

CNL Final Project Report

Team 1

周紀寧 林宗興 吳肇中 鍾毓安 傅冠儒 賴志得

Outline

1. Introduction
 - a. Hanabi rules (give reference)
 - b. Issues:
 - i. Information leak
 - ii. Privacy
2. Framework
 - a. Hanabi server
 - b. HTTP server
 - c. Radius server
3. Implementation
 - a. Hanabi server
 - i. server
 - ii. client
 - iii. judge
 - b. HTTP server
 - c. Radius server
4. Tutorial
5. Analysis
 - a. Rate of information leak
 - b. Attack on private network
6. Appendix
7. Contribution

I. Introduction

Hanabi is a famous board game invented in 2010 by German designer Antoine Bauza. Unlike traditional board game, the goal of Hanabi is collaboration not competition. Namely, players have to work with each other and strike for higher score for the team. There's no winner or loser in Hanabi. It's a game for

teamwork. More details about Hanabi can be found in this document:

<https://boardgamegeek.com/boardgame/98778/hanabi>

After playing lots of Hanabi, we found out that there are two fatal defects: *Information leak* and *Privacy*. Players tend to accidentally expose additional information about the game besides legal action (hint). Although it helps them get higher score, this kind of “accidents” destroy the spirit of Hanabi. On the other hand, Hanabi requires the collaboration among players. As a result, players need to know more about each other to achieve higher score. However, in the real life, people are too shy to turn down other’s invitation to play game. Resulting in poor cooperation in Hanabi.

In order to solve these two critical issues in Hanabi, we decided to develop the online version of it: *Hanabi-online*. In the online Hanabi, we first limit the communication between players to lower the probability of information leak with well-designed protocols. Moreover, we build up private network who only allows members in the network to access the game. Anyone that is not in the private list cannot join the Hanabi. Last but not least, to avoid players exchange information through other internet applications, we use VLAN to block the communication among nodes in the private network. As a result, player cannot send any message to others as long as he/she connects to the private network.

II. Framework

In *Hanabi-online*, there are three servers: Hanabi server, HTTP server, and Radius server.

Hanabi server is written in Python, and is responsible for every services related to playing the game such as register, login, lobby control, matching, game handle, etc.

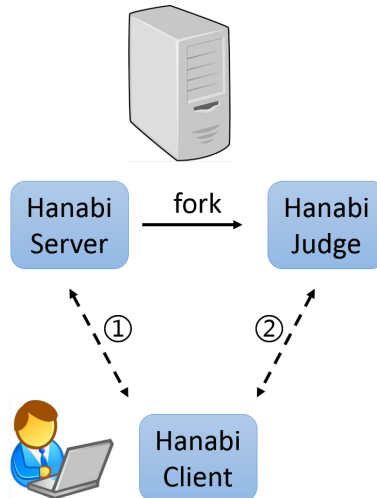
HTTP server provides the platform for administrator to manage the AP service.

Radius server realizes the AAA service to guarantee the privacy.

III. Implementation

The following section will illustrate the implementation details of each server.

1.) Hanabi Server



Below are some commands you will need in the Hanabi Game.

login

Just type your nickname and login!

croom/groom

You could use 'croom [room ID] [max player]' to create a new room or 'groom [room ID]' to join a room

ready/unready/start

Each player should use 'ready' to change his STATE to READY. After all players are READY, the player who creates this room could then use 'start' to start the game. Also, user could use 'unready' to change his current back to UNREADY.

hit/hint/throw

When it's your turn, you could choose one of the hit hint throw command.

hit: Use 'hit [card_index]' to hit a card. After that server will give you a new card automatically.

hint: Use 'hint [player ID] [hint_type] [number]' to give other player some information.

throw: Use 'throw [card_index]' to throw a card. After that server will give you a new card automatically.

2.) HTTP Server

We set up Apache HTTP server and SQL server on the computer where Hanabi server also is. By using web UI(PHP and ajax), we let user login/logout our AP, or register a new account conveniently.

Because we set up our Hanabi server in our lan network, user who wants to participate in this Game should first connect to our AP. Just as what we do in LAB2, chilispot will redirect user to our login page after user connect to AP successfully. User could then login(Radius server) or register a new account(record on MYSQL).

3.) Radius Server

Just as LAB2 did, we use freeradius for authentication. When user try to login to use our AP, the website will set what user type to radius server.(actually the server itself) Freeradius check the

password with the record on SQL database. After that Freeradius sends Accepted or Reject back and website displays result on website.

4.) Firewall

Although we already put our server in private network, we still try blocking those connection from outside network. We could build our iptables on Hanabi server to block those outside ip simply.

IV. Tutorial

In this section, we will give a tutorial on how to play with Hanabi-online. The following will be divided into two parts. The first part is for server, and the second part is for client.

Part 1: Server

To start a Hanabi server, just simply enter the following command:

```
> python3 server.py [port number]
```

After you can start the server, you simply don't have to do anything. The program will handle all the tasks for you. However, there are still two commands you can use to see the configuration of the server. They are `userlist`, and `roomlist`. You can see user status of current Hanabi server and the room status respectively.

Part 2: Client

There are four stages for clients: Login, Lobby, Room, Game. But in the very beginning, client should start the client program just as the server does. And the command is a slight different:

```
> python3 client.py [server IP] [port number]
```

1.) Login

After you start the client, you will see the login page:

[illegible]

Enter your name and login to Hanabi server!

2.) Lobby

After you login, you will go to the lobby:

Lobby				
Room			User	
ID	Num	Max	Name	Status
0	0	4	jerry	IDLE
1	0	4	jack	GAME
2	0	4	mid	IDLE
3	0	4	Andy	IDLE
4	4	5	Frank	GAME
5	0	4	Tim	GAME
6	0	4	Cindy	GAME
7	0	4		
8	0	4		
9	0	4		

In the lobby, you can see the status of the rooms and players in Hanabi server.

You can do four things: quit, go to a room, create a room, or update the status of the lobby.

- quit

quit


- go to a room

```
groom [room number]
```

- create a room

```
room [room number]
```

- update the status of the lobby

update 

3.) Room


When you go into a room, you will see the following page:

Player3		Room4	
Host			
Ready	Ready	Ready	Ready
Frank	Tim	Cindy	
Ready	Ready	Ready	
jack			


It shows the room status for you. You can see your ID in the room on the left hand side. And you can see whether the players in this room are ready for the game. In the demo picture above, we can see that every player in the room are ready.

In the room stage, you can do four things: ready, unready, leave, start:

- ready

ready 


- unready

unready 

- leave

leave 

- start

start 

Notice that, only the **room host** can start the game!

4.) Game

In the game, there will be a judge to connect each player and hold the configuration of the game.

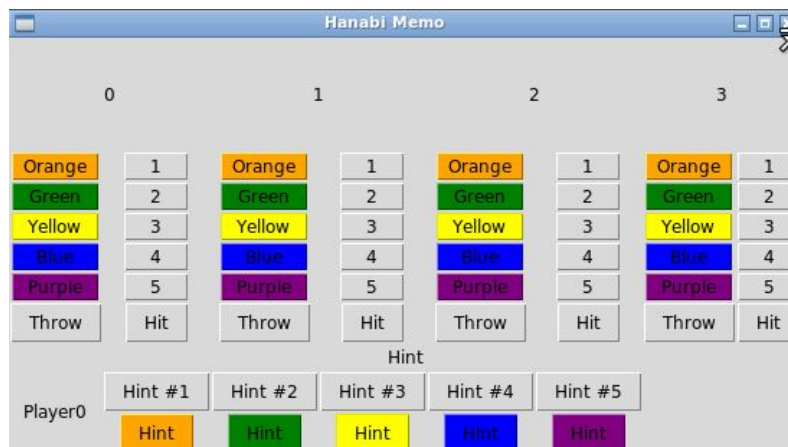
```

[Player 0] # Players: 4 Hint: 8 Fail 0
Garbage

+---+ +---+ +---+ +---+ +---+
| 0 | | 0 | | 0 | | 0 | | 0 |
+---+ +---+ +---+ +---+ +---+
Hint_list:
>>> player 0
+---+ +---+ +---+ +---+
| ? | | ? | | ? | | ? |
+---+ +---+ +---+ +---+
>>> player 1
+---+ +---+ +---+ +---+
| 1 | | 2 | | 1 | | 5 |
+---+ +---+ +---+ +---+
>>> player 2
+---+ +---+ +---+ +---+
| 1 | | 1 | | 2 | | 1 |
+---+ +---+ +---+ +---+
>>> player 3
+---+ +---+ +---+ +---+
| 3 | | 2 | | 4 | | 1 |
+---+ +---+ +---+ +---+
[judge 4] inside game loop XD
Your turn: █

```

Moreover, if your system have installed the tkinter module, then you will have a GUI memo to help you play the game:



Whenever it's your time to play, the judge will send you "yourturn", and then you can do the three operations in the Hanabi game: hit, throw, hint. The details of each operation are listed follow:

- hit

```
Your turn: hit [card number]█
```

The card number will from 0 to 3

- throw

```
Your turn: throw [card number]█
```

The card number will from 0 to 3

- hint

```
Your turn: hint [player ID] [hint type] [hint number]█
```

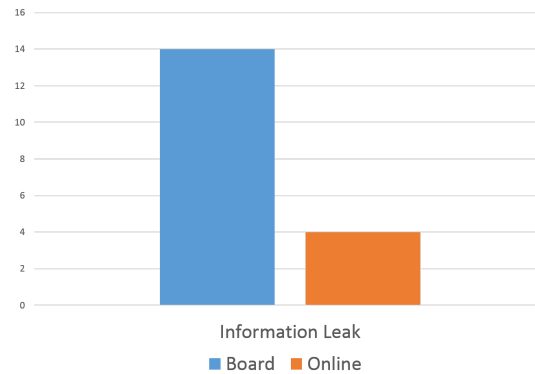
The player ID will start from 0, and you can see the corresponding ID in the screen. The hint type is 0 or 1. 0 means “color”, and 1 means “number”. The hint number is the color/number you want to hint. For example, if you want to hint player 0 the color Blue. You can see that Blue is the fourth color, so that you can enter **hint 0 0 4** to execute your hint.

V. Analysis

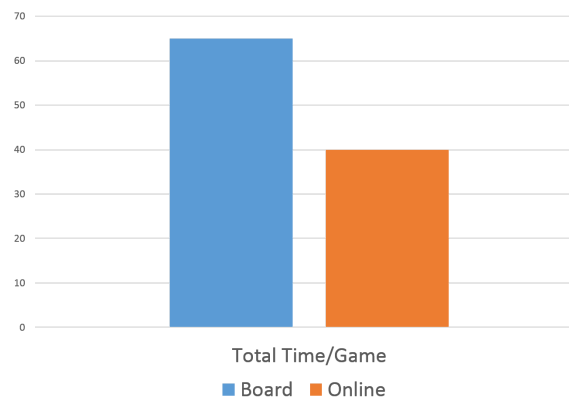
1.) Performance

	Computer-based (Hanabi-online)	Card-based (Board version)	Corresponding Reasons
Frequency of Information Leak	low	high	Using computer prevents players from communicating with each other via emotion or facial expressions, which play important roles in information leak.
Score	high	low	Computer-based version provides players with hint history, therefore, there's need for players to pay extra effort to keep in mind all the details, which usually results in miss-judgement.
Speed of Decision Making	fast	slow	The reason is quite similar with the former one, since the more details are handy, the faster you can make your decision.
Conclusion	even better :)	good	It's trivial !

- Information Leak



- Speed



2.) Attack on Private Network

Actually we put our server in a private network, so it's somehow safe enough. However, we believe that there are some tricky way to access into the internal side of NAT, such as that users in the private network can build up a reverse SSH tunnel to public network. What we do to improve is setting up iptables on Hanabi server. We could drop all package when source ip does not belong to 192.168.1.0/24, and also, we drop all packets that try to access to both outside and inside of NAT.

command line :

```
iptables -P FORWARD -j DROP
iptables -A INPUT -s 0.0.0.0 -j ACCPET
iptables -A INPUT -s 192.168.1.0/24 -j ACCEPT
iptables -A INPUT -j DROP
```

potential attack: attacker may disguise it's IP address as private IP address, then it could access our AP...
advanced setting

another ideal goal is to prevent clients' communication, at first we consider if the router itself can achieve the goal by setting the forward policy of firewall, but as we know later that firewall only works for the communication with "outside" of NAT.

Then, we had considered a vlan solution. It can effectively separate every user's IP subnet, and then in each virtual router interface (since vlan setting may divide subnet 192.168.1.x/24 to

192.168.x.x/24), it can be simply solved by firewall solutions (because at each subnet 192.168.x.0/24, it need routers to forward to other subnets, and firewalls work in such situation). We, however, fail to do so because our hardware, AP: DIR615 dd wrt, does not support vlan on wireless network and port-based LAN, so we only solve the problem that can prevent the entire game environment from attacking.

VI. Appendix

- Github: <https://github.com/jerrychou82/Hanabi>
- Slides: <http://www.csie.ntu.edu.tw/~b01902046/CNL/#/>

VII. Contribution

	周紀寧	林宗興	吳肇中	鍾毓安	傅冠儒	賴志得
周紀寧	35	30	20	7	7	1
林宗興	30	30	15	10	10	5
吳肇中	33	18	18	18	11	2
鍾毓安	35	30	15	10	10	0
傅冠儒	40	19	16	10	10	5
賴志得	NULL	NULL	NULL	NULL	NULL	NULL
平均	35	26	17	11	10	2