin{pmatrix} 0.98 & 0.02 \\ 0 & 1 \end{pmatrix}, B_3^1 = \begin{pmatrix} 0.99 & 0.01 \\ 0 & 1 \end{pmatrix},$$

The state transition matrices P_n^a can be computed by: $A_n^a \otimes B_n^a$ for all the biosensors. The observation matrices are defined as:

$$O_s^1 = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}, O_s^2 = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix}, O_s^3 = \begin{pmatrix} 0.97 & 0.03 \\ 0.03 & 0.97 \end{pmatrix},$$

$$O_e^1=\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}, O_e^2=\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}, O_e^3=\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix},$$

and the observation probability matrices $O(l)$ can be computed by: $O_s^i \otimes O_e^i$ for all the biosensors.

The cost matrices for the simulation are defined as follows: $c(1) = (4, 8, 10, 12)$, $c(2) = (3, 7, 11, 14)$, $c(3) = (1, 4, 15, 16)$.

The data is defined based on the assumption: Iris sensor is the most expensive one while it is also the most powerful authentication facility, which provides the highest security authentication. IDS consumes the least resources while it mainly used for state detection. Fingerprint provides intermediate security authentication while its energy cost is between IDS and Iris sensor.

6.2.1 Performance Improvement over the Existing Scheme

We first compare the cost in the proposed scheme and the existing scheme. Fig. 6.15 shows the result of the first 200 steps. It is shown that the proposed scheme always has lower cost than the existing scheme, therefore the information leaking of the proposed scheme is lower than existing scheme, thus the node is more secure under the proposed scheme.

A sample of biosensor policy is shown in Fig. 6.16, we can see that IDS is more frequently selected when the system is secure since it has the lowest cost; when the system become unsecured, more reliable biosensors, e.g. fingerprint or iris will be used. Through optimal scheduling, the cost of biosensor usage is decreased.

To verify the dynamic stability of the proposed scheme, we consider different

transition probabilities for the nodes in our scheme. Fig. 6.17 shows the cost comparison with the existing scheme when the first component in the security transition probability matrix changes from 0.75 to 0.95. With the increases of the transition probabilities, the system becomes more secure and the cost is decreased while the proposed scheme always has lower cost than the existing scheme. For each configuration we average 40 runs ensure adequate confidence of our results. The 95% confidence interval is within 3% to 10% of the mean.

6.2.2 Node Lifetime Improvement

We investigate the node lifetime improvement of the proposed scheme over the existing scheme. We check the performance when different energy transition probabilities are used. The energy transition probability is set from 0.85 to 0.98. As shown in Fig. 6.18, the proposed scheme always have longer network lifetime than the existing scheme because the proposed scheme selects nodes considering their energy states. The 95% confidence interval is within 3% to 9% of the mean. Therefore through optimal biosensors scheduling, the energy can be used more efficiently.

6.3 Summary

In this chapter we provided simulation results for verification of the proposed schemes. For the proposed hierarchical key management scheme, simulation results show that the examples using optimal algorithms always have lower cost than the existing scheme. The compromising probability of the proposed scheme is lower, and the network lifetime is extended due to the optimal node selection. We also performed parameters sensitivity analysis of the proposed scheme, the proposed scheme always show better performance than existing scheme.

Another scheme tested is the proposed multimodal biometric authentication. With

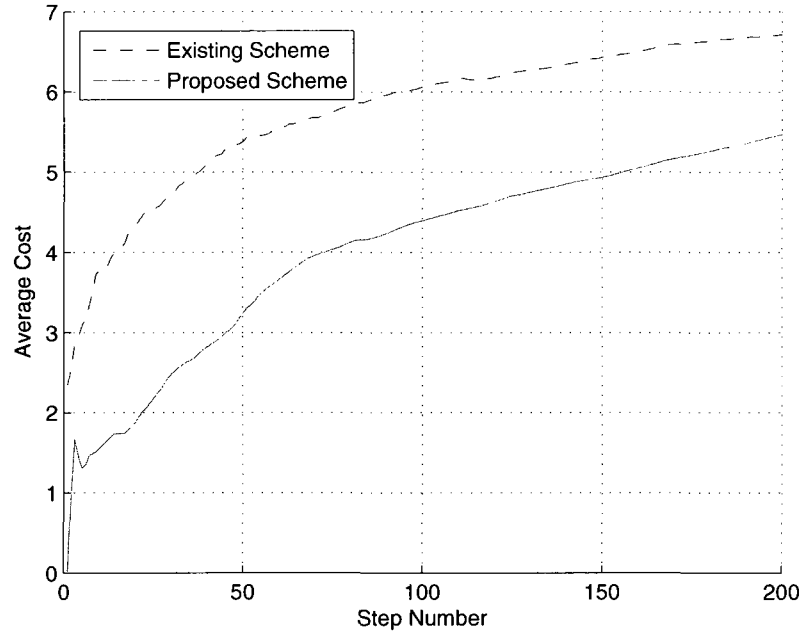

Figure 6.15: Cost comparison of the proposed scheme and existing scheme.

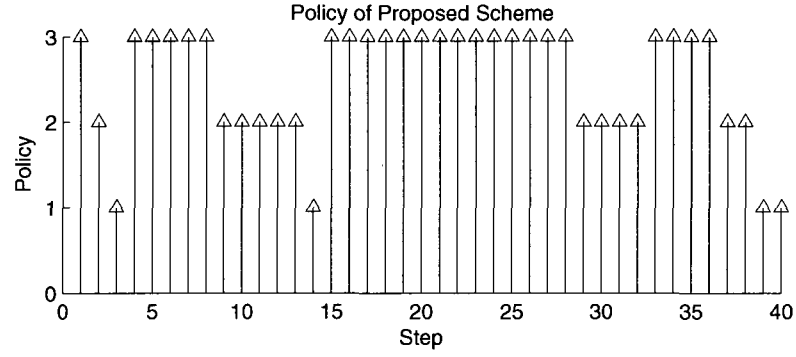

Figure 6.16: Policy of the proposed scheme.

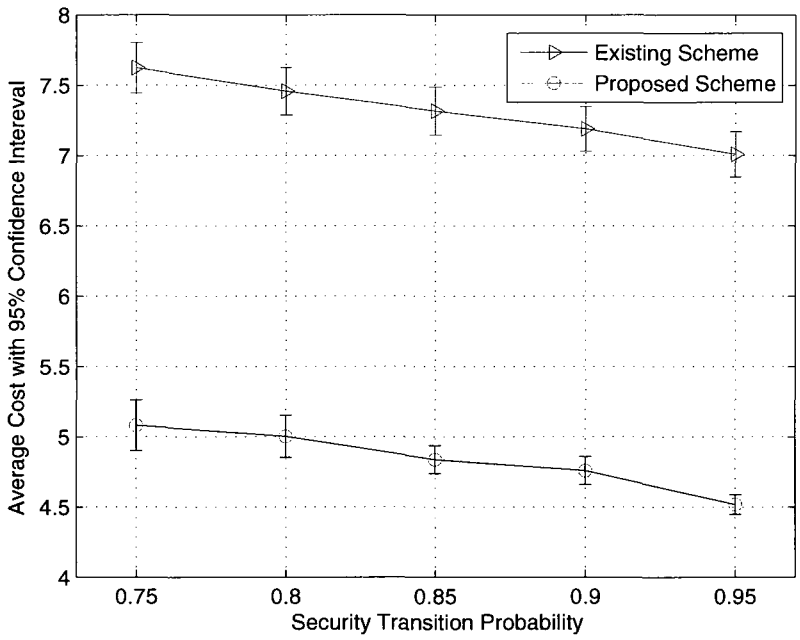

Figure 6.17: Cost comparison of the proposed scheme and existing scheme under different transition probabilities.

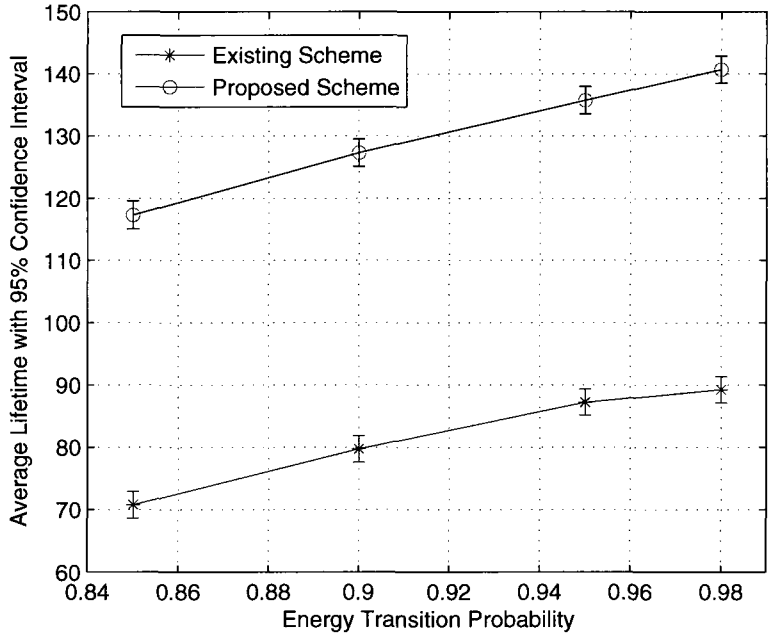

Figure 6.18: Lifetime comparison of the proposed scheme and existing scheme under different energy transition probabilities.

the optimal biosensor scheduling algorithms, the continuous authentication process can select the biosensor dynamically according to the system status. The proposed scheme always has the lower cost than the existing scheme, therefore system security is improved. It is also shown that the device lifetime is improved for the proposed scheme.

Chapter 7

Conclusions and Future work

In this thesis, we first proposed a hierarchical ID-based key management scheme for tactical MANETs. The proposed scheme can dynamically select the best nodes to work as the PKG while taking account into node security conditions and energy states, which are not considered in existing schemes. The node selection is formulated as a restless bandit problem, which in essence is a MDP. We use a primal dual heuristic to solve the restless bandit problem, which reduces the computation complexity significantly. Simulation results show that the proposed scheme can improve network security and decrease the network compromising probability. One advantage of the proposed scheme is it can work with any existing hierarchical schemes, without changing their algorithms. We also discussed some issues with the implementation of the hierarchical key management scheme.

Another scheme we proposed is a distributed multimodal biometric authentication. Multimodal biometrics provides the possibility to meet all the requirements of authentication in MANETs. However, biometric authentication is expensive in resource consumption, which includes computation and energy. The proposed scheme can select a biometric sensor based on current security state and energy conditions,

therefore reduces the overall cost and improves device lifetime. Due to the inaccuracy of the system states, the proposed scheme is formulated as a multi-armed bandit problem. With the Gittins index algorithm, the optimal biometric sensor selection is just selecting the sensor with the maximum index. Simulations examples were provided for validation of the scheme.

In the future work, it is interesting to consider more node states, such as wireless route and channel states in hierarchical key management for tactical MANETs. Another interesting direction is to find some structured solutions to the key management and authentication problems.

List of References

[1] S. Balfe, K. D. Boklan, Z. Klagsbrun, and K. G. Paterson, "Key refreshing in identity-based cryptography and its applications in MANETs," in *Proc. of the MILCOM 2007*, (Orlando, FL, USA), pp. 1–8, Oct. 2007.

[2] R. L. Rivest, A. Shamir, and Y. Tauman, "How to share a secret," *Comm. ACM*, vol. 22, pp. 612–612, Nov. 1979.

[3] Y. Desmedt and Y. Frankel, "Threshold cryptosystems," in *Proc. of CRYPTO'89*, (Santa Barbara, CA, USA), pp. 307–315, Aug. 1989.

[4] G. Hanaoka, T. Nishioka, Y. Zheng, , and H. Imai., "A hierarchical non-interactive key-sharing scheme with low memory size and high resistance against collusion attacks," *Comput. J.*, vol. 45, no. 3, pp. 293–303, 2002.

[5] M. Ramkumar, N. Memon, and R. Simha, "A hierarchical key pre-distribution scheme," in *Proc. of EIT 2005*, (Lincoln, NE, USA), May. 2005.

[6] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt, and S. D. Wolthusen, "Strongly-resilient and non-interactive hierarchical key-agreement in MANETs," in *Proc. of the ESORICS '08*, (Berlin, Heidelberg), Springer-Verlag, 2008.

[7] B. Carlo, D. S. Alfredo, V. Ugo, H. Amir, K. Shay, and Y. Moti, "Perfectly secure key distribution for dynamic conferences," *Inf. Comput.*, vol. 146, no. 1, pp. 1–23, 1998.

[8] A. Weimerskirch and G. Thonet, "A distributed light-weight authentication model for ad-hoc networks," *Lecture Notes in Computer Science*, vol. 2288, pp. 341–354, ISBN: 3-540-43319-8 2001.

[9] Q. Xiao, "A biometric authentication approach for high security ad-hoc networks," in *Proc. IEEE Info. Assurance Workshop*, (West Point, NY), June 2004.

[10] T. Sim, S. Zhang, R. Janakriaman, and S. Kumar, "Continuous verification using multimodal biometrics," *IEEE Trans. Patten Analysis and Machine Intelligence*, vol. 29, pp. 687–700, Apr. 2007.

[11] J. Koreman, A. C. Morris, D. Wu, S. A. Jassim, and et. al, "Multi-modal biometrics authentication on the securephone PDA," in *Proc. Second Workshop on Multimodal User Authentication*, (Toulouse, France), May 2006.

[12] J. Muncaster and M. Turk, "Continuous multimodal authentication using dynamic bayesian networks," in *Proc. Second Workshop on Multimodal User Authentication*, (Toulouse, France), May 2006.

[13] A. Altinok and M. Turk, "Temporal integration for continuous multimodal biometrics," in *Proc. Workshop Multimodal User Authentication*, pp. 131–137, Dec. 2003.

[14] J. Liu, F. Yu, C.-H. Lung, and T. H., "Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 806–815, Feb. 2009.

[15] P. Whittle, "Multi-armed bandits and the Gittins index," *J. R. Statist. Soc. B*, vol. 42, no. 2, pp. 143–149, 1980.

[16] D. Berstimas and J. Nino-Mora, "Restless bandits, linear programming relaxations, and a primal dual index heuristic," *Operations Research*, vol. 48, no. 1, pp. 80–90, 2000.

[17] J. L. Ny, M. Dahleh, and E. Feron, "Multi-agent task assignment in the bandit framework," in *Proc. 45th IEEE Conf. Decision and Control*, (San Diego, California), pp. 5281–5286, Dec. 2006.

[18] J. L. Ny and E. Feron, "Restless bandits with switching costs: Linear programming relaxations, performance bounds and limited lookahead policies," in *Proc. 2006 American Control Conf.*, (Minneapolis, Minnesota), pp. 1587–1592, June 2006.

[19] P. Whittle, "Restless bandits: activity allocation in a changing world," in *A Celebration of Applied Probability* (J. Gani, ed.), vol. 25 of *J. Appl. Probab.*, pp. 287–298, Applied Probability Trust, 1988.

[20] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Upper Saddle River, NJ: Prentice Hall, 2004.

[21] H. Yang, H. Y. Luo, F. Ye, S. W. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," *IEEE Wireless Comm.*, vol. 11, pp. 38–47, Feb. 2004.

[22] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proc. of CRYPTO'84*, (Santa Barbara, CA, USA), Aug. 1984.

[23] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Computing*, vol. 32, pp. 586–615, Mar. 2003.

[24] R. Saka, K. Ohgishi, , and M. Kasahara, "Cryptosystems based on pairings," in *Proc. of SCIS 2000)*, (Okinawa, Japan), Jan. 2000.

[25] R. Dupont and A. Enge, "Practical non-interactive key distribution based on pairings," in *Proceedings of the International Workshop on Coding and Cryptography (WCC*, 2002.

[26] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, (London, UK), pp. 514–532, Springer-Verlag, 2001.

[27] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Networks*, vol. 13, no. 6, pp. 24–30, 1999.

[28] H. Deng, A. Mukherjee, and D. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *Proc. of ITCC'04*, (Washington, DC, USA), Apr. 2004.

[29] K. Hoeper and G. Gong, "Bootstrapping security in mobile ad hoc networks using identity-based schemes with key revocation," tech. rep., University of Waterloo, Canada, 2006.

[30] D. S. K., A. Afrand, and B. Kalyan, "Security in wireless mobile and sensor networks," pp. 531–557, 2004.

[31] E. Laurent and D. G. Virgil, "A key-management scheme for distributed sensor networks," in *Proc. of the 9th ACM conf. on Computer and communications security*, (Washington, DC, USA), ACM, 2002.

[32] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and robust access control for mobile ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 12, pp. 1049–1063, Dec. 2004.

[33] A. Ross and A. K. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, pp. 2115–2225, Sep. 2003.

[34] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai, "An efficient hierarchical identity-based key-sharing method resistant against collusion-attacks," in *ASIACRYPT '99: Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security*, (London, UK), pp. 348–362, Springer-Verlag, 1999.

[35] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," 2002.

[36] A. R. Cassandra, *Exact and Approximate Algorithms for Partially Observed Markov Decision Process.* PhD thesis, Brown University, 1998.

[37] R. Smallwood and E. Sondik, "Optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, pp. 1071–1088, 1973.

[38] N. L. Zhang and W. Liu, "Planning in stochastic domains: Problem characteristics and approximation," tech. rep., Department of Computer Science, Hong Kong University of Science and Technology, 1996.

[39] A. R. Cassandra, "Tony's pomdp webpage." available online: http://www.cs.brown.edu/research/ai/pomdp/index.html.

[40] A. Mishra, K. Nadkarni, , and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Comm.*, vol. 11, pp. 48–60, Feb. 2004.

[41] P. Hu, Z. Zhou, Q. Liu, and F. Li, "The hmm-based modeling for the energy level prediction in wireless sensor networks," in *Proc. IEEE 2nd Conf. on Industrial Electronics and Applications*, (Harbin, P.R. China), pp. 2253–2258, May 2007.

[42] Q. Dong, "Maximizing system lifetime in wireless sensor networks," in *Proc. 4h Int. Symp. Inform. Proc. in Sensor Netw.*, (Los Angeles, California), pp. 13–19, Apr. 2005.

[43] Y. Chen, Q. Zhao, and V. Krishnamurthy, "Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach," *IEEE Trans. Signal Proc.*, vol. 55, no. 5, pp. 2294–2309, 2007.

[44] W. P, "Multi-armed bandits and the gittins index," *Roy. Statist. Soc. Ser*, no. 2, pp. 143–149, 1980.

[45] J. Gittins, "Bandit processes and dynamic allocation indices," *J.R. Statist. Soc. B*, vol. 41, no. 2, pp. 148–177, 1979.

[46] F. d'Epenoux, "Sur un problème de production et de stockagedans l'aléatoire," *RAIRO Rech. Opér*, vol. 14, pp. 3–16, 1960.

[47] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in *IEEE Workshop on Information Assurance and Security*, pp. 85–90, 2001.

[48] P. Mell and M. Mclarnon, "Mobile agent attack resistant distributed hierarchical intrusion detection systems," in *Second International Workshop on Recent Advances in Intrusion Detection*, (Purdue University), Sep. 1999.

[49] D. G. Marks, P. Mell, and M. Stinson, "Optimizing the scalability of network intrusion detection systems using mobile agents," *J. Netw. Syst. Manage.*, vol. 12, no. 1, pp. 95–110, 2004.

[50] V. Krishnamurthy, "A value iteration algorithm for partially observed markov decision process multi-armed bandits," *Math. of Oper. Res.*, pp. 133–152, May 2005.

Appendix A

Matlab Programs

```
% evaluate a policy
%@param total_step: total executed steps
%@param random: true/false
%@param gittins_sel_num: how many servers to be selected
%@initial_state: initial server states
%@info_state: information state at initial
function [policy_hist, reward_hist, server_state, gittins_hist, obs_hist, run_step,inf_leakage]
    = eval_policy(total_step, ...
  random, server_number, ...
  state_number, gittins_sel_num,initial_state,info_state,trans_matrix,trans_matrix_passive,
      ...
  obs_matrix,reward_matrix,gittins_reward,alpha_filename,restless, node_master_tag,
      s_reward_matrix)
clear policy_hist ;
clear reward_hist ;
clear server_state ;

%initialize data
for kk = 1:server_number
  if ( ~random )
    fid = fopen(alpha_filename{kk}, 'r');
    alpha_data{kk} = fscanf(fid, '%d %g %g %g %g %g %g', [7 inf]);
    fclose(fid);
  end
  %keep server states
  server_state(kk,:) = zeros(1, total_step) ;
  %keep observe history
  obs_hist(kk,:) = zeros(1, total_step) ;
  %keep gittins index history
  gittins_hist(kk,:) = zeros(1, total_step) ;
  info_state_hist{kk} = zeros(state_number,total_step);
end
```

```
% keep server gittins index and corresponding action/policy index
gittins_index = zeros(1,server_number) ;
action_index = zeros(1,server_number) ;
inf_leakage = zeros(1,total_step) ;
% all server policies history
policy_hist = zeros(gittins_sel_num,total_step);
% all server reward history
reward_hist = zeros(gittins_sel_num,total_step);
%initialize system states
for kk = 1:server_number
   %info_state(:,kk) = [rand_temp; 1-rand_temp]; %this should change if there are more
       than 2 states
   info_state_hist{kk}(:,1) = info_state(:,kk) ;
   current_state(kk) = initial_state(kk) ;
   % put the initial state into observation history for late update
   obs_hist(kk,1) = initial_state(kk) ;
   server_state(kk,1) = initial_state(kk) ;
   if ~random
      gittins_index(kk) = calc_index(info_state(:,kk), alpha_data{kk}, gittins_reward(kk));
   end
end
run_step = 0 ;
for step = 1:total_step
   run_step = run_step +1 ;
   if ~random
      active_server_list = get_max_index(gittins_index, gittins_sel_num ) ;
      gittins_hist(:,step) = gittins_index(1,:) ;
   else
      done = 0 ;
         active_server_list = random_select(server_number, gittins_sel_num) ;
         % if master node is in the selected list, use the master node
         for i = 1:length(active_server_list)
            if node_master_tag(active_server_list(i)) == 1
               active_server_list = [active_server_list(i)] ;
               break ;
            end
         end
   end
   % duplicate information state for passive nodes
   passive_node_list = setdiff([1:server_number],active_server_list) ;
   for passive_nodex_index = 1:size(passive_node_list,2)
      passive_node = passive_node_list(passive_nodex_index);
      % the following just copy the states!!!!
      if ~restless
         if step > 1
            % current state does not change
            server_state(passive_node,step) = current_state(passive_node) ;
            info_state_hist{passive_node}(:,step) = info_state_hist{passive_node}(:,step-1) ;
            obs_hist(passive_node, step) = obs_hist(passive_node, step-1) ;
         end
```

```
else
   % Restless way!!!!
   rn = rand(1);
   state_tmp = 0;
   if step > 1
      server_old_state = server_state(passive_node,step-1) ;
   else
      server_old_state = current_state(passive_node) ; % in fact, it is default initial
          state
   end
   if step > 1
      transit_success = 0 ;
      for kkk = 1:state_number
         state_tmp = state_tmp + trans_matrix_passive{passive_node}(
             server_old_state ,kkk);
         if rn <= state_tmp
            current_state(passive_node) = kkk;
            server_state(passive_node,step) = kkk ;
            transit_success = 1 ;
            break ;
         end

      end
      if kkk == state_number && ~transit_success
         fprintf('transit_fails:_rn:%f:_state_tmp:%f.server_old_state:%d_,kkk:%d\n',
             rn, state_tmp,server_old_state ,kkk) ;
         trans_matrix_passive{passive_node}
      end
   end
   if ~random
      % update 'now' state, random simulation does not
      o_now(passive_node) = obs_update(current_state(passive_node),obs_matrix{
          passive_node});
      obs_hist(passive_node, step) = o_now(passive_node) ;
      % Compute the current info state based on observation
      down_tmp = 0;
      is_tmp = [];
      for i = 1:size(info_state, 1)
         for k = 1:size(info_state, 1)
            down_tmp = down_tmp + info_state(i, passive_node)*
                trans_matrix_passive{passive_node}(i,k)*obs_matrix{passive_node}(k,
                o_now(passive_node));
         end
      end
      for j = 1:size(info_state, 1)
         up_tmp = 0;
         for i = 1:size(info_state, 1)
            up_tmp = up_tmp + info_state(i, passive_node)*trans_matrix_passive{
                passive_node}(i,j)*obs_matrix{passive_node}(j,o_now(passive_node));
         end
```

```
            is_tmp = [is_tmp; up_tmp/down_tmp];
        end
        info_state(:,passive_node) = is_tmp;
        % remember all information history, currently no use
        info_state_hist{passive_node}(:,step) = is_tmp ;
    end
    if ~random
        gittins_index(passive_node) = calc_index(info_state(:,passive_node), alpha_data{
            passive_node}, gittins_reward(passive_node)) ;
    end
  end %end of restless processing
end
activce_node_number = server_number - length(passive_node_list) ;
% get compromise probability
inf_leakage_sum = 0 ;
worst_sum = 0 ;
for gittins_sel_index = 1: activce_node_number
  active_server = active_server_list(gittins_sel_index) ;
  inf_leakage_sum = inf_leakage_sum + reward_matrix{active_server}(current_state(
      active_server)) ;
  worst_sum = worst_sum + reward_matrix{active_server}(state_number) ;
  if current_state(active_server) < 4
    xt = 1 ;
  else
    xt = 2 ;
  end
  inf_leakage_sum = inf_leakage_sum + s_reward_matrix{active_server}(xt) ;
  worst_sum = worst_sum + s_reward_matrix{active_server}(2) ;
end
inf_leakage(step) = inf_leakage_sum/worst_sum ;
for gittins_sel_index = 1: activce_node_number
  % new active server
  active_server = active_server_list(gittins_sel_index) ;
  % save the active server into history
  policy_hist(gittins_sel_index, step) = active_server ;
  % get reward
  if step > 1
    reward_hist(gittins_sel_index,step) = reward_matrix{active_server}(server_state(
        active_server,step-1));
  else
    reward_hist(gittins_sel_index,step) = reward_matrix{active_server}(server_state(
        active_server,1));
  end
  % state transition after action
  rn = rand(1);
  state_tmp = 0;
  % Update all server states, for inactive server, the states does not change
  server_old_state = current_state(active_server) ;
  transit_success = 0 ;
  for kkk = 1:state_number
```

```
        state_tmp = state_tmp + trans_matrix{active_server}( server_old_state ,kkk);
        if rn <= state_tmp
            current_state(active_server) = kkk;
            server_state(active_server,step) = kkk ;
            transit_success = 1 ;
            %update passive nodes codes removed from here!
            break; % this is a must!!!
        end
    end
    if kkk == state_number && ~transit_success
        fprintf('active_transit_fails:_state_number:%d,rn:%f:_state_tmp:%f:_active_server
            :%d:_server_old_state:%d\n', state_number,rn, state_tmp, active_server,
            server_old_state) ;
        trans_matrix{active_server}( server_old_state ,kkk)
    end
    % update 'now' state
    if ~random
        o_now(active_server) = obs_update(current_state(active_server),obs_matrix{
            active_server});
        obs_hist(active_server, step) = o_now(active_server) ;
        % Compute the current info state based on observation
        down_tmp = 0;
        is_tmp = [];
        for i = 1:size(info_state, 1)
            for k = 1:size(info_state, 1)
                down_tmp = down_tmp + info_state(i, active_server)*trans_matrix{
                    active_server}(i,k)*obs_matrix{active_server}(k,o_now(active_server));
            end
        end
        for j = 1:size(info_state, 1)
            up_tmp = 0;
            for i = 1:size(info_state, 1)
                up_tmp = up_tmp + info_state(i, active_server)*trans_matrix{active_server}(
                    i,j)*obs_matrix{active_server}(j,o_now(active_server));
            end
            is_tmp = [is_tmp; up_tmp/down_tmp];
        end
        info_state(:,active_server) = is_tmp;
        % remember all information history, currently no use
        info_state_hist{active_server}(:,step) = is_tmp ;
    end
    if ~random
        gittins_index(active_server) = calc_index(info_state(:,active_server), alpha_data{
            active_server}, gittins_reward(active_server)) ;
    end
end % select N maximum gittins index server
% after node selection, if the master node or threshold number of nodes are
    compromised, return
comprised_count = 0 ;
for kk = 1:server_number
```

```
      if node_master_tag(kk) == 1 && current_state(kk) >= 5
         fprintf('master_node_compromised\n');
         return ;
      else
         if node_master_tag(kk) == 0 && current_state(kk) >= 5
            comprised_count = comprised_count + 1 ;
         end
      end
   end
   if comprised_count >= gittins_sel_num
      % threshold number of nodes are compromised
      fprintf('child_node_compromised\n');
      return;
   end
 end %cycle of steps
end %function
```

```
% evaluate a policy, restless bandit
%@param total_step: total executed steps
%@param random: true/false, random selection or gittins selection
%@param gittins_sel_num: how many servers to be selected
%@initial_state: initial server states
%@info_state: information state at initial
%@node_master_tag: indicate if the node is a 'master' node, then only
%select one master server; if server_master_tag = 0, needs to select
%threhold number of server
function [policy_hist, reward_hist, server_state, gittins_hist, run_step, inf_leakage] =
    eval_restless_policy(total_step, server_number, ...
   state_number, threshold,initial_state,trans_matrix_active,trans_matrix_passive, ...
   reward_matrix,gittins_reward, restless, node_master_tag,s_reward_matrix )
   % discount factor, this is only used in restless bandit
   Beta = 0.8 ;
   %initialize data: server states etc.
   for kk = 1:server_number
      % maintain server states
      server_state(kk,:) = zeros(1, total_step) ;
      %keep gittins index history
      gittins_hist(kk,:) = zeros(1, total_step) ;
   end
   % keep server gittins index and corresponding action/policy index
   gittins_index = zeros(1,server_number) ;
   action_index = zeros(1,server_number) ;
   restless_index = zeros(server_number, state_number) ;
   StepReward = zeros(total_step) ;
   inf_leakage = zeros(1, total_step) ;
   step_ave = 0 ;
   ste_pave = 0;
   % all server policies history
   policy_hist = zeros(threshold,total_step);
   % all server reward history
```

```
reward_hist = zeros(threshold,total_step);
%transform reward matrix (map) and transit matrix (also a map) to a real matrx
reward_mtx = zeros(server_number, state_number) ;
reward_mtx2 = zeros(server_number, state_number) ;
trans_mtx = zeros(state_number, state_number, server_number) ;
trans_mtx2 = zeros(state_number, state_number, server_number) ;
for i = 1:server_number
   for ttt=1:state_number
      reward_mtx2(i,ttt) = reward_matrix{i}(ttt) ;
   end
   trans_mtx(:,:,i) = trans_matrix_active{i} ;
   trans_mtx2(:,:,i) = trans_matrix_passive{i} ;
end
%initialize system states
AlphaNow = zeros(server_number, state_number) ;
for kk = 1:server_number
   %setup alpha state
   AlphaNow(kk, initial_state(kk)) = 1 ;
   State(kk) = initial_state(kk) ;
   server_state(kk,1) = initial_state(kk) ;
end
%define index table
index_table = {} ;
index_table_num = [] ;
run_step = 0 ;
for step = 1:total_step
   run_step = run_step+1 ;
   %compute restless priority index for current state: State(..)
   temp_index = 0 ;
   for tempi = 1:server_number
      temp_index = temp_index + State(tempi)*10^(server_number-tempi) ;
   end
   index_found = find(index_table_num == temp_index) ;
   if length(index_found) <= 0
      %setup the parameters, then calculate restlesss gittins index
      [Delta,Result] = calc_restless_index(threshold, server_number, Beta, ...
            state_number, AlphaNow, reward_mtx, reward_mtx2, trans_mtx, trans_mtx2)
                ;

      if Result == 0
         %no result found
         fprintf('can_not_find_the_restless_index.\n');
% return ;
      end
      index_table_num = [index_table_num temp_index] ;
      index_table{length(index_table_num)} = Delta ;
   else
      Delta = index_table{index_found} ;
   end
   %get node with maximum restless index
```

```
for kkk = 1:server_number
   State(kkk) = find(AlphaNow(kkk,:) == 1);
   Index(kkk) = Delta(kkk, find(AlphaNow(kkk,:) == 1));
end
Sorted = sortrows([Index;1:server_number]');
ActiveSet = Sorted(1:threshold,2);
% if there is a master node in ActiveSet, use the master node, then
% ActiveSet will have only one element, that is the master node
master_node_index = find(node_master_tag) ;
if length(master_node_index) > 0
   use_master = 0 ;
   for i = 1:length(ActiveSet)
      if node_master_tag(ActiveSet(i)) == 1
         ActiveSet = ActiveSet(i) ;
         use_master = 1 ;
         break ;
      end
   end
   %try to compare cost sum of selected nodes with single master node
   if use_master == 0
      ActiveSet_sum = 0 ;
      for i = 1:length(ActiveSet)
         ActiveSet_sum = ActiveSet_sum + reward_mtx2(ActiveSet(i),State(
             ActiveSet(i))) ;
      end
      if ActiveSet_sum < reward_mtx2(master_node_index(1),State(
          master_node_index(1) )) ;
         ActiveSet = master_node_index(1) ;
      end
   end
end
% get compromise probability
inf_leakage_sum = 0 ;
worst_sum = 0 ;
for sel_index = 1: length(ActiveSet)
   active_server = ActiveSet(sel_index) ;
   inf_leakage_sum = inf_leakage_sum + reward_mtx2(active_server, State(
       active_server) ) ;
   worst_sum = worst_sum + reward_mtx2(active_server, state_number) ;
   if State(active_server) < 4
      xt = 1 ;
   else
      xt = 2 ;
   end
   inf_leakage_sum = inf_leakage_sum + s_reward_matrix{active_server}(xt) ;
   worst_sum = worst_sum + s_reward_matrix{active_server}(2) ;
   end
inf_leakage(step) = inf_leakage_sum/worst_sum ;

% update state and calculate reward
```

```
server_tmp_index = 0 ;
for kk = 1:server_number
   RandNum = rand(1);
   if length(find(kk == ActiveSet)) > 0
      %active node, find reward, and update state
      server_tmp_index = server_tmp_index + 1;
      reward_hist( server_tmp_index,step) = reward_mtx2(kk, State(kk));
      policy_hist( server_tmp_index, step) = kk ;
      %update states
      for kkk = 1:state_number
         if RandNum <= sum(trans_mtx(State(kk),1:kkk,kk))
            State(kk) = kkk;
            break ;
            %else continue, untill find the state the node
            %will transit to
         end
      end
  else
      % passive node update states
      if restless
         %update states of passive nodes
         for kkk = 1:state_number
            if RandNum <= sum(trans_mtx2(State(kk),1:kkk,kk))
               State(kk) = kkk;
               break;
            end
         end
      end
      %else do nothing
   end
   AlphaNow(kk, :) = zeros(1, state_number);
   AlphaNow(kk, State(kk)) = 1;
   server_state(kk, step) = State(kk) ;
end
% after node selection, if the master node is compromised, return
comprised_count = 0 ;
for kk = 1:server_number
   if node_master_tag(kk) == 1 && State(kk) >= 5
      fprintf('master_node_compromised\n');
      return ;
   else
      if node_master_tag(kk) == 0 && State(kk) >= 5
         comprised_count = comprised_count + 1 ;
      end
   end

end
if comprised_count >= threshold
   % threshold number of nodes are compromised
   fprintf('child_node_compromised\n');
```

```
        return;
      end
    end %cycle of steps
end %evaluate restless bandit function
```

```
show_step = 40 ;
server_number = 6 ;
gittins_sel_num = 2 ;
run_time = 60 ;
restless = true ;
state_number = 3 ;
for run_index = 1:run_time*100
    for kk = 1:server_number
        if node_master_tag(kk) == 1
            init_state(run_index,kk) = 1 ;
        else
            %init_state(run_index,kk) = random_select(state_number,1) ;
            init_state(run_index,kk) = 1 ;
        end
    end
end
run_step_rand = zeros(1, run_time) ;
% random simulation
total_ave_result_rand = zeros(1, total_step) ;
run_random_success_count = 0 ;
run_index = 0 ;
while run_random_success_count < run_time
    run_index = run_index + 1 ;
    fprintf('Random_Recursion_time:_%d/%d.\n', run_index,run_time);
    save_file_name = 'gittins_restless_simu.mat' ;
    gittins_random_simu( total_step, server_number , state_number, ...
        gittins_sel_num,trans_matrix,passive_trans_matrix, obs_matrix, ...
        reward_matrix,gittins_reward, save_file_name, restless, init_state(run_index,:) ,
            node_master_tag ) ;
    load(save_file_name) ;
    run_step_rand(run_index) = run_step ;
    if run_step < total_step
        continue ;
    else
        total_ave_result_rand = total_ave_result_rand + total_step_ave_rand ;
        run_random_success_count = run_random_success_count +1 ;
    end
end
% Draw restless bandit cost
%server_high_ratio_data_trans85
total_ave_result = zeros(1, total_step) ;
run_step_restless = zeros(1, run_time) ;
run_restless_success_count = 0 ;
run_index = 0 ;
while run_restless_success_count < run_time
```

```
    run_index = run_index + 1 ;
    fprintf('Restless_Recursion_time:_%d/%d.\n', run_index,run_time);
    save_file_name = 'gittins_restless_simu.mat' ;
    gittins_restless_simu( total_step, server_number , state_number, ...
       gittins_sel_num,trans_matrix,passive_trans_matrix, reward_matrix,gittins_reward, ...
       save_file_name, init_state(run_index,:), true , node_master_tag ) ;
    load(save_file_name) ;
    run_step_restless(run_index) = run_step ;

    if run_step < total_step
       continue;
    else
       run_restless_success_count = run_restless_success_count +1 ;
       total_ave_result = total_ave_result + total_step_ave ;
    end
end
%use cost to replace reward, need to negtiviate
total_result = (total_ave_result /run_restless_success_count );
total_result_rand = (total_ave_result_rand /run_random_success_count ) ;
hold on
%title('Cost Comparasion Between Existing Scheme and Proposed Scheme');
plot([1:total_step], total_result_rand, '--b');
hold on
plot([1:total_step], total_result, '-r');
grid on;
legend('Existing_Scheme','Proposed_Scheme');
xlabel('Step');
ylabel('Average_Cost');
for ttt = 1:show_step
  if sum(reward_hist2(:,ttt)) == reward_hist2(1,ttt)
     plot(ttt, reward_hist2(1,ttt) , 'ro');
  else
     for kkk = 1:gittins_sel_num
        plot(ttt, reward_hist2(kkk,ttt) , '*');
     end
  end
  hold on ;
end
xlabel('Step');
ylabel('Cost');
grid on;
subplot(2,1,2);
%title('Policy of Restless Bandit Scheme');
hold on
for ttt = 1:show_step
  if sum(policy_hist2(:,ttt)) == policy_hist2(1,ttt)
     plot(ttt, policy_hist2(1,ttt) , 'ro');
  else
     for kkk = 1:gittins_sel_num
        plot(ttt, policy_hist2(kkk,ttt) , '*');
```

```
    end
  end
  hold on ;
end
xlabel('Step');
ylabel('Policy');
grid on;
```