Universidade de Aveiro
2022

Diogo
Martins Mota

Suporte para pseudonimato em jogos de mesa na Internet

Support for pseudonymity in online table games

Universidade de Aveiro
2022

Diogo
Martins Mota

Suporte para pseudonimato em jogos de mesa na Internet

Support for pseudonymity in online table games

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Doutor André Ventura da Cruz Marnôto Zúquete, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Hélder José Rodrigues Gomes, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro.

Dedico este trabalho ao meu pai.

o júri / the jury

presidente / president

Professor Doutor Paulo Jorge Salvador Serra Ferreira
Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Professora Doutora Ana Rita Costa Bonifácio Selores dos Santos
Professora Adjunta da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro

Professor Doutor André Ventura da Cruz Marnôto Zúquete
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos / **acknowledgements**

O meu maior agradecimento aos meus pais por me terem educado, apoiado e suportado, de forma incondicional, ao longo deste percurso. Deixo também um agradecimento a toda a minha família que me acompanha desde sempre. Um obrigado a todos os meus amigos que sempre estiveram presentes durante este percurso. E que estarão presentes durante os próximos desafios da minha vida. Deixo também o meu profundo agradecimento ao Professor André Zúquete, orientador desta tese, e ao Professor Hélder Gomes, co-orientador desta tese, por todas as horas semanais disponibilizadas com a finalidade de ajudar a completar esta dissertação da melhor maneira possível, pela partilha de conhecimento e pelas valiosas orientações que me deram.

Palavras Chave

anonimato, conluio, jogos online, pseudónimos, ataque de Sybil, jogos de tabuleiro

Resumo

Os jogos online são uma forma interessante de reunir jogadores de vários locais sem a necessidade de presença física. Mas levantam várias questões de segurança, uma das quais é a possibilidade de ataques de conluio. Tais ataques podem envolver diferentes jogadores a trabalhar em conjunto para um objetivo comum ou um único jogador usando múltiplas contas (um ataque chamado Sybil, ou auto-conluio). Uma forma de combater ataques de conluio é impedir que os jogadores saibam com quem estão a jogar. No entanto, isto significa que os jogadores devem ser anónimos para os outros. Isto exigiria algum tipo de pseudónimo de longo prazo que não pudesse ser apresentado a outros jogadores, porque poderia ser identificado por possíveis atacantes de Sybil. Para resolver este problema, implementámos uma arquitetura com a prevenção de ataques de Sybil em mente. Para criar uma arquitetura capaz de cumprir o nosso objetivo, era necessário um sistema que tivesse o anonimato ao fazer o matching através de pseudónimos, mas que também pudesse conter dados dos jogadores. A utilização de pseudónimos, tanto de longa como curta duração, para combater os ataques de Sybil (manter os ataques de Sybil no mínimo) foi a ideia utilizada para evitar um matching forçado entre os jogadores.

Keywords anonymity, collusion, online games, pseudonyms, sybil attack, table games

Abstract

Online gaming is an interesting way to bring players together from various locations without the need for physical presence. But it raises several security issues, one of them being the possibility of collusion attacks. Such attacks can involve different players working together for a common goal or a single player using multiple accounts (an attack called Sybil, or self-collusion). One way to combat collusion attacks is to prevent players from knowing whom they are playing with. However, this means that players must be anonymous to others. This would require that even some kind of long-term pseudonym could not be presented to other players, because it could be identified by possible Sybil attackers. To solve this problem, we implemented an architecture with Sybil attack prevention in mind. To create an architecture capable of accomplishing our goal, a system that has anonymity when matching via pseudonym but can also hold valuable data of the players was necessary. Using pseudonyms, both long and short-term, to counter Sybil attacks (keep Sybil attacks at a minimum) was the idea to avoid forced matches between players.

Contents

List of Figures

List of Tables

Glossary

MMORPG	Massive Multiplayer Online Role Play Game
IGN	In-Game Nickname
AS	Authentication Server
PDB	Profile Database
MM	Matchmaker
BS	Board Server
PP	Personal Pseudonym
OTP	One-time Pseudonym
MVC	Model–View–Controller
UA	User Agent

CHAPTER 1

Introduction

The purpose of this chapter is to give an introduction to the dissertation by describing the problem, the objectives and the structure of the document.

1.1 Introduction

Table manners have been around since the fifteenth and sixteenth centuries. Nothing is as revealing about one's social environment and adaptation as good table manners. A person with bad table manners usually has bad manners in other areas of life. Thus, good etiquette at the table leads to a good reputation with its peers [19].

There are over a hundred rules contained within table manners and obliging to these rules results in a better environment at the table. And just like in table manners, games have rules that need to be followed with the intent of everyone having a good time playing them while maintaining the fairness.

The rules of a game can be divided in two categories, the standard set of rules (the socially known rules) and competitive ruling (extra set of rules normally written in a competitive rule book for each competition). The games themselves also follow these two categories, some are more focused on the social aspect and other more focused on the competitive aspect. In chess, for example, a standard rule is how the pieces move, a pawn can only be moved to the square in front of itself (two squares if the piece has not been moved before) and can only take another piece diagonally. In tournaments some rules may be added, like a timer to each player, forcing games to end within a time frame established by the tournament's organization.

Competitive games where a group of players is matched against each other, such as chess, card games, mah-jong and others, are the games that this dissertation wants to explore. These are the types of games where the competitive aspect (knowing who wins and who loses) is more prominent than the social aspect of the game, even though some of the games mentioned can be played just for the sake of social interaction on some occasions.

The board games mentioned above can also be divided into games where the players can see everything but also games where the player does not know what the others hold. For example, chess allows for players to know exactly what moves the opponent has available and thus playing accordingly. On the other hand, card games normally have dealers that give a number of cards to each player and those cards are not known to anybody else, if there was a way to know the cards of the opponent the fairness of the game would be ruined.

Games that thrive on the social aspect of the game, like Massive Multiplayer Online Role Play Game (MMORPG) [29], are basically parallel worlds and building your character inside that game is the main goal, instead of trying to win against another player. For that reason, the focus will be on board games, with a competitive view in mind.

1.2 Motivation

Games today mostly use usernames or nicknames as a way to authenticate and identify players while not revealing the person who owns them. They already use Pseudonymity but do not push it to its limits.

Pseudonymity is the use, by an entity, of fictitious identity attributes that do not allow the entity using them to be identified, that fictitious identity is a pseudonym. Pseudonyms are interesting, instead of a identification with attributes recognized in other contexts, such as name or email, they allow the player's integrity to be protected from threats directed at him or her.

Owning a long term pseudonym allows for hiding the identity of the user and at the same time holding their data in a profile, for example. They are very useful in this situation, particularly when revealing some kind of information about the user may lead to harmful actions from other players.

Conversely, the use of pseudonyms, if not accounted for, may open the doors to Collusion and Sybil attack (self-collusion), the latter is possible with the acquisition and making use of several pseudonyms by the same entity in a given context.

A player who can control several real entities, for example, family members or friends, or otherwise manages to create several pseudonyms for himself, is also able to carry out Sybil attacks, since it is the same person that controls all those entities.

In order to stop self-collusion from happening, it is important to know why it happens in the first place or what is to be gained by doing it. Some in depth analyses in [29] shows the impact it can have on games:

- Card games - where players can collude to get more information about the cards in the play than they should, and thus earn unfair advantages over their honest opponents.
- Games with ranking ladder systems - where players or teams of players battle each other to gain rating and progress on the ladder. The players or teams look for times where the number of players playing is low and simply take turns winning, so both gain ranking points and climb the ranking ladder, this is called win-trading. A simple fix is making the losing team also lose points in a manner that even if they manage to match against each other there would be no positive point gain.
- Games with a vote kick system - where players can vote to kick someone from a group. This type of system exist to protect groups from being held hostage when a certain

player does not want to cooperate with the group. But it can be exploited with collusion, if a game requires five players to form a group and the attacker has access to three entities (or more, to form a majority), the attacker can then vote to kick others.

These examples can be associated with some of the types of Sybil attacks explained in [2], [13]:

- Fair Resource Allocation - Attackers through Sybil attacks may unfairly and disproportionately obtain much of the resources that were intended to be distributed equally among all users in the network. For example, in a Card Game, an attacker controls more cards than a legitimate user.
- Tampering with Voting and Reputation Systems - A Sybil attack can be particularly dangerous in the case of any environment where there is schema voting in place for purposes such as reporting and identifying user misbehavior in the system, reputation updating, and so on. As an example, in Games with a vote kick system, to repeatedly report and subsequently remove legitimate users from the network, an attacker can create various malicious identities. Alternatively, these malicious users can protect themselves from ever being removed because they are in collusion.

Papers [2], [13] also proposed some methods to counter Sybil attacks, that could happen in an online game environment, such as Trusted Certification, Resource Testing, Recurring Costs, Privilege Attenuation, Incentive-based Detection and Location/Position Verification. But these methods have a small impact on Sybil attacks, cannot minimize the number of attacks and can be avoided by the attacker if it has the right resources available.

Therefore, the cornerstone of this dissertation will be on using long term pseudonyms for maintaining profile, while hiding them in the process of matchmaking (one time use pseudonyms). This way, we will have some level of anonymity when matching players, in order to prevent Sybil attacks. In other words, the goal is to stop users from being able to force certain match-ups.

Sybil attack detection is also an interesting topic related to Sybil attack prevention, it will be mentioned briefly but will not be explored to the same depth.

1.3 Objectives

The main objective of this dissertation is to design a matchmaking system for online players, based on pseudonyms, which avoids collusion and self-collusion (Sybil attacks) and, at the same time, allows the existence of player profiles.

With regard to the system design, the requirements for said system need to be established. To understand which requirements are needed there is a need to analyse and understand Anonymity, both the positive and negative points, it is also important to study what Pseudonymity and Pseudonyms are, also what current models are used. After that the study of Sybil attacks, which types of attacks are possible and what techniques are available to prevent them.

It is also of upmost importance to understand the games in question, existing matchmaking systems and ratings of the concerned games. The reputation of the players is also a topic that need to be revised.

The two main questions that need to be answered are:

- How can a gaming system maintain anonymity while preserving information about users?
- Can an anonymous system, that matches players without revealing their information, counter Collusion and Sybil attacks?

In a ordered manner, the objectives of this dissertation were the following:

1. Being able to maintain a gaming profile using Pseudonyms.
2. Maintaining the highest possible level of anonymity of the user.
3. Avoid Collusion and Sybil attack through Pseudonymity.
4. Implement the system with the previous objectives.
5. Evaluate the results of said system.

1.4 Structure

To maintain the organization of the document, its structure is arranged as follows.

Chapter 2 contains the state of the art, the necessary knowledge needed beforehand to be able to properly understand the following chapters.

Chapter 3 expresses a thought out architecture of a system with anonymity through pseudonyms and Sybil attack prevention.

Chapter 4 holds the decisions made in the implementation of the architecture and what libraries were used to accomplish the solution.

Chapter 5 is about the results of the implementation and comparisons with the real probability numbers.

Finally, chapter 6 presents a conclusion about the discussed problem, the solution's description and advantages and future work perspectives.

CHAPTER 2

State of the Art

In this chapter it will be described the state of the art related with Anonymity in online games and what comes with it, understand the user's perspective and fears; the actions that happen in online games; Pseudonymity, the current models, and Pseudonyms; Matchmaking and Ratings in online games; Reputation in online games; Sybil Attacks and techniques that prevent them.

2.1 Introduction

According to [16], many systems claim to have tools in place to detect collusion and other types of unethical behavior.

Even though collusion detection is a very interesting topic, it can only happen after the collusion already happened. That is why the focal point will be on preventing collusion from happening, or maintaining it to as unlikely as possible.

In this related work, we also want to look into anonymity in table games, how to maintain profiles of users that get together to play anonymously and how to use profile data to match players. In order to do that, its important to understand how matching players is done and how to match them without revealing their identity, through ranking or other data element but the identity.

With anonymity in matchmaking being established, it is interesting to analyze users' experience when playing anonymized games and what kind of behaviour they display when anonymous. Furthermore, it is also interesting to find out what kind of feedback is held when the behaviour of the players is judged by the others.

With anonymity being in play, knowing what it is needed in terms of security is key, understanding the techniques that may be needed to maintain anonymity throughout the system and being able to implement them is of utmost importance.

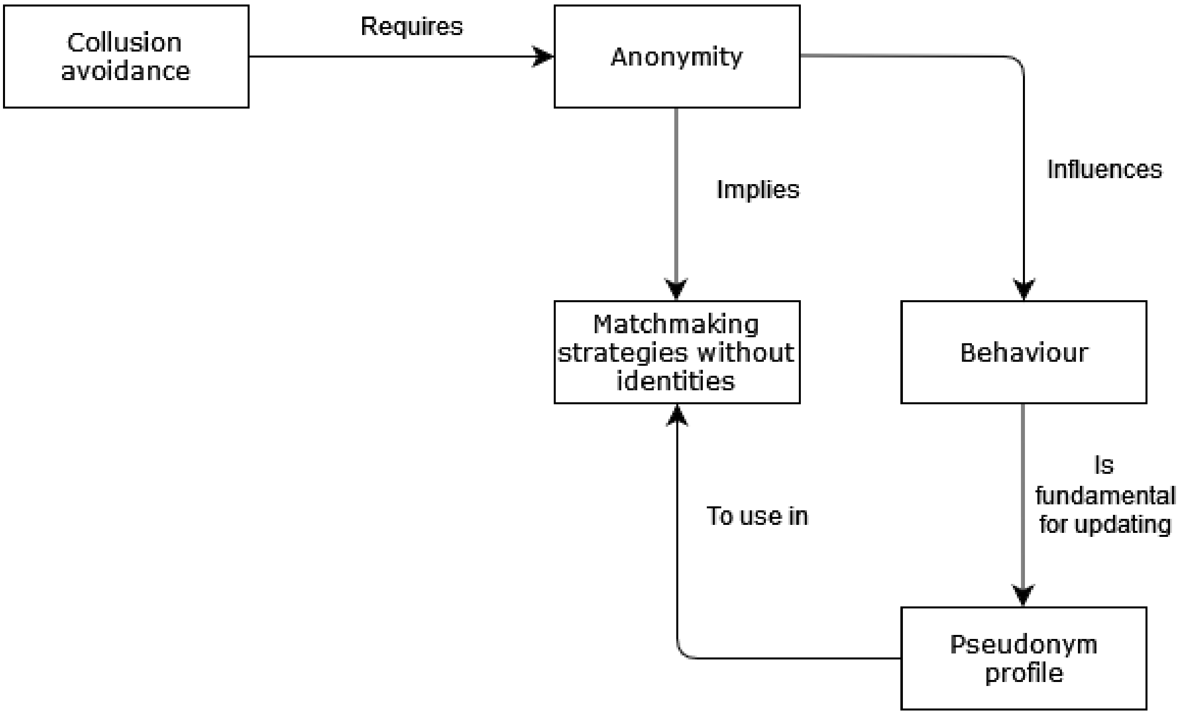


Figure 2.1: Related work dynamic.

2.2 People's behaviour with anonymity

Anonymity is the state of being not identifiable within a set of subjects.

Anonymous users are known to show reduced inhibition of antisocial, reckless and impulsive behavior. They are easily irritable and their sense of identity can be overwhelmed [18], resulting in diffusion of responsibility and excitement, contributing to depersonalization and antisociality [10].

People display reduced self-consciousness, reduced reliance on internal standards that normally qualify their behavior, and little self-awareness. Sometimes anonymous people go beyond verbal abuse and seem to be willing to inflict harm on others [32].

Anonymity also increases the likelihood that people will transgress rules and laws [8] and in crowds becomes even worse, since people lose the grasp of oneself and everything that comes out of the group will be pointed at the cluster of people and not individually.

Multiple examples from [14] show that real life activities that involve some kind of anonymity or hide the identity of someone in a group may result in behaviour changes. The anonymity examples are more related with the topic but it is also important to know that grouping might lead to undesirably behaviour.

The examples from [14] are as follows:

- People in traffic do not view the cars as an object with a human inside but only as an object (the car) and display actions like the ones described above (if every car would be the exact same, this would happen way less, creating a global anonymity)
- People in football matches while in the crowd become way more hostile, seeing the other team and referees as an enemy target and not as a group of people (this happens because

there is a sense of minority, one group against two, but if the referee and opponent were not known entities to the singular group, the hostile levels would be lower).

It is crucial that anonymity systems provide a sense of oneself, show the human side of each user, and do not create groups.

2.2.1 Toxicity - People's behaviour in games with anonymity

Toxicity, also known as Trolling or Dark Participation, can fall into two large categories of verbal (actions via communication methods) and behavioral (actions via game related mechanics or related to real life) [5].

It can also be divided into two types, transient when an action is often committed in the spur of the moment and strategic when some time was taken to gather information and formulate a plan before executing the action.

The spectrum of the actions is pretty large, some of the actions can be basically harmless, but on the other hand others can even go against the law and put people in danger/problems outside of the gaming world.

Table 2.1 and Table 2.2 distinguish verbal/behavioral actions and also transient/strategic action types.

Verbal Actions	Description	Type
Trash talking	Putting down or making fun of other players	Transient
Misinformation	Repeating game-unrelated chat	
Spamming	Repeatedly engaging in an action, such as sending the same verbal message or using the same in-game move, often to the consternation of others	
Griefing	Irritating and/or harassing other players by using the game in unintended ways	
Sexual harassment	Insults or comments based on gender, including threats, the criticism of women and their interests, and stalking	
Hate speech	Insults based on religion, ethnicity, nationality, or other personal information	
Threats of violence	Threats of physical abuse, vandalism, arson, sabotage, possession, or use of weapons or other dangerous act	
Flaming	Presenting emotionally fuelled or contrary statements with an instrumental purpose	Strategic

Table 2.1: Negative verbal actions, their description and type [17].

The motivation behind toxic actions can be divided as an attack (a direct attack on the other players' enjoyment of the game or gameplay), sensation-seeking (it is neither inherently good nor bad, but simply a behaviour which leads to enjoyable consequences for the troll) or interaction-seeking (an unorthodox method of communication designed to make players get involved in both the conversation and the game), the three are not mutually exclusive.

Toxic players, or trolls, normally require a trigger that precedes the toxic action, those triggers can be social (the top social trigger according to [5] is actually being trolled first),

Verbal Actions	Description	Type
Inappropriate role-playing	Pretending you are a different person to obtain a specific reaction or not abiding by role playing norms of the game and/or community	Strategic
Contrary play	Playing the game outside of what it is intended by most players	
Inhibiting team	Inhibiting your own team from being successful in winning	
Aiding the enemy	Behaving in a way that strategically aids the opposing team	
In-game cheating	Using methods to create advantage beyond normal gameplay in order to make the game easier for oneself	
Hate raiding	Purposefully infiltrating the gaming space of another with the intention of spreading hate/harassment	
Doxxing	Publicly sharing and/or publishing another player's identifying information	
Swatting	Prank calling emergency services in an attempt to dispatch armed police officers to a particular address	
Spamming (behavioral)	Repeatedly engaging in an action, such as using the same in-game move, often to the consternation of others	Transient

Table 2.2: Negative behavioral actions, their description and type [17].

internal (the second most common according to [5] is being ”tilted” or just bored) and circumstantial (something that happened pre-game or early in the game), the least popular.

The goals of the trolls can also be grouped in three categories: (i) personal enjoyment (joy from the simple pleasure of the trolling act itself or use trolling in order to disable or weaken the opponent and enjoy winning the game), (ii) revenge (seek either the misery and/or failure of the initial troll, or trying to reform said troll by showing them how their behavior affects others) and (iii) thrill-seeking (complete disregard for the impact of their actions, seeking the most outrageous reaction possible by any means necessary, verbal or behavioral).

The following are actions that are ingrained in the gaming communities even with anonymity they will probably still happen. These types of actions are very hard to stop:

- Trash talking [23] — The use of aggressive or toxic language in order to intimidate or distract opponents. Happens frequently in sports.
- Griefing [4] – Deliberately disturbing another player's gaming experience for own personal pleasure or advantage.
- In-game cheating – Cheating has a large spectre, but it can be summed up to a way to gain advantage in-game.

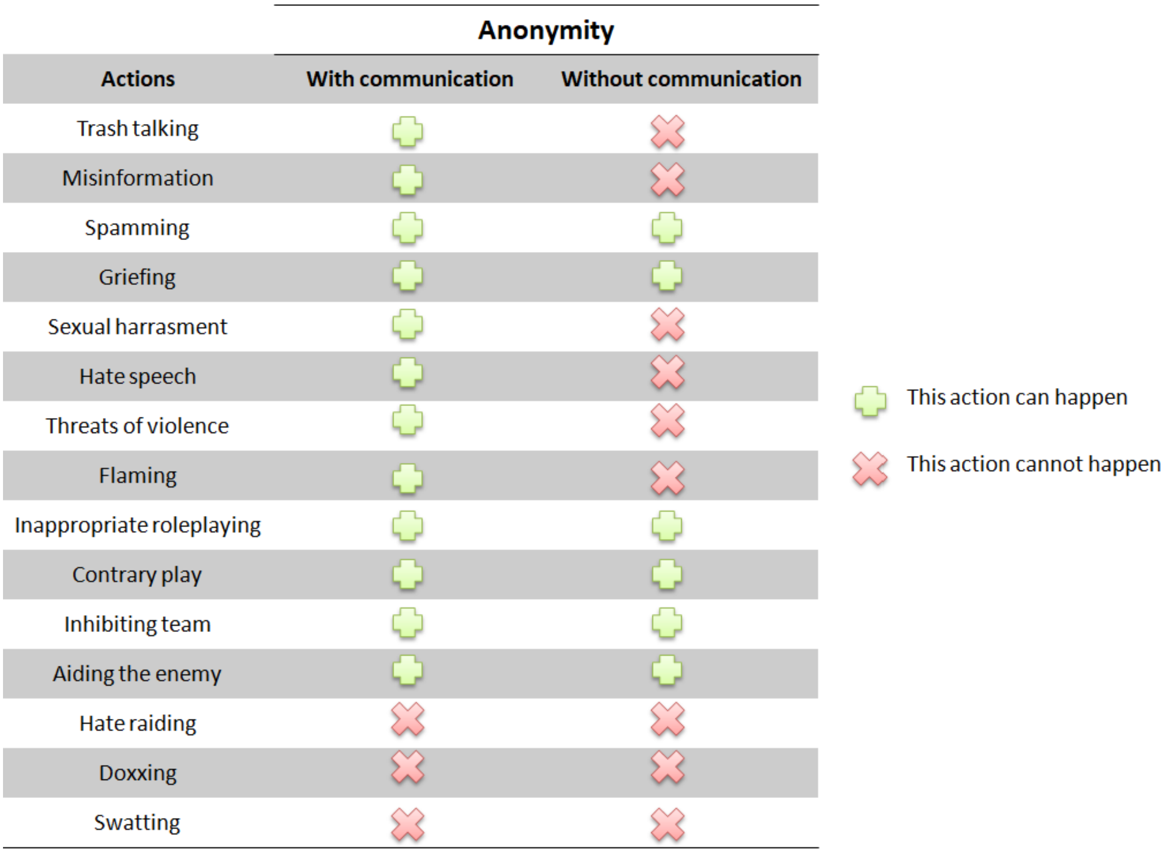
The following are more serious actions that can happen or not with anonymity depending on the existence of communication channels. These types of actions may be less frequent with anonymity since the target's identity is not known:

- Sexual harassment [30] – Insulting a player because of its gender. Men are often the aggressors and women the victims, but the opposite is also possible.
- Hate speech [6] – Similar to the previous segment, insults based on religion, ethnicity, nationality.

The following are Illegal actions. These types of actions may still happen with anonymity but just as threats that wont come to fruition, since the information about the player is not known:

- Doxxing [9] – Sharing personal information of a player.
- Swatting [21] – Calling the authorities to a certain player's address.

A summarized view of these action can be seen in Figure 2.2, with communication channels, the users may be able to able communicate via audio or chat, and without any type of communication.



	Anonymity	
Actions	With communication	Without communication
Trash talking	✚	✖
Misinformation	✚	✖
Spamming	✚	✚
Griefing	✚	✚
Sexual harrasment	✚	✖
Hate speech	✚	✖
Threats of violence	✚	✖
Flaming	✚	✖
Inappropriate roleplaying	✚	✚
Contrary play	✚	✚
Inhibiting team	✚	✚
Aiding the enemy	✚	✚
Hate raiding	✖	✖
Doxxing	✖	✖
Swatting	✖	✖

✚ This action can happen

✖ This action cannot happen

Figure 2.2: Actions that may be influenced by implementing anonymity.

2.2.2 Anonymity in online table games

According to [16], anonymous game tables have already been tested in poker due to its close relation with money and people wanting to not show their true identity.

Single game tables attract a lot of players and are rising in popularity, due to the fact that players can play one game and just leave afterwards. On the other hand networks (in this instance poker networks refer to sites or applications with people that group up to play) seem to not enjoy single tables in an anonymous format, because player retention is lower and there is no way to create fair tables.

There is only one network doing tournaments in the anonymous format, the reason behind this is obviously monetary, it is harder to gather media attention/sponsors with players/teams with no identity. Players also do not enjoy anonymous tournaments as much they do single tables, being able to brag about winning a famous tournament is something that motivates players. Winning a big tournament and not being able to claim the trophy or the winner being recognized as player X instead of their name pushes away players from this format.

Poker, is a competitive game, but like any table game it also has social aspects and it is one of the reasons poker networks thrived due to personality enhancement and customization. There is a list of add-ons that networks use to target both previous points, with chat boxes, player profiles, avatars, country flags and screen names. None of these exist in a anonymous table.

As reported by [16], players play poker for three reasons: monetary, competitive and entertainment. Players who play for monetary reasons should feel more positive about anonymous tables, since it was proven by a study [27] that winners actually win more in anonymous tables.

In the same study, the results showed that in anonymous tables there is more action, more bets are played, more money is bet on each hand, the average size of each bet is larger, and the average pot size is larger. This ultimately results in more money changing hands at anonymous tables, per hand dealt, which means that winners win more and losers lose more.

Even if there is evidence of players choosing anonymous tables to cheat, if the tables know the player's identity behind the anonymity, the system can detect any attempt of collusion.

2.3 Pseudonymity

A pseudonym is a fictitious name that a person or group assumes for a particular purpose, which differs from their original or true name.

Pseudonyms are used to conceal people's identity in business, criminal activity, literature, medicine and science, but the focus of this paper will be on pseudonyms within online activity [14].

Individuals using an online computer may adopt or be required to use a form of pseudonym known as "handle", "username", "login name", "avatar", "Gamertag", "In-Game Nickname (IGN)" or "nickname".

On the Internet, pseudonym remailers use encryption to achieve a persistent pseudonym, so that two-way communication can be achieved and reputations can be established but without linking physical identities to their respective pseudonyms [15]. Pseudonymity has become an important phenomenon on the Internet and other computer networks. In computer networks, pseudonyms have various degrees of anonymity, ranging from highly connectable public pseudonyms (the connection between the pseudonym and a human being is known publicly or easy to discover), potentially connectable non-public pseudonyms (the connection is known to the system operators but not publicly disclosed), and non-connectable pseudonyms (the connection is not known to the system operators and cannot be determined).

2.3.1 Pseudonymity Models in Games

According to [11] these are the Pseudonymity Models in Games:

- Basic Model – In this model users may create identifiers at will, which means that one single user may create several identifiers. Each identifier has a correspondent profile, as if they belong to different players.

  Players are informed of the adversaries' identifiers and are not able to know if two or more identifiers belong to one single player. However, the system may maintain public profiles of each identifier, allowing players to study the actions of each other.

  By playing using an identifier the user enriches the associated profile which creates a reputation. In opposition, this model facilitates playing maliciously against an adversary by allowing an user to create identifiers specifically for that purpose, avoiding possible penalties or acts of revenge.
- Malicious players and Errors Model - This is a variation of the Basic Model.

  Malicious players choose actions that may cause an increase in the overall level of discontent. These types of players make up a small but not zero portion of the player population.

  On the other hand, well-meaning casual players can also do bad actions, these mistakes can happen due to errors in judgment, unstable network connections (a player performing poorly in a game may lose the game due to a lost network connection, which may appear to the opponent as a bad sport), or simply because a person mistakenly hits the wrong key on a keyboard.

  Considering the case where players cannot change their identifiers, the player who commits negative actions will be known to have bad behavior.

  But if players can freely change their identifiers, malicious players can cause havoc in the system.
- Payments Model – The simplest method to achieve complete efficiency in the game with persistent identities and malicious or error-prone players is to make the payment of fees explicit, such as with the imposition of an entry fee. It is easy to see that if such a fee is chosen accordingly, then players will have a sufficient incentive not to leave their current progress and start over with a new identifier since they would then face a new entry fee.

  While attractive, this method suffers from some problems. Redistribution payouts can introduce incentives for players to stay in the game beyond their natural interests. Thus, the redistribution of entry fees would make the exit process as exogenous. Even without this problem, this solution does not work if the life expectancy of the players varies.

  These problems can be eliminated if entry fees are not redistributed to players. If, however, players' payoffs are heterogeneous, some players will opt-out.
- Identifier Model – A system that could achieve full efficiency even in the presence of heterogeneous payouts, allowing players to credibly commit to not changing identifiers, but without revealing their true identities.

  Assuming that there is an intermediary, an entity that all players trust. The intermediary assigns identifiers to players when they request it, but promises never to reveal which players have received which identifiers. Suppose the intermediary also offers a special class of identifiers, which can be called once in a lifetime identifiers, but for each social arena, it will issue at most one identifier to each player. A player with a once in a lifetime identifier is not prevented from returning with a regular identifier, although regular identifiers may be viewed with suspicion by other players.

In this context, players must trust the intermediary and not reveal their true identities, even if the intermediary knows the correspondence between players and identifiers. The trust requirement can be reduced somewhat by an encryption technique.

2.4 Matchmaking criteria

According to [12], "*the key idea behind skill-based ranking and matchmaking is that a game is fun for the participating players if the outcome is uncertain in the sense that each of the participating teams of players has a fair chance of winning*".

In matchmaking an head to head game is assumed, according to [26], a matchmaker can be skill-based, behavior-based or location-based (an hybrid form of two of them or even all three can work), Table 2.3.

Parameter	**Type**	**Description**
Skill	Statistics Distribution	Rating players' abilities using statistical methods to calculate skill representation fairly
Behavior	Unsupervised Learning	Grouping players based on playing habits which shows how experienced the player is based on playing activities
Location	Unsupervised Learning	Grouping players by location to find out how close the distance between players in the real world is, is a consideration of network latency

Table 2.3: Parameters used for matchmaking algorithms [26].

In order to not break anonymity only skill-based and behavior-based matchmakers can be considered, since location-based would reveal the user's location. Skill-based and behavior-based matchmaking can be made without breaking the state of being anonymous, if the matchmakers utilize the skill and behavior data of the users without knowing their identity (pseudonym).

Matchmakers that are skill-based try to match the users according to their abilities in the game, to make the game as close as possible. But it can also allow for more challenging games or easier ones if need be or if users want a specific type of game.

Matchmakers that are behavior-based try to match the users according to their good or bad behavior. In order to not corrupt players with good behavior the matchmaker tries to match good behavior with good behavior. This leads to bad behavior being matched with bad behavior, not as a punishment, but in order to not influence players with good behavior.

In a perfect world where every player has played enough games so that the system has a correct estimation of the exact skill level of the players, every game should have a 50% chance for every team to win.

2.5 Skill-based rating systems

Rating systems, such as Ingo, Elo, Glicko, and Edo, have emerged with modern chess rating systems, and lately have been widely applied in the gaming industry as a whole [25].

Competitive games use skill-based rating systems for several practical purposes:

- to qualify players for tournaments.
- to match players of similar abilities for tournaments.
- to monitor players' progress.

In Elo rating systems, players' rating is viewed as a way to estimate the players' abilities and it is possible to calculate through probabilities the outcome of a game knowing the Elo of the players. This system was adopted by the World Chess Federation (FIDE) in 1970. Figure 2.3 shows the chess Elo ratings distribution in 2018.

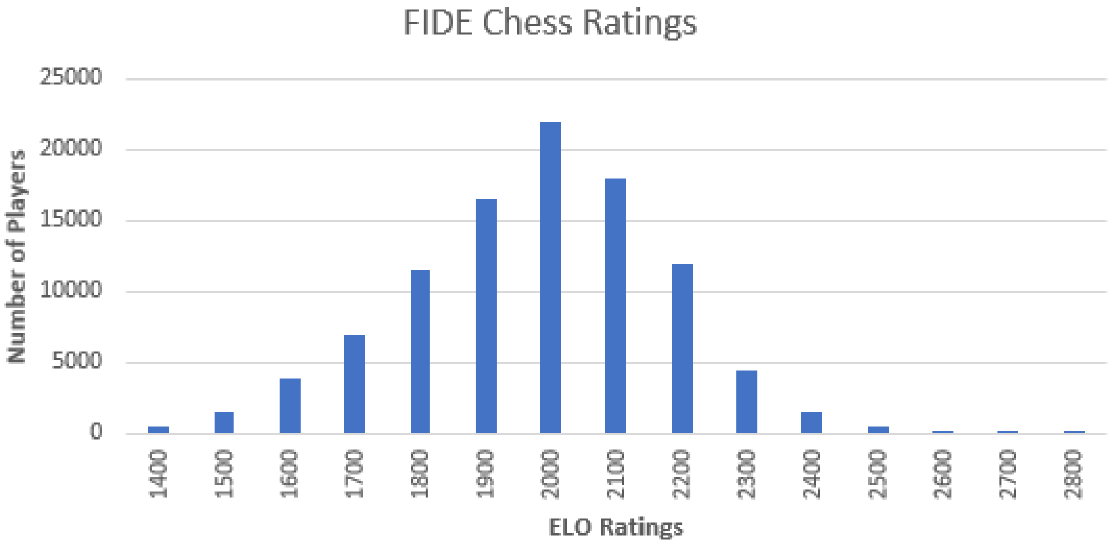


Figure 2.3: FIDE chess ratings distribution in 2018 [31].

To fully appreciate the benefits of the Elo rating system, it is useful to contrast it with alternative approaches to skill estimation. The basic item response theory models assume constant skill. These models are thus applicable only for short tests where no learning is expected or for modeling very basic skills where learning as a whole is slow [22].

For Two-Player and/or Multiplayer Games an generalization of the Elo system was implemented, called TrueSkill. The TrueSkill system makes it possible to immediately update the ranking of players and/or teams after each game [25].

2.6 Behavior-based systems and Reputation

Reputation spreads information about users' behaviours so that expectations of future interactions can influence the behaviour of said user and others around him, even if the future interactions may be with different people than the current ones. The way reputation spreads can affect its ability to influence behaviour, and it is especially interesting to consider situations where people exercise some control over the spread of their reputation, a situation that is common on the Internet [11].

The Internet has created numerous social and business environments that allow for frequent and meaningful interactions between strangers. This leads to many problems and properties that do not normally arise in other social environments. However, the flexibility of the Internet as a social structure also allows for a large degree of engineering, which is also more difficult in normal social environments, allowing many of these problems to be solved.

The key aspect of Internet reputation that does not normally arise in non-electronic environments is the ability to change identity. While in real life, this is a complex process, on the Internet a person can choose between interacting anonymously, constantly changing identifiers, or maintaining a persistent identity.

A system like the one shown in the Figure 2.4 is a good example of a possible reputation system for online games. This system has three groups, the one main group with the "self" user and two other friendly users, F1 and F2, which "self" has a good relationship with. This group has good reputation, so "self" trusts F1 and F2. F2 has a good relationship with F3, which belongs to another group with F4 and F5, since F3 trusts F4 and F5, the whole group (F3, F4 and F5) will be trusted by the main group. On the other hand, F1 has a bad relationship with A1, an antagonist of F1, but since A1 is in the same group as A2 and A3, the main group will also not trust them.

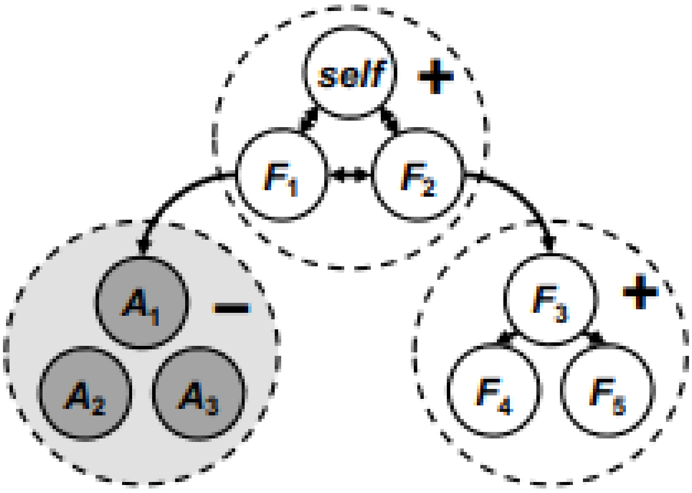


Figure 2.4: Example of a reputation system [24].

The Internet highlights the issue of being able to change one's name as a way of erasing reputation, making name changes almost free. This creates a situation where users with positive reputations are expected to have good behavior, but negative reputations do not stick. It is natural to ask to what extent cooperation can be sustained based on positive reputations alone. The answer is plenty, but not full cooperation. A natural convention is to distrust or even mistreat strangers until they establish a positive reputation.

Users may try to Sybil attack by creating additional accounts. If identifiers are expensive or linked to an ID, it becomes more difficult. Some paid online games usually have an experimental system in place. That system allows users to create a trial account to try the game before buying it. If many accounts are able to be created, even if only for a short period of time, that enables Sybil attacks in that short period.

Reputation as currency is an interesting way to make sure users behave within the game. Users are given a certain amount of reputation as a coin and they lose or gain that currency according to their actions [28].

Reputation systems with recency bias are systems that value the latest reputation data more but also take older data into account. These are systems that allow the users to redeem themselves but at the same time keep some kind of background behaviour.

- A Time Interval system will consider the reputation gathered by the user within a certain time-frame (week/month, may vary) established by the system and put more weight on that interval that on the rest of the reputation.
- A Numeral system will take N (number defined by the system) most recent reputation points received by the user and put more weight on them.
- A Weighted Arithmetic Mean will put more weight on a certain group of reputation points defined by the system [20].

It is important to note that when talking about reputation, the age of the user's account should be taken into consideration, a user that only recently joined the system and has a good reputation should not be on the same level as a veteran user with a good reputation.

CHAPTER 3

System Architecture

This chapter will convey the requirements of the system and what types of necessities the architecture of the solution has.

3.1 Requirements

After understanding the keys points of the state of the art, it is clear that an anonymity system has certain requirements. It is important to analyze them and understand which ones are important and relevant for the development of the architecture and future solution.

- For each game each player has a pseudonym to keep information but this pseudonym is not revealed to the rest of the players at the table.
- The player must be anonymous to the other players and to the architecture system.
- Architecture components must know the bare minimum about users for the system to work.
- Each system component must not break the anonymity of the users by itself.
- The system must include mechanisms to prevent Sybil attacks.

By having a different pseudonym for every game, the possibility of a match history is non existent. There is no knowledge of who played against who, the players do not know who they really played against and the system does not know who they matched against each other.

A player will only know the one-time-use pseudonym of the other players.

The architecture components should not receive information about the user that are not required for their functions or that may somehow reveal some kind of information about the original pseudonym.

The system should not in any moment break the anonymity of the players, the players should remain anonymous throughout the whole process.

Mechanisms that do not allow Sybil attacks, for example not matching players right away and updating profiles at certain timings, to make sure the players cannot force a match between themselves and even the architecture components are not aware of who matched against who.

These requirements come from the basis that the possible attackers are users that have negative intentions. With that in mind there are some assumptions needed for the system to work:

- Users have different IP addresses when connecting, otherwise anonymity would be lost.
- Architecture components do not ally with users, to avoid users getting the match they want for example.
- Architecture components are honest, in other words, components always do what they are meant to do and do not give incorrect information or try to corrupt the fairness of the system.

3.2 Selection of Pseudonymity Model

If pseudonym identifiers are cheap or easy to get, which is the case for the Basic Model and the Malicious players and Errors Model, one single person may be able to get various identifiers in his possession and that can eventually result in Sybil attack.

Considering the fact that allowing multiple identifiers to belong to an unique owner is not a good way to counter Sybil attacks, there is a necessity to prevent one to be able to get various identifiers.

The two models that make obtaining more than one identifier harder are the Payments for identifiers Model and the Identifier commitment Model. One requires currency and the other the commitment of the user through some kind of identification.

The Payments for identifiers Model will stop the attackers that do not have the funds to purchase multiple identifiers and will also make attackers wonder if the attack will be worth the investment. The numeral value of the identifier can be important, but any attacker with large sum of currency or truly dedicated to make this attack happen will be able to do it.

Which leads us to the Identifier commitment Model, where the user will commit to the system with an identification and trust the system to protect it. The person can obtain one identifier through an exchange with the system, for example the user can give its telephone/identification number to the system expecting it to protected with an encryption technique and in return receive an identifier. This makes the attainability of having multiple identifiers much more difficult and will also help with further Sybil attack prevention.

3.3 Architecture

The Architecture is constituted by five components: a single Authentication Server (AS) that authenticates the users, a single PDB that preserves the profile of the users, multiple Matchmaker (MM) that match players accordingly, multiple Board Server (BS) that allow the users to play together and the User Agent (UA)

All components belong to the same organization, to make sure they are legit before the system starts running.

3.3.1 User Agent

The UA is a web application that fundamentally communicates with the components of the system for the user, it acts as a mediator.

In this instance, the UA processes the user's instructions, transfers them and receives the requested data. The user utilizes the UA to manage to get in the system and to be able to play without having to deal with the components itself.

The UA allows for the user to only worry about playing the game and not about about the communications and data transfers happening in the system.

3.3.2 Pseudonym Management

The UA must authenticate themselves with the AS before being allowed into a game and the MM has to authenticate with the PDB to receive data from the profiles.

The AS recognizes the UA trough an identifier (identification/telephone number), that the UA employed to commit to the system. But it does not recognize the pseudonym of any user.

The UA validation uses blind signing. The UA will blind the ID (hash of public key), the AS will then sign it and send it back to the UA, the UAwill unblind it and send it to the PDB, so it can be verified. The status of the UA is changed "Authenticated" and only then it can start looking for a match. This allows the UA to verify the ownership of the public key with the private key.

This process has to be done on a daily basis, this means that the token received only works for 24 hours.

The procedure for the authentication of the UA, according to Figure 3.1, is the following:

1. The UA does the whole authentication process with the AS
2. The UA blinds ID (hash of public key) and requests the AS to sign it blindly.
3. The AS signs the ID blindly and creates a 24 hour (daily) token, sends both the UA.
4. The AS blind ID and daily token to the UA.
5. The UA unblinds the ID and signs the daily token.
6. The UA signed ID and signed daily token to the UA.
7. The PDB verifies the signed ID from the AS and checks validity of signed daily token.

Since, during step 6 trough 7, the PDB changes the status of the UA to "Authenticated" even if an attacker is able to capture the information sent in step 6, they cannot claim the pseudonym. Because the PDB will not allow more than one verification per pseudonym per day.

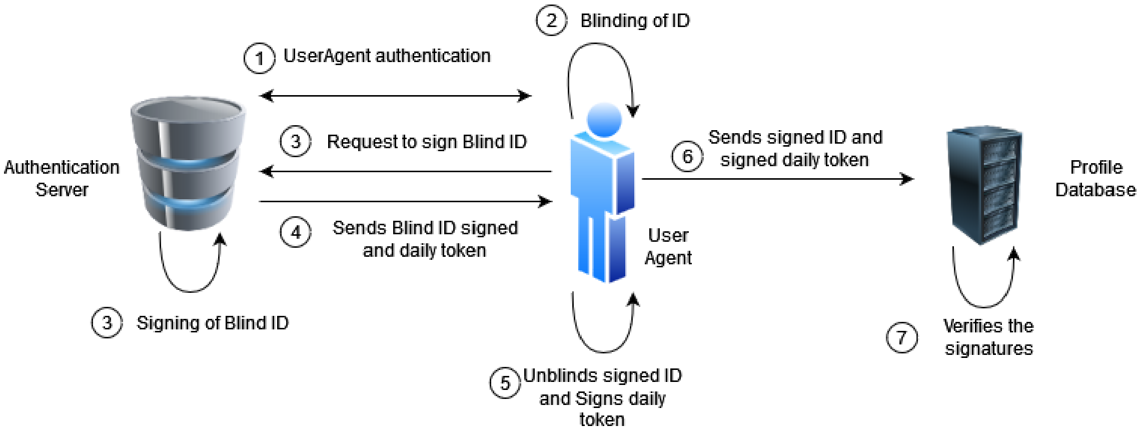

1 UserAgent authentication
2 Blinding of ID
3 Request to sign Blind ID
4 Sends Blind ID signed and daily token
3 Signing of Blind ID
5 Unblinds signed ID and Signs daily token
6 Sends signed ID and signed daily token
7 Verifies the signatures
Authentication Server
User Agent
Profile Database

Figure 3.1: UserAgent authentication and ID Verification Flow.

3.3.3 Matchmaker Authentication

The MM are also required to authenticate themselves daily to make sure none of them is compromised. The authentication of the MM is done through encryption of the MM ID via their private key and decryption via public key by the PDB. The MM receives a token that can be used for the day, so it is not needed to authenticate more than once per day.

The procedure for the authentication of the MM, according to Figure 3.2, is the following:

1. MM encrypts ID to PDB using Private Key.
2. PDB uses Public Key to validate authentication. And creates a token. The token is only usable for the day.
3. The MM receives the token to finalize the authentication.

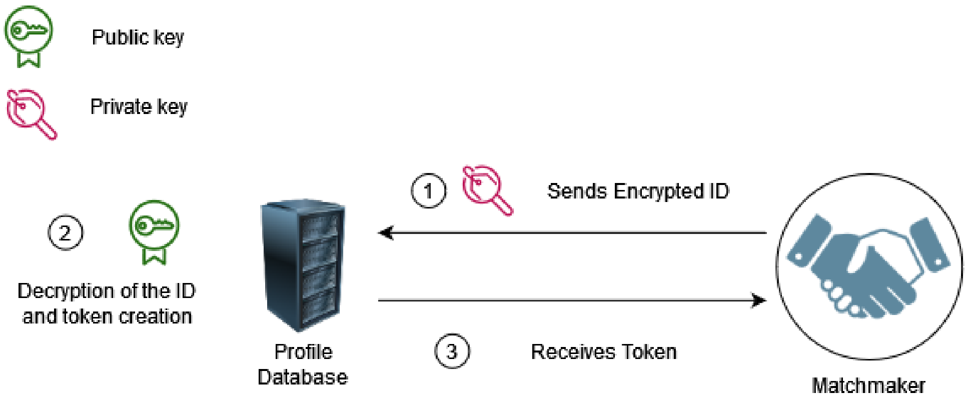


Figure 3.2: Authentication of the Matchmakers.

3.3.4 Architecture Model

Figure 3.3 shows the interactions between the user and the components that leads the user to a game, it also shows the exchange of the user's data between said components.

The user interacts only once with each component to maintain minimal interactivity between them. The interaction with the PDB is to create a One-time Pseudonym (OTP) to

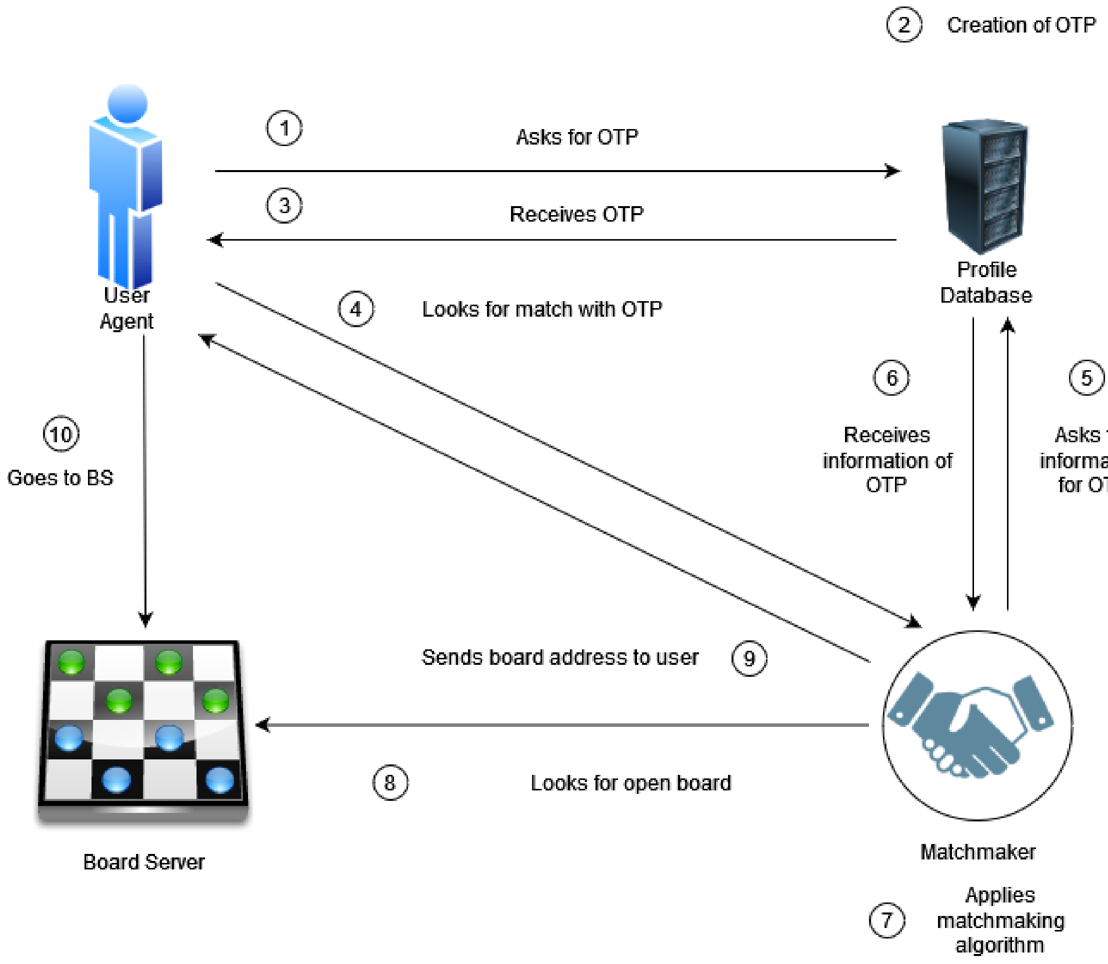

2 Creation of OTP
User Agent
1 Asks for OTP
3 Receives OTP
Profile Database
4 Looks for match with OTP
6 Receives information of OTP
5 Asks for information for OTP
10 Goes to BS
Sends board address to user 9
8 Looks for open board
Board Server
Matchmaker
7 Applies matchmaking algorithm

Figure 3.3: Player flow to participate in the game.

use once for this game, with the MM to know the right BS for the game and with the BS to play the actual game.

The MM authentication, Figure 3.2, happens before step 5 of Figure 3.3.

All the data exchanges between the components the user is completely unaware of.

- User asks PDB for OTP, this step is only allowed if the authentication of the user was successful and current status is “Authenticated”.
- Creates OTP and changes user’s status to “Looking for match”. Multiple OTP creations are impossible since the status wont allow it.
- User receives OTP signed by the PDB so its content cannot be modified and can now chose to play anytime.
- The user looks for a match, choosing the wanted MM.
- The MM asks for the reputation and rating (if needed) of the OTP from the PDB.
- The PDB sends the information needed only if status of the OTP is “Looking for match”.
- The MM will match players according to the ranking and reputation and send an update status do the PDB so the profile statues is “In Game” and does not force another match. This status update has to be done individually so the PDB does not realise who is playing against who.

- The MM will look for a Board that has a “Open” status and give the BS the list of OTP that are going to play there.
- The User receives the board address from the MM.
- The User goes to the correct BS. If the user tries to join a random BS since it is not on the list sent by the MM to the BS it wont be allowed to join.

Figure 3.4 shows what data from the PDB is shared with the other components and with the user. Obviously, the goal is to have the least possible data shared between them to avoid Sybil attacks. For that reason the only point of data that they all share is the OTP. The OTP will be used only for one game and it is what enables anonymity during the matchmaking process and then updating the profile of the real pseudonym. The data that each component has is also the minimum possible so that the system is able to work.

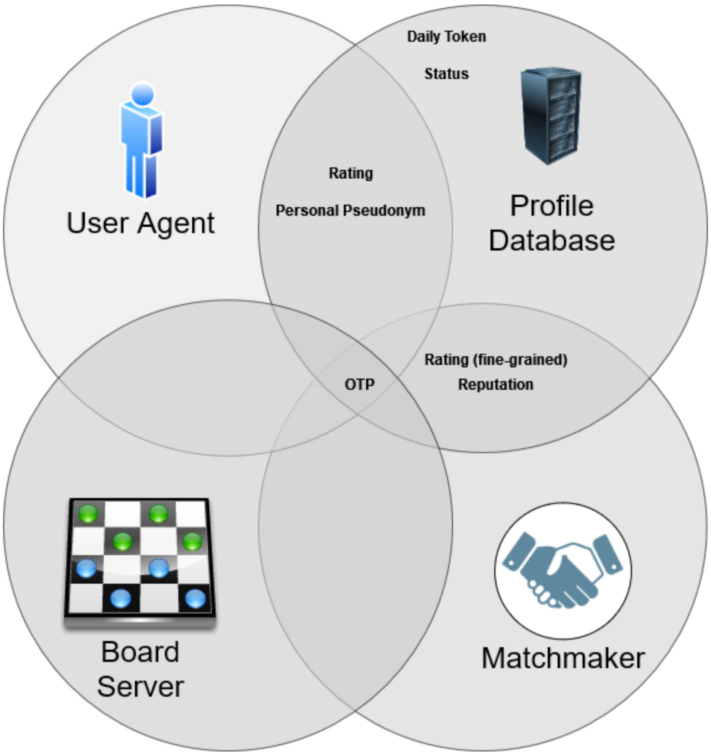


Figure 3.4: What each component but Authentication Server knows about the user. And also what the user knows about his profile.

The AS is responsible for authenticating the user without knowing its pseudonym. After authenticating, the AS creates a token usable for the day that the user must then use to claim its profile from the PDB. If multiple authentication attempts are tried during the time that the token is valid, the AS will resend the previous token.

The PDB is responsible for keeping and managing the profiles. The first thing the PDB is responsible for is attributing the right profile once an user checks in with the daily token. The PDB keep the status of the profile updated and modifies it after every game. When the user asks to play, the PDB must create a smaller version of the user’s profile with only necessary information for matchmaking and OTP. The PDB is also responsible for authenticating the MM as shown in Section 3.3.2 and Figure 3.2.

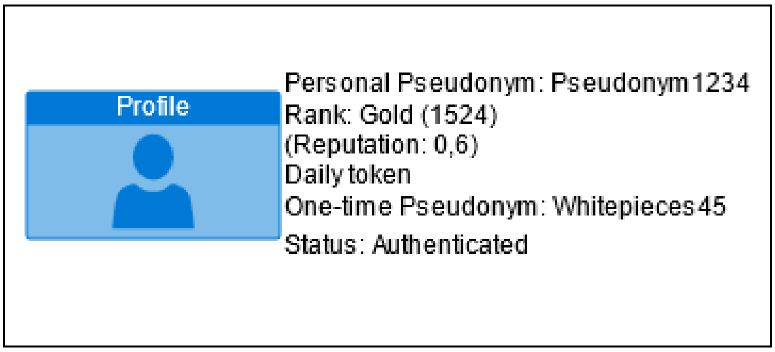
The MM is responsible for matching the players and sending them to the board. The MM receives a smaller profile with just an OTP, numeral rank and numeral reputation. Applies the matchmaking algorithm and then sends a status update to the PDB, so the users cannot look for anymore games. Also sends the OTP and only the pseudonyms to the BS. After the game is over the small profile is updated and sent to the PDB.

The BS is responsible for the game itself. Receives a list of OTP that it does not know who they are. At the end of the match gives MM the information about who won and who lost without ever knowing their true identity. In order to receive players the board has to be open (with zero players) but at the same time once the game start must close to any other player, so there is no interference. A board status is sufficient to resolve this issue, "Open" when there are no players and no game and "Closed" when there is a game going on. The status changes once a game start and ends accordingly. The relation of a Board Server with a ongoing game is one to one, one server takes care of only one game.

3.3.5 Profiles

In order to maintain the fairness of the games, there is a need to link the player's game skill and behaviour with their pseudonyms.

To resolve this issue a profile must be created, Figure 3.5 shows a profile that the system needs to keep safe. No one but the owner of the Pseudonym should be able to see the profile and even him should not see all of it. In regards to the Elo of the player the user should only be able to see the rank and not the number and the reputation should be totally secret. This is a necessity because if players know who they can play against they can manipulate the matchmaking system to create a Sybil attack.



Personal Pseudonym: Pseudonym1234
Rank: Gold (1524)
(Reputation: 0,6)
Daily token
One-time Pseudonym: Whitepieces45
Status: Authenticated

Figure 3.5: Profile concept.

Profile analysis of Figure 3.5:

- Personal Pseudonym - Fixed Pseudonym of the user.
- Rank - Skill level metric of the user, the numeral rank shouldn't be displayed but just used for matchmaking.
- Reputation - Behavioral metric of the user, shouldn't be displayed but just used for matchmaking.
- Daily Token - Authentication token can only be used for 24 hours.

- One-time Pseudonym - Pseudonym created for the sole reason of playing one single game.
- Status - The status' purpose is preventing players from playing without being authenticated and from trying to be in multiple matches.

3.3.6 Profile Creation

The first time an user is authenticated is a critical instance because it still does not hold a pseudonym. If the system does not provide a way to authenticate the user before the pseudonym and consequent profile is created, users would be able to create multiple profiles.

To avoid this from happening the first time the user shares it's public key with the AS, the user receives a different kind of token that it will give to the PDB. The PDB then understands that the token is not to claim a profile but to actually create a new one.

The PDB will then create a Personal Pseudonym (PP) for the user and also create a profile for that PP. This new profile has to have standard rank and standard reputation. The status of that profile is changed to "Authenticated" so the user can start playing right away.

3.3.7 Matchmaking

It is possible to match players anonymously [7], but at the cost of the fun and fairness of the matches, which will lead to lower player longevity.

Matches can only be fair if all players involved have similar skill ratings (Elo), but making sure player with bad attitude do not ruin games for others is also of upmost importance. The matchmaker needs to make use of both attributes from the player's profile in order to make the best match possible.

There is a need to create OTP before they gather at the virtual board. This is a necessity because some Pseudonym may gain reputation (good or bad) from outside the system, for example a professional player's Pseudonym is known from taking part in some tournaments. And also to prevent system components from knowing who played with who.

The first algorithm (ranking algorithm) is a hybrid algorithm based on the skill and behavior parameters. It will firstly try to match players according to their abilities and then try match their reputations. The priority will always be of the ability of the players, if the MM finds a good skill match but cannot find matching reputations in the queue the game will go ahead.

The ranking algorithm explained step by step:

1. Gathers a list of at least 10 players from the same rank, not necessarily the same Elo number. If the list does not contain 10 players the algorithm wont start.
2. Compares the Elo numbers of each players and tries to match them.
3. If a player has a negative reputation compared to the other matched against him, the algorithm will look for a better replacement. Otherwise the game goes ahead.
4. After a time frame, if the algorithm does not find a replacement, the match will go forward even with a player with less reputation.

The second algorithm (non-ranking algorithm) does not take skill into consideration, it will only try to match players according to their reputation.

The non-ranking algorithm explained step by step:

1. Gathers a list of 10 players. If the list does not contain 10 players the algorithm wont start.
2. Compares reputation numbers and matches players accordingly.
3. Players with higher reputation will have priority for matching.
4. Players with lower reputation will be left with a slower matching.

3.3.8 Match run

The system concept of the game after the MM matches the two players is demonstrated in Figure 3.6.

After the matchmaker chooses the best possible match in the pool of available players according to the chosen algorithm and finds a open BS, it will send the players to the correct BS so the game can be played.

During this whole process the users will be using the OTP so anonymity is not lost at any point. And during the game they wont even see their opponents OTP but a placeholder (Player1 and Player2, if the game is between two users).

The use of the OTP, besides maintaining anonymity, is also for the BS to be able to tell who won and who lost and for MM to be able to update the PDB according to the information received from the BS.

Explanation step by step:

1. MM sends the board address to the OTP.
2. The players go to the right BS.
3. The game is played.
4. The BS gives the result of the match and reputation evaluation to the MM.
5. The MM updates the ranking and reputation of the OTP and sends it to the PDB. The PDB updates the profile of the PP.

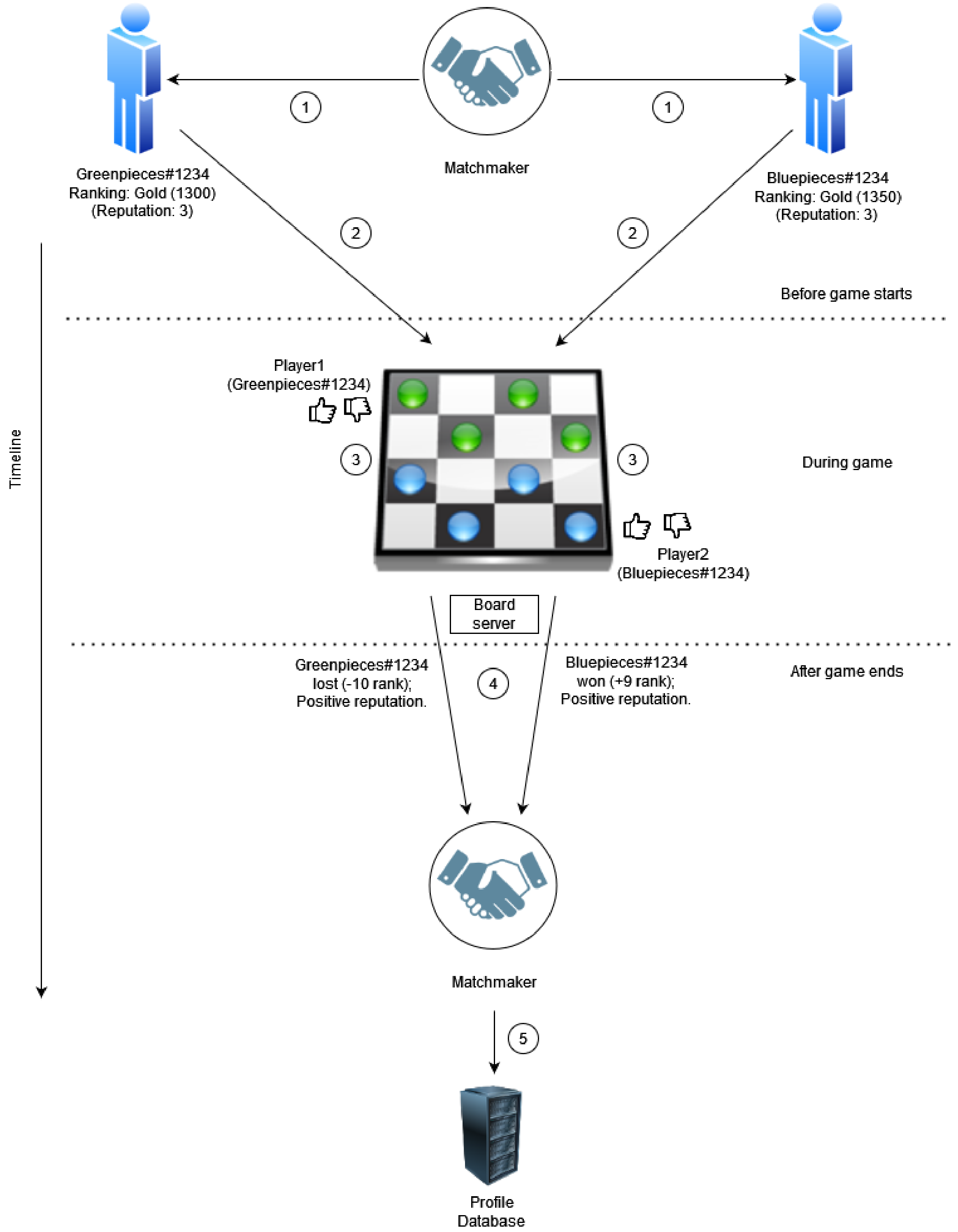


Figure 3.6: Match run concept.

3.3.9 Profile status

The Profile has four status to force authentication and to prevent Sybil attacks. The first two status are “Not Authenticated” and “Authenticated”, these two status are needed so players cannot play a game without authentication. The “Authenticated” status always comes after the “Not Authenticated” status, it is not possible to change from “Not Authenticated” to any other status besides “Authenticated”. The next two status are “Looking for match” and “In game”, these two status prevent the possibility of the player trying to be in various matches at the same time, because users will be locked from trying to find matches when they have this status. When a game is over the status will revert do “Authenticated” again and the player can then look for another match.

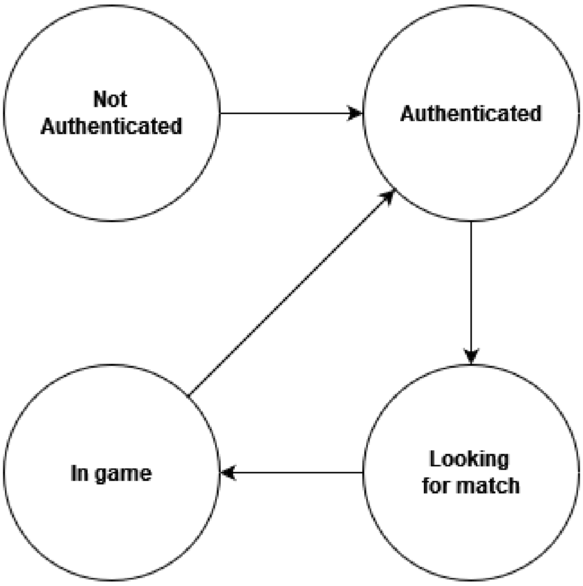


Figure 3.7: Profile status sequence.

3.3.10 Profile update

The profile update from the MM to the PDB has to be done carefully, so the database does not know who matched against who. The MM will send batches of profile updates, with a limit of 100 players per batch, if the requirement number of players is not met the batch is not sent and has to wait for some matches to end to send them.

This solves the problem of the PDB discovering who matched against who, but may cause some time of wait to the players. To minimize the waiting time in case a player wants to play again immediately after a match, the user can ask the for his profile to be updated individually.

3.3.11 Board Server status

The BS only has two status to prevent multiple games from happening in the same server.

The status are “Open” when the server is completely free and ready to receive a list of OTP from the MM to start a game. Once the server receives the list the status changes to “Closed” and stays this way until the match ends. When the game ends the BS sends the results of the match to the MM and the status changes to “Open” again.

Players only join the server when the status is “Closed” and their OTP are on the list that the server received. If a player has the server address but the status is “Open” or they are not on the list and the players tries to join the BS, the server won’t allow it because the player is possibly trying a Sybil attack.

CHAPTER 4

Implementation

In this chapter it will be explained how the solution was implemented, the problems faced during the implementation and how they were overcome.

4.1 System

As explained in Chapter 3, the system's main purpose is to maintain anonymity while playing using pseudonyms.

In this prototype, the first step of being able to preserve anonymity is having an Authentication mechanism that allows authentication without revealing the true identity of the user. The second step is being able to match users without giving away their identity to others.

4.2 Framework

The chosen framework for the project is Django because it works well with small projects and it is good for demonstrations, like this one, due to its Model–View–Controller (MVC) architecture. It is object-relational mapper eases the interaction with the database which is key, since one part of the architecture is exactly a database that must be secure to maintain anonymity, Figure 4.1.

By default the Django configuration uses SQLite, which is perfect for demonstrations since SQLite is already included in Python, so there is not a need to install anything else to support the database.

Django also has an admin page that is useful too see the database, check if the models are working properly and if needed manually adjust some data within the model, Figure 4.2.

The Django MVC architecture is a variant of the usual MVC architectures, instead of having Models, Views and Controllers it has Models, Views and Templates, but the idea is very similar, Table 4.1.

The Model is the logical data structure that is connected to the database and has the whole data that can be used by the view.

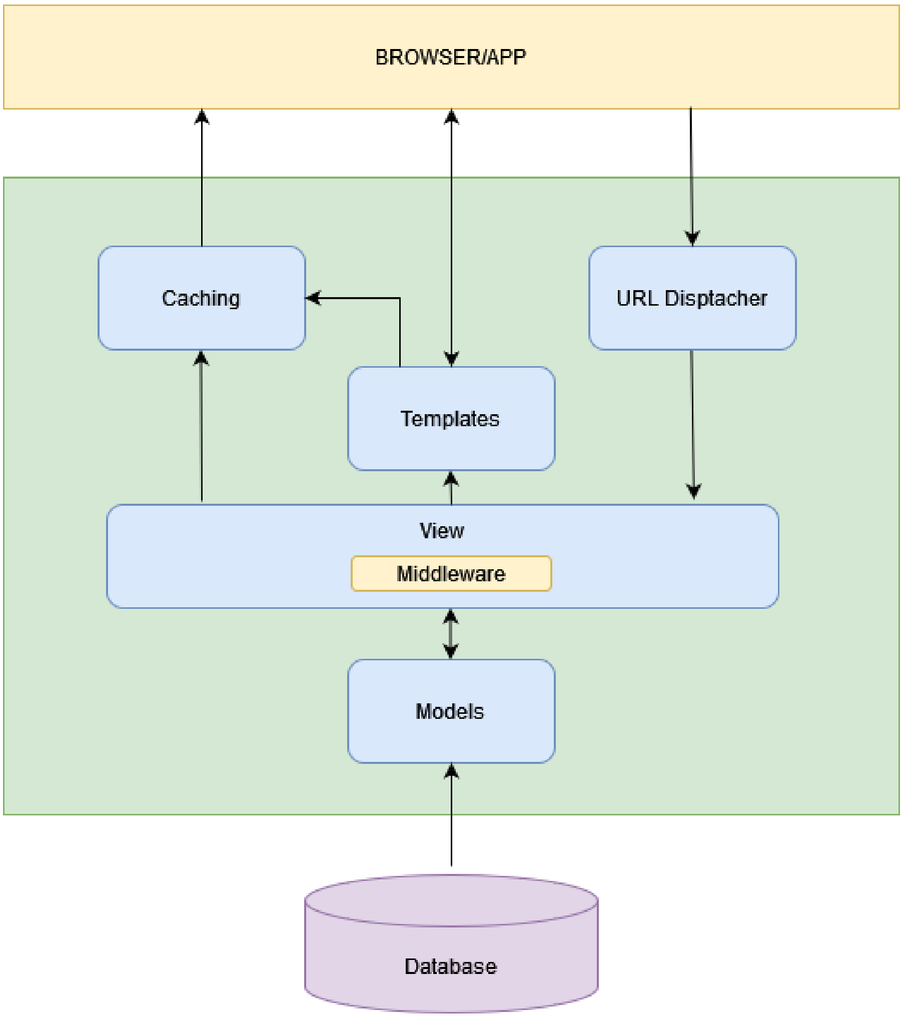
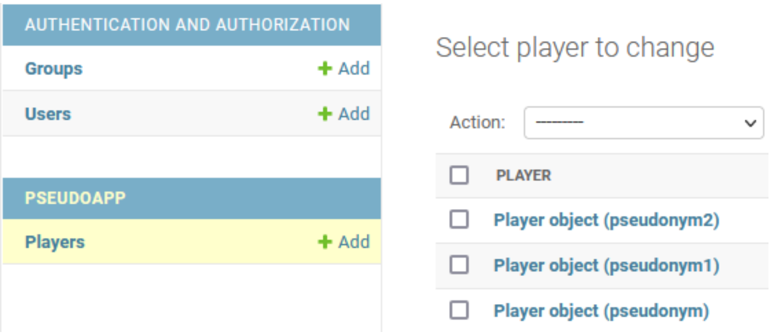


Figure 4.1: Django architecture [1].



AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PSEUDOAPP

Players + Add

Select player to change

Action: ---------

□ PLAYER

□ Player object (pseudonym2)

□ Player object (pseudonym1)

□ Player object (pseudonym)

Figure 4.2: Admin page and some Player objects.

The View, unlike the usual MVC, is used for data formatting, it can insert, update and delete data of the models and sends data to be displayed on the template.

The Template is the presentation, it is what the user sees and what the user can utilize to input data.

Both the Model and View are server sided and the Template is client sided.

To integrate the planned architecture into the framework the MVC architecture was

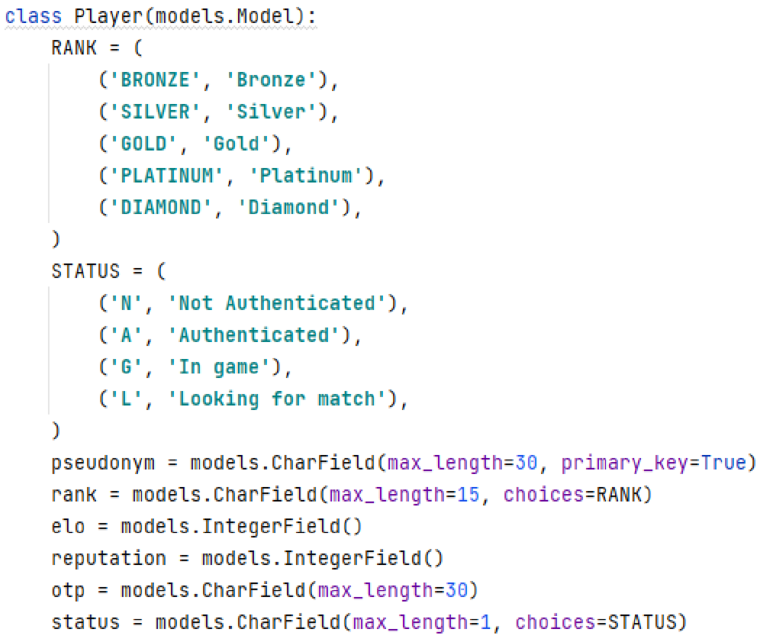
Django	Traditional MVC
Model	Model
Template	View
View	Controller

Table 4.1: Django MVC variant.

devised as follows in Section 4.2.1, 4.2.2 and 4.2.3.

4.2.1 Models

Only the class Player was created in the Model. Even though a class Board and class Matchmaker could easily be implemented with a One-to-Many relationship with the Player, the models are directly connected with the database are keeping the data in it would not allow the requirements (Section 3.1) to be met afterwards.



```
class Player(models.Model):
    RANK = (
        ('BRONZE', 'Bronze'),
        ('SILVER', 'Silver'),
        ('GOLD', 'Gold'),
        ('PLATINUM', 'Platinum'),
        ('DIAMOND', 'Diamond'),
    )
    STATUS = (
        ('N', 'Not Authenticated'),
        ('A', 'Authenticated'),
        ('G', 'In game'),
        ('L', 'Looking for match'),
    )
    pseudonym = models.CharField(max_length=30, primary_key=True)
    rank = models.CharField(max_length=15, choices=RANK)
    elo = models.IntegerField()
    reputation = models.IntegerField()
    otp = models.CharField(max_length=30)
    status = models.CharField(max_length=1, choices=STATUS)
```

Figure 4.3: Player Model.

Figure 4.3 shows the implementation of the PDB, the data that it contains and their limitations: the maximum length of the data, if the data is an integer or a char, and if the data is an enumerate.

The pseudonym is a the primary key of the profile and is a random generated string of number and letters (lower and uppercase) with a maximum length of 20, the OTP is generated the same way.

The rank is a choice of 5 ranks (dividing the player population in 5 quadrants), the status is also a choice and works in sequence, Figure 3.7.

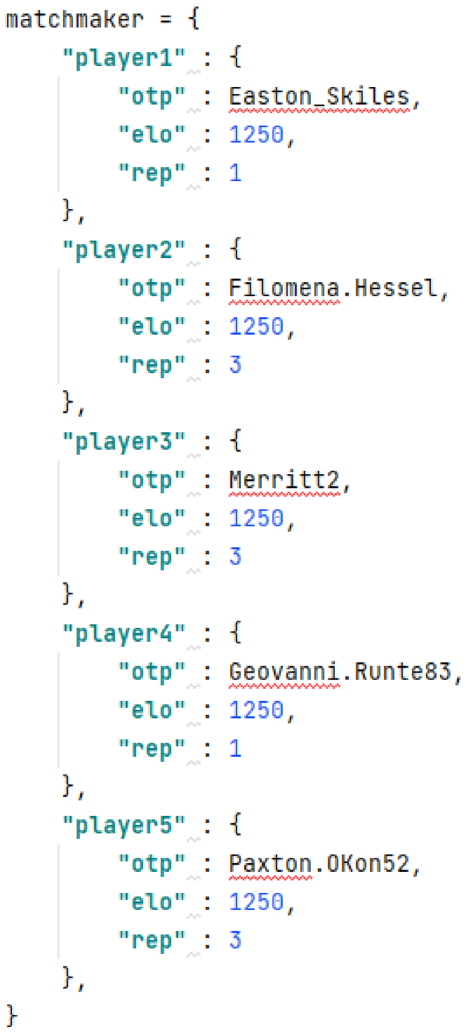
The Elo is an integer ranging from 0 to 2500, 500 is the range for every rank, and the reputation is a 1 trough 5 integer, the higher the integer values the better.

Since the MM and BS cannot be models, they work as functions in the app logic.

The MM functions contain a nested dictionary of 10 players, with the respective OTP, Elo and reputation, that are matched and then sent to a BS, as shown in Figure 4.4.

The BS, since this is a demonstration and not a full on application, does not actually allow the players to play the game but randomly chooses one to win and the other to lose and then sends that information to the MM

The MM after receiving the winners/losers information updates the respective dictionary and then sends it to the PDB individually and with a certain time difference. After that the main dictionary is cleared and the MM is ready to receive a new batch of 10 players.



```
matchmaker = {
    "player1" : {
        "otp" : Easton_Skiles,
        "elo" : 1250,
        "rep" : 1
    },
    "player2" : {
        "otp" : Filomena.Hessel,
        "elo" : 1250,
        "rep" : 3
    },
    "player3" : {
        "otp" : Merritt2,
        "elo" : 1250,
        "rep" : 3
    },
    "player4" : {
        "otp" : Geovanni.Runte83,
        "elo" : 1250,
        "rep" : 1
    },
    "player5" : {
        "otp" : Paxton.OKon52,
        "elo" : 1250,
        "rep" : 3
    },
}
```

Figure 4.4: Example of a matchmaker dictionary with some players.

4.2.2 Views

- auth_view - Authenticates the user via blind signing, also allows for pseudonym creation. If authentication succeeds it moves on to the profile_view.
- profile_view - Displays pseudonym some information from the PDB and redirects the player to chosen matchmaker. After queueing up for a game the game_view takes over.
- game_view - Automated gameplay with a random winner, with the goal of being able to visualize results. When the game ends it returns to the profile_view.

4.2.3 Templates

- auth - Authentication of a already established pseudonym or pseudonym creation.
- auth_error - Error display in case the authentication failed.
- profile - Some information from the pseudonym and possibility to choose matchmaker before the game. The user gets a pop up while in search for a game.
- game - Simple board with the player's OTP showing.

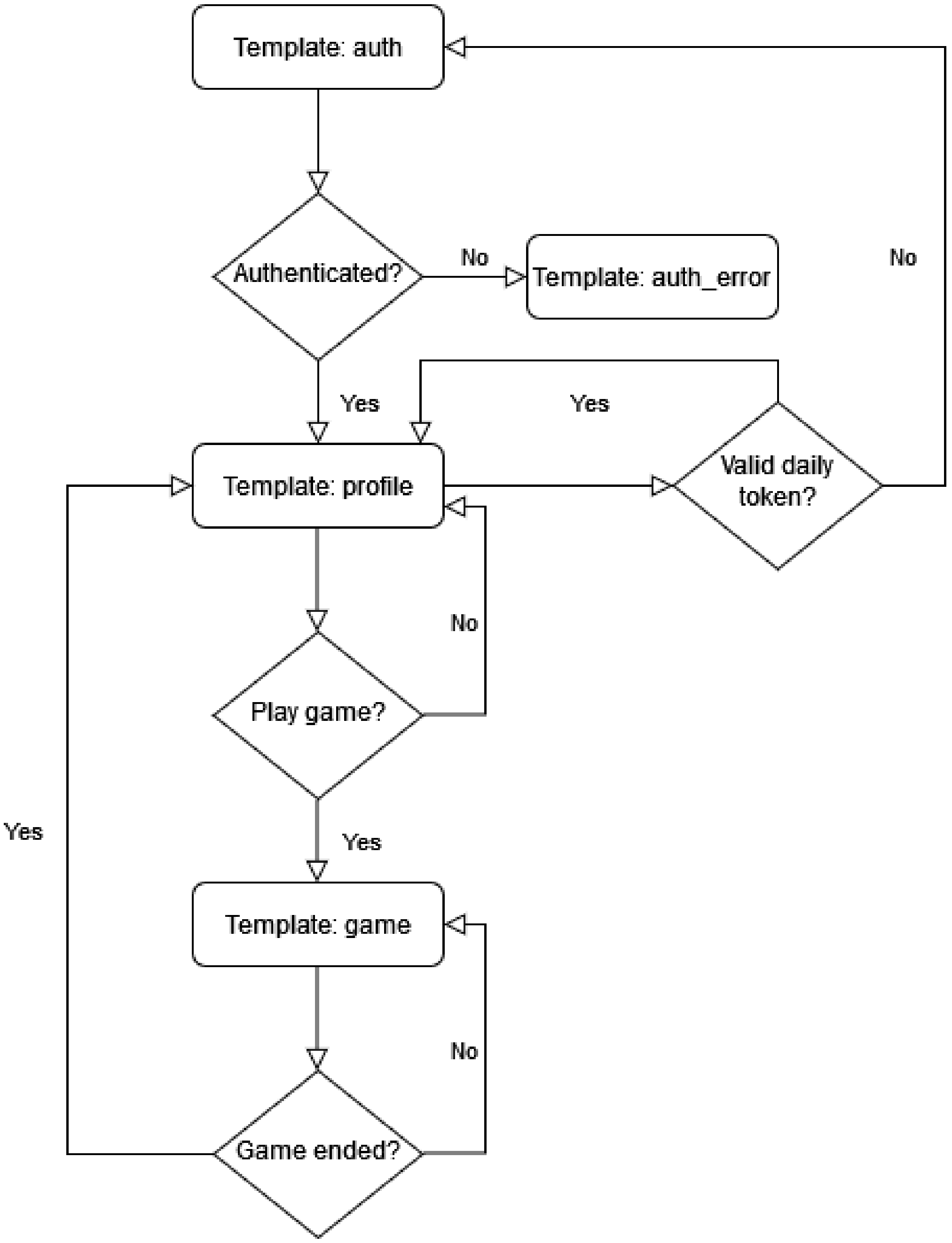


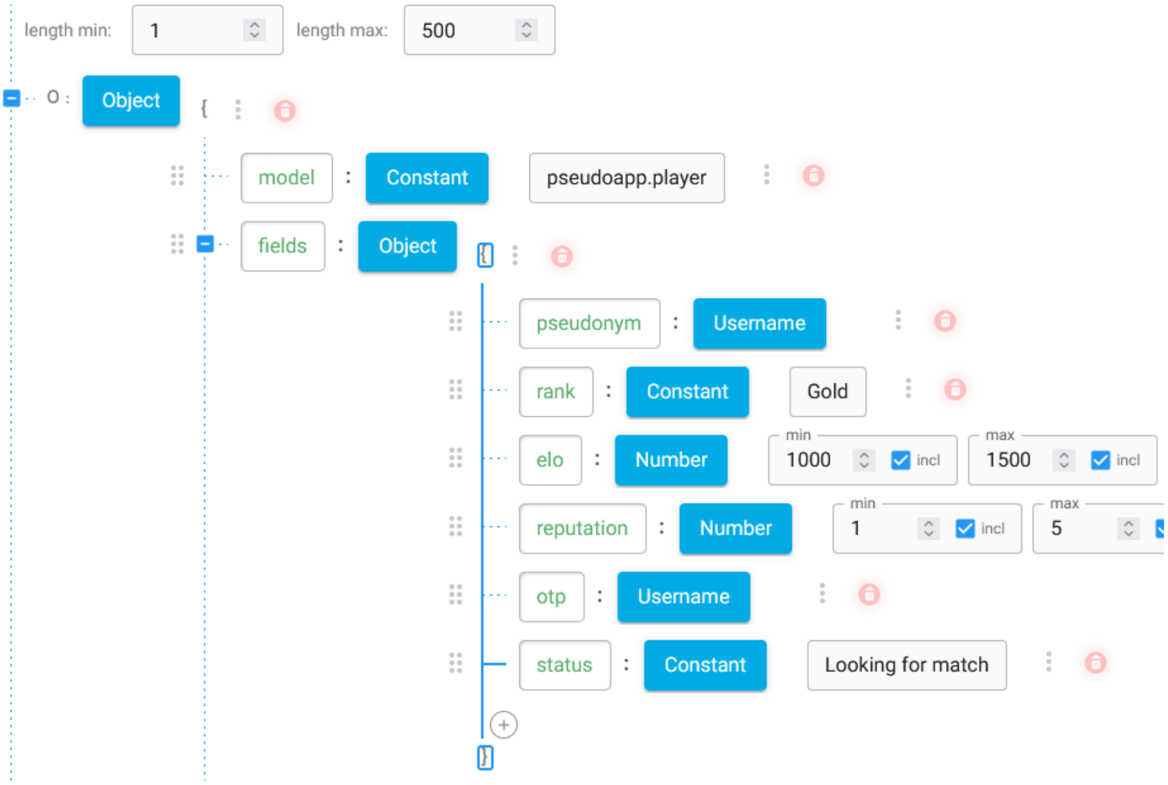
Figure 4.5: Decision graph of the templates.

4.3 Populating the Database

Four json files were created in order to populate the database.

These four files are variants of each other, this is a necessity to obtain good results and to try to understand which characteristic suits Sybil attack prevention better.

- **500players1250elo.json** - This json file contains 500 players with exactly 1250 Elo (middle of the Gold rank).
- **1000players1250elo.json** - This json file contains 1000 players with exactly 1250 Elo (middle of the Gold rank).



length min: 1 length max: 500

0 : Object {

model : Constant pseudoapp.player

fields : Object {

pseudonym : Username

rank : Constant Gold

elo : Number min 1000 ☑ incl max 1500 ☑ incl

reputation : Number min 1 ☑ incl max 5

otp : Username

status : Constant Looking for match

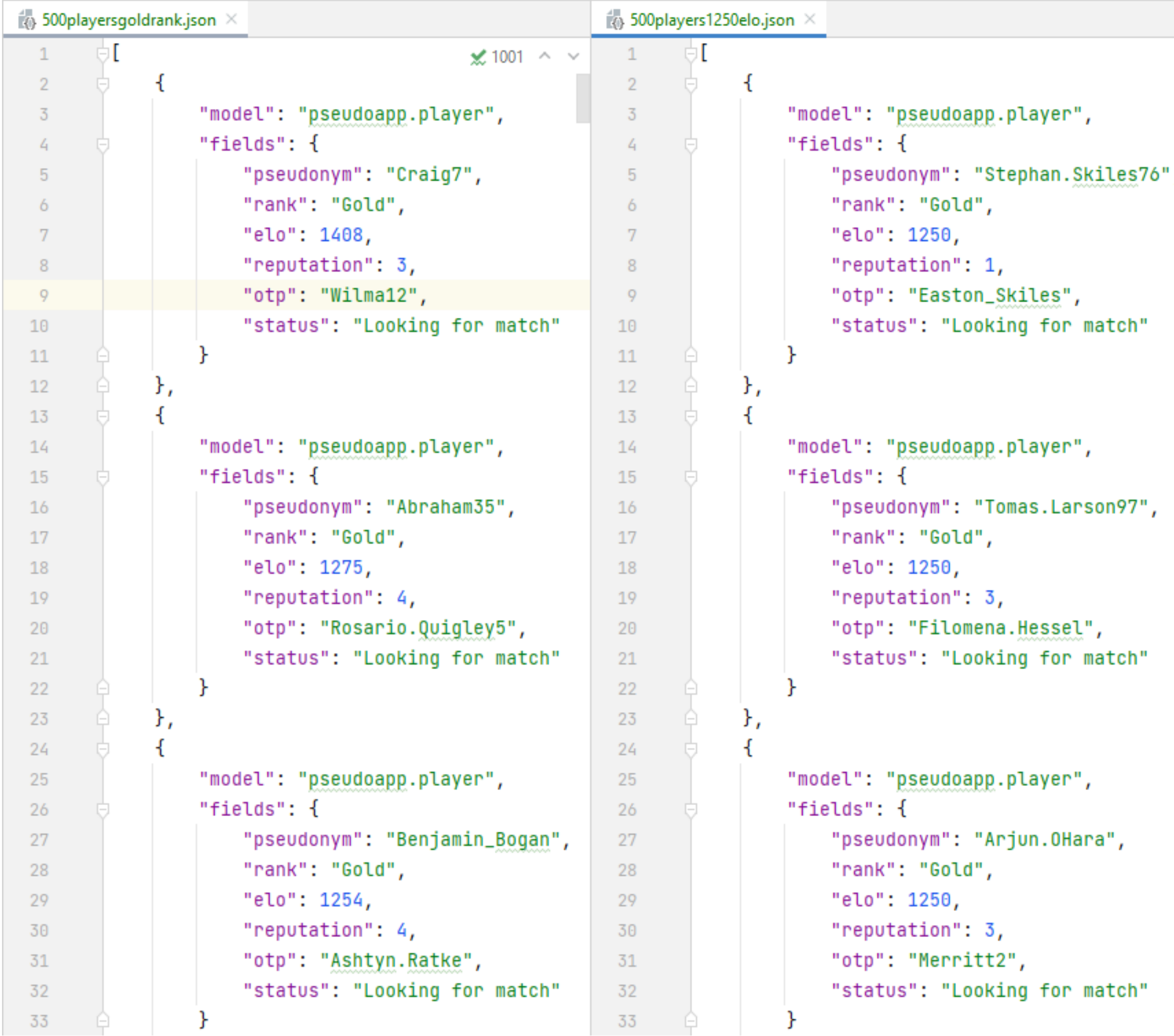
}

Figure 4.6: Json mock to create json files.

- **500playersgoldrank.json** - This json file contains 500 players with exactly Elo between 1000 and 1500 (Gold rank Elo range).
- **1000playersgoldrank.json** - This json file contains 1000 players with exactly Elo between 1000 and 1500 (Gold rank Elo range).

The idea behind having multiple json files to populate the database is to see how a less or more populated database influences Sybil attacks and the same to the Elo spread.

Figure 4.6 shows the mock of the 500playergoldrank.json (data shown in Figure 4.7), where each object has a constant for the model which is required to properly migrate the json into the database. The model constant needs to be the application name, a dot and then the name of the model in question. The field is an object that contains the data needed for the model in the PDB.



500playersgoldrank.json

```json
[
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Craig7",
            "rank": "Gold",
            "elo": 1408,
            "reputation": 3,
            "otp": "Wilma12",
            "status": "Looking for match"
        }
    },
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Abraham35",
            "rank": "Gold",
            "elo": 1275,
            "reputation": 4,
            "otp": "Rosario.Quigley5",
            "status": "Looking for match"
        }
    },
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Benjamin_Bogan",
            "rank": "Gold",
            "elo": 1254,
            "reputation": 4,
            "otp": "Ashtyn.Ratke",
            "status": "Looking for match"
        }
```

500players1250elo.json

```json
[
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Stephan.Skiles76"
            "rank": "Gold",
            "elo": 1250,
            "reputation": 1,
            "otp": "Easton_Skiles",
            "status": "Looking for match"
        }
    },
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Tomas.Larson97",
            "rank": "Gold",
            "elo": 1250,
            "reputation": 3,
            "otp": "Filomena.Hessel",
            "status": "Looking for match"
        }
    },
    {
        "model": "pseudoapp.player",
        "fields": {
            "pseudonym": "Arjun.OHara",
            "rank": "Gold",
            "elo": 1250,
            "reputation": 3,
            "otp": "Merritt2",
            "status": "Looking for match"
        }
```

Figure 4.7: Json files, side to side comparison.

4.4 Authentication

To be able to achieve an authentication without revealing the true identity of someone there is a need to authenticate trough blind signing.

Since the latest cryptography library in Python (Pycryptodome) does not have the blind and unblind functions, like the previous versions did, both functions were implemented in C and used through wrapping.

The blind process according to [3]:

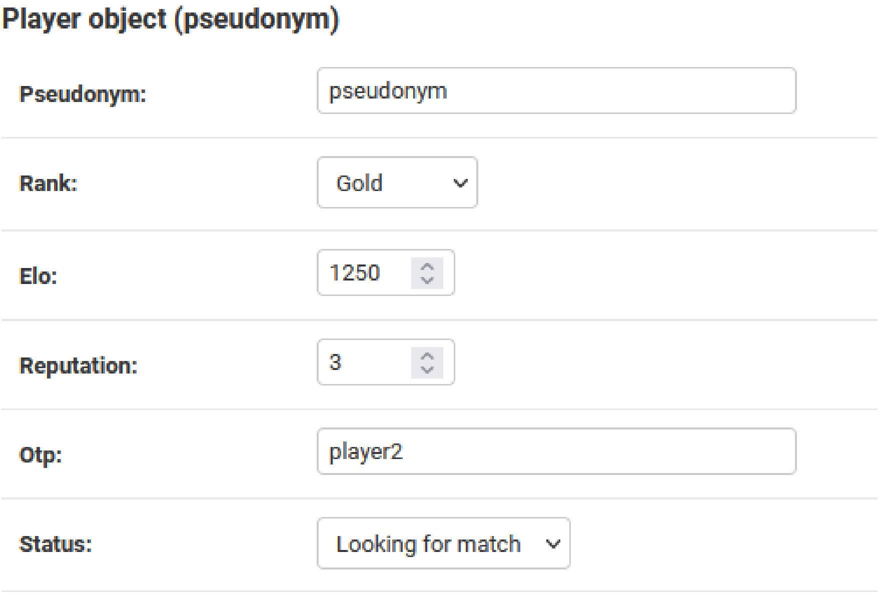
- **blind()** - The "message" that the user wants to blind, using the blind factor, before sending it to the AS is their own pseudonym. The pseudonym is generated automatically for the user in the first instance.
- **sign()** - The AS uses his private key to encrypt the blind message of the user (pseudonym).
- **unblind()** - The user recovers the digital signature of the AS, using the blind factor on the blind message.

- **verify()** - To finalize the authentication and to claim the profile within the PDB, the PDB verifies the signature of AS using its public key.

After the first authentication the user must save its pseudonym to be able to authenticate in the future.

The AS generates the key pair with Pycryptodome, using the private key for the **sign()** and creating a certificate with the public key for easy access.

The user profile is stored in the PDB, example:



Player object (pseudonym)

Pseudonym:	pseudonym
Rank:	Gold
Elo:	1250
Reputation:	3
Otp:	player2
Status:	Looking for match

Figure 4.8: Example of a Player object in the PDB.

4.5 Matching Pseudonyms

No algorithm, steps in order:

1. Queue size of 10, will not start without 10 or more players wanting to play with that matchmaker.
2. Matches randomly within the Queue of 10.
3. No Elo is taken into account.
4. No Elo is gained or lost.
5. Does not update profile in PDB.

Betting algorithm, steps in order:

1. Queue size of 10, will not start without 10 or more players wanting to play with that matchmaker.
2. Each player bets 1% of its current Elo value.
3. Matches as fair as possible (tries to match players with similar Elo values)
4. Elo is taken into account.

5. Elo is gained or lost, according to the bet of the players matched.
6. Updates profile in PDB.

Regular algorithm, steps in order:

1. Queue size of 10, will not start without 10 or more players wanting to play with that matchmaker.
2. Matches as fair as possible (tries to match players with similar Elo values)
3. Elo is taken into account.
4. Elo is gained or lost, -15 from a lose and +15 from a win.
5. Updates profile in PDB.

4.6 Anti Sybil Attack mechanisms

PDB mechanisms:

- Only shows minimal information in the profile, so the users cannot try to manipulate the system into matching them against their Target.

MM mechanisms:

- Queue size of 10 is a must, games will not start without 10 or more players wanting to play with that MM, so players cannot brute force a Sybil attack.
- When the MM receives the 10 OTP from the queue and queries the PDB about their Elo values and reputation values, the queries are done individually (instead of all 10 at the same time), so the PDB does not know who is about to play.
- Updates profile in PDB of each player within the queue individually (instead of all 10 at the same time), so the PDB will not know who played against who.
- Only sends OTP of the players to the BS, so the board will not know who is playing against who.

BS mechanisms:

- Only shows placeholder name of the adversaries or teammates (if they exist), so the users will not know if they matched with their Target.

CHAPTER 5

Results

In this chapter some screenshots are shown and explained, in order to display the results obtained during the tests of the system. The chapter will be divided in two major sections, the anonymity of the system and how well the system prevents Sybil attacks.

5.1 Anonymity

Figures 5.1, 5.2, 5.3, 5.4, 5.5 are all UA interfaces and Figure 5.6 is the BS interface. Figures 5.3, 5.4, 5.5 are also complement by the PDB background data, on the right side of the figure.

Before the user is authenticated he sees two options. One to randomly generate a new pseudonym (Figure 5.1) and another one to authenticate an existing pseudonym (Figure 5.2).

When the user clicks "Generate", Figure 5.1, the UA will ask the PDB to generate a new pseudonym, after receiving the new pseudonym the UA will automatically authenticated the user.

When the user fills the pseudonym placeholder, Figure 5.2, the UA uses the pseudonym to initiate the authentication with the AS. If the authentication fails the AS notifies the UA and the image in the bottom right of the figure is displayed, otherwise the user is redirected to the profile. The user only needs to fill the pseudonym placeholder since the UA already knows all the other information needed for the authentication and does that automatically for the user.

Both options (Generate and Authenticate) will be subjected to the blind signing authentication through the AS.

For this example, the pseudonym “pseudonymTest” was used, as seen in Figure 5.2.

The profile page shows some information of the pseudonym within the PDB, left side of Figure 5.3 displays the pseudonym of the user (pseudonymTest), it is rank (platinum) and the OTP (Schuyler.Glover62). Exactly the same data that the profile holds in the PDB, right side of Figure 5.3.

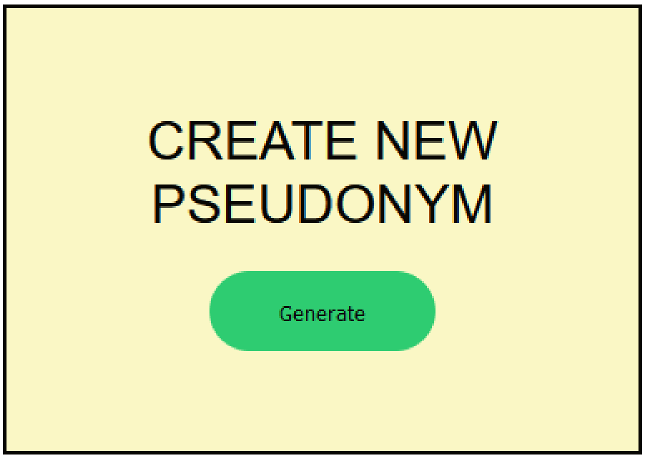

CREATE NEW PSEUDONYM

Figure 5.1: User's view for pseudonym creation.

The user has three options (matchmakers) to select before starting to play. The No Algorithm matchmaker is default, the user can then switch to the other Algorithms if needed (Regular Algorithm or Betting Algorithm)

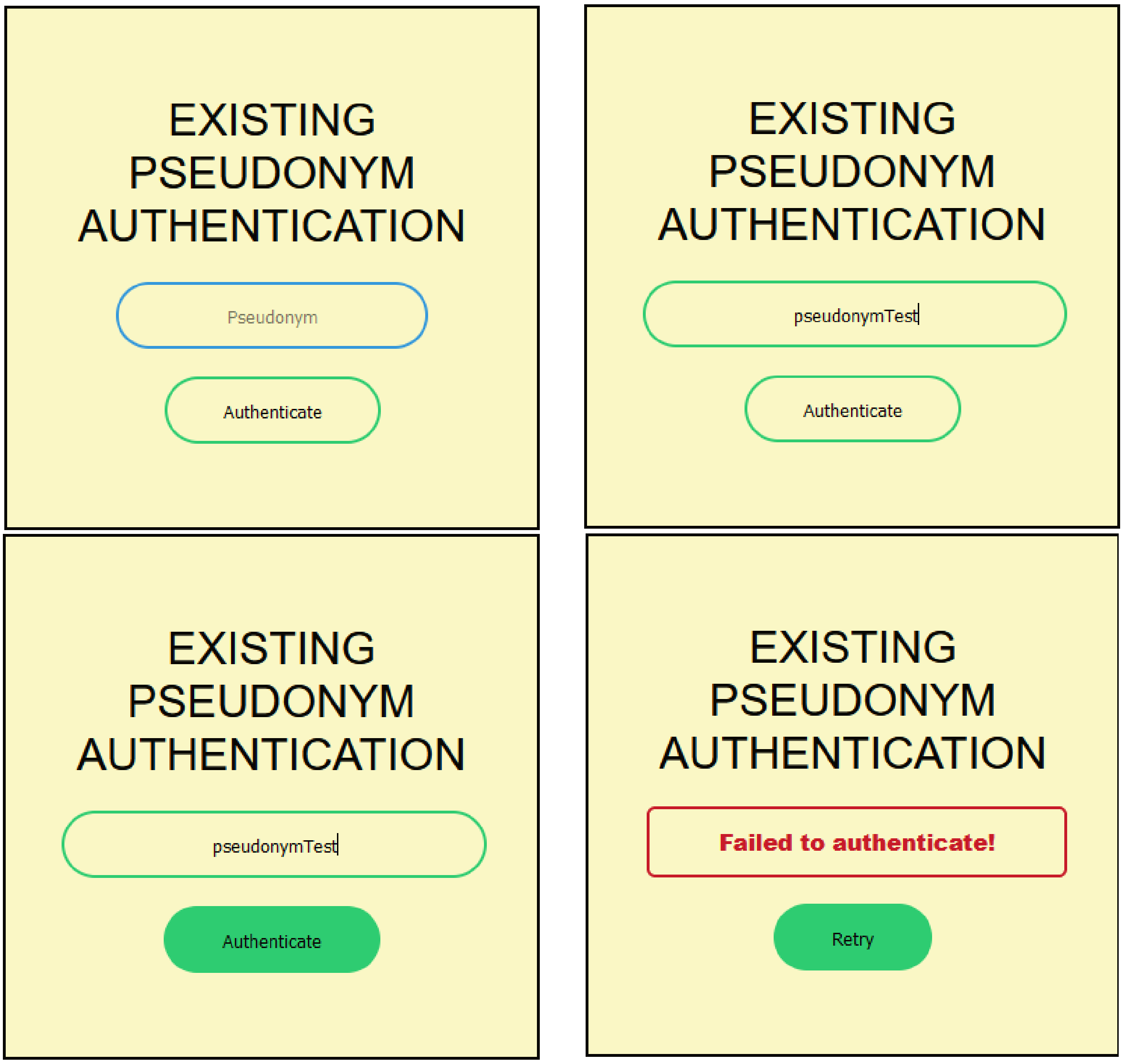
After the user starts looking for a game (click the "Play now!" button) the PDB updated the status of the profile, this way the user is locked from playing multiple games. Figure 5.4 shows the PDB while the profile is looking for a game and also the profile of pseudonymTest while the user is waiting in queue, the user can cancel the match if needed.

Since we did not implement the game itself, Figure 5.6 is a simple placeholder of the BS showcasing how a game would be displayed. The OTP would be replaced by a standard name (Player and a number), to make it harder for players trying to Sybil attack to know if they managed to match against their Target.

The left side of Figure 5.6 shows a match between the user (always Player 1) and another player (which will always be Player 2). This way the user will always see his opponent as Player 2, making it harder to recognize if a Sybil attack was successful. The right side of the figure is exactly the same scenario but for 4 players.

Figure 5.6, also has the reputation system implemented, with a simple thumps up for good behaviour and thumps down for bad behaviour, which will after the game allow for the user's reputation to be updated.

After the game is finished, the user is redirected to the profile page.



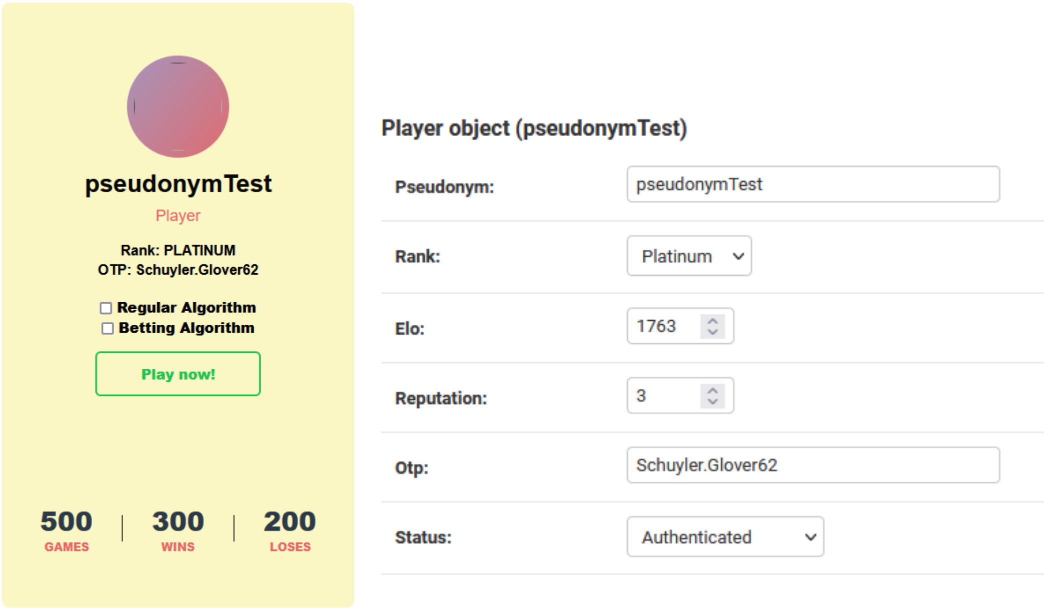
EXISTING PSEUDONYM AUTHENTICATION
Pseudonym
Authenticate
EXISTING PSEUDONYM AUTHENTICATION
Authenticate
EXISTING PSEUDONYM AUTHENTICATION
Authenticate
EXISTING PSEUDONYM AUTHENTICATION
Retry

pseudonymTest

pseudonymTest

Failed to authenticate!

Figure 5.2: Pseudonym authentication with placeholder (top left), inserting pseudonym (top right), "Authenticate" button (bottom left) and failed authentication (bottom right).



pseudonymTest

Player

Rank: PLATINUM
OTP: Schuyler.Glover62

☐ Regular Algorithm
☐ Betting Algorithm

Play now!

500 GAMES | 300 WINS | 200 LOSES

Player object (pseudonymTest)

Pseudonym: pseudonymTest

Rank: Platinum

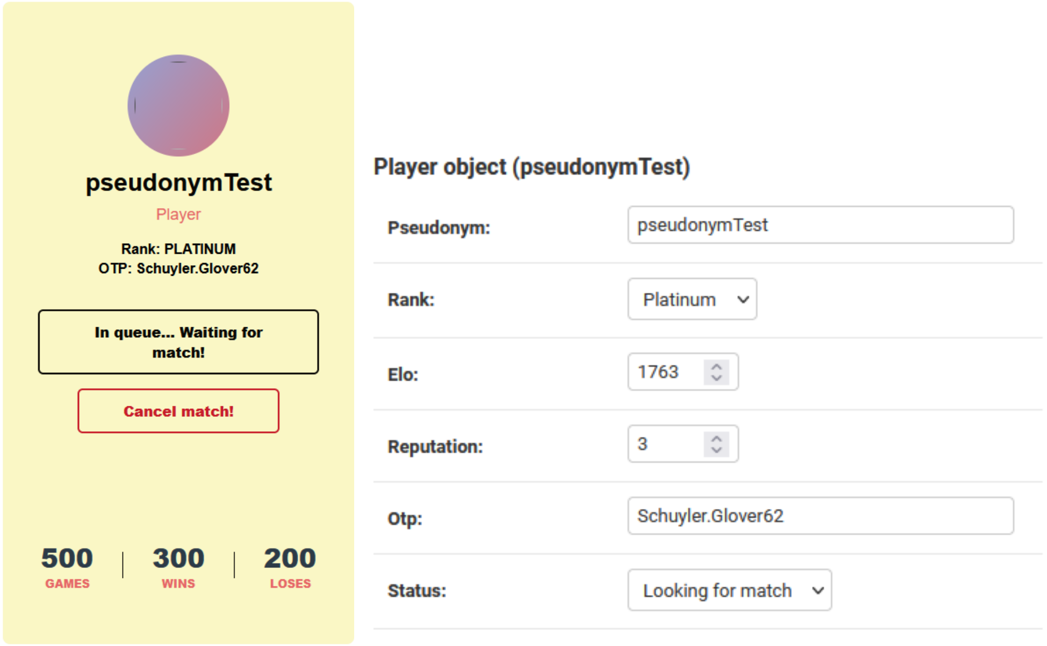
Elo: 1763

Reputation: 3

Otp: Schuyler.Glover62

Status: Authenticated

Figure 5.3: User's profile with available information (left) and their profile in the profile database (right).



pseudonymTest

Player

Rank: PLATINUM
OTP: Schuyler.Glover62

In queue... Waiting for match!

Cancel match!

500 GAMES | 300 WINS | 200 LOSES

Player object (pseudonymTest)

Pseudonym: pseudonymTest

Rank: Platinum

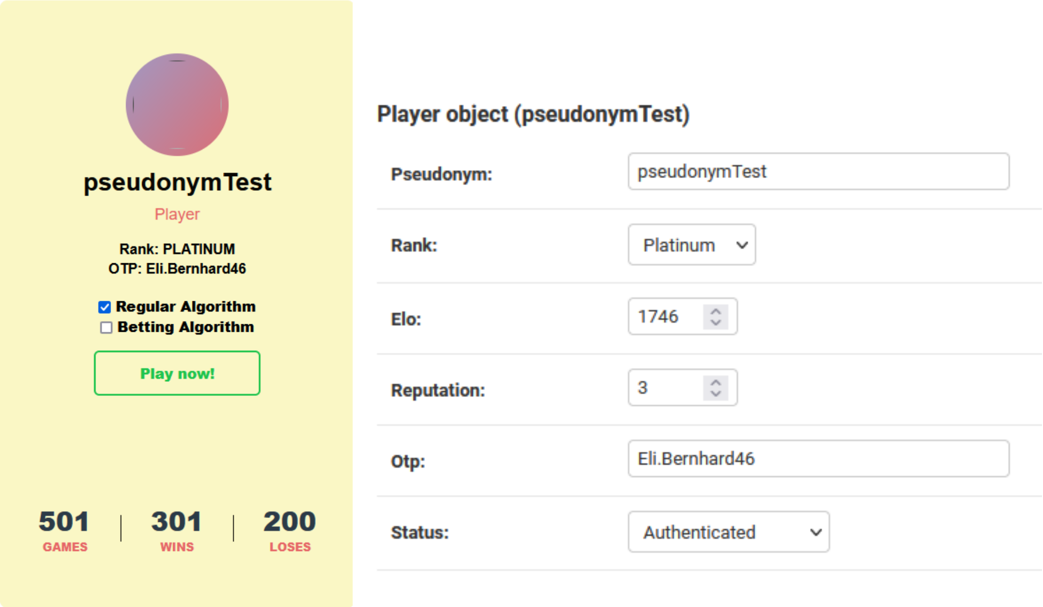
Elo: 1763

Reputation: 3

Otp: Schuyler.Glover62

Status: Looking for match

Figure 5.4: Profile while in queue for match (left) and profile database while user is looking for a game (right).



pseudonymTest

Player

Rank: PLATINUM
OTP: Eli.Bernhard46

☑ Regular Algorithm
□ Betting Algorithm

501 GAMES | 301 WINS | 200 LOSES

Player object (pseudonymTest)

Pseudonym:	pseudonymTest
Rank:	Platinum
Elo:	1746
Reputation:	3
Otp:	Eli.Bernhard46
Status:	Authenticated

Figure 5.5: User's profile with available information after having played a game (left) and profile database after user finishes a game (right).

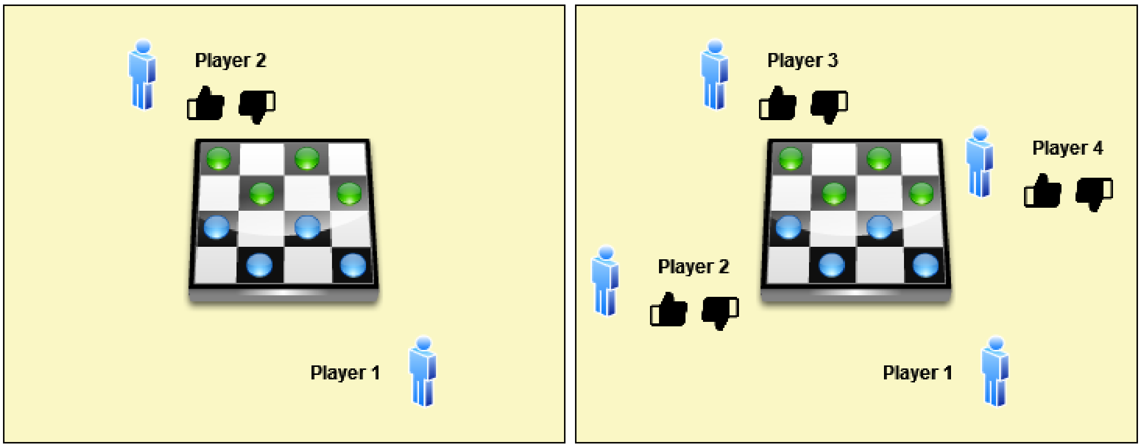


Figure 5.6: Match example with 2 players (left) and match example with 4 players (right).

To determine if the system can indeed avoid Sybil attacks multiple datasets, algorithms and variations were tested.

To test this a Target was defined from within the dataset in every single test, this Target can be a pseudonym of our own or a pseudonym of someone that we want to attack.

Every test was done against the range of 1 through 4 pseudonyms with the Target playing a million games (1.000.000), the results will be the median value of 10 repetitions of the same exact test. That median value will be compared to the real probability value.

In the tables and boxplots the target and number of pseudonyms in question will be represented in this way: T+1, T+2, T+3 and T+4. Where the T stands for Target and the number following the amount of pseudonyms owned.

The base test will be done in a 500 population dataset, with 10 queue size, no ranking algorithm and be a game of two. The games will also be instant.

What will be varying:

- Datasets.
- Algorithms - Changing algorithms will allow for algorithms comparison as well as testing against no algorithm usage (base test).
- Players available trough game time - With varied game time within the player population will show if shortening the player pool will affect the results.
- Queue size - A bigger queue size will display results when the Target will have more suitable matches.
- Player population size - A bigger population size will show results if a game is more popular.
- Types of games (2 players and 4 players) - Different games (games with 2 players and games with 4 players) will display how different games can be affected by the system.
- Player's Elo proximity - Changing the proximity of the player's Elo will showcase if it has major interference in the results.

5.2.1 Histogram of the datasets

The following histograms, Figure 5.7 and Figure 5.8, show the distribution of the players within the gold rank, 1000 to 1500 Elo range.

The histograms show the starting point of both datasets before testing. The Target will be always a player around the 1250 Elo mark. the other pseudonyms selected will always be within the range of 1000 to 1500 Elo and have to respect the requirements of each test.

The datasets with a default Elo (1250) value for every player have no histogram since it would be pointless.

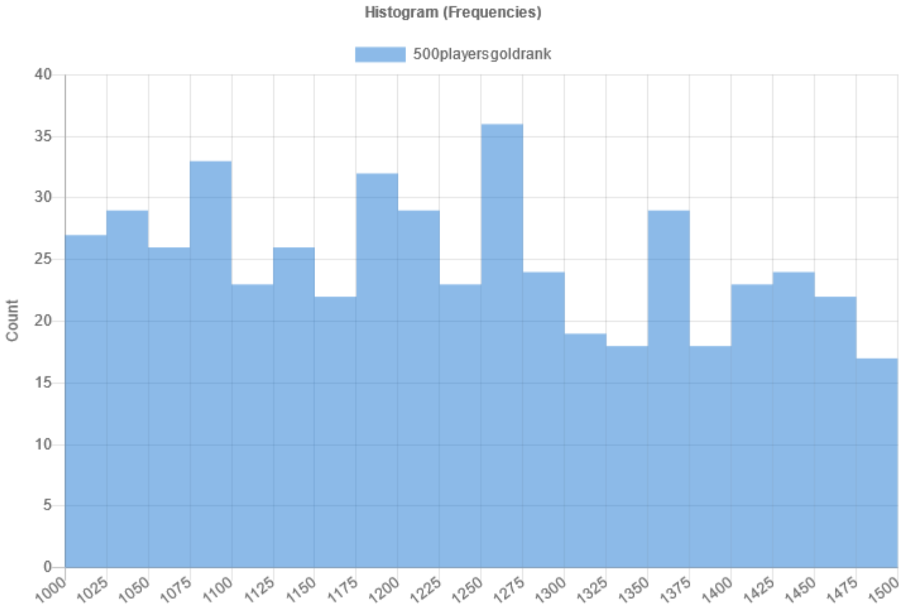


Figure 5.7: Histogram of the players rankings from the 500playersgoldrank.json.

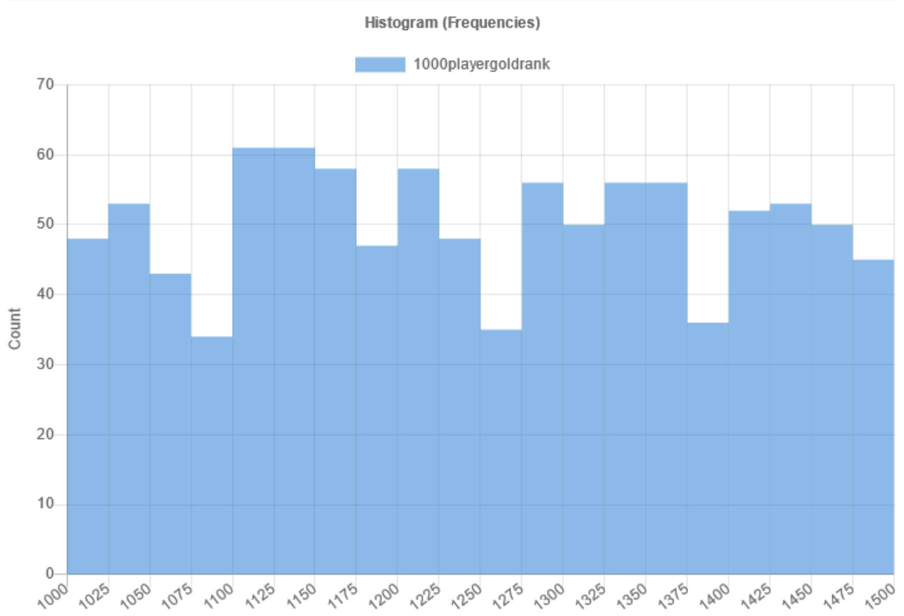


Figure 5.8: Histogram of the players rankings from the 1000playersgoldrank.json.

5.2.2 Results of testing a 500 player population without a ranking algorithm (Base test)

The results of this test will be used as base for comparison will all the others.

As expected, in a population of 500 players and without a algorithm that matches them with Elo values into account the percentage obtained will be really close to the real percentage.

The Table 5.1 shows how close the results were compared to the real values.

Figure 5.9, shows that all examples are within the range of a hundred, more or less, games than the real value.

These boxplots will be the base to compare with the other results.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Theoretic
T+1	1 980	1 959	1 992	2 035	1 937	$1.980 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 996	3 947	4 049	4 114	3 930	$3.996 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 954	5 930	6 002	6 088	5 912	$5.954 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	8 031	7 933	8 062	8 282	7 886	$8.031 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.1: Results from varying the number of pseudonyms, testing a 500 player population without a ranking algorithm, 500players1250elo.json file population.

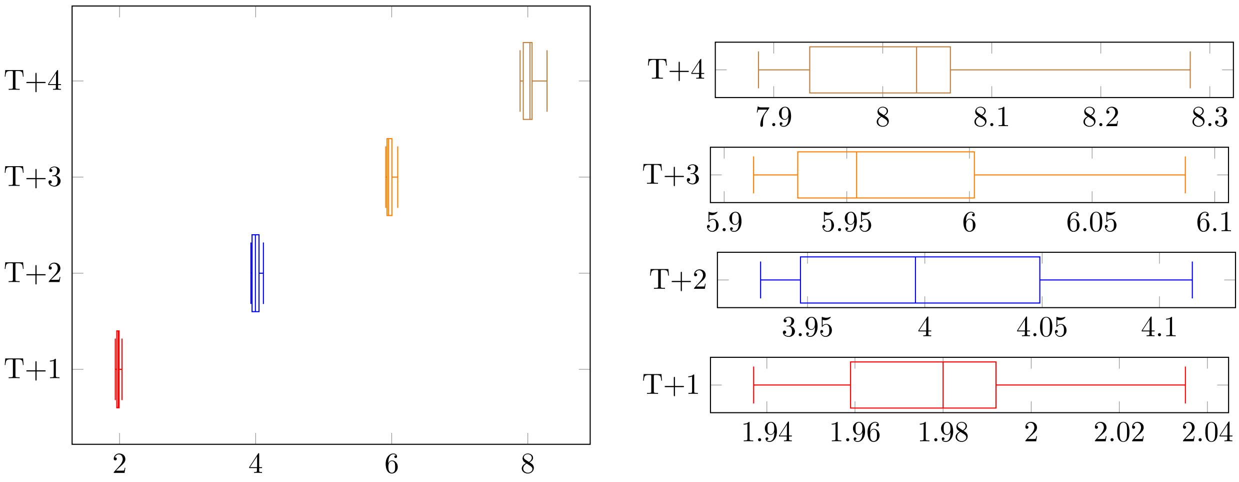


Figure 5.9: Boxplots of Table 5.1 results.

5.2.3 Results of testing a 1000 player population without a ranking algorithm

After doubling the population of the dataset and with the same exact algorithm the results are as expected, the percentage was halved.

This indicates that the bigger the population of players the harder it becomes to have a successful Sybil attack.

The Table 5.2 shows how close the results were compared to the theoretic values.

Figure 5.10, shows that T+1, T+2 and T+3 are all within the range of a hundred, more or less, games than the real value. T+4 has some more variance but it is to be expected since there are more pseudonyms to work with in a bigger dataset.

All Figures also show how when the population is multiplied by a value the result of matches played against target will be divided by that number.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Theoretic
T+1	991	969	1 021	1 048	950	$0.991 \cdot 10^{-3}$	$\frac{1}{999} = 1001 \cdot 10^{-3}$
T+2	2 004	1 944	2 031	2 093	1 937	$2.004 \cdot 10^{-3}$	$\frac{2}{999} = 2.002 \cdot 10^{-3}$
T+3	3 003	2 937	3 062	3 089	2 887	$3.003 \cdot 10^{-3}$	$\frac{3}{999} = 3.003 \cdot 10^{-3}$
T+4	3 997	3 979	4 124	4 167	3 937	$3.997 \cdot 10^{-3}$	$\frac{4}{999} = 4.004 \cdot 10^{-3}$

Table 5.2: Results from varying the number of pseudonyms, testing a 1000 player population without a ranking algorithm, 1000players1250elo.json file population.

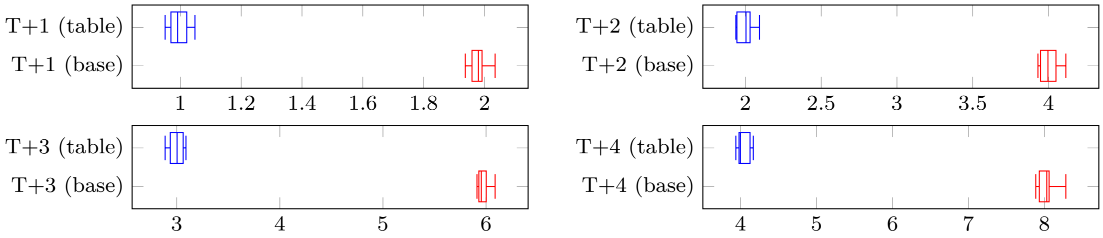


Figure 5.10: Boxplots of Table 5.2 results against the base from Table 5.1.

5.2.4 Results of testing a 500 player population without a ranking algorithm but with a bigger queue list (20 instead of 10)

Having the queue size doubled resulted in slightly worse results, but nothing too major.

With this results, it is obvious that apart from blocking games when there are less than a certain number of players, that the queue does not actually block Sybil attacks.

The Table 5.3 shows how close the results were compared to the theoretic values.

Figure 5.11 shows that all tests are all within the range of a hundred, more or less, games than the real value.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Theoretic
T+1	2 004	1 969	2 032	2 033	1 880	$2.004 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 996	3 947	4 049	4 114	3 930	$3.996 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 954	5 930	6 002	6 088	5 912	$5.954 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	8 031	7933	8 062	8 282	7, 886	$8.031 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.3: Results from varying the number of pseudonyms, testing a 500 player population without a ranking algorithm, but with a bigger queue list (20 instead of 10), 500players1250elo.json file population.

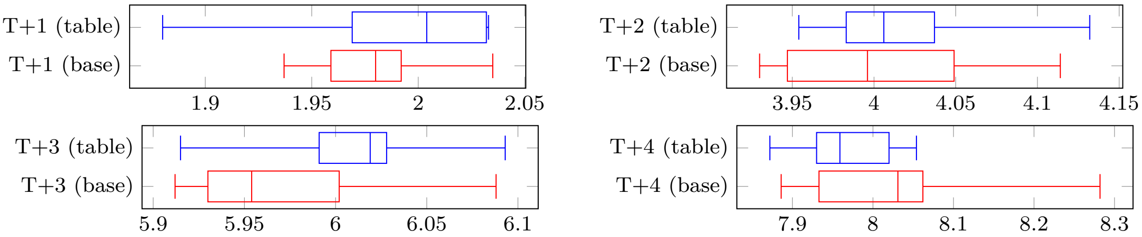


Figure 5.11: Boxplots of Table 5.3 results against the base from Table 5.1.

5.2.5 Results of testing a 500 player population with game time being different

For this test, each player after a game will be given a number, between 0 and 2, to simulate games ending at different times. Players can only be matched if the sum of their time numbers are the same. The queue limit is still considered, a player cannot start a game without 9 other players in the same matchmaker.

Figure 5.12 shows that all tests maximum values can go up to three times the base value, since there were three different match times being tested this makes sense. T+2 and T+4 maximum value does not quite reach three times the base value, but it can happen.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Theoretic
T+1	4 308	2 836	4 875	5 618	2 029	$4.308 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	7 075	5 128	8 615	9 670	3 883	$7.075 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	10 679	6 892	14 132	16 960	5 796	$10.679 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	11 546	10 406	14 833	18 977	8 408	$11.546 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.4: Results from varying the number of pseudonyms, testing a 500 player population with match times being different, 500players1250elo.json file population.

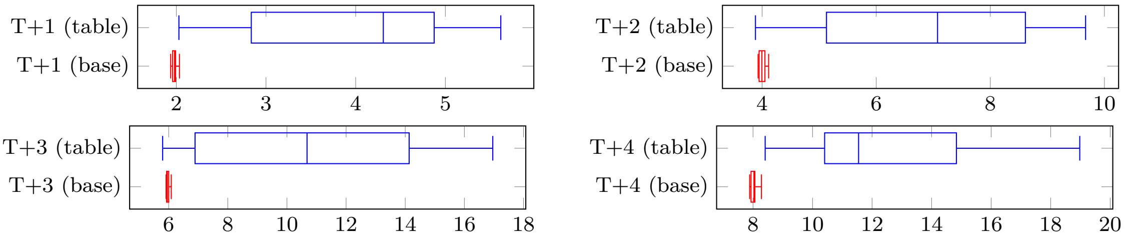


Figure 5.12: Boxplots of Table 5.4 results against the base from Table 5.1.

5.2.6 Results of testing a 500 player population without a ranking algorithm but in a game of four players instead of two

When raising the number of players allowed within a game, the number of matches where our pseudonyms play against the Target increases significantly.

This happens because, in this case, instead of a vacant spot to match our Target there are now three spots. One to play with the Target and two to play against it.

Every spot increases the percentage of a possible game against a owned pseudonym, consequently the percentage is three times higher.

The Table 5.5 shows how close the results were compared to the real values.

Figure 5.13 shows that T+1 is within the range of a hundred, more or less, games than the real value, but T+2, T+3 and T+4 all vary a little more from the the expected value, but the median value still holds true to the real value.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Theoretic
T+1	6 011	5 977	6 022	6 064	5 924	$6.011 \cdot 10^{-3}$	$\frac{1}{499} + \frac{1}{498} + \frac{1}{497} = 6.024 \cdot 10^{-3}$
T+2	11 980	11 883	12 067	12 178	11 790	$11.980 \cdot 10^{-3}$	$\frac{2}{499} + \frac{2}{498} + \frac{2}{497} = 12.048 \cdot 10^{-3}$
T+3	18 043	17 926	18 085	18 300	17 818	$18.043 \cdot 10^{-3}$	$\frac{3}{499} + \frac{3}{498} + \frac{3}{497} = 18.072 \cdot 10^{-3}$
T+4	24 119	23 924	24 140	24 363	23 671	$24.119 \cdot 10^{-3}$	$\frac{4}{499} + \frac{4}{498} + \frac{4}{497} = 24.096 \cdot 10^{-3}$

Table 5.5: Results from varying the number of pseudonyms with the 500players1250elo.json file population in game of four players.

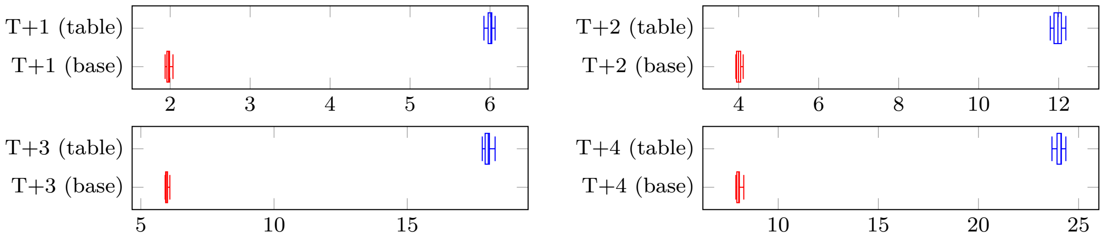


Figure 5.13: Boxplots of Table 5.5 results against the base from Table 5.1.

5.2.7 Results of testing a 500 player population with a regular ranking algorithm with Target and pseudonyms with close Elo values

The regular algorithm was applied during this test. The pseudonyms chosen to "attack" the Target all started the tests within 50 Elo of the target.

Comparing to Table 5.1 where no algorithm was applied, there is a slight decrease in games played against the Target. This comes from the fact that the pseudonyms and the Target with the Elo increases/decreases can become closer or not in Elo. When they become too distant the probability of matching is lower.

Figure 5.14, shows an increase is variance during the tests.

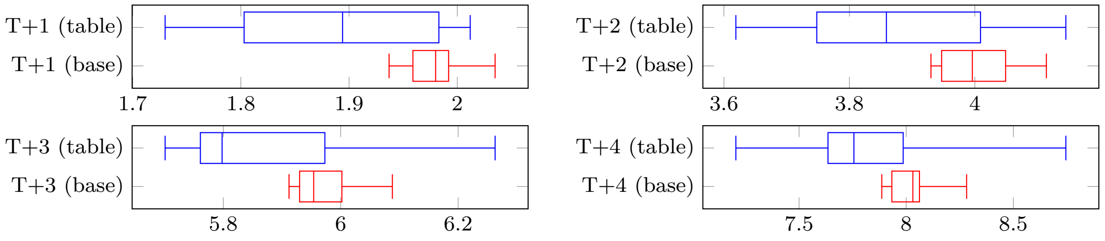


Figure 5.14: Boxplots of Table 5.6 results against the base from Table 5.1.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Base
T+1	1 894	1 803	1 983	2 012	1 730	$1.894 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 859	3 748	4 009	4 145	3 619	$3.859 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 798	5 761	5 973	6 263	5 701	$5.798 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	7 756	7 635	7 986	8 745	7 205	$7.756 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.6: Results from varying the number of pseudonyms with the 500players1250elo.json file population.

5.2.8 Results of testing a 500 player population with a regular ranking algorithm with Target and pseudonyms with distant Elo values

The regular algorithm was applied during this test. The pseudonyms chosen to "attack" the Target all started the tests further than 200 away of the target.

Comparing to Table 5.6 where the pseudonyms were closer in Elo, there is a slight decrease in games played against the Target, but nothing much. This happens because the number of games is big enough and the Elos values increase and decrease each game, leading to the pseudonyms distance in Elo value to become closer and distant multiple times during the testing.

Figure 5.15 shows a variance similar to their counterparts in 5.2.7, meaning that starting with close or distant Elo values may not have a very big effect on the overall results.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Base
T+1	1 885	1 843	1 928	1 953	1 742	$1.885 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 840	3 674	3 967	4 204	3 391	$3.840 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 790	5 740	5 914	6 201	5 645	$5.790 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	7 639	7 450	7 907	8 856	7 134	$7.639 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.7: Results from varying the number of pseudonyms, testing a 500 player population with a regular ranking algorithm with Target and pseudonyms with close Elo values, with the 500players1250elo.json file population.

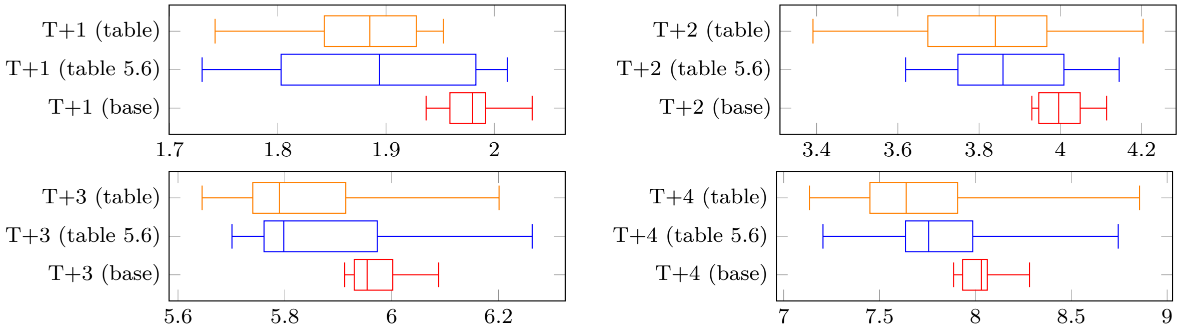


Figure 5.15: Boxplots of Table 5.7 results against Table 5.6 and the base from Table 5.1.

5.2.9 Results of testing a 500 player population with a betting ranking algorithm with Target and pseudonyms with close Elo values

The betting algorithm was applied during this test. The pseudonyms chosen to "attack" the Target all started the tests within 50 Elo of the target.

Comparing to Table 5.1 where no algorithm was applied, there is a slight decrease in games played against the Target. This comes from the fact that the pseudonyms and the Target with the Elo increases/decreases can become closer or not in Elo. When they become too distant the probability of matching is lower.

Comparing to Table 5.6 where the regular algorithm 5.2.7 was applied, the tests with one and four pseudonyms got better results but the tests with two and three were worse, meaning that both algorithms improve Sybil attack prevention but it is hard to say which is better.

Figure 5.16 shows similar results to the previous Figure 5.15 of the regular algorithm.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Base
T+1	1 880	1 829	1 998	2 026	1 782	$1.880 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 893	3 779	3 983	4 167	3 725	$3.893 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 812	5 763	5 952	6 225	5 572	$5.812 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	7 729	7 678	7 986	8 668	7 307	$7.729 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.8: Results from varying the number of pseudonyms, testing a 500 player population with a betting ranking algorithm with Target and pseudonyms with close Elo values, with the 500players1250elo.json file population.

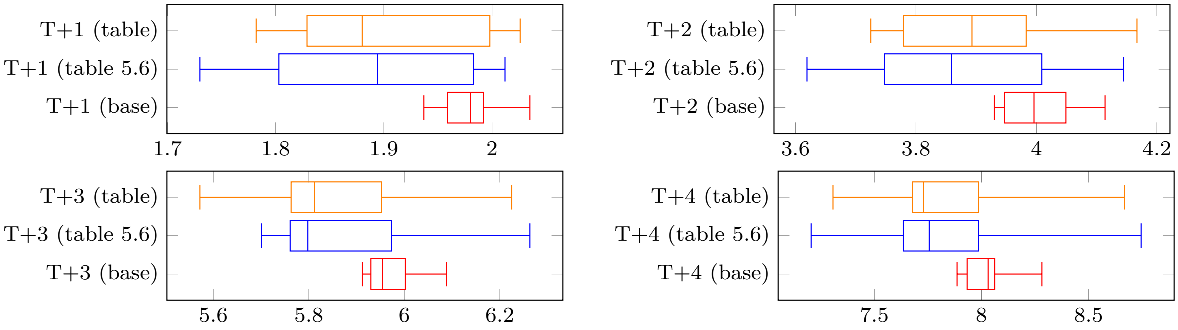


Figure 5.16: Boxplots of Table 5.8 results against Table 5.6 and the base from Table 5.1.

5.2.10 Results of testing a 500 player population with a betting ranking algorithm with Target and pseudonyms with distant Elo values

The betting algorithm was applied during this test. The pseudonyms chosen to "attack" the Target all started the tests further than 200 away of the target.

Comparing to Table 5.8 where the pseudonyms were closer in Elo, there is a slight decrease in games played against the Target, but nothing much, just like with the regular algorithm. This happens because the number of games is big enough and the Elo values increase and decrease each game, leading to the pseudonyms distance in Elo value to become closer and distant multiple times during the testing.

Again comparing the both algorithms seems to lead to a dead end. Both are a improvement when put against a no-algorithm (base test) but there does not seem to be a clear winner when comparing the two (regular and betting algorithms).

Figure 5.17, shows a variance similar to their counterpart in 5.2.9.

Pseudonyms	Observed results					Probability	
owned	Median	Q1	Q3	Maximum	Minimum	Obtained	Base
T+1	1 890	1 835	1 926	1 983	1 785	$1.890 \cdot 10^{-3}$	$\frac{1}{499} = 2.004 \cdot 10^{-3}$
T+2	3 849	3 774	3 993	4 184	3 263	$3.849 \cdot 10^{-3}$	$\frac{2}{499} = 4.008 \cdot 10^{-3}$
T+3	5 811	5 773	5 956	6 186	5 675	$5811 \cdot 10^{-3}$	$\frac{3}{499} = 6.012 \cdot 10^{-3}$
T+4	7 679	7 450	7 936	8 723	7 234	$7.679 \cdot 10^{-3}$	$\frac{4}{499} = 8.016 \cdot 10^{-3}$

Table 5.9: Results from varying the number of pseudonyms, testing a 500 player population with a betting ranking algorithm with Target and pseudonyms with distant Elo values, with the 500players1250elo.json file population.

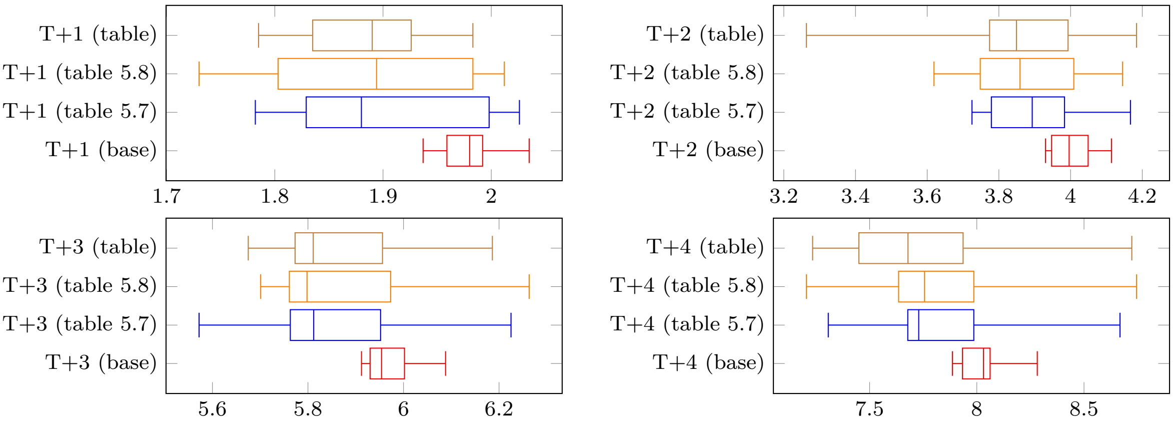


Figure 5.17: Boxplots of Table 5.9 results against Table 5.7, Table 5.8 and the base from Table 5.1.

5.2.11 Discussion of the results

After observing the results of the previous sections and having Section 5.2.2 as a comparison to all the variations tested, it is possible to elaborate upon each topic pointed out in the main Section 5.2.

First of all, in Section 5.2.2, it is clear that if the number of pseudonyms owned increases the chances of matching with the Target also increases.

Regarding the algorithms, both the regular algorithm and the betting algorithm got better median results than the base test. But from the results it is hard to conclude which one is better, because the margins are so small between the two and there is not one clear algorithm outperforming the other. Since there are changes of Elo values the minimum and maximum values also decrease and increase respectively, because players' Elo rankings become distant or close accordingly. On topic with the distant and close Elo values of the pseudonyms, there does not seem to have much impact on the results if the pseudonyms start close or distant because it eventually evens out across the million matches.

When the game time is different, Section 5.2.5, the amount of players that are available decreases, consequently the median and maximum values also increase even though the minimum values may even be lower than the base minimum value.

The results obtained from testing a bigger queue, Section 5.2.4, are very similar to the base results with minimal discrepancies between the two. Although the queue is necessary to avoid player forcing matches, the actual size of the queue does not really affect the results too much.

Changing the population size of the dataset used led to obvious results, Section 5.2.3. With a dataset with double the population size of the base dataset all values halved across the board, which was expected.

When testing a game of 4 players instead of a game of 2 players (base test), Section 5.2.6, the probability increased by three fold in every result. Taking the T+1 results as an example, the pseudonym owned will have three chances of getting matched against the target instead of only one. So the number of players being matched to a game proportionally increases the probability of being matched against the target.

CHAPTER 6

Conclusion

This chapter presents a final conclusion about this thesis, describing the problem, the solution and what can be looked upon as future work.

Table games are usually played in groups of people, players play against each other to see who can best the other. Table games are associated with hands on and physically present games but with the growth of online gaming, they have also moved on to online platforms. This allows the games to grow in popularity and for players to play against players from across the world and with various skill levels. Most online games nowadays have some type of competitive ladder, normally using the Elo ranking system. If there is something to gain players will try their upmost to win but sometimes they may test the system and attempt to cheat.

In this dissertation, we tried to minimize the probability of a player being able to cheat trough Sybil Attack. In order to easily win games and gain competitive advantage, the players may attempt to force game against themselves. If a player owns more than one way to play the game or has control over multiple people's gaming user, they can try to force a Sybil Attack.

To counter this fact and to prevent Sybil Attacks from happening we created a system using pseudonyms and having anonymity in mind. Our system uses both long term pseudonyms, to save data from the users, and short term pseudonyms, to maintain anonymity during the matching process. The system has four components that work independently and share has little data of the user has possible, the AS knows the user's identity but does not know any of his pseudonyms, the PDB knows all of the user's pseudonyms but does not know his identity and the MM and BS only know the short term pseudonym, the OTP. This way even if the system is compromised matches can not be forced.

Even though matches can not be forced, the system can not fully stop one player to play against another. So we needed to minimize the chances of that happening. So in order to claim a pseudonym the user has to identify himself with the AS and receives a daily token in order to avoid the use of multiple pseudonyms by the same person. Also, from observing the results, some key variations can be pointed out as clear Sybil Attack prevention methods. Such as a big player population, a queue restriction for the matchmaker even though its size does not really impact the end result, having a matchmaker with an algorithm that takes into account Elo values. But we were also able to visualize what can increase the probability of a Sybil Attack, such as games with more players and more pseudonyms owned by the attacker.

For future work, the psychology behind the players' actions could be studied and taken into account to try and improve the system already implemented. Since changing the number of players per match is impossible in certain games and completely stopping someone from gathering multiple pseudonyms is also difficult.

References

[1] S. P. Adithela, M. Christie, S. Marru, and M. Pierce, “Django Content Management System Evaluation and Integration with Apache Airavata,” in *Proceedings of the Practice and Experience on Advanced Research Computing*, 2018, pp. 1–4.

[2] N. Balachandran and S. Sanyal, “A review of techniques to mitigate sybil attacks,” *arXiv preprint arXiv:1207.2617*, 2012.

[3] D. Chaum, “Blind signatures for untraceable payments,” in *Advances in cryptology*, Springer, 1983, pp. 199–203.

[4] T. Chesney, I. Coyne, B. Logan, and N. Madden, “Griefing in virtual worlds: Causes, casualties and coping strategies,” *Information Systems Journal*, vol. 19, no. 6, pp. 525–548, 2009.

[5] C. Cook, J. Schaafsma, and M. Antheunis, “Under the bridge: An in-depth examination of online trolling in the gaming context,” *New Media & Society*, vol. 20, no. 9, pp. 3323–3340, 2018.

[6] J. Daniels, “Race, civil rights, and hate speech in the digital era,” 2008.

[7] J. Deb, T. Sugaya, and A. Wolitzky, “The folk theorem in repeated games with anonymous random matching,” *Econometrica*, vol. 88, no. 3, pp. 917–964, 2020.

[8] E. Diener, S. C. Fraser, A. L. Beaman, and R. T. Kelem, “Effects of deindividuation variables on stealing among halloween trick-or-treaters.,” *Journal of personality and social psychology*, vol. 33, no. 2, p. 178, 1976.

[9] S. Eckert and J. Metzger-Riftkin, “Doxxing, Privacy and Gendered Harassment. The Shock and Normalization of Veillance Cultures,” *M&K Medien & Kommunikationswissenschaft*, vol. 68, no. 3, pp. 273–287, 2020.

[10] L. Festinger, A. Pepitone, and T. Newcomb, “Some consequences of de-individuation in a group,” *The Journal of Abnormal and Social Psychology*, vol. 47, no. 2, Suppl, pp. 382–389, 1952.

[11] E. J. Friedman and P. Resnick, “The social cost of cheap pseudonyms,” *Journal of Economics & Management Strategy*, vol. 10, no. 2, pp. 173–199, 2001.

[12] T. Graepel and R. Herbrich, “Ranking and Matchmaking,” *Game Developer Magazine*, vol. 25, p. 34, 2006.

[13] R. John, J. P. Cherian, and J. J. Kizhakkethottam, “A survey of techniques to prevent sybil attacks,” in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, IEEE, 2015, pp. 1–6.

[14] M. Kabay, “Anonymity and pseudonymity in cyberspace: Deindividuation, incivility and lawlessness versus freedom and privacy,” in *Paper presented at the Annual Conference of the European Institute for Computer Anti-virus Research (EICAR)*, vol. 16, 1998, p. 8.

[15] J. Kang, R. Yu, X. Huang, and Y. Zhang, “Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2017.

[16] P. Kekäläinen, “Evaluating the impact of anonymous tables in the world of online poker,” 2016.

[17] R. Kowert, “Dark Participation in Games,” *Frontiers in Psychology*, vol. 11, 2020.

[18] G. Le Bon, *The crowd: A study of the popular mind.* Courier Corporation, 2002.

[19] M. Lininger, *Etiquette Scholar, Dining Etiquette History*, Last accessed 14 December 2021. [Online]. Available: https://www.etiquettescholar.com/dining_etiquette/potpourri/dining_etiquette_history.html.

[20] S. K. Mishra, "Median as a weighted arithmetic mean of all sample observations," *Available at SSRN 555021: https://ssrn.com/abstract=555021*, 2004.

[21] A. Nellist, "Swatting: Protecting the Individual," Ph.D. dissertation, Utica College, 2018.

[22] R. Pelánek, "Applications of the elo rating system in adaptive educational systems," *Computers & Education*, vol. 98, pp. 169–179, 2016.

[23] N. T. Pujante Jr, "Speech for Fun, Fury, and Freedom: Exploring Trash Talk in Gaming Stations," *Asian Journal of Language, Literature and Culture Studies*, pp. 1–11, 2021.

[24] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[25] O. Rotou, X. Qian, and M. von Davier, "Ranking systems used in gaming assessments and/or competitive games," *ETS Research Memorandum Series, ETS RM-15-03*, 2015.

[26] M. W. N. Sakti, A. Basuki, and A. R. Barakbah, "Implementation of elo rating system and player clustering for competitive matchmaking in trivia education game," *Jurnal Mantik*, vol. 5, no. 4, 2022.

[27] A. Scott, *A World Without HUDs*, Last accessed 07 February 2022. [Online]. Available: https://www.linkedin.com/pulse/world-without-huds-alex-scott.

[28] L. Silva, C. Senna, and A. Zúquete, "Using Reputation as a Coin to Bet on Information Items Distributed in a Smart City," in *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*, IEEE, 2019, pp. 1–2.

[29] R. van Summeren, "Security in online gaming," *Radbound University Nijmegen: Bachelor Thesis Information Science*, 2011.

[30] W. Y. Tang and J. Fox, "Men's harassment behavior in online video games: Personality traits and game factors," *Aggressive behavior*, vol. 42, no. 6, pp. 513–521, 2016.

[31] *What does chess ELO ratings mean?* Last accessed 14 December 2021, AskFoxes, 2018. [Online]. Available: http://www.askfoxes.com/58/what-does-chess-elo-ratings-mean.html.

[32] P. G. Zimbardo, "The human choice: Individuation, reason, and order versus deindividuation, impulse, and chaos.," in *Nebraska symposium on motivation*, University of Nebraska press, 1969.