Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Min-Chieh Yu

Entitled

A SECURE MOBILE AGENT E-COMMERCE PROTOCOL

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Brian King
Chair

Paul Salama

Mohamed El-Sharkawy

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Brian King

Approved by: Brian King 12/8/2015

Head of the Departmental Graduate Program Date

A SECURE MOBILE AGENT E-COMMERCE PROTOCOL

A Thesis

Submitted to the Faculty

of

Purdue University

by

Min-Chieh Yu

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

December 2015

Purdue University

Indianapolis, Indiana

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere appreciate to my advisor Dr. King for the support of my research and study, for his patience, encouragement, and knowledge. His navigation and teaching helped me in all the time of research, study, and writing of this thesis. I could not have thought up having a better teacher and mentor for my Master study.

Also, I would like to thank my thesis committee: Prof. Salama, and Prof. El-Sharkawy, for their accurate and thoughtful opinion, comment, and assistance, but also for the question which provides me with an incentive to widen my research from various aspect.

Last but not the least, I would like to thank my family: my parents and to my uncle Peter and to my brother and sister and to my best friend for supporting me all the time.

TABLE OF CONTENTS

Page

LIST OF TABLES vi

LIST OF FIGURES vii

SYMBOLS viii

ABSTRACT xi

1 INTRODUCTION 1

2 BACKGROUND MATERIAL 3

2.1 Computer Security Objectives 3

2.2 Cryptography 4

2.2.1 Symmetric Key Cryptosystem 4

2.2.2 Public Key Cryptosystem 5

2.2.3 Key Exchange 6

2.2.4 EL Gamal Encryption 6

2.2.5 Hash value 6

2.2.6 Hash Chaining 7

2.2.7 Digital Signatures 7

2.2.8 Secret Sharing 8

2.2.9 (t, n) Threshold Signature 9

2.2.10 Environmental Key Encryption 11

2.3 Pairing Based Cryptography 13

2.4 Elliptic Curve Cryptography (ECC) 14

2.4.1 Elliptic Curve Arithmetic 14

2.4.2 Elliptic Curve El-Gamal 15

3 BACKGROUND OF MOBILE AGENT 17

3.1 Mobile Agent 17

Page

3.1.1 Basic Terminology 17
3.1.2 The Structure of Mobile Agent 17
3.2 Mobile Agent Migration 18
3.3 Mobile Agent Communication 18
3.4 Advantages of a Mobile Agent 20
3.4.1 Delegation of Task 20
3.4.2 Asynchronous Processing 20
3.4.3 Adaptable Server Interface 21
3.4.4 Code shipping versus Data shipping 21
3.5 Disadvantages of a Mobile Agent 22
3.5.1 Delegation of Tasks 22
3.5.2 Asynchronous Processing 22
3.5.3 *TTP* Issues 22
3.5.4 Price Discrimination 23
3.5.5 Profiling 23
3.5.6 Unauthorized Access 23
3.5.7 Unauthorized Dissemination 23
3.6 Possible Applications of Mobile Agent 24
3.6.1 Electronic Commerce 24
3.6.2 Information Retrieval 24
3.6.3 Network Management 24
4 RELATED WORK 25
4.1 Secure E-commerce using Mobile Agents on Untrusted Hosts 25
4.1.1 Hash Chaining 25
4.2 Secure Web Transaction with Anonymous Mobile Agent Protocol . 27
4.2.1 Setup 27
4.2.2 Registering a User 28
4.2.3 User releases Mobile Agents 28

Page

4.2.4 Host Authenticate and Execute Agent 29
4.2.5 Signature and Verification 30
4.2.6 Tracing and Revoking Malicious Users 31
5 SECURITY DESIGN OF MOBILE AGENT 32
6 MOBILE AGENTS IN ELECTRONIC COMMERCE 35
6.1 Motivation 35
6.2 General Setup 37
6.3 Registering with the TTP 38
6.4 Construction and Use of Single Mobile Agent 39
6.4.1 User Wishes to Use Single Mobile Agent E-commerce System 40
6.4.2 Generating the Agent 40
6.4.3 Host Authentication 44
6.4.4 Host executes Instruction Set 47
6.4.5 Host writes Offer in Dynamic Data 49
6.4.6 Host sends the Dynamic Data to Bulletin Board 51
6.4.7 User verifies the Signature and the Hash Chaining 51
6.5 Construction and Use of Multiple Mobile Agents 51
6.5.1 User wishes to use Multiple Mobile Agent Ecommerce System 52
6.5.2 Generating the Agent $\mathcal{A}_i$ 53
6.5.3 Host of Mobile Agents make the Decision 59
6.6 TTP Tracing and Revoking the Malicious Users 63
6.7 Bulletin Board Store Signature, Output, and Time 64
7 CONCLUSION 65
REFERENCES 66

LIST OF TABLES

Table		Page
2.1	Key Sizes for Encryption Algorithms in bits [16]	14
4.1	Symbols used in Singelee's hash chaining	26
4.2	Cryptographic notation used in Singelee's hash chaining	27
4.3	General Mobile Agent Data Structure	29
6.1	Data Structure of TTP	39
6.2	Data Structure of Single Mobile Agent	43
6.3	Single Mobile Agent Data Structure	45
6.4	Symbols used in Hash Chaining Protocol	50
6.5	Cryptographic notation used in the Hash Chaining Protocol	51
6.6	Data Structure of Dynamic Data	51
6.7	Data Structure of Multiple Mobile Agents	56
6.8	Multiple Mobile Agent Data Structure	58
6.9	Revocation List	64

LIST OF FIGURES

Figure		Page
2.1	Shamir Threshold Sharing Scheme	9
2.2	Threshold Signature Distribution Phase	12
2.3	Threshold Signature Construction Phase	12
3.1	Migration Scheme	19
6.1	Model for our scheme	36
6.2	User and TTP	38
6.3	Portion of Instruction Set is encrypted	42
6.4	Chaining Relation	50
6.5	Multiple Mobile Agents Scheme	52
6.6	(m, n) Threshold Secret Sharing Scheme	53
6.7	Host of agents upload the Dynamic Data to Bulletin Board	59
6.8	Host of agents download all the Dynamic Data from Bulletin Board	60
6.9	Host of agents upload the partial key with the best offer to Bulletin Board	61
6.10	Hosts downloads all the partial keys to Bulletin Board	62

SYMBOLS

$\mathcal{A}$	Single Mobile Agent
$\mathcal{A}_i$	Multiple Mobile Agent
$\mathcal{A}_{ID}$	Mobile Agent Identity
$\mathcal{B}$	Subset of Mobile Agents
BB	Bulletin Board
BB_{ID}	Bulletin Board Identity
C	Client
D	Session Parameters
DD	Dynamic Data
E	Elliptic Curve
ENC	Data Encryption
G	Point of Order q
$\mathbb{G}$	Group
H	Hosts
H_i	Host Identity
I	Targeted Item
K	Environmental Key
L	One-Off Public Key
M	Static Data of Mobile Agent
$\mathcal{M}$	Message
$\mathcal{O}$	Big O
P	Point
$\mathcal{Q}_{\mathcal{A}}$	One-Off Public Key
R	Threshold Signature Parameter

R_i	Threshold Signature Parameter
S	Servers
SIG	Digital Signature
SYM	Symmetric Key
T	Virtual Certificate Parameter
$\mathcal{T}$	Threshold Signature Verification Parameter
TTP	Trusted Third Party
U	Users
V_{cert}	Virtual Certificate
$\mathcal{V}$	Authentication Variable
W	Host Authentication Parameter
Y_U	Public Key of User
Y_{TTP}	Public Key of TTP
Z	Threshold Sharing Signature Parameter
$\mathbb{Z}$	Integer
c_i	Chaining Relation
e	Pairing Function
e_i	Encrypted Offer
$f(\cdot)$	Polynomial Function
g_t	Authentication Parameter
g_r	Authentication Parameter
g_z	Authentication Parameter
h_i	Hash Function
i	Index
j	Index
k_i	Secret Key of Signature Generation
m	Threshold Number
n	Total Number of Agents
o_i	Offer

q	Large Prime Number
r_i	Index of Polynomial Function
s_i	Share of Signing Key
$sigKey$	Threshold Signature
$sigKey_{\mathcal{A}_i}$	Partial Threshold Signature
t	Virtual Certificate Parameter
u_i	Share of Signing Key
v	Threshold Signature Parameter
x	One-Off Private Key
$x_{\mathcal{A}}$	One-Off Private Key
x_U	Private Key of User
x_{TTP}	Private Key of TTP
z	Host Authentication Parameters
α	Virtual Certificate Parameter
β	Virtual Certificate Parameter
Δ	Uploading Time
δ	Downloading Time
λ	Virtual Certificate Parameter
ψ	Virtual Certificate Parameter
σ	Verification Threshold Signature
ω	Secret Parameter
τ	Expiration Time
$\|\|$	Concatenation

ABSTRACT

Yu, Min-Chieh. M.S.E.C.E., Purdue University, December 2015. A Secure Mobile Agent E-commerce Protocol. Major Professor: Brian King.

There are many advantages of mobile agent such as delegation of tasks, asynchronous processing, adaptable service in interfaces, and code shipping. Mobile agents can be utilized in many areas such as electronic commerce, information retrieval, network management, etc. The main problem with mobile agents is security. The three basic security design goals of a system are confidentiality, integrity, and availability. The goal of this thesis concerns the property of secure purchasing by mobile agents. First present Jalal's anonymous authentication protocol. Next, we construct our single mobile agent protocol based on Jalal's authentication technique. Also, we add some addition cryptography techniques to make the data more secure during its migration. Lastly, we build a multiple mobile agent protocol based on the single mobile agent protocol. Here, the multiple mobile agents are capable to make the decision and purchase the item for user.

1. INTRODUCTION

Mobile agents are mobile code that migrate from one host to another in the network. A mobile agent can be used to solve many network problems such as reducing bandwidth, network traffic, and saving space. Mobile agents can be implemented to perform electronic trading, distributed information system, network management, social networking, etc. One of the important applications for a mobile agent will be e-commerce. The mobile agents can perform autonomously. Mobile agents are able to migrate more than once. After visiting the first host, the mobile agent might migrate further to other hosts in order to continue to collect data. The migration of mobile agents is usually done so the agent can access resources from the hosts in the network. When we compare mobile agents to the client-server approach we see that the mobile agents reduce the network load and processing time. This is because the code is moved to the data instead of the data being moved to the code. However, there are some concerns of mobile agents such as security issues. The goal of this paper is to construct a secure mobile agent e-commerce protocol.

Today, many people are overworked. They are working, exercising, etc. Waiting in long lines, to shop, searching for the "best price" for items with "best quality" can cause further frustrations. Shopping and searching for bargains today may be done by collecting information from websites or shopping store to store. Nevertheless, it can be time-consuming. Instead of client-server shopping approach, with the shopper controlling the client, one can use mobile agents, which can be more efficient and save time. We can use mobile agents to retrieve information from servers. User can simply set requirements for their mobile agents and let the mobile agent to complete the tasks. The user has no need to monitor mobile agents, he or she can focus on their work, things that mobile agents cannot accomplish. For example, Alice is preparing to attend an international conference. She needs to book the flight tickets, hotel, rental

car. In addition, she needs to schedule some meetings with the members and prepare the reports. Alice needs to finish all these items on her to-do list in one day. She assigns mobile agents to finish several of her tasks. Alice sets up the requirements such as the city, leaving date, departure time, returning date, location, brand, price, etc. She sends out the mobile agent to search via the network based on these requirements. Now, she has some more time to call the members, schedule meetings, and prepare her reports. Mobile agents will use Alice's detailed information about the flight ticket, hotel, and rental car that they purchased from the merchants. During the searching time, Alice has no need to worry about the tasks that she assigned to mobile agents.

In this thesis, we will focus on the privacy issues and provide technical solutions for mobile agents. In addition, we enable mobile agents not only to searching, collecting, comparing the particular item but also to purchase the particular item in e-commerce system. The organization of this thesis is as follows: In Chapter 2, we provide mathematical and cryptographic tools, as well as, the security mechanisms and analysis. In Chapter 3, we discuss the background of mobile agents. We describe the mobile agent system's basic terminology, background definition, and structure. Also, we discuss the advantages and disadvantages about the mobile agents. In addition, we address several applications of mobile agent. In Chapter 4, we address different previous secure mobile agent protocol designs and approaches. We first present Singelee's secure collection of dynamic data approaches. After that we describe Jalal's mobile agents electronic commerce protocol. In Chapter 5, we describe the secure design goal of mobile agent. We present the mobile agents privacy issues along with the concept of protocol and methods that we are using to construct a secure mobile agent e-commerce protocol. In Chapter 6, we present a secure protocol of mobile agent in e-commerce system. Further, we address the motivation, setup, registration, authentication, and tracing in this protocol. In Chapter 7, we conclude this thesis and discuss the open problems in the future work.

2. BACKGROUND MATERIAL

2.1 Computer Security Objectives

Typical computer security concerns includes: authentication, confidentiality, integrity, accountability, availability, and anonymity.

Integrity is the service that assures that the information has not been modified. Potential concerns include data that has been exchanged, removed, or added in time. Integrity contains two related concepts, which is data integrity and system integrity. The goal of data integrity is to assure that information is changed by authorized manner. The system integrity means a system is free from unauthorized manipulation.

Authentication is an important area, concerning the integrity of the identity of an entity, is a major concern and requirement in the security, especially in an electronic commerce application. We need a secure authentication between mobile agents and agencies. Authenticity demands that a mobile agent needs to be able to prove the claim that they make during the communication. This means that the mobile agent needs to be able to identify itself to others to allow agencies to determine whether it can be trusted. Furthermore, the agency needs to prove the authenticity of itself so that the agent can be sure whether it is on the correct agency.

Confidentiality implies restrictions on system access and disclosure [1]. This demands that information cannot be allowed unauthorized access between communication partners. Further, this covers two concepts: data confidentiality and privacy. The concept of data confidentiality is to protect the private information not be disclosed to unauthorized parties.

Accountability means the partners during any action need to be responsible and cannot deny responsibility later on. Accountability generates the requirement for actions of an party to be able traced in the system [1]. The reason is that a secure system must enable to trace the responsible party. A secure system must keep records of their activities to permit later analysis or to help the transaction disputes [1].

Availability demands that the authorized partner can access to resources reliable and rapidly ways. The goal of availability is to ensure that the service cannot be unauthorized prohibit. The system and service cannot deny to authorized users. For example, a malicious host can reject to execute a mobile agent. On the other hand, the malicious agency could deny letting the mobile agent to migrate to different platforms.

Anonymity means that users do not have to identify themselves. However, anonymity is contrast to authenticity. There are some solutions for anonymity.

2.2 Cryptography

Cryptography is concerns a collection of a variety of tools and security mechanisms, it also includes analyzing protocols to improve the secure communication in the network. Cryptography solves various needs in information security such as data confidentiality, data integrity, authentication, and non-repudiation [2].

2.2.1 Symmetric Key Cryptosystem

Symmetric key cryptosystem is a scheme using the same key for both encryption and decryption the text.

$$E_{Key}(P) = C$$

and,

$$D_{Key}(C) = P$$

where E_{Key} is the encryption using key Key to encrypt the plaintext P. Further, D_{Key} is the decryption using key Key to decrypt the ciphertext C. Common symmetric key schemes are the data encryption standard (DES) and advanced encryption standard (AES).

Data Encryption Standard (DES) DES is a symmetric key cryptosystem [3]. DES can encrypts a length of 64 bits plaintext P by using length of 56 bits key Key to compute a length of 64 bits ciphertext C. DES uses a Feistel-like encryption structure.

Advanced Encryption Standard (AES) AES is a iterated cipher which is based on permutation and substitution [4]. AES cipher can three different sizes of key which is 128, 192, or 256 bits. There are four basic operations in an AES round.

1. ByteSub (BS): This is a non-linear layer for resistance to differential and linear cryptanalysis attacks.
2. ShiftRow (SR):This is a linear mixing layer cause diffusion of the bits over multiple rounds.
3. MixColumn (MC): This is the layer that similar to shiftrow SR.
4. AddRoundKey (ARK): The round key is XORed with the result of all the layers.

DES is no longer considered secure due to its short key size, however triple DES (TDES) is still secure.

2.2.2 Public Key Cryptosystem

Public key cryptography, also known as asymmetric key cryptosystems. A Public key cryptosystem uses two separate keys, one is a public key and the other one is the secret key. The secret key is used to decrypt ciphertext or to create a digital signature. The public key is used to encrypt plaintext or to verify digital signature [5].

2.2.3 Key Exchange

The goal of a key exchange is to generate a common key between two participants by using a public channel. The two participants can use the common key in the cryptosystem. There are several methods that can be used to achieve a key exchange, two common key exchanges are public key encryption and Diffie-Hellman key exchange [6].

2.2.4 EL Gamal Encryption

EL Gamal encryption is a public key encryption cryptosystem [6]. The security of EL Gamal encryption security is related to the difficulty of computing discrete logarithm. In this setting p is a large prime and α a primitive element of $\mathbb{Z}_p$. Let a denote the secret key, the corresponding public key β is computed as $\beta \equiv \alpha^a (\bmod\ p)$. Made the information (p, α, β) public(see Algorithm 1 and Algorithm 2).

Algorithm 1 El-Gamal Encryption

1: Alice who wishes to send a message to Bob downloads Bob's publick key(p, α, β)
2: Alice chooses a random integer k and computes $r \equiv \alpha^k (\bmod p)$
3: Alice computes $t \equiv \beta^k (\bmod\ p)$
4: Alice sends (r, t) to Bob

Algorithm 2 El-Gamal Decryption

1: Bob computes $tr^{-a} \equiv m (\bmod\ p)$
2: Bob returns (m)

2.2.5 Hash value

A hash function is a one-way function [7]. It is a function which takes as an input a message of any size and computes a fixed length output. The output is called a

digest or hash. The function is one way function computed over the message, easy to compute and computationally hard to invert. Hash function $h(\cdot)$ maps the any arbitrary size of byte to a fixed size sequence. There is an important feature of hash function $h(\cdot)$ which is it is hard to find the original value when only the hash value is known. In addition, it is also to find a different value that will be collision after the hash value. In other words, if two hash values are the same that is the two original values are the same value. Some other properties a hash function should satisfy:

Collision Resistance: It is computationally hard to find two inputs a, b where $a \neq b$ and $h(a) = h(b)$.

One way function: Given the hash $h(x)$ is hard to find the input x.

Ease of Computation Given x it is easy to compute $h(x)$.

2.2.6 Hash Chaining

A hash chaining protocol has seven security properties: confidentiality, nonrepudiation, strong forward integrity, publicly verifiable chain, insertion resilience, deletion resilience, and truncation resilience [8]. Using data collection by hash chaining is one of the ways to prevent malicious hosts change the information that a visiting agent has already collected from other hosts. The author is using the protocol that the user, denoted by S_0, sends out one mobile agent to n different hosts $S_1 \cdots S_n$. the protocol is used to protect the information that the agents collected in previous hosts. We discuss this in greater detail in Section 4.1.1.

2.2.7 Digital Signatures

A digital signature is used for authentication, integrity, and non-reputation [2]. A digital signature binds the data and the secret key. There are two common signature schemes, RSA signatures and EL-Gamal signatures [9], [10]. Also, there is a hash and signing method to instead of the RSA signatures and EL-Gamal signature. Here

we are using the hash and signing in our later protocol. The hash function $h(\cdot)$ is made public. Using the $(M, sign(h(M)))$ and the public keys are inputs to the digital signature verification algorithm (see Algorithm 4).

Algorithm 3 Hashing and Signing

1: Hash the message M, $h(M)$
2: Calculates the signed message $sig(h(M))$
3: Uses $sig(h(M))$ as the signature of the message M
4: Returns $(M, sign(h(M)))$

Algorithm 4 Verifcation of

1: Given $(M', sign(h(M)))$
2: Hash the message M', $h(M')$
3: Calculate Verify($sig(h(M), \text{public key})$
4: **if** Verify($sig(h(M), \text{public key})$ equals $h(M')$ **then**
5: Return true
6: **else**
7: Return false

2.2.8 Secret Sharing

Secret sharing is a method to share out the secret key in such a way that only authorized subsets can reconstruct the key [11, 12]. Let t, n be positive integers with $t \leq n$. Then a (t, n) threshold sharing scheme scheme is a method that is used to share a secret key K among a set of n parties, such that any subset of t or more parties can reconstruct the secret key K, but any subset $t - 1$ or less parties cannot

determine any information concerning the secret key K. This can be achieved by using Shamir Secret Sharing:

$$f(x) = K + \sum_{j=1}^{t-1} a_j x^j \bmod p. \tag{2.1}$$

For each i $(i = 1, ..., n)$ let x_i be distinct public nonzero values. Then participant P_i receives the share $f(x_i) \bmod p$. Observe that substituting $x = 0$ in equation (2.1),

$$f(x) = K.$$

Thus if $P_{i_1}, ..., P_{i_t}$ wish to reconstruct the secret then,

$$K = \sum_{j=1}^{t} f(x_{i_j}) \prod_{\substack{w=1 \\ w \neq j}}^{t}.$$

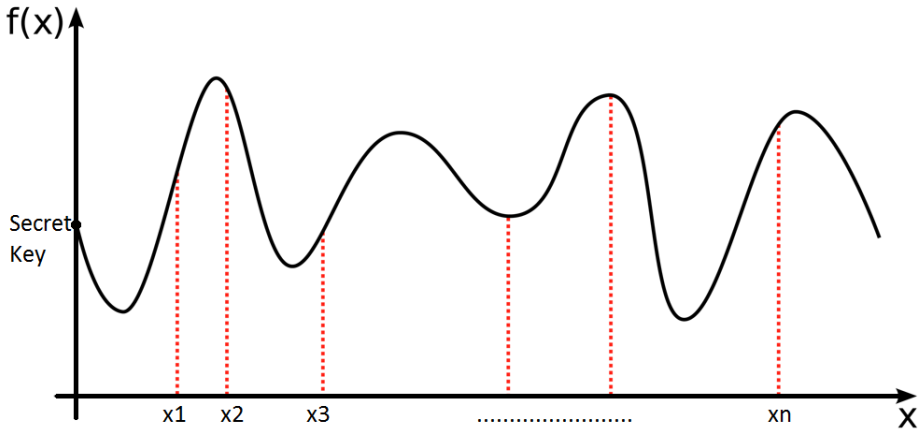

Fig. 2.1.: Shamir Threshold Sharing Scheme

2.2.9 (t, n) Threshold Signature

Threshold signature scheme is used sharing mechanism to protect the signing key [13]. Here any subset of t or more parties can construct a signore to some message m and any subset of $t - 1$ or less cannot construct a signature.

The El-Gamal Threshold signature scheme is a threshold secret sharing scheme that can be used to construct El-Gamal signatures [14].

Initialization: see Algorithm 5

Algorithm 5 Initialization of (t, n) El Gamal threshold signature

1: TTP choose a hash function h
2: TTP choose a secure elliptic curve $\mathcal{E}$ so that it has a subgroup of order q where q is suitably large
3: User U decides the polynomial $f(x)$ of degree at most $(t-1)$ with public coordinates associated with each agents $\mathcal{A}_i$
4: The group of secret key $k = f(0)$
5: The group of public key $Y = kG \bmod p$
6: For each i, the share s_i is calculated as $s_i = u_i + f(x_i)$ where $u_i \in_R \mathbb{Z}_q \setminus 0$, and the x_i are public $(i = 1, 2, ..., n)$. s_i and U_i are placed in the agent $\mathcal{A}_i$'s data structure. All values $x_1, \ldots, x_n$ are also placed on the data structure.
7: User U calculates the public elements Y_i and Z_i where $Y_i = s_i G \bmod p$ and $Z_i = u_i G \bmod p$ these are associated with mobile agent $\mathcal{A}_i \in \mathcal{A}$. The points $Y_1, \ldots, Y_n, Z_1, \ldots, Z_n$ are placed in agent $\mathcal{A}_i$'s data structure
8: The parameters (h, q, G, Y) together with $\{(Y_i, Z_i) | \mathcal{A}_i \in \mathcal{A}\}$

Signing: see Algorithm 6 Partial Signature Generation of (t, n) ElGamal threshold signature.

Verification: see Algorithm 7 Verification (t, n) ElGamal threshold signature.

Algorithm 6 Partial Signature Generation of (t, n) ElGamal threshold signature

1: For each i, agent $\mathcal{A}_i$ generates a secret key $k_i \leq q-1$. Now compute $R_i = k_i G \bmod p$. The host of agent $\mathcal{A}_i$ posts R_i on BB
2: agent $\mathcal{A}_i$ downloads $R_1, ..., R_n$ from BB
3: Once active agent subset $\mathcal{B} \subset \mathcal{A}$ is known, each agent $\mathcal{A}_i \in \mathcal{B}$ computes R and e, where $R = \sum_{\mathcal{A}_i \in \mathcal{B}} R_i$ and $e \equiv h(\ \textit{best offer}, R) \bmod q$
4: Agents $\mathcal{A}_i$ generates their partial signature c_i by (s_i, k_i) Note $c_i \equiv s_i \prod_{A_{\mathcal{A}_i} \in \mathcal{B}; j \neq i} \frac{-x_j}{x_i - x_j} + k_i e \bmod q$
5: Agent $\mathcal{A}_i$ sends (offer, c_i) on the BB.
6: Agent $\mathcal{A}_i$ downloads all partial signatures $(offer, c_j)$ from $\mathcal{B}$ that are on the bulletin board BB and verify the partial signatures by checking $c_i G$
7: $c_i G \equiv \prod_{\mathcal{A}_j \in \mathcal{B}; j \neq i} \frac{-x_j}{x_i - x_j} Y_i + e R_i$
8: If All partial signature are congruence
9: Computes $\sigma = \sum_{\mathcal{A}_i \in \mathcal{B}} c_i \bmod q$
10: The $(\mathcal{B}, R, \sigma)$ is the signature of best offer

Algorithm 7 Verification (t, n) ElGamal threshold signature

$$T = \sum_{\mathcal{A}_i \in \mathcal{B}} \prod_{\substack{\mathcal{A}_j \in \mathcal{B} \\ j \neq i}} \frac{-x_j}{x_i - x_j} Z_i$$

Check $\sigma G \stackrel{?}{\equiv} Y + T + eR$ where $e \equiv h(\ \text{best offer}, R) \bmod q$

2.2.10 Environmental Key Encryption

The goal of environmental key encryption is to let the host remain clueless about the message unless the host possesses a specific environmental item [8]. Environmental key encryption can allow the host determine the item if it has the key otherwise making the host clueless about the item. In this way, this could possibly protect the private code and data of mobile agent from being analyses by the hosts. Environmental key encryption use a cipher item and a method to search environment for the data needed to generate the key decryption. The host can generate the key to decipher

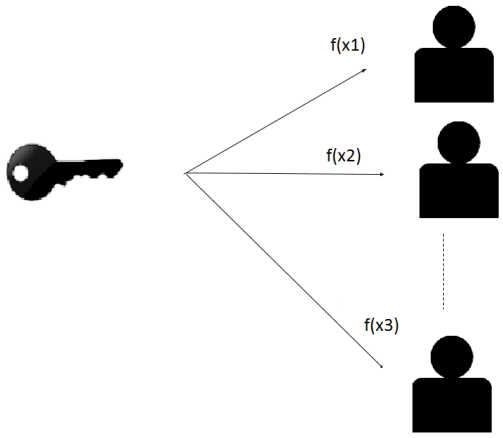

Fig. 2.2.: Threshold Signature Distribution Phase

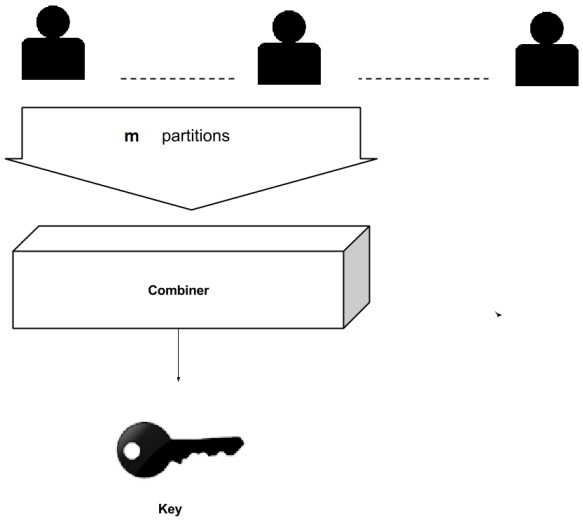

Fig. 2.3.: Threshold Signature Construction Phase

the item when the cipher item is found. Without this key, the host cannot know the item that the user is searching for. The general way is to give the mobile agent the hash value of the item or information and compare this hash value to all the hashes

values of items the host possesses. If they match, the hash value is used as a key to decrypt the rest of the information to be process. The goal here is as follows:

$$KEY = h(ITEM)$$

$$X = h(h(ITEM))$$

If $h(h(ITEM)) = X$ then reveal the key to the host and decrypt rest of the code of mobile agent.

Algorithm 8 Environmental Key Encryption

1: Message M is encrypted by $ITEM$

2: Encrypt the $ITEM$ by hash function $h(\cdot)$,

$$KEY = h(ITEM)$$

3: Encrypt the KEY by second hash,

$$X = h(h(ITEM))$$

4: Return X

Algorithm 9 Environmental Key Decryption

1: Computes $Y = h(h(PRODUCT))$

2: **if** $Y == X$ **then**

3: **if** $PRODUCT{=}{=}ITEM$ **then**

4: Decrypt the message M by $PRODUCT$

2.3 Pairing Based Cryptography

Let $\mathbb{G}$ be an additive ableian group of prime order q and $\mathbb{G}_1$ to a multiplicative group of order q. Pairing based cryptography [15] is a cryptographic tool that describes a mapping e which maps elements from $\mathbb{G} \times \mathbb{G}$ to a multiplicative group $\mathbb{G}_1$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, that satisfies the following properties:

Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$

Non-degeneracy: $e(P, Q) \neq 1$

Computability: There exist an efficient algorithm that can compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$

2.4 Elliptic Curve Cryptography (ECC)

Here one constructs an elliptic curve E such that number of points on E, represented by $\#E$ has a large prime divisor q. Then there is a natural addition on E, so that E is an additive abelian group. Thus there will be a subgroup of E of order q. Elliptic Curve Cryptography (ECC) [6] is where one construct cryptographic primitives over the elliptic curve E. Many of the cryptographic primitives that are based on the difficulty of the discrete logarithm , can be redefined over an elliptic curve. One of the ECC advantages is that it can use a smaller key size to make the same level of security compared to the classical cryptosystems. Curve E is chosen so that it has a subgroup of large prime order q.

Table 2.1.: Key Sizes for Encryption Algorithms in bits [16]

Year	Symmetric	Factoring (modulus)	Discrete Log Key	Discrete Log Group	Elliptic Curve	Hash
2033	96	2698	169	2698	181	191
2034	96	2768	171	2768	182	192
2035	97	2840	172	2840	184	194
2036	98	2912	173	2912	185	195
2037	99	2986	175	2986	186	197

2.4.1 Elliptic Curve Arithmetic

One of the ECC advantages is that it can use a smaller key size to make the same level of security compared to the classical cryptosystems. Curve E is chosen so that

it has a subgroup of large prime order q. An Elliptic curve E can be defined over a prime field $\mathbb{F}_p$ over $\mathbb{Z}_p$. Then all points (x, y) that satisfy:

$$y = x^3 + ax + b$$

where a, b $\in \mathbb{F}_p$, together with the special point $\mathcal{O}$, which is called the point of infinity, belong to E. As mentioned early there is a natural addition on E.

Negative of a point: $P = (x_P, y_P)$ $-P = (x_P, -y_P)$

Doubling a point: Doubling one point P, where $P = (x, y)$. $R = 2P = (x', y')$

$$s = \frac{3(x)^2+a}{2y} \text{ where } y \neq 0.$$
$$x' = s^2 - 2x \text{ and}$$
$$y' = -y + s(x - x')$$

Adding two distinct points P and Q: Adding two points P and Q, where $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)R = P + Q = (x_R, y_R)$.

$$s = \frac{y_P - y_Q}{x_P - x_Q}$$
$$x_R = s^2 - x_P - x_Q \text{ and}$$
$$y_R = -y_P + s(x_P - x_R)$$

where s is the slope of the line through P, Q.

Now, we discuss the computing of a scalar multiple, kP where k is an integer with $0 < k < q$ and P is a point on ECC(see Algorithm 10).

2.4.2 Elliptic Curve El-Gamal

Suppose G is a generator of the prime subgroup of E.Let $y + kG$ then k is the secret key and Y is a public key.

Now suppose M is a plaintext message and encrypted M by adding to kY where Y is a public key and k is a random scalar. Here C_1 and C_2 are denote as a ciphertext (see Algorithm 11 and Algorithm 12) [6].

Algorithm 10 Scalar Multiple kP in ECC

1: $Q \leftarrow \mathcal{O}$
2: **for** i from $n-1$ down to 0 **do**
3: $\quad Q \leftarrow 2Q$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad Q \leftarrow Q + P$
6: Returns Q

Algorithm 11 Elliptic Curve El-Gamal Encryption

1: Selects private key x and computes $Y = xG$
2: Encrypt M where M belongs to the prime subgroup of E
3: Selects k and Computes $R = kG$
4: The ciphertext is $(C_1, C_2) = (kG, M + kY)$
5: Returns (C_1, C_2)

Algorithm 12 Elliptic Curve El-Gamal Decryption

1: Receives the ciphertext (C_1, C_2)
2: Computes $-xC_1 + C_2 = M$ by using her secret key

3. BACKGROUND OF MOBILE AGENT

3.1 Mobile Agent

Mobile agents are software entities which can act autonomously like a user. Mobile agents can migrate to different platforms in order to find out the specific items or services that are requested by the user. The mobile agents can do browsing, comparing, and negotiating for the user during the searching. Users do not need to monitor the workstation to wait for the result or supervise the mobile agents until mobile agents get the job done. Mobile agents can save bandwidth, avoid network traffic, and react to the dynamic environment fast.

3.1.1 Basic Terminology

Agent migration refers to transferring a mobile agent from one host to another. A mobile agent is a software program and be executed by systems. The user who starts the agent is called agent's owner. Agent's codes will be held by the host. The host executes the agent's code and provides some functionality for agent communication, security and migration. The host is reachable by URL.

3.1.2 The Structure of Mobile Agent

Mobile agent consists of two components: code and data. The code contains the mobile agent instructions serve as the logic of mobile agent. Data can be divided into two parts which are static data and dynamic data. Static data is restricted to be changed, on the other hand, dynamic data is allowed to be changed.

Static Data The static part is unchangeable during the migration. Public static data is for all the visited hosts to read such as the name of the flight. Private static data is only read by the agent itself.

Dynamic Data The dynamic part of data is modifiable during the migration. One example of dynamic data is the list of visited hosts with the price information.

3.2 Mobile Agent Migration

The Migration Framework is (see Figure 3.1) [7]:

1. Initialize agent. The user starts with the special command, migration command, and announce the intention to migrate to another host.
2. Capture data and state. Agents date and state information is written to agent.
3. Transfer the agent. Transfer the agent to the receiver host.
4. Receive the agent. The receiver host checks if the agent can be accepted. The receiver host verifies the information about the user and the sender host and rejects the agent which is unknown or not trusted.
5. Execute the agent. The variables and execution state are restored from the agent.
6. Start agent execution.

3.3 Mobile Agent Communication

Mobile agents need to work together to solve a single task when the user sends out more than one mobile agents. In this case, it is important to have a technique to solve the problem of agent coordination and communication. For example, mobile

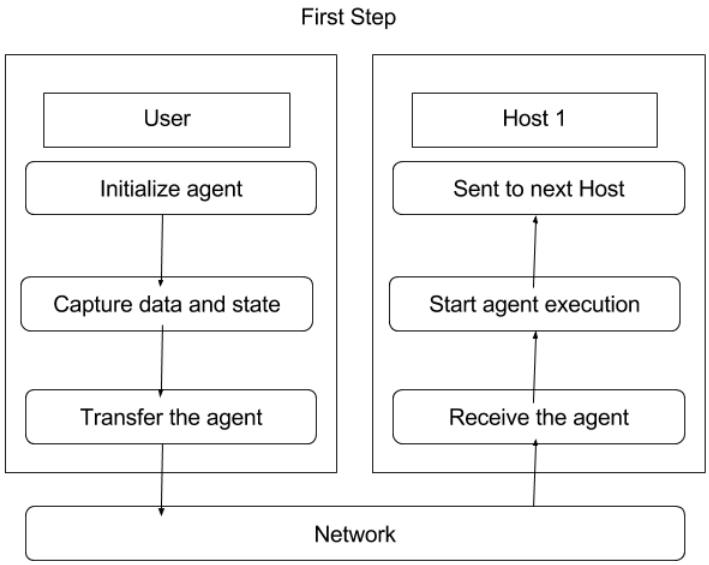

Fig. 3.1.: Migration Scheme

agents need to synchronize their results, to exchange the data that they collected separately.

There are two types of communication techniques [7]: message passing and information space. We use the later one. We focus the discussion on the problem of communication, in particular, how frequently do migrating mobile agents migrating communicate to each other in a reliable way.

Information Space Information space solves the communication problem between the agents. There is no need for writer agents and reader agents to synchronize to communicate. The goal is to write and read the information from the information space. Mobile agents need to register their requirements with the information space if they want to make sure that they get notified when new information is published on the information space.

Blackboard System All the information on blackboard is stored under an identifier. This allow the reader agents to know which agent wrote the information on the blackboard. Blackboard system is good for collaborating softwares to exchange information among agents. All the agents must have some common knowledge so that they can share the information on the blackboard.

3.4 Advantages of a Mobile Agent

There are some advantages of mobile agent such as delegation of tasks, asynchronous processing, adaptable service interfaces, and code shipping versus data shipping [7].

3.4.1 Delegation of Task

A mobile agent represent the user to delegate tasks. Instead of using computer systems as interactive tools that only work under direct control by users, we are using mobile agents that can take care of the task on their own. As a result, the user can spend time and attention to other things. Mobile agents are a good way to deal with the information. For example, Alice is going to arrange a business trip where she needs to book a flight ticket and a hotel. She dispatches a mobile agent to deal with the bookings so that she can have more time to arrange the meeting with her business partners. After setting up the mobile agent, she can go offline and let the mobile agent find the best offer with the information regarding flights and hotel then help her to purchase the products.

3.4.2 Asynchronous Processing

Once mobile agents have been initialized and set up for a task, they leave the user's computer system and migrate through the Internet. Only the first migration requires a network connection. This is also a feature that allows the user to operate

with their portable devices. The user can start the mobile agents from mobile devices such as personal digital assistant and notebook computer which have the feature of mobility and portability for mobile user to connect to the network. However, mobile devices offer limited bandwidth, limited transmission speed, higher cost, lower computing power, and volatile network links. Because mobile agents become independent of the process after the user sent out the mobile agent, it will be more stable than client-server-based applications. Mobile agents can operate asynchronously and autonomously without the user.

3.4.3 Adaptable Server Interface

In a distributed system application service interfaces are usually like a collection of functions [7]. Most of the interface functions are more or less primitive. On the other hand, mobile agents can simulate a highly specialized interface for the user while talking to each host in its own language to allow hosts to become simpler and more generalized. Mobile agents can execute environment and react autonomously to changes. Also, they have the ability to distribute themselves among the hosts and collect information during the migration.

3.4.4 Code shipping versus Data shipping

A mobile agent stands in close relationship to adaptable service interfaces which offer only primitive functions to access databases. Instead of transferring data to the user, where it will be executed, filtered, and cause a new request, this code can be transferred to the location of the data by mobile agent. Only the relevant data is sent back to the user, which reduces network traffic and save time because of the filtering.

3.5 Disadvantages of a Mobile Agent

The main problem with mobile agent is security such as delegation of tasks, asynchronous processing, TTP issues, price discrimination, profiling, unauthorized access, and unauthorized dissemination.

3.5.1 Delegation of Tasks

Mobile agents are autonomous and migrate without direct control by the user, it is hard to govern mobile agents, in terms of what is allowed or not allowed to do during the process. There is an important issue of delegation of authentication that the mobile agent carries proofs of its authorizations which are private information and knowledge such as passwords.

3.5.2 Asynchronous Processing

Mobile agents can operate asynchronously and autonomously without the user which means the host gains full control over the agents which is possible that the code of mobile agent could be modified. For example, the mobile agent could be modified to attack on other hosts. In addition, the collected data from previous hosts or dynamic information could also be read, copy, or delete the information from the current host.

3.5.3 TTP Issues

It is difficult to disseminate mobile agent execution environments to large numbers of TTP services. In addition, TTP providers in the face of security concerns. Also, there is a commercial issue whether the TTP supports the computational load of mobile agent and to permit users ability to customize server behavior [3].

3.5.4 Price Discrimination

Host could charge different price for the same product based on the client profile. For example, if the host knows that some product is of great interest to one sort of clients they may charge more money for this product than other clients. Take Amazon for example, in 2000, Amazon started to charge customers different prices for same DVD title. Companies usually divide their potential customers into similar groups based on customer's characteristic.

3.5.5 Profiling

Profiling is the process of discovering patterns in data that can be used to identify a user and application of correlated data. For example, the hosts can obtain information of the user and modify their offers regarding the preference of the user in e-commerce environment.

3.5.6 Unauthorized Access

The attacker could attack the machine where stores sensitive information about principles without consent if the machine without appropriate protection. For instance, the malicious host can be listening to transferred information over the network such as emails, files, messages and gather the information flowing in the network.

3.5.7 Unauthorized Dissemination

Malicious hosts can transfer the collected information to other parties without consent of the user of this information. For example, a mobile agent $\mathcal{A}$ collects the information that it receives about another mobile agent $\mathcal{B}$. Mobile agent $\mathcal{A}$ can transfer information about mobile agent $\mathcal{B}$ to another mobile agent $\mathcal{C}$.

3.6 Possible Applications of Mobile Agent

Mobile agent build distributed application more efficiently. We can look at difference application to understand that mobile agent enable a new level of networked software.

3.6.1 Electronic Commerce

Mobile agent support for automation and higher coverage of information from different resources. The customer only need to state what he or she wants and no need to manually to implement how this is done. Mobile agent offer delegation and asynchronous task execution to process the requirement for the user without any interaction.

3.6.2 Information Retrieval

Instead of moving large amounts of data to a single point where it is searched, information retrieval moves the data searching code to the data. This is a new way for the search engine, Web pages will analyzed locally by the mobile agent that was sent to the Web Server. Mobile agents will be able to unify that interface from the client's perspective and offer a higher and well-adapted level of functionality. The main idea is to move code close to a large database instead of transferring lots of data to a client [7].

3.6.3 Network Management

Network management involves collecting and analyzing data from the devices to monitor and control the devices [17]. Mobile agents solve the problems of performance management such as network delay and information bottleneck by its ability of migrating from node to node. Mobile agents can analyze the node locally instead of analyzing all the collected data from the nodes in the management station.

4. RELATED WORK

We now discuss some related work concerning the use of mobile agent in e-commerce systems. In our work, our goal is to enable the mobile agents to purchase the best offer of the targeted item. Achieving this while maintaining privacy.

We will first discuss the work of Singelee [8]. Later, we will discuss a protocol constructed by Jalal [18]. This latter protocol inspired much of our work.

4.1 Secure E-commerce using Mobile Agents on Untrusted Hosts

Here we discuss some work concerning secure and efficient mobile agent application in e-commerce.

4.1.1 Hash Chaining

Secure data collection by using hash chaining [8] is one way to prevent malicious hosts from changing the information that an agent has already collected from other hosts. Here the user, denoted by S_0, sends out one mobile agent to n different hosts, denoted by $(S_1 \cdots S_n)$. To protect the information that the agent collects from previous hosts as it migrates, one can use the hash chaining protocol protocol [8]. There are several steps: hash chaining, encapsulating the offer and the transmission protocol.

The user S_0 of the mobile agent initializes the "first offer" o_0 by assigning at the initial information such as the identity of the mobile agent. Using the hash function $h(\cdot)$ to compute the initial chaining relation h_0 with the initial offer o_0 and the identity of the first host S_1. The user S_0 selects a random number r_0 and encrypts the random number r_0 and the public key of the user (S_0). Next, the user S_0 constructs the

encapsulated offer O_0 by signing the encryption and the chaining relation h_0, see Equation 4.1. Then they send *encapsulated offer* O_0 to the first host S_1. All the hosts S_i, for $i = 1,\cdots,n$, performs nearly the same steps. First, the host S_i calculates the chaining relation h_i using the encapsulated offer O_{i-1}, see Equation 4.2, which they received from the previous host S_{i-1} and the identity of the next host S_{i+1}. Next, the host will select a random number r_i and use this random number r_i to encrypt the offer o_i with the public key of the user S_0. Finally, the host S_i constructs the encapsulated offer O_i by signing the encrypted(o_i, r_i), the chaining relation h_i, and all the encapsulated offers $O_1 \cdots O_{i-1}$ from the previous hosts sent to the next host S_{i+1}. At the conclusion, the last host S_n will sent the encryption(o_n, r_n), chaining relation h_n, and all the encapsulated offers $O_1 \cdots O_{n-1}$ from the previous hosts sent to the user S_0, i.e. $S_{n+1} = S_0$.

Chaining Relation: Here o_i represents the of host S_i. In the case $i = 0$, $o_0 = \mathcal{A}_{ID}$.

$$h_0 = h(o_0, S_1) \tag{4.1}$$

$$h_i = h(O_{i-1}, S_{i+1}), 1 \leq i \leq n \tag{4.2}$$

Table 4.1.: Symbols used in Singelee's hash chaining

$S_0 = S_{n+1}$	Identity of the user
$S_i, 1 \leq i \leq n$	Identity of host i
o_0	Initial information(e.g. identity of the agent $\mathcal{A}_{ID}$)
o_i, $1 \leq i \leq n$	Offer from host i
O_i, $1 \leq i \leq n$	Encrypted offer from host i H_i
$O_0, O_1, O_2, \cdots, O_n$	Chain of encapsulated offers

Encapsulated Offer: In this case, host S_i, for $0 \leq i \leq n$, encapsulates its offer by:

$$O_i = SIG_i(ENC_0(r_i, o_i)h_i).$$

Table 4.2.: Cryptographic notation used in Singelee's hash chaining

r_i	Random number generated by host S_i
$ENC_0(m)$	Message m encrypted with the public key of user S_0
$SIG_i(m)$	Message m digitally signed by host S_i
$H(m)$	Hash value calculated on message m
$S_i \rightarrow S_{i+1} : m$	Message m is sent from host S_i to host S_{i+1}

Transmission Protocol: Here hiost S_i transmits to host S_{i+1}.

$$S_i \rightarrow S_{i+1} : \{o_k, O_k | 0 \leq k \leq i\}, 0 \leq i \leq n\}$$

4.2 Secure Web Transaction with Anonymous Mobile Agent Protocol

Jalal's protocol [18], the e-commerce systems consists of users U, servers S, and a trusted third party TTP. The mobile agent that is prepared by the user U is denoted by $\mathcal{A}$.

4.2.1 Setup

The TTP generates a secure elliptic curve E, which a subgroup $\mathbb{G}_1$ of order p, where p is suitably large prime. The TTP then computes a generator G for the subgroup of E of order q where q is prime. Let $\mathbb{G}_2$ denotes a multiplicative group of order q. We assume there is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Further, there is secure cryptographic hash function denoted by $h(\cdot)$. The TTP publishes $E, \mathbb{G}_1, \mathbb{G}_2, G$, and h.

Next the TTP selects a random number $s \in_R \mathbb{Z}_q^*$ then publishes sG. Each user U generates a private key k_U and public key $Y_U = k_U G$ to use in a digital signature scheme for authentication in the system. The TTP also generates a private key k_{TTP} and a public key $Y_{TTP} = k_{TTP} G$. We denote is $H_0 = H(R_x(Y_{TTP})||R_y(Y_{TTP}))$ where

$R_x(Y_{TTP})$ is the x coordinate of Y_{TTP} and the $R_y(Y_{TTP})$ is the y coordinate of Y_{TTP} and $||$ is the concatenation. Each user U has an identity ID_U which is bounded to their public key Y_U by a trusted party Certificate Authority CA.

4.2.2 Registering a User

The user U registers with TTP then receives secret parameters γ_U and A_U from the TTP, where γ_U and A_U are computed as described in Algorithm 13. The TTP records the public key Y_U, identity ID_U, and the secret parameters:γ_U and A_U of the user C into the database (see Algorithm 13).

Algorithm 13 The TTP registration of user U [8]

1: The TTP performs the following operations.
2: Selects random $\omega_U \in_R Z_q^*$
3: Computes $A_U = \omega_U G - Y_U$
4: Computes $\gamma_U = H_0\omega_U + R_x(A_U)k_{TTP} \bmod q$
5: Stores $\gamma_U, A_U, Y_U, \omega_U$, and ID_U in TTP database
6: Returns A_U and γ_U to user U

4.2.3 User releases Mobile Agents

The user U prepares the agent's data structure (see Table 4.3) and incorporates the requirements of its intended "targeted item" into the instruction set of the data structure of mobile agent. The user then dispatches the mobile agent. The mobile agent will work autonomously to locate the targeted item and return the best offer for that item that as required (see Algorithm 14).

Table 4.3.: General Mobile Agent Data Structure

Session Number	User's Request
Routing Table	
Time Stamp	
Instruction Set	
Static Data	
Dynamic Data	
Virtual Certificate V_{cert}	

Algorithm 14 User U Prepares Mobile Agent [8]

The user performs the following operations.

1: Prepares request and routing table of the mobile agent

2: Embeds the code and instruction on the mobile agent

3: Uploads the static data on the mobile agent

4: Prepares the virtual certificate

5: Selects $\psi \in_R \mathbb{Z}_q^*$

6: Computes elliptic curve point $T = \psi G$

7: Selects $t \in_R \mathbb{Z}_q^*$ and computes tG

8: Computes $z = tR_x(A_C) \bmod q$ and computes zG

9: Computes $B = t(A_C + T)$

10: Computes $\lambda = \gamma_C t + (\psi t - k_C t + xH(M))H_0 \bmod q$

11: Computes tx and $tL = txG$

12: Returns the $V_{cert} = (zG, tG, B, \lambda, L, tx, tL)$

4.2.4 Host Authenticate and Execute Agent

When a server S receives the mobile agent, it will check if the mobile agent is authentic or not, before it executes the instruction set on the server site. If the mobile agent $\mathcal{A}$ is not authentic, the server S will reports this mobile agent to the

TTP. Further, if the mobile agent $\mathcal{A}$ is authentic but the instruction set turns out to be malicious then the TTP can trace the user U who dispatch this malicious mobile agent. There are several steps for the host to authenticate the mobile agent. Recall the virtual certificate parameters:$zG, tG, B, \lambda, L, tx$, and tL.

First, the server S computes $\lambda_1 G$ and zY_{TTP} from Equation (4.3)

$$\lambda G \stackrel{?}{=} H_0(B + H(M)L) + zY_{TTP}. \tag{4.3}$$

The server S now computes tY_{TTP}, the passed parameter tG,and the public information G, and Y_{TTP} to check the validity of the relation (see Equation (4.4), and Equation (4.5)).

$$W = \lambda G - H_0(B + H(M)L). \tag{4.4}$$

$$e(W, G) \stackrel{?}{=} e(zG, Y_{TTP}). \tag{4.5}$$

If the Equation (4.5) return a yes then the mobile agent $\mathcal{A}$ is authenticated. The server S will then execute the instruction set of mobile agent $\mathcal{A}$ and the request. The server S will store the result and the output in the dynamic data structure of mobile agent $\mathcal{A}$.

4.2.5 Signature and Verification

The server S will sign its output S_{bid} so that no one can forge the output in dynamic data structure. The server S signs the output $Sign_S(m)$ where $m = H(H(D)||ID_S||S_{bid})$. The signature $Sig = \{K, r\}$ where $k \in_R \mathbb{Z}_q^*$ and computes $K = kG$. The mobile agent also computes $r = H(H(m)||H(D)||H(V_{cert,C}k + R_x(K)tx \bmod q))$ so that the S_{bid} and the authentication parameters $V_{cert,U}$ have been bound into a signature (see Algorithm 15).

Algorithm 15 Signature Verification [8]

1: INPUT: $m = H(ID_S, S_{bid})$, data structure D, and $V_{cert,U}$
2: **if** $Sign_S(m)$ is not a valid signature **then**
3: Return FALSE
4: **if** The routing table is listed and $ID_S \notin routingtable$ **then**
5: Return FALSE
6: **if** $e(tL, G) \neq e(L, tG)$ **then**
7: Return FALSE
8: Computes rG
9: **if** $rG \neq H(H(m)||H(D)||H(V_{cert,C}))K + R_x(K)tL$ **then**
10: Return FALSE
11: Return TRUE

4.2.6 Tracing and Revoking Malicious Users

If TTP receives complains about mobile agent $\mathcal{A}$ from server S. The servers reports to the TTP that they have received an authenticated agent which is malicious. The TTP traces the malicious user U by using session parameter and revokes the user U(see Algorithm 16).

Algorithm 16 Tracing and Revoking Malicious Users [8]

1: Server S complains to TTP about malicious mobile agents and sends the session parameters of the mobile agent.
2: TTP checks its database for an $\mathcal{A}$ that satisfies $R_x(A_U)tG = zG$
3: TTP publishes the corresponding A_U in the network
4: Each server S adds the published A_U to its revocation list, so that this malicious user is not authenticated anymore.

5. SECURITY DESIGN OF MOBILE AGENT

Confidentiality, integrity, and availability are the three basic components of security. There are a number of security services from these three compoenets that impact our work, they include: authentication, access control, nonrepudiation, tracibility, and revocation and accountability [18].

Confidentiality: Confidentiality concerns that private and sensitive information should not be disclosed to unauthorized individuals. We need to protect: the data between mobile agent to host, the data between host to mobile agent, and the data between blackboard to host. It is possible that the user data can be attacked by a malicious host. For example, when the host is executing a mobile agent which carries some private information about the user could be hack by malicious host. On the other hand, the host could be attacked by the agent with malicious code which is sent by the user, so that the user can attain sensitive documentation from the host. The host is the one that has responsibility to send the mobile agent to the blackboard. Thus the data and information on the blackboard could be manipulated by the host.

Integrity: Integrity includes data integrity, and system integrity [19]. By integrity we are concerned with protecting data and information so that it is not modified, nor is data inserted, deleted, or replayed by an unauthorized entity during the migration from host to host as well as host to blackboard. Consider the the dynamic data section of the mobile agent (see Section 3.1.2). A possible integrity scenario: a current host could modify the previous data to make the current host's offer the best in comparison to previous host. In addition, the information on the blackboard could be changed only in authorized manner. As the result the host is the party that has responsibility to send the mobile agent

to the blackboard. Thus, the data and information on the blackboard should only be modified by the last host who is authorized to sent mobile agent to the blackboard.

Authentication: The authentication should provide a proof of an authenticated identity and verify the authentication information in the system. Peer entity authentication and data-origin authentication [1] are needed to provide the connection and the assurance of the data. Two entities are consider as peers, for example mobile agent to host, host to mobile agent, and host to blackboard. A mobile agent will need to prove that it is honest so that the host can trust it and execute it. On the other hand, the host needs to prove that it is an honest host so that it has the right to write the offer in the dynamic part of mobile agent. The mobile agent needs to prove that it has the right to write on the blackboard and the host needs to prove that it is authorized to access into the blackboard.

Access Control: The goal of an access control feature is to prevent the unauthorized access of resource under some conditions. In practice, an access control feature needs three main elements fulfilled: authentication, authorization, and audit [19]. In our protocol, we need to consider the access control between the agent and host, between host and agent, between host and blackboard, and between agent and blackboard. For example, it is possible that a host could be attacked by a malicious mobile agent which carries malicious operation such as unauthorized data access on local system. The audit (i.e. log of transaction) is needed, auditing is outside the scope of this thesis.

Nonrepudiation: Nonrepudiation prevents either sender or receiver from denying a transmitted message [1]. To overcome this, the receiver can prove that the sender sent the mobile agent and when the mobile agent is sent, vice versa, the sender can prove that the receiver received the mobile agent when the mobile agent is received.

Traceable: Tractability is service, the user or host finds out malicious entities if so then they should report it to TTP and the TTP is able to determine the identity of the user. For example, the host reports to TTP about the malicious mobile agent. It must be possible to trace the event and the party to who was malicious.

Revocation and Accountability: Revocation is a service that if the user or host is verified to be malicious and report to TTP, the TTP will then revoke these parties and their access to this ecommerce system.

Availability: The availability needs to ensure that the system exists for usage. A system is available if it it is able to provide the service requested. For example, the current host could restrict the agents and even made its offer be the best in the comparison. Availability depends on access control service and other security service [1]. The use of a event logs and courts can help redeem. This service is outside the scope of this thesis.

We now discuss some special services that pertain to our work.

Privacy of Purchasing: The goal of privacy of purchasing is that the targeted item (what to buy), what the mobile agent is looking for, should be private only to those that need to know. We should hide this information from hosts that do not have the targeted item. We can hide the search item that the user is searching for when it arrive at a host. Only if the host possess the item should they know about the item. If the host does have the item then it will know what target item the user is searching for, otherwise it has no idea about the item but only know that it does not provide the item.

6. MOBILE AGENTS IN ELECTRONIC COMMERCE

Here we develop a secure e-commerce protocol. The security tools that we utilize include elliptic curve cryptography, paring based cryptography and secret sharing. The structure of our mobile agent system will have three types of parties and the bulletin board. The parties include: trusted third party TTP, users U, and hosts H (see Figure 6.1).

Trusted Third Party (TTP): The trusted third party TTP controls the access to the e-commerce system. The users need to register with TTP first if they want to join the system.

User (U): The user U is the party that utilizes the e-commerce system to make purchase. They will constructs and release mobile agents.

Host (H): The host H is the party in network that the mobile agent visits. They execute the instructions of mobile agent.

Bulletin Board (BB): The bulletin board BB provides a site for mobile agents to exchange information by granting to read and write access to it.

Here the user will construct several mobile agents. In order to simplify this part of the protocol, we will first limit our discussion to the construction of a single mobile agent. Of course a single mobile agent will not be able to "decide" the final purchase. The decision aspect of our protocol will be discussed in Section 6.5

6.1 Motivation

Mobile agent $\mathcal{A}$ registers with TTP to receive security parameters for authentication between the mobile agents and the hosts.

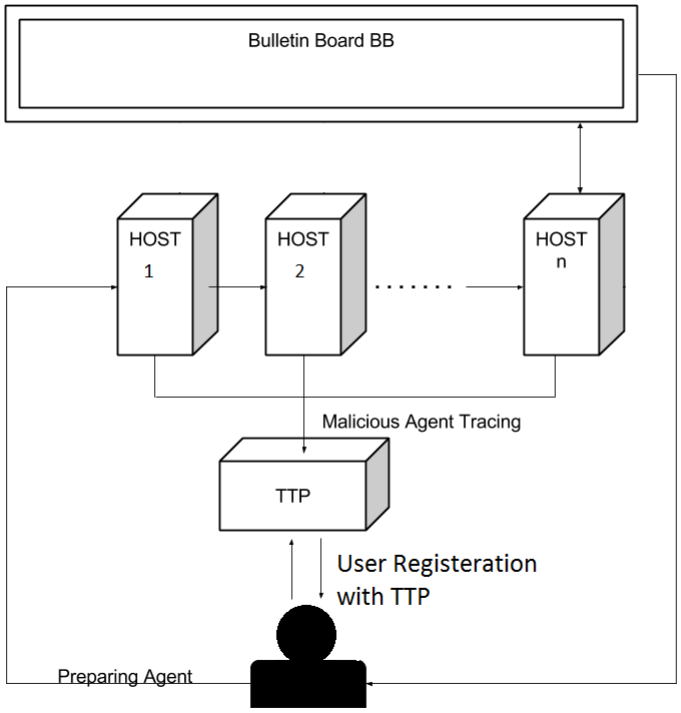

Fig. 6.1.: Model for our scheme

Setup: The TTP generates parameters for each member in that system. Each member generates a pair of private and public keys in the system.

Registration: Each user U who wants to participate in the system needs to register with the TTP. The TTP sends the secret parameters to the user.

User Access a System in the Network: The user U generates the virtual certification parameters to access a system in the network.

Server Authenticates User: A user U who wants to access a system in the network needs to be authenticated by the host H. The host H will verify the virtual certification parameters' authenticity before it executes the mobile agent $\mathcal{A}$.

Tracing and Revocation: The host H makes a complaint about the user U to TTP if it determines the mobile agent $\mathcal{A}$ is malicious. The host H reports the session parameters of the malicious mobile agent $\mathcal{A}$ to TTP. Then the TTP

will trace the user U by using the session parameters that provided by the host H and revokes and restricts the user U from using any system in the network in the future.

6.2 General Setup

We assume there is a secure elliptic curve E generated for all to use (for example one of the NIST recommended elliptic curves for federal government [20]. Furthermore, we assume that a suitably large prime q divides the $\#E$ (the number of points on the elliptic curve E) and that base point G is a point of order q. Then E, G and q have been published for all to use. The trusted third party TTP has the private and public key pair denotes as x_{TTP} and Y_{TTP} where $Y_{TTP} = x_{TTP}G$. If P is a point of the elliptic curve E then we will use $R_x(P)$ to denote the x-coordinate of point P and $R_y(P)$ to denote the y-coordinate of point P. We will use H_0 in the following:

$$H_0 = h(R_x(Y_{TTP})||R_y(Y_{TTP}))$$

TTP also computes,

$$\gamma_U = H_0\omega_U + R_x(A_U)x_{TTP} \bmod q. \tag{6.1}$$

In addition, we assume each user U, as well as the trusted third party TTP, has an identity ID_U which is bound to their public key Y_U. Here x_U denotes the secret key and $Y_U = x_U G$. Y_U and x_U are the long term public and private keys of the user U. Moreover, we assume that there is a pairing function $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{Z}_q$ (see Section 2.3). Lastly, we assume that a secure hash function $h(\cdot)$ has been selected and publicized for all to use (see Algorithm 17) [2].

Algorithm 17 General Setup

1: TTP publishes elliptic curve E, base point G, large prime number q, pairing function e, and hash function $h(\cdot)$

2: TTP selects private key x_{TTP} and computes public key $Y_{TTP} = x_{TTP}G$

3: TTP computes $H_0 = h(R_x(Y_{TTP})||R_y(Y_{TTP}))$

4: TTP computes $\gamma_U = H_0\omega_U + R_x(A_U)x_{TTP} \bmod q$

5: User U selects private key x_U and computes public key $Y_U = x_UG$

6.3 Registering with the TTP

User U must registers with the TTP to get the access to the ecommerce system to release mobile agents and make purchases. User U sends ID_U and public key Y_U to the TTP(see Figure 6.2). The TTP selects ω_U randomly from $\mathbb{Z}_q$ and computes

$$A_U = \omega_U G - Y_U. \tag{6.2}$$

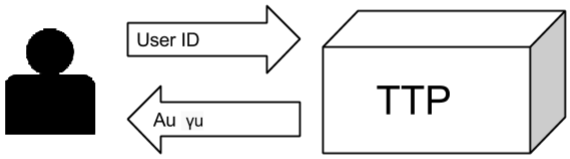

Fig. 6.2.: User and TTP

Then the TTP stores A_U, γ_U, Y_U, ID_U, ω_U, and $R_x(A_U)^{-1}G$ in hash table which can make the tracing be $\mathcal{O}(1)$(see Table 6.1) and returns long term certification: A_U and γ_U to User U(see Figure 6.2). The User U can check the certification from the TTP by computing $\gamma_U G$ and verifying the Equation (6.3).

$$\gamma_U G \stackrel{?}{=} H_0(A_U + Y_U) + R_x(A_U)Y_{TTP}. \tag{6.3}$$

The $\gamma_U G$ represents the parameter that allows user U to be authenticated by hosts in the system. Because $\gamma_U G$ binds the A_U and Y_{TTP}, which are generated by TTP and the long term public Y_U of user U.

Table 6.1.: Data Structure of TTP

A_U	γ_U	Y_U	ID_U	w_U	$R_x(A)^{-1}G$
A_{U_1}	γ_{U_1}	Y_{U_1}	ID_{U_1}	w_{U_1}	$R_x(A_1)^{-1}G$
A_{U_2}	γ_{U_2}	Y_{U_2}	ID_{U_2}	w_{U_2}	$R_x(A_2)^{-1}G$
A_{U_3}	γ_{U_3}	Y_{U_3}	ID_{U_3}	w_{U_v}	$R_x(A_3)^{-1}G$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Algorithm 18 Registering with the TTP

1: User U sends ID_U to TTP
2: TTP selects $\omega_U \in_R \mathbb{Z}_q^*$
3: TTP computes $A_U = \omega_U G - Y_U$
4: TTP computes $\gamma_U = H_0\omega_U + R_x(A_U)x_{TTP} \bmod q$
5: TTP stores A_U, γ_U, Y_U, ID_U, ω_U, and $R_x(A_U)^{-1}G$ in database
6: TTP sends A_U and γ_U to user U
7: User U computes $\gamma_U G \stackrel{?}{=} H_0(A_U + Y_U) + R_x(A_U)Y_{TTP}$ to check the certification from the TTP

6.4 Construction and Use of Single Mobile Agent

We will first limit our discussion to the construction of a single mobile agent $\mathcal{A}$. Of course if we use a single mobile agent $\mathcal{A}$ then they will not be able to "decide" and "make" the final purchase. The decision aspect of our protocol will be discussed in Section 6.5. The users need to provide a signature to prove that they have the

authorization to access and use the system. Each user U needs to generate virtual certificate parameters and one-off key pair $(x_{\mathcal{A}}, Q_{\mathcal{A}})$ [21] to prove the authorization to access and use the system.

6.4.1 User Wishes to Use Single Mobile Agent E-commerce System

The mobile agent $\mathcal{A}$ (user U) needs to provide a signature to prove that they are registered and trusted. There will be a public timer for this single mobile agent system. The host H needs to send the mobile agent $\mathcal{A}$ to the bulletin board BB when the timer reaches **expiration time** τ. This **expiration time** τ will be included in the mobile agent data structure. The single mobile agent $\mathcal{A}$ is using an environmental key generation to encrypt the targeted item I. The host H can compare the ciphertext item with their inventory database (see discussion of environmental key). If the host H does have the targeted item I they decrypt the rest of the instructions in mobile agent $\mathcal{A}$ and execute it. Otherwise the host H needs to send the mobile agent $\mathcal{A}$ to the next host. In this way, the environmental key K can decrease the risk that the mobile agent $\mathcal{A}$ being attacked and protect the user U privacy. Also, if the host H does possess the targeted item I they can write their offer in dynamic data structure DD of the mobile agent $\mathcal{A}$ using **hash chaining**. The concept of hash chaining prevents previous offer modification by malicious hosts(see discussion on hash chaining).

Note: The single mobile agent will not able to purchase the item by themselves. In section 6.5 we use multiple mobile agents and we are able to achieve this.

6.4.2 Generating the Agent

User U will generate mobile agent $\mathcal{A}$ and its virtual certificate parameters V_{cert} as follows (see Table 6.2). The user U generates a unique agent number $\mathcal{A}_{ID}$ using hash function $h(\cdot)$, a random nonce and the the time of the day. They will also generate a unique bulletin board number BB_{ID}, title of the agent, routing table of possible hosts,

and current time stamp. They then place the instruction set, the static data, and a location for the dynamic data DD. Lastly, they place the virtual certificate V_{cert}. The parameters in the virtual certificate V_{cert} are generated as follows, we modify the virtual certificate parameters V_{vert} that Jalal used in [18].

One-Off Key: The use of the term *one-off key* was introduced by Wang in [21], which protects the identity of the user U from the host H and prove the user U is authorized to sign without revealing the long term private key x_U. The user U selects $x_{\mathcal{A}} \in_R \mathbb{Z}_q^*$ for private key and computes $Q_{\mathcal{A}} = x_{\mathcal{A}}G$ for public key of one-off key pair. Unlike Jalal [18] we make $x_{\mathcal{A}}$ secret. The user U creates the one-off keys $\{x_{\mathcal{A}}, Q_{\mathcal{A}}\}$ [21]

Mobile Agent Data Structure: The single mobile agent data structure is illustrated in Table 6.2. The single mobile agent data structure includes M which denote the static data that the user U put in the mobile agent $\mathcal{A}$, which contains the agent number $\mathcal{A}_{ID}$, bulletin board number BB_{ID}, title of agent, routing time, time stamp, instruction set, and static data. Also, the single mobile agent data structure includes dynamic data DD, virtual certificate V_{cert}, public one-off key $Q_{\mathcal{A}}$, and expiration time τ.

Virtual Certificate: The γ_U represents the ability of user U to be authenticated by the host H. But we cannot send γ_U with the agent $\mathcal{A}$ otherwise this would reveal the identify of the user ID_U. As the result, the user U needs to hide their sensitive information and provides enough information for host H to verify the mobile agent $\mathcal{A}$ at the same time. We denote $h(D)$ to represent the hash of some session parameter in the mobile agent data structure with static data M(see Equation(6.4)). The user U needs to generate a virtual certificate V_{cert} by Algorithm 20.

$$h(D) = h(h(M)||h(g_r)||h(g_t)||h(g_z)||h(\alpha)||h(\beta)||h(Y_U)) \tag{6.4}$$

The virtual certificate parameters V_{cert} includes: $g_r, g_t, g_z, \alpha, \beta, Q_{\mathcal{A}_i}$, and λ is described in Algorithm 20.

Environmental Key Generation: We are using the environmental key K to protect the privacy of data. In the real world, customers will only ask sales about the detailed information of the targeted item I in the store if available or go to the next shop to search for it. The goal is to hide part of the instruction set of mobile agent so that the mobile agents do not completely know what their behavior will be. The mobile agent $\mathcal{A}$ uses the targeted item I as a secret key and generates environmental key K for hosts to compare against their database. The environmental key K is targeted item I encrypts with hash function $h(I)$ and encrypt part of the code by targeted item I. The host needs to have the product that has $h(Product) = h(I)$, which means the $Product$ that the host provides is the same as targeted item I that the user U is searching for. In this case the host H can decrypt the rest of the code by $Product$ which is the same as targeted item I in mobile agent $\mathcal{A}$ and executes it(see Figure 6.3 and Algorithm 19).

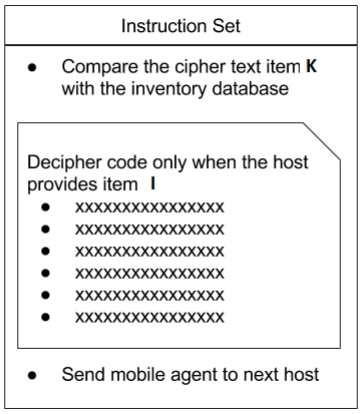

Fig. 6.3.: Portion of Instruction Set is encrypted

Algorithm 19 Environmental Key Generation [8]

1: User U requests the targeted item I
2: User U computes the environmental key $K = h(I)$
3: User U Encrypts the Instruction Set of Mobile Agent $\mathcal{A}$ by the targeted item I
4: Host H looks into the database of goods to checks for every $Product$ in its database
5: **if** $h(Product) = K$ **then**
6: Host H decrypts the Instruction Set of Mobile Agent $\mathcal{A}$ by $Product$

Table 6.2.: Data Structure of Single Mobile Agent

Agent No. $\mathcal{A}_{ID}$	Bulletin Board No. BB_{ID}	Title of Mobile Agent
Routing Table		
Time Stamp		
Environmental Key K		
Instruction Set		
Static Data		
Expiration Time τ		
Dynamic Data		
Virtual Certificate V_{cert}=$\{g_r, g_t, g_z, \alpha, \beta$ and $\lambda\}$		
Public One-Off Key $Q_{\mathcal{A}}$		

Algorithm 20 Virtual Certificate Parameter Generation V_{cert} [18]

1: Input: A_U, x_U, one-off key$\{x_{\mathcal{A}}, Q_{\mathcal{A}}\}$ and static data M
2: User U computes $g_r = R_x(A_U)^{-1}G$
3: User U selects $\psi \in_R \mathbb{Z}_q^*$
4: User U computes $T = \psi G$
5: Let $t = R_x(T) \bmod q$
6: **if** t=0 **then**
7: Go to 2 and User U selects new $\psi \in_R \mathbb{Z}_q^*$
8: User U computes $g_t = tG$
9: Let $z = tR_X(A_U) \bmod q$
10: User U computes $g_z = zG$
11: Let $\alpha = A_U + T$
12: Let $\beta = t(A_U + T)$
13: Let $\lambda = \gamma_U t + \psi t - x_U t + x_{\mathcal{A}} h(D)$
14: Output: $V_{cert} = \{g_r g_t, g_z, \alpha, \beta, \lambda\}$.

Thus the V_{cert} is $\{g_r, g_t, g_z, \alpha, \beta, \lambda\}$. Table 6.2 illustrates the single mobile agent data structure.

6.4.3 Host Authentication

The mobile agent $\mathcal{A}$ selects a starting host H_1. In general the host H_i will authenticate $\mathcal{A}$, and execute it. H_i will check the expiration time τ in mobile agent to see if it is time to send mobile agent $\mathcal{A}$ to bulletin board BB. Also, the host H_i will check the list of malicious mobile agents by using certificated parameters V_{cert}. If the expiration time τ is expired and the $\mathcal{A}_{ID}$ is not in the revoke list then host H_i will send the agent $\mathcal{A}$ to bulletin board BB. If the expiration time τ is not expired, the

Table 6.3.: Single Mobile Agent Data Structure

$\mathcal{A}_{ID}$:d4d5db87449hdkwa	BB_{ID} :ab859g	"Book an flight ticket"
ar.jal.com 213.171.166.525 delta.com 198.193.60.50 aa.com 151.174.224.81 china-airlines.com 205.116.44.33		
11:32 PM EST		
Environmental Key $K = h(\text{"}flight\ ticket\text{"})$		
Instruction Set		

if(A_{ID} is in revoking list)
reports to TTP
if(Expiration time τ is reached)
sends agent to bulletin board BB
if($h(Product) == K$){
decrypt the code in below by using "$Product$"{
for m in Host which is flies from Michigan to Chicago do{
if preference(m) > 0 then
write offer in dynamic data by using hash chaining}}
sends agent to next host		

Static Data		
Expiration Time τ		
Dynamic Data		
Virtual Certificate V_{cert}=$\{g_r, g_t, g_z, \alpha, \beta, \lambda\ \}$ g_r 114 44 2 35 39 155 135 2 65 60 74 57 18 98 8 237 g_t 191 14 214 13 194 213 82 165 169 178 216 34 162 178 39 102 g_z 209 47 145 134 206 97 131 118 32 109 221 69 19 51 209 161 α 42 0 180 18 204 19 7 93 116 242 84 26 8 138 159 21 β 65 202 243 141 237 66 195 234 254 187 210 0 5 95 255 251 λ 158 10 245 94 250 223 124 141 91 82 45 163 192 124 237 123		
Public One-Off Key $Q_{\mathcal{A}}$		

H_i will execute the instruction set of agent $\mathcal{A}$ and send it to the next host H_{i+1} (see Algorithm 21).

The host H computes the mobile agent $\mathcal{A}$ by using Equation (6.5).

$$\mathcal{V} = \lambda G - H_0(\beta + h(D)Q_{\mathcal{A}}). \tag{6.5}$$

Also, host H checks if the agent $\mathcal{A}$ is valid by using Equation (6.6), Equation (6.7), and Equation (6.8) by using pairing function e.

$$e(\mathcal{V}, G) \stackrel{?}{=} e(g_z, Y_{TTP}). \tag{6.6}$$

Note that $\mathcal{V} = zY_{TTP}$ and g_z which is zG is published in the V_{cert} thus it is known by the host H.

$$e(\beta, G) \stackrel{?}{=} e(\alpha, g_t). \tag{6.7}$$

Note that g_t which is tG is published in the V_{cert} thus it is known by the host H.

$$e(g_z, g_r) \stackrel{?}{=} e(G, g_t). \tag{6.8}$$

Algorithm 21 Host Authentication

1: Checks expiration time τ
2: **if** Expiration time τ is expired **then**
3: Sends agent $\mathcal{A}$ to bulletin board BB
4: Checks revoking list
5: **if** A_{ID} is in the revoking list **then**
6: Stops executing agent $\mathcal{A}$
7: Computes $\mathcal{V} = \lambda G - H_0(\beta + H(D)Q_{\mathcal{A}})$
8: Computes $e(\mathcal{V}, G) = e(g_z, Y_{TTP})$
9: **if** $e(\mathcal{V}, G) \neq e(g_z, Y_{TTP})$ **then**
10: Stops executing agent $\mathcal{A}$
11: Reports to TTP
12: Computes $e(\beta, G) = e(\alpha, g_t)$
13: **if** $e(\beta, G) \neq e(\alpha, g_t)$ **then**
14: Stops executing agent $\mathcal{A}$
15: Reports to TTP
16: Computes $e(g_z, g_r) = e(G, g_t)$
17: **if** $e(g_z, g_r) \neq e(G, g_t)$ **then**
18: Stops executing agent $\mathcal{A}$
19: Reports to TTP
20: Executes instruction set of agent $\mathcal{A}$

6.4.4 Host executes Instruction Set

The host H_i searches its product database to see if it has the *Product* that the user U is searching for. If the host H_i has the *Product* then it is allowed to execute the instruction set of mobile agent $\mathcal{A}$, i.e. it has the environmental key so it can decrypt. If the host H_i does not have the *Product* then it needs to send the mobile agent $\mathcal{A}$ to the next host H_{i+1}.

After the host H_i executes instruction set of the mobile agent $\mathcal{A}$, it computes its host identity H_{ID} with its first 8-byte as a identification number, should sign with its digital signature Sig_{host} in the dynamic data part of mobile agent $\mathcal{A}$ and use the hash chaining to protect the offer. The dynamic data DD of mobile agent stores the current host identity H_{ID}, next host identity H_{ID+1}, current host public key Y_H, current host certificated $cert_H$, and current host signature Sig_H (see Table 6.6). The overall execution of the host H is as following (see Algorithem 22).

Algorithm 22 Host executes Mobile Agent

1: Uses V_{cert} to verify the agent $\mathcal{A}$
2: Checks the revoking list of malicious users
3: **if** Agent $\mathcal{A}$ belongs to the malicious user U **then**
4: Stops executing and reports to TTP
5: Check the Expiration Time τ
6: **if** τ expired **then**
7: Sends agent $\mathcal{A}$ to bulletin board BB
8: **else**
9: Executes instruction set of agent $\mathcal{A}$
10: Checks the environmental key K in database to see if any $Product$ matched the environmental key K.
11: **if** $h(Product) \neq K$ **then**
12: Sends agent $\mathcal{A}$ to the next host H_{i+1}
13: **else**
14: Uses the matching item $Product$ as a decrypt key to decrypt the instruction set and executes it
15: Signes the offer in the dynamic data DD of mobile agent
16: Sends agent $\mathcal{A}$ to the next host H_{i+1}

6.4.5 Host writes Offer in Dynamic Data

The mobile agent $\mathcal{A}$ does not immediately know which host H_i provides the best offer. In fact the mobile agent $\mathcal{A}$ visits many hosts and collects several offers before the expiration time τ. If host H_i possesses product item I, it uses the environmental key to decrypt the instruction set to write an offer in the dynamic data DD section of the mobile agent $\mathcal{A}$. However, a malicious host could try to change the offers from previous hosts that the agent $\mathcal{A}$ has already visited. We will apply the hash chaining [8] to prevent this problem and secure data collection in dynamic data DD.

Hash Chaining: We bind the previous offers, next host identity, current host identity, and current offer into one value by hashing them together. In hash chaining the digital signature can tie the signer and the message. In this way we could protect the price discrimination because the signature cannot be separated from the message and the signer (see Table 6.4 and Table 6.5).

Symmetric Key (AES_i)**:** The host H_i selects an AES symmetric key AES_i and signs offer by using their signing key (see Equation (6.9).

$$AES_i = rand(128bits)$$

$$ENC_{AES_i}(\text{Offer}). \tag{6.9}$$

Encrypted Offer (e_i)**:** Using message $\mathcal{M}$ digitally signed by host H_i $SIG_i(\mathcal{M})$ and message $\mathcal{M}$ encrypted with public key of agent $ENC_{key}(\mathcal{M})$

$$e_i = \{ENC_{AES_i}(\text{Offer}), ENC_{Q_\mathcal{A}}(AES_i), SIG(ENC_{AES_i}(\text{Offer})||ENC_{Q_\mathcal{A}}(AES_i)||c_i)\}, \tag{6.10}$$

$0 \leq i \leq n$

Chaining Relation (c_i)**:**

$$c_0 = h(\mathcal{A}_{ID}, H_2)$$

$$c_i = h(e_{i-1}, H_{i+1}),\ 1 \leq i \leq n$$

We compute a hash value on message $\mathcal{M}$ $h(\mathcal{M})$(see Figure 6.4). Host H_i can verify the previous host H_{i-1} by the chaining relation c_i to see if the previous host H_{i-1} cheated.

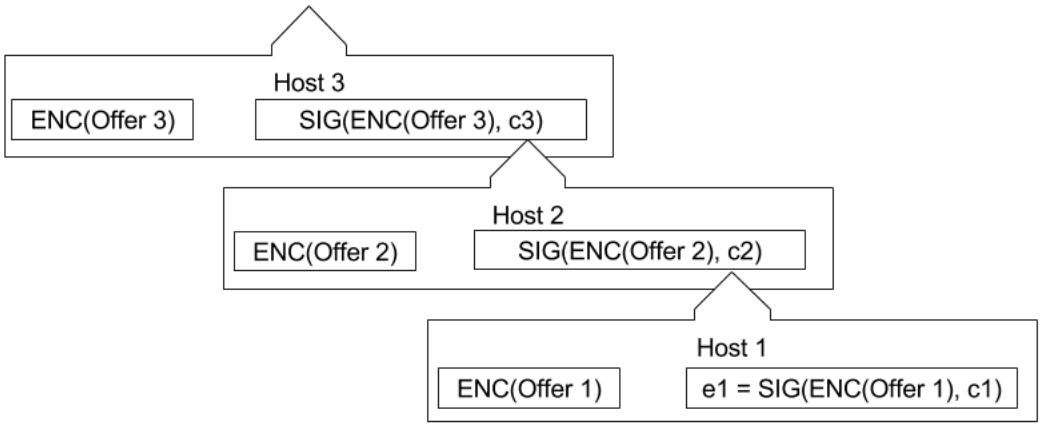

Fig. 6.4.: Chaining Relation

Table 6.4.: Symbols used in Hash Chaining Protocol

H_i, $1 \leq i \leq n$	Identity of the host
$Q_{\mathcal{A}}$	Public one-off key of mobile agent
o_0	Initial information(Identity of the agent A_{ID})
o_i, $1 \leq i \leq n$	Offer from host i H_i
c_i, $1 \leq i \leq n$	Chaining relation
e_i, $1 \leq i \leq n$	Encrypted offer from host i H_i
$e_1, e_2, ..., e_n$	Chain of encrypted offers

Data Structure of Dynamic Data: Agent identity $\mathcal{A}_{ID}$, chaining relation c_i, and encrypted offer e_i (see Table 6.6).

Table 6.5.: Cryptographic notation used in the Hash Chaining Protocol

$ENC_{key}(Data)$	Data encrypted with the key	ElGamal Cryptosystem
$SIG_i(Data)$	Data digitally signed by host i H_i	ElGamal Signature
$h(Data)$	Hash value calculate on Data	...
SYM_i	Symmetric key generates by host i H_i	...

Table 6.6.: Data Structure of Dynamic Data

Host ID	Chaining Relation	Encrypted Offer
H_{ID}	c_i	e_i

6.4.6 Host sends the Dynamic Data to Bulletin Board

When expiration time τ is reached, the last host sends the dynamic data DD to the bulletin board BB.

6.4.7 User verifies the Signature and the Hash Chaining

After waiting for Δ time, the user U can download all the offers from the bulletin board BB and determine the best offer then purchase it.

6.5 Construction and Use of Multiple Mobile Agents

In this section we constructed multiple mobile agents (see Figure 6.5). Each of the mobile agents is generated like a single mobile agent (Section 6.4.2) except we use threshold signatures in this protocol and slightly change the data structure of the mobile agent. Also, it shares many concepts with single mobile agent (Section 6.4): the host H authentication (Section 6.4.3), the mobile agent execution (Section 6.4.4), and the "host-offer" writing (using hash chaining) in the dynamic data DD (Section

6.4.5). Unlike the single mobile agent, multiple mobile agents will be able to securely determine the best offer decision and execute the final purchase in ecommerce system. Here we will highlight the changes between a single and multiple mobile agents system.

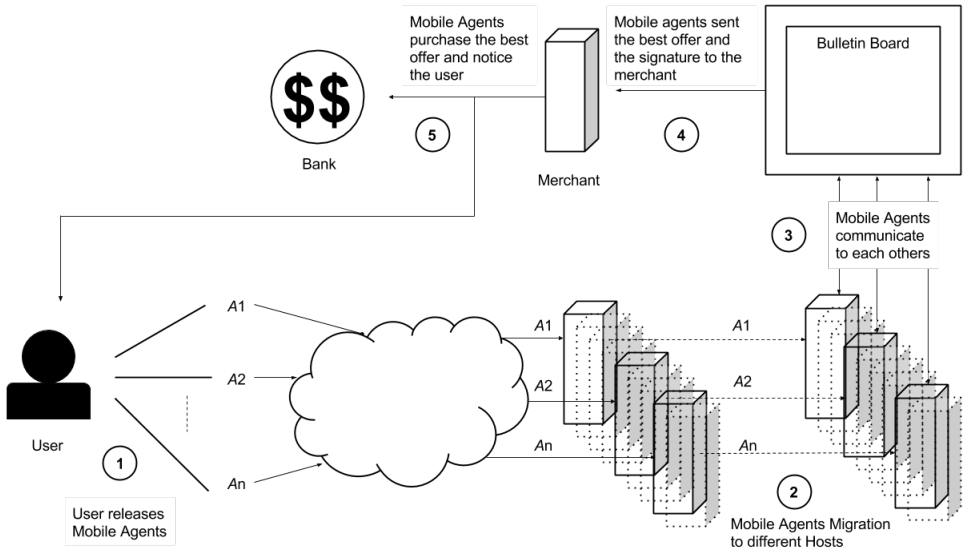

Fig. 6.5.: Multiple Mobile Agents Scheme

6.5.1 User wishes to use Multiple Mobile Agent Ecommerce System

The user U generates a group of agents $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$. Furthermore, the user U partitions the routing table of hosts into total number n of agents disjoint groups, where group i has $n_{\mathcal{A}_i}$ hosts.

$$\sum_{i=1}^{n} n_{\mathcal{A}_i} = \ \# \text{ hosts.}$$

Here the n is security parameter related to how many agents there are.

We are using the ECC El-Gamal threshold signature scheme [14] to generate digital signature to make a purchase offer. This implies they made the purchase.

Threshold/Secret Sharing

The user U needs to decide the on the minimum number m of agents that are needed to construct a signature. This "threshold" protects the user U from a malicious hosts capturing agents and trying to generate a signature. Since $m \leq n$, user U is using (m, n) threshold secret sharing scheme (see Figure 6.6). The higher m offers higher security but but also raises the number of "good agents" that are needed. The user U first shares out the user's signature key (a signature key is analogous to signature for a credit card purchase) x_U in n shares to each agent $\mathcal{A}_i$.

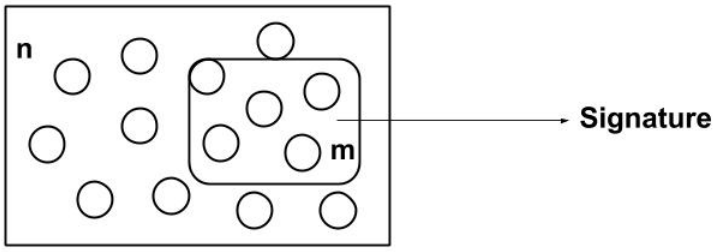

Fig. 6.6.: (m, n) Threshold Secret Sharing Scheme

6.5.2 Generating the Agent $\mathcal{A}_i$

We modified the agent generation of the single mobile agent $\mathcal{A}$. In addition, we slightly modify the data structure of mobile agent $\mathcal{A}_i$ (see Table 6.7) and add a share of user signing key (s_i, u_i) and parameters $\{(S_i, Z_i)|\mathcal{A}_i \in \mathcal{A}\}$.

Mobile Agent Data Structure The multiple mobile agent data structure is similar to signal mobile agent $\mathcal{A}$ except the share of user's signing key (s_i, u_i) and

parameters $\{(S_j, Z_j)|\mathcal{A}_j \in \mathcal{A}\}$ for $j = 1, ..., n$ are added to the static data section of the agent. We illustrate multiple mobile agent data structure in Table 6.7. The multiple mobile agent data structure includes M, which denotes the static data that the user U places in the mobile agent $\mathcal{A}_i$, which contains the agent number $\mathcal{A}_i$, bulletin board number BB_i, title of agent, routing time, time stamp, instruction set, parameters $\{(S_j, Z_j)|\mathcal{A}_j \in \mathcal{A}\}$, and static data. Also, the multiple mobile agent data structure includes the dynamic data DD, virtual certificate V_{cert_i}, public one-off key $Q_{\mathcal{A}_i}$, and share of user signing key (s_i, u_i).

Share of User Signing Key In this protocol we are using secret sharing. The user U wishes that a threshold with many mobile agents $\mathcal{A}_i$ can make the decision and purchase the targeted product item I. The user U divides the secret key x_U into n shares so that any m of them can reconstruct the key. We are using Algorithm 23 to initialize the partial signature. The user U calculates the partial signature (s_i, u_i) and parameters S_j and Z_j then places partial signature (s_i, u_i) in data structure and places $(S_1 \dots S_n)$ and $(Z_1 \dots Z_n)$ in static data section. Each of the mobile agent $\mathcal{A}_i$ will be migrate to different hosts to collect information. At the expiration time τ, all the mobile agents $\mathcal{A}_i$ will send the chaining of the "offers" from the dynamic data section to the bulletin board.

Algorithm 23 Initialization of (m, n) ElGamal threshold signature

1: TTP choose a hash function h

2: TTP choose a secure elliptic curve $\mathcal{E}$ so that it has a subgroup of order q where q is suitably large

3: User U selects the polynomial $f(r) = x_U + a_1 r + a_2 r^2 + \cdots + a_{m-1} r^{m-1}$ of degree at most $(m-1)$

4: The user U secret key $x_U = f(0)$

5: The user U public key $Y_U = x_U G$

6: For each i, the share s_i is calculated as $s_i = u_i + f(r_i)$ where $u_i \in_R \mathbb{Z}_q \setminus 0$, and the r_i are public $(i = 1, 2, ..., n)$. s_i and u_i are placed in the agent $\mathcal{A}_i$'s static data section. All values $r_1, \ldots, r_n$ are also placed on the static data section of mobile agent $\mathcal{A}_i$.

7: User U calculates the public elements S_i and Z_i where $S_i = s_i G$ and $Z_i = u_i G$ these are associated with mobile agents $\mathcal{A}_i \in \mathcal{A}$. The points $S_1, \ldots, S_n, Z_1, \ldots, Z_n$ are placed in agent $\mathcal{A}_i$'s static data section

8: The parameters (h, q, G, Y_U) together with $\{(S_i, Z_i) | \mathcal{A}_i \in \mathcal{A}\}$

Table 6.7.: Data Structure of Multiple Mobile Agents

Agent No. A_i	Bulletin Board No. BB_i	Title of Agent
Routing Table		
Time Stamp		
Environmental Key K		
Share Signing Key Parameters (r_i, S_i, Z_i)		
Instruction Set		
Static Data		
Expiration Time τ		
Dynamic Data DD		
V_{cert}=$\{g_r, g_{t_i}, g_{z_i}, \alpha, \beta_i,$ and $\lambda_i\}$		
Share of Signing Key (s_i, u_i)		
Public One-Off Key $Q_{\mathcal{A}_i}$		

How the host H executes mobile agent $\mathcal{A}_i$ is presented in Algorithm 24.

Algorithm 24 Host executes Mobile Agent

1: Uses V_{cert_i} to verify the mobile agent $\mathcal{A}_i$
2: Checks the revocation list of malicious users
3: **if** Mobile agent $\mathcal{A}_i$ belongs to the malicious user U **then**
4: Stops executing and reports to TTP
5: Checks the expiration time τ
6: **if** Expiration time τ is reached **then**
7: Sends mobile agent $\mathcal{A}_i$ to bulletin board BB
8: **else**
9: Executes the mobile agent instructions set
10: Check the environmental key K in database to see if any $Product$ matches the environmental key K.
11: **if** There is no match **then**
12: Send mobile agent $\mathcal{A}_\rangle$ to the next host H_{i+1}
13: **else**
14: Uses the environmental key K as a decrypt key to decrypt the instruction set and execute it
15: Signs the offer in the dynamic data DD of mobile agent by using hash chaining to protect the offer
16: Send the mobile agent $\mathcal{A}_i$ to the next host H_{i+1}

Table 6.8.: Multiple Mobile Agent Data Structure

$\mathcal{A}_{ID}$:d4d5db87449hdkwa	BB_{ID} :ab859g	"Book an flight ticket"
ar.jal.com 213.171.166.525 delta.com 198.193.60.50 aa.com 151.174.224.81 china-airlines.com 205.116.44.33		
11:32 PM EST		
Environmental Key $K = h(\text{"}flight\ ticket\text{"})$		
Share Signing Key Parameters (r_i, S_i, Z_i)		
Instruction Set		

if(A_{ID} is in revoking list)
reports to TTP
if(Expiration time τ is reached)
sends agent to bulletin board BB
if($h(Product) == K$) then
decrypt the code in below by using "$Product$" {
for m in Host which is flies from Michigan to Chicago do{
if $preference(m) > 0$ then
write offer in dynamic data by using hash chaining}}
sends agent to next host		

Static Data		
Expiration Time τ		
Dynamic Data		
Virtual Certificate V_{cert}=$\{g_r, g_t, g_z, \alpha, \beta, \lambda\ \}$ g_r 114 44 2 35 39 155 135 2 65 60 74 57 18 98 8 237 g_t 191 14 214 13 194 213 82 165 169 178 216 34 162 178 39 102 g_z 209 47 145 134 206 97 131 118 32 109 221 69 19 51 209 161 α 42 0 180 18 204 19 7 93 116 242 84 26 8 138 159 21 β 65 202 243 141 237 66 195 234 254 187 210 0 5 95 255 251 λ 158 10 245 94 250 223 124 141 91 82 45 163 192 124 237 123 Share of Signing Key (s_i, u_i)		
Public One-Off Key $Q_{\mathcal{A}_i}$		

6.5.3 Host of Mobile Agents make the Decision

There are six phases to purchase the targeted item. Host of agent uploads the dynamic data to bulletin board, host of agent downloads all the offers from the bulletin board, host of agents generates partial signature and uploads to bulletin board, host of agent downloads all the partial secret key from bulletin board, purchases and contact the merchant host H_m and user U, and get signature verified by merchant and bank.

Phase One: Agents upload the Dynamic Data to BB All the hosts wait for Δ time after the expiration time τ is reached and send the dynamic data DD to bulletin board BB (see Figure 6.7).

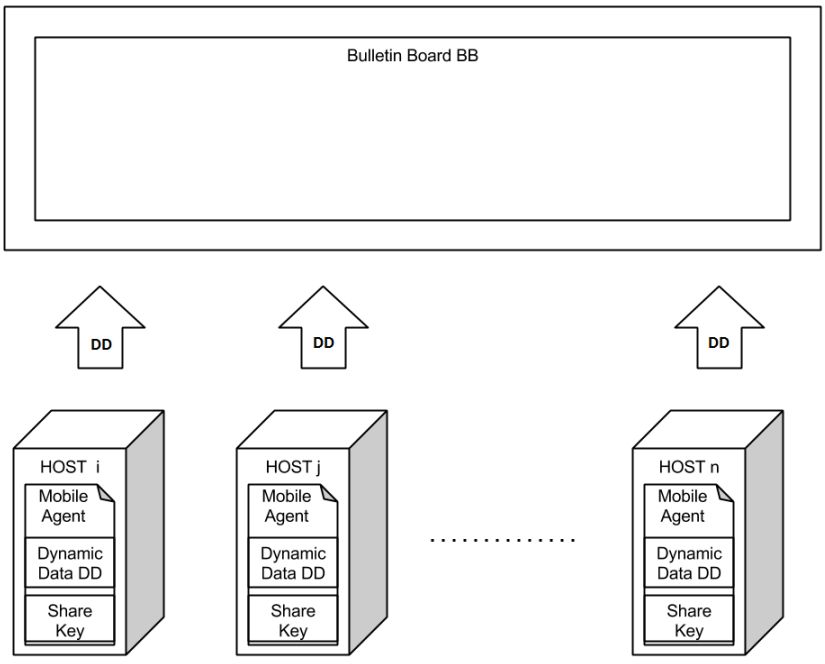

Fig. 6.7.: Host of agents upload the Dynamic Data to Bulletin Board

Phase Two: Agent download all the offers from BB Host of agents downloads all the offers from the bulletin board BB after Δ time. All host of agents verify all of the the hash chains c_i (see Figure 6.8, Algorithm 25, and Figure 6.9).

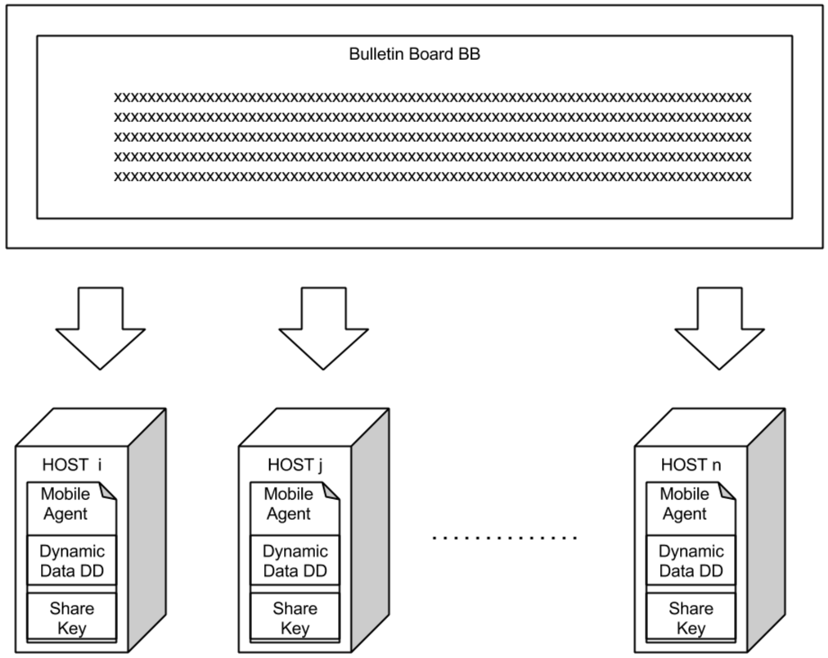

Fig. 6.8.: Host of agents download all the Dynamic Data from Bulletin Board

Phase Three: Agents Generate Partial Signature and upload to BB Host of agents computes the best offer and generate the partial signature for δ time. Each agent $\mathcal{A}_i$ collects all the offers from all agents. The mobile agents $\mathcal{A}_i$ constructs a partial signature which is the user U sharing key using Algorithm 25. All the host of agents should compute the same best offer, since the best offer is a deterministic computation (part of the instruction set), so no extra communication is needed. They then partially sign the best offer using their share of the signing key, which is in the data structure of the agent. Lastly, the host of agent uploads the best offer with the partial signature sigKey on bulletin board BB.

Algorithm 25 Partial Signature Generation of (m, n) ElGamal threshold signature

1: For each i, agent $\mathcal{A}_i$ generates a secret key $k_i \leq q-1$. Now computes $R_i = k_i G$. The host of agent $\mathcal{A}_i$ posts R_i on bulletin board BB

2: Agent $\mathcal{A}_i$ downloads $R_1, \cdots, R_n$ from bulletin board BB

3: Once active agent subset $\mathcal{B} \subset \mathcal{A}$ is known, each agent $\mathcal{A}_i \in \mathcal{B}$ computes R and v , where $R = \sum_{\mathcal{A}_i \in \mathcal{B}} R_i$ and $v \equiv h(\ best\ offer, R) \bmod q$

4: Agents $\mathcal{A}_i$ generates their partial signature by (s_i, k_i) Note $\mathsf{sigKey}_{\mathcal{A}_i} \equiv s_i \prod_{\mathcal{A}_i \in \mathcal{B}; j \neq i} \frac{-r_j}{r_i - r_j} + k_i v \bmod q$

5: The partial signature $\mathsf{sigKey}_{\mathcal{A}_i}$ is placed in agent $\mathcal{A}_i$.

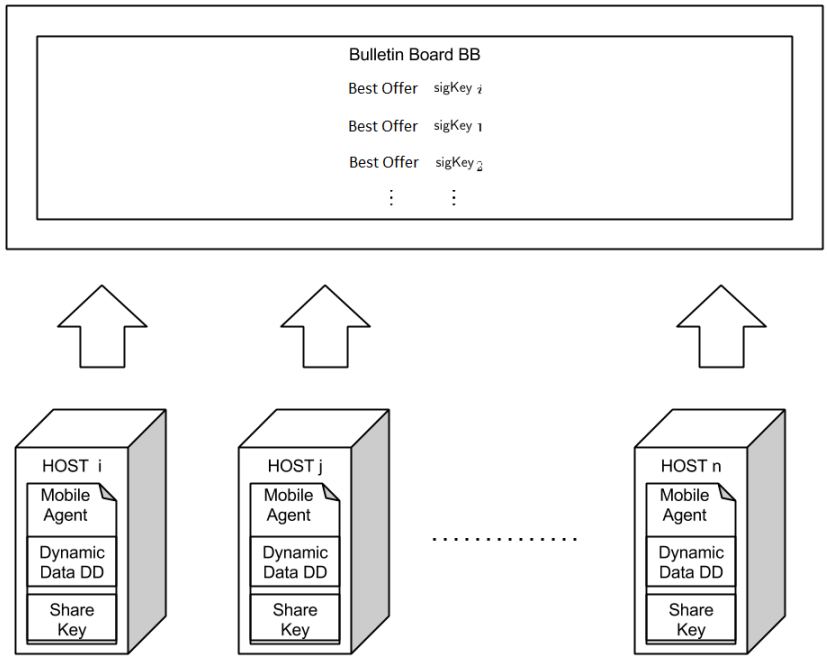

Fig. 6.9.: Host of agents upload the partial key with the best offer to Bulletin Board

Phase Four: Agents downloads all the Partial Signatures from BB Host of agents downloads all the partial signatures $\mathsf{sigKey}_{\mathcal{A}_i}$ and the best offer from bulletin board BB.

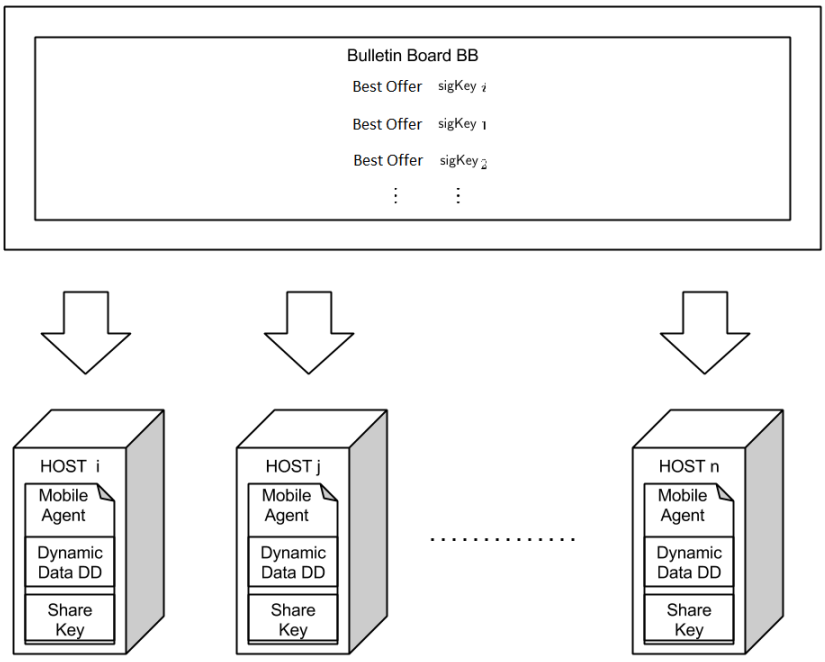

Fig. 6.10.: Hosts downloads all the partial keys to Bulletin Board

Algorithm 26 Final Signature (m, n) ElGamal threshold signature

1: Agent $\mathcal{A}_i$ downloads all partial signatures $(\text{offer}, \mathsf{sigKey}_{\mathcal{A}_i})$ from $\mathcal{B}$ that are on the bulletin board BB and verify the partial signatures by checking $\mathsf{sigKey}_{\mathcal{A}_i} G$

2: $\mathsf{sigKey}_{\mathcal{A}_i} G \equiv \prod_{\mathcal{A}_j \in \mathcal{B}; j \neq i} \frac{-r_j}{r_i - r_j} S_i + v R_i$

3: If All partial signature are congruence

4: Computes $\sigma = \sum_{\mathcal{A}_i \in \mathcal{B}} \mathsf{sigKey}_{\mathcal{A}_i} \bmod q$

5: The $(\mathcal{B}, R, \sigma)$ is the signature of best offer

Phase Five: Purchasing and Contacts the Host and User The host of agents construct the final signature sigKey (see Algorithm 26). The signature that the host of agents reconstruct is the user's signature of the best offer with their secret key sigKey. The host sends the signature to the merchant who provides the best offer.

Phase Six: Merchant and Bank verify the Signature The merchant who made the best offer in the hosts receives the signature sigKey with the best offer. It will verify the signature sigKey by using Algorithm 27. If the signature is authentic then it will send the product to the user U and take the signature sigKey to the bank to get the money. The bank will also use the same Algorithm 27 to verify the signature sigKey.

Algorithm 27 Verification (m, n) ElGamal threshold signature

1: Compute $T = \sum_{\mathcal{A}_i \in \mathcal{B}} \prod_{\substack{\mathcal{A}_j \in \mathcal{B} \\ j \neq i}} \frac{-r_j}{r_i - r_j} Z_i$

2: Check $\sigma G \stackrel{?}{\equiv} \mathcal{Y} + T + eR$ where $v \equiv h(\text{ best offer}, R) \bmod q$

6.6 TTP Tracing and Revoking the Malicious Users

In order to reduce risk and protect the hosts, the hosts can submit the V_{cert} of the malicious agent to TTP. Since the TTP stored the $ID_U, Y_U, A_U, w_U,\ and\ R_x(A_U)^{-1}G$ into the database during the registration, it has the ability to find out who is the one releases the malicious mobile agents in $\mathcal{O}(1)$ using hash table method. TTP would penalize the user U and announce the malicious user ID to revocation list to prevent more damage. See Algorithm 28

Algorithm 28 Host complains Mobile Agent

1: TTP received complaint from the host with the virtual certificate of mobile agent V_{cert}=$\{g_r, g_t, g_z, \alpha, \beta, Q_{\mathcal{A}}, \lambda\}$

2: TTP checks the database to find $R_x(A_U)^{-1}G$.

3: TTP announces the malicious user ID and A_U

Table 6.9.: Revocation List

User ID	A_U
ID_{U1}	A_{U1}
ID_{U2}	A_{U2}
.	.
.	.
.	.
ID_{Un}	A_{Un}

6.7 Bulletin Board Store Signature, Output, and Time

When the expiration time τ is reached, all the mobile agents would be sent to the bulletin board BB by the hosts. At most $n-m$ mobile agents can reside on a host H and m mobile agents must be sent to the bulletin board BB. All the host of mobile agents will use the bulletin board BB to communicate to each others.

7. CONCLUSION

In this thesis we applied the "An Anonymous Authentication Protocol: Achieving Privacy with Trust" to construct a "Secure Mobile Agent E-commerce Protocol". We described the preparation of mobile agents and presented the mobile agent data structure. Also, we presented the hash chaining to secure the collected information that mobile agents store the servers' offers in dynamic data and protect the offers from modifying during the migration. Moreover, we described the environmental key to protect the user privacy. In addition, we use a threshold sharing signature approach to allow the mobile agents to decide and purchase the targeted item for the user from the merchant. In the future work, we could add a two steps verification to verify the final transaction with the costumers. Also, there is a long term parameter in virtual certificate parameters. This long term parameter never change when user reconstructs a mobile agent but it helps us to complete the tracing, however, this could let the hosts have an ability to profiling an anonymous customer. The host could know that someone bought some items before but they never knew the details about the costumer. However, we might want to be more anonymous, secure, and still enable the tracing in this protocol.

REFERENCES

REFERENCES

[1] W. Stallings, *Network security essentials: applications and standards*. Pearson Education India, 2007.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC Press, 1996.

[3] P. FIPS, "46-3: Federal information processing standards publication," *DATA ENCRYPTION STANDARD (DES), Reaffirmed*, 1999.

[4] N.-F. Standard, "Announcing the advanced encryption standard (aes)," *Federal Information Processing Standards Publication*, vol. 197, pp. 1–51, 2001.

[5] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] W. Trappe and L. C. Washington, *Introduction to cryptography with coding theory*. Pearson Education India, 2006.

[7] P. Braun and W. R. Rossak, *Mobile agents: Basic concepts, mobility models, and the tracy toolkit*. Elsevier, 2005.

[8] D. Singelée and B. Preneel, "Secure e-commerce using mobile agents on untrusted hosts," *Computer Security and Industrial Cryptography (COSIC) Internal Report*, 2004.

[9] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[10] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," In Advances in cryptology, pp. 10-18. Springer Berlin Heidelberg, 1985.

[11] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[12] D. R. Stinson, *Cryptography: Theory and Practice*. CRC Press, 2007.

[13] Y. G. Desmedt, "Threshold cryptography," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 449–458, 1994.

[14] C.-M. Li, T. Hwang, and N.-Y. Lee, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders," in *Advances in CryptologyEUROCRYPT'94*. Springer, 1995, pp. 194–204.

[15] B. Lynn, "On the implementation of pairing-based cryptosystems," Ph.D. dissertation, Stanford University, 2007.

[16] "BlueKrypt," http://www.keylength.com/, [LAST DATE ACCESSED: February 23, 2015].

[17] I. Adhicandra, C. Pattinson, and E. Shaghouei, "Using mobile agents to improve performance of network management operations," 2003.

[18] S. Jalal and B. King, "An attack and repair of secure web transaction protocol for anonymous mobile agents," in *Information Security and Cryptology*. Springer, 2010, pp. 426–442.

[19] L. Brown and W. Stalling, "Computer security: Principles and practice," *William Stallings*, 2008.

[20] M. Brown, D. Hankerson, J. López, and A. Menezes, *Software implementation of the NIST elliptic curves over prime fields*. Springer, 2001.

[21] C. Wang, F. Zhang, and Y. Wang, "Secure web transaction with anonymous mobile agent over internet," *Journal of Computer Science and Technology*, vol. 18, no. 1, pp. 84–89, 2003.