Security Issues in Online Games

Jianxin Jeff Yan and Hyun-Jin Choi

Abstract-- **The traditional target of computer game security is mainly copy protection. The emergence of online games fundamentally changes the security requirements for computer games. Although computer game development often utilizes the cutting edge technology in computer graphics, artificial intelligence, human computer interaction and programming, game providers (developers or operators) do not pay much attention to security techniques. In this paper, we look into security failures that have happened or might happen in online games, and discuss some key security issues that online game providers have to concern. Specifically, we look into various online cheatings, and introduce security techniques to deal with cheating prevention, though meanwhile other security issues are also discussed.**

Keywords-- **online games, security, online cheating**

I. INTRODUCTION

With the emergence of the technology in recent years, the way of playing games has changed. Now PC or console games are not any more the shining star in the game world, and we are going towards the era of online games.

Online games are networked games. The network can be low or high bandwidth, wired or wireless, LAN, the Internet or other wide area networks. The game device may be a computer, a game console or a mobile phone. Different from the traditional PC (or console) games, where each player play one game on his own device and his opponent is a virtual player simulated by game software, or two players play simultaneously against each other or partner together to against the virtual player on the same game device, online games allow much more users, who may be in different places over the world, to play together over the network, and thus create an exciting virtual game community where players can have lots of fun.

Meanwhile, security issues with computer games are changing. For PC games, the main security target of computer games is copy protection, i.e., how to make it difficult to manufacture illegal copies. Online games, however, propose a fundamentally different requirement for security. Firstly, online games are born to be one of distributed E-commerce applications, which concern more complicated security issues. On the other hand, online games have their own unique security challenges. Some online game vendors even deliberately ignore copy protection, since they might be using a different business model: game client software is freely distributed, and a fee is charged only when a player logs on to play on their game server.

Author address: Computer Laboratory, University of Cambridge, William Gates Building, JJ Thomson Avenue, Cambridge CB3 0FD, UK. Email: {Jeff.Yan, Hyun-Jin.Choi}@cl.cam.ac.uk

Although game development often utilizes the cutting edge technology in computer graphics, artificial intelligence, human computer interaction and programming, our gaming experience and literature survey show that game providers (developers or operators) do not pay much attention to (new) security techniques. Unfortunately, ignorance of game security issues in the era of online games will easily bring game providers and players into trouble.

In this paper, we look into some security issues that online games should concern. They are either specific to online games, or may be generic to distributed E-commerce applications but have special interests in the context of online games. Specifically, we discuss security failures that have happened or might happen in online games, and introduce relevant security techniques to address those security threats. For those issues that pure technical mechanisms cannot provide good solutions, we introduce social or procedural means to tackle them.

II. CHEATING IN ONLINE GAMES: FACTS AND PREVENTION

Online cheating is an important security issue that distinguishes online games from other E-commerce applications, though some cheats in online games may find similar exploits in other E-commerce applications.

Playing with other gamers over the Internet can be real fun at the first time, because each time it may bring each player a totally new gaming experience, interesting and exciting. Unfortunately, many players, due to the widespread use of various cheats, soon realize that it may also be unsatisfying to play online. In a 1997 online survey, 35% of 594 players that responded in a two-week period admitted to cheating, and 55% had seen cheating while playing online, only 10% had neither ever cheated nor seen a cheater. Though the cheaters were in the minority, 35% was believed to be a significant enough number to constantly deteriorate game playing for the honest players (Greenhill, 1997). As some players say, online cheats ruin good games, and result in (new) users giving up.

One of the main security targets for online games will be to deal with various online cheating. In this section, we define what constitutes online cheating, categorise various cheatings that have happened or might happen, and discuss their prevention.

A. What's A Cheat?

Though online cheating is popular, there is not a generally accepted definition on what a cheat is. Different companies use different criteria to determine which behaviour is cheating. For example, some allow players to use macros to replay a series of keystrokes, or mouse moves and clicks, while others regard the use of macros as cheating.

Sometimes it is very difficult to distinguish smart play, e.g. good use of tactics, from cheating. A good example is a so-called "camping" behaviour in some online war games where players can act as snipers. A camper is a player who would be sitting with a gun in a place (e.g. a corner) where other players must pass by. He acts like setting up a camp, and never moves from his place, while what he does is only to wait for people to come by and shoot them. It is hard or impossible for other players

to kill the camper. Since camping usually ruins the fun of other players, it is an annoying behaviour in those games, and some people regard it as cheating. Others argue that this is not cheating but normal sniping, since the game is a simulation of real world, and if it was real combat you couldn't complain that the enemy played too cunningly.

The lack of a common definition for online cheating even caused conflicts between game providers and players. It is necessary for game providers to define clearly what constitutes a cheat, and communicate promptly and clearly with players about their definitions.

In this paper, we define online cheating as follows. Any behaviour that a player may use to get an unfair advantage, or achieve a target that he is not supposed to is cheating. In the case of "camping", if a player camps in a place where he will never be killed by hiding there due to a design flaw or programming bug, according to our definition, he is cheating. Otherwise, it is not necessarily cheating.

B. A Taxonomy of Online Cheating

A recent paper summarized six different categories of online cheating as follows (Pritchard, 2001).

1. *Reflex Augmentation*: exploiting a computer program to replace human reaction to produce superior results
2. *Authoritative Clients*: exploiting compromised clients to send modified commands to the other honest clients who blindly accept them
3. *Information Exposure*: exploiting access or visibility to hidden information by compromising client software
4. *Compromised Servers*: modifying server configurations to get unfair advantages
5. *Bugs and Design Loopholes*: exploiting bugs or design flaws in game software
6. *Environmental Weaknesses*: exploiting particular hardware or operating conditions

Unfortunately, this categorization was ad hoc, and it only covered a small number of types of cheats. Many cheats cannot fit into any of these categories. In this section, we categorise online cheatings that have happened or might happen. Though our taxonomy might be incomplete either, we try to cover all online cheatings known to us. Moreover, our taxonomy categorises online cheatings in a structured way so that it helps security specialists understand the threats, and look for countermeasures.

1) Cheating by Collusion

It is common that players collude to cheat. Take online Bridge game as an example, one infamous collusion cheating is like this: one cheater uses two computers, through each of which he logs on to a Bridge server with a distinct ID, then he partners his two IDs on a same Bridge table, just like two distinct players partnering each other. The cheater will play by himself both hands of cards, which are supposed to be held by two different players and blind to each other! Therefore, he can always make a precise contract; or when he turns to be a defender, he knows exactly every card the declarer has after the dummy – the partner of the declarer – places his cards face-up on the table as required by the rule of Bridge. Two cheaters,

who do not stay in a same place and cannot see each other, can also cheat by collusion, since they can communicate to exchange card information via email, online chatting or instant messenger software, or telephone etc.

2) Cheating by Abusing Procedure or Policy

Some players make use of game procedure or policy to cheat. One example is *scoring cheat*, which might happen when two players are scoring a Go (WeiQi or Baduk) game. Because artificial intelligence research is not mature enough to identify dead stones and then decide who wins at the end of each game, online Go players must identify and remove the dead stones by themselves before their software can count the result. Some cheaters may stealthily remove live stones in the scoring process, and "overturn" the result.

A common example of abusing policy is *escaping*. In some online games, each game one player plays will affect his rank. When a cheater is going to lose his game, he will disconnect himself from the system so that his game is unfinished and thus non-scorable. The type of cheater is commonly called an "escaper". Different online games may use different schemes to deal with this cheating. For example, some online Go games implemented a penalty policy: the player who disconnects his game will lose that game if he does not finish it in a limited say 20 days. *StarCraft* used a different method: for each player, the system can show the number of games he has dropped so that each player may check that information and determine whether to play a game with a specific opponent.

Cheaters may also exploit the above policy of escaping prevention to cheat in online Go games. This cheating may be called *scapegoating* (or *hit-then-hide*) and works as follows. One cheater uses some attacks[1] to disconnect his opponent so that the game is recorded as disconnected by the opponent. Afterwards he does not log on until the opponent automatically loses that game, because nobody will show up to finish that game in the limited period.

3) Cheating Related with Virtual Assets

Trading of virtual characters and items (e.g. clothing, weapons, homes and magical objects) acquired in games is a new and real business created by online games. Many players would like to have good characters, or improve the status of their own characters by getting some items in the game. Nonetheless, it is not easy for every player to get good characters and items, which require gaming skills and time. Where there is demand, there is supply, and then there is a market! Now virtual characters and items become virtual assets, or real assets in a virtual world, and many of them have been auctioned for real money on eBay.

Where there is money flow, there is cheating. Trade cheating also has happened to virtual characters and items, since no security mechanism was implemented to protect them. For example, players *Alice* and *Bob* would like to exchange virtual items owned by each other. After receiving the item from *Alice*, *Bob* does not give his item to *Alice*. There is no way for *Alice* to get her own item back or claim her right over the item that *Bob* has promised to pass to her.

[1] Details please refer to "*5) Cheating by Service Denial of Peer Player*".

It was reported recently that a person earned around US$11,000 through this type of trade cheating: he kept the money but never sent any item to the payers. Another case reported that a person lost his item bought at around US$220 and called the police, and the investigation showed that the theft was the person who sold the item - he hacked the game site, and stole the item back after selling it (Hankyoreh, 2001).

Digital signature can provide non-repudiation, which helps each player to claim his possession of valuable characters and items. Fair-trading can be implemented to prevent trade cheating.

4) Cheating by Compromising Passwords

Historically, user passwords, as an easy target with high payoff potential, have been among the first targets on which an attacker would focus attention when he attacks a system. Passwords used for online games are not an exception. A password used by a player, as usual, is often the key to all the data and authorization that this player has in the game system. An attacker can exploit a compromised password to do various cheatings. Therefore, good password management and practice is also essential to prevent online game cheating.

One of the main threats upon passwords comes from offline or online dictionary attack. Due to the limitation of human memory, people like choosing easy-to-remember passwords such as phone numbers, birthdays, or names of friends or family, or words in human languages. A dictionary attack tries each of a list of word and other possible weak passwords, and simple transformations such as capitalizing, prefixing, suffixing or reversing a word, as a candidate until the hashed value of the candidate matches a password hash. It has been very often successful in attacking many easy-to-remember passwords. Anybody who gets a password file can launch an offline dictionary attack against all passwords in the file, while anybody who knows a player ID can do an online dictionary attack by simulating his logon procedure.

In the case an online game system is designed to be resilient to online dictionary attack, e.g., by limiting the number of logging that a player is allowed to try, a malicious attacker may easily block any user, whose ID is known, by continuing logging with wrong passwords. Very few designers are willing to tolerate this consequence, and the convenience of players is a more important issue to them, consequently, their systems will be vulnerable to online dictionary attack on passwords.

Good password management and practice are needed to protect players from online or offline password attacks.

5) Cheating by Denying Service from Peer Players

As discussed above, in an online game system that is designed to be resilient to online dictionary attack, a malicious player may easily block peer players whose IDs are known to him so that they cannot have access to the game. This is one type of denial of service (DoS) attack against peer players. In some games, a cheater might be able to get unfair advantages by blocking more experienced players in this way.

Another common cheating of this category is that a cheater floods an opponent's network connection by launching network DoS attacks, and gets unfair advantages.

For example, when playing timing-critical games like Chess or Go, each side has a same time limit, and who runs out of the time will lose the game. A cheater can exploit a DoS attack to slow down the network connection of his opponent to run out of his time. For instance, many Go players choose to play 25 stones in 10 minutes, and it is not unusual for them to play 5 stones in the last 10 seconds. During the 10 seconds, the *running-out-of-timing* (or *timing out*) cheat is always deadly. Most victims lose their games, and they are not sure whether they are attacked or the network connection is jammed due to other "normal" reasons.

Another example may happen in real time strategy games. A player ping–floods his opponent to heavily delay responses from the opponent. Other players will be cheated to believe that there is something wrong with the connection of the victim, and kick him out from the game.

Some systems hide IP addresses of players to avoid network DoS attacks.

6) Cheating due to Lack of Secrecy

Since all packets exchanged among players and servers, or among peer players are in plaintext, a player can easily cheat by eavesdropping packets and inserting, deleting or modifying game events or commands transmitted over the network. Password eavesdropping is one example of many this type of cheatings. Encryption is commonly needed to address this type of cheats.

7) Cheating due to Lack of Authentication

Password authentication only provides assurance that one is a registered or legitimate game player. In many cases, two-way authentication between game servers and clients are also needed to authenticate that both servers and clients are genuine. For online game systems that do not implement this two-way authentication, it is easy for a cheater to collect many ID-password pairs of legitimate players by setting up a bogus game server.

It is also important to re-authenticate a player before any password changing operation. Otherwise, when a player temporarily leave his computer with his game session unclosed - that is not unpopular in many internet cafe where online gamers are active - a cheater who can physically access to the player's machine may stealthily change his password, and exploit the new password later. It is also probable for a technically capable cheater to replay the game session and force a password change.

8) Cheating Related with Internal Misuse

Online games also suffer from internal misuses. Since game operators (sometimes they are game developers as well) have almighty power of system administrators, they can do many things that cheaters dream to do. For example, they can summon whatever character or weapon they like, and they can do offline dictionary attacks and then exploit any compromised password. An insider was recently fired in Korea because he abused his privilege to generate super-characters by modifying the game database (Chosun Ilbo, 2001). Many similar cases were also reported.

In a place where trading virtual characters or items for real money is profitable, there are enough economic incentives to tempt the insiders to misuse their privilege.

Although it may be hard to prevent inside misuse from happening in the first place, logging privileged operations into CD-ROMs provides a good solution to catch the insider cheaters. Sometimes, data mining techniques may be needed to dig out the specific misuse evidence.

9) Cheating by Social Engineering

Social engineering is often used to steal passwords. There are many variations of this scam but all of them aim the same: to trick players to happily reveal their ID-password pairs. Often these social engineers – password scammers – will attempt to trick a player into believing something attractive or annoying has happened to the player and his ID and password are needed for that purpose. They may approach a victim by phone, email, online chatting channels, or whatever they may exploit.

The solution to deal with social engineering is simple: to educate players to simply ignore any such password requests, since password is designed only for them to log on a specified system, and should not be revealed at any other time for any reason (For online games that do not appropriately implement the two-way authentication, any password request is suspicious).

10) Cheating by Modifying Game Software or Data

This has been a traditional cheating since the beginning of the PC game era, and many tools available to enable cheaters to modify either program file or memory. The difference is that cheaters were cheating the computer before, while now they are cheating human players sitting at the other ends of the network. Both *information exposure* and *compromised servers* cheats defined by Pritchard are of this category.

Cheaters may use debuggers to reverse engineer game programs and customize them to get various unfair advantages for different purposes. For example, they may remove validating routines, modify configuration parameters, or rewrite some parts of game software to optimise the weapon loading time. Program obfuscation, anti-debugging code, and integrity checking are the traditional methods to deal with this cheat.

Memory scanning tools such as Game Buster were also developed to help cheaters look for critical variables in the memory. Cheaters do not need to modify game files, but modify those variables on the fly when game software is running. As discussed by Pritchard, program obfuscation can make life harder for cheaters, but cannot prevent this memory data modification. Encryption can be used to encrypt critical values in the memory all the time so that a cheater cannot locate the variables that he is looking for. For some games based on the client-server model, critical variables may be kept on the server when this does not have severe performance impact on the game (Pritchard, 2001).

Security protocols can be designed to validate client software and critical data in an encrypted way. The protocols can be run when a client initialises a connection, and run periodically when the game is running. When the validation process fails, the server can take appropriate actions, e.g. disconnecting the client.

11) Cheating by Exploiting Bug or Design Flaw

Some online cheats exploit bugs or design flaws found in game software to get an unfair advantage. This category is the same as defined in (Pritchard, 2001) and the practical solution is to patch the bugs, though some may argue that good software engineering will provide a reasonable solution.

C. Cheating Mitigation: Prevention, Detection and Management

Traditional security mechanisms such as encryption, authentication, integrity checking, digital signature and cryptographic protocol all can find plenty of applications in online games. Nonetheless, they do not solve all online cheating problems. There is no silver bullet. A systematic approach is needed to mitigate online cheating. Some means are required for preventing cheatings from happening in the first place, and others needed for detecting cheatings after they happen. Pure technical mechanisms cannot provide a complete solution; management and policy means are also needed. In this section, we highlight some important means that we do not discuss in details in section *B*.

1. Built-in cheating detection

While an intrusion detection system can be used to detect hackers that break in a system, which hosts one online game or more, a cheating detection engine can be designed and implemented as one built-in component of each game software.

Some game providers proposed to use experienced game developers to police their online games by randomly monitoring player behaviours. No matter whether or not this method is as effective as expected, it is however very expensive. A carefully designed built-in cheating detection engine will provide a cheap alternative, since it can automatically detect and prevent many cheating behaviours by monitoring critical game events and variables. This engine can be shared by different games, though triggering events may be specific to each game. In case game providers cannot guarantee good security for game client software, the built-in detection should be implemented in a game server, which is typically installed in a protected environment where it is difficult for cheaters to tamper the software.

Take a popular cheat of item duplication as an example, the number of items generated by the system should always be equal to the number of items that are possessed and consumed by all players. Therefore, when a gaming behaviour violates this principle, a triggering event will be thrown to the built-in detection engine, which will take appropriate actions, e.g., void the game change that the cheating behaviour would like to achieve, and record cheater's ID.

Another example is in *Ultima Online*, where shopkeepers buy in from players. The built-in detection may define the buy-in frequency and buy-in amount per time, and then check whether a buy-in behaviour violates these predefined values before each buy-in is allowed.

For an online game where there is a ranking scheme, rank tracking can be implemented as an effective part of the built-in detection to alert some cheatings, e.g., collusion cheating in online Bridge, that are difficult to be caught by other technical means. It is highly likely for a policing Bridge veteran to identify the abnormal when he happens to watch some games played by colluding cheaters. Nonetheless, there are thousands of games played in a busy Bridge server every day,

and it takes much time to monitor even only a few of them. Instead, it is easy for rank tracking to alert that two IDs always partner together, making many – if not always – precise contracts or defence, or defeating top-ranked players surprisingly many times. Though alerts triggered by rank tracking may not be correct all the time, this approach narrows down a game operator's focus onto only potential cheaters, and it saves time and cost.

2. Make players be security-aware

Security is not a problem solvable by technical means alone. Human factor plays a very important role in achieving good security. It is unimaginable to have a good security for online games without the cooperation of game players. Game providers need to educate players about security, e.g., what potential security threats exist, and what to do when they face a potential security threat.

3. Good password practice and management

Though it is commonly believed that secure passwords are difficult to remember and easy-to-remember passwords are insecure, a recent experiment showed that passwords based on mnemonic phrases can provide both good memorability and security (Yan et al, 2000a). Therefore, game providers may instruct players to choose this type of passwords. On the other hand, non-compliance with good password selection advices is a main threat to password security (Yan et al, 2000a). A proactive password checker may be integrated with online games to enforce good password selection policies and prohibit easily guessable passwords. The proactive checking is done online and a player will be immediately responded whether his password choice is acceptable or not.

Although dictionary-based proactive checkers will fail to filter some weak passwords with low entropy such as `12a34b5`, the entropy-based proactive checking was a good remedy (Yan, 2001).

Briefly, some thumb rules for password security are:
- *Never transmit passwords in plaintext.*
- *Re-authenticate before changing passwords.*
- *Use memorable and secure passwords.*
- *Prohibit easily guessable passwords by integrating a proactive checker with the password mechanism.*
- *Never respond to any password request unless logging on to a trusted machine.*

4. Fair trading

This fair-trading of virtual assets can be achieved by introducing a trusted third party (TTP). Players may negotiate deals by themselves, and then pass their items to the TTP, and the TTP will help the players complete the exchange in a trusted way. The TTP approach, however, is expensive, since it requires much human interference. Fair protocols initiated by Ben-Or et al. (1990) can provide an automatic solution, though where a TTP may or may not be needed.

5. The bug patching approach

No developer can fix all bugs before software release. The traditional bug patching approach in security still works here.

6. An active complain-response channel

A complain channel should be maintained, so that players can report new bugs, potential cheatings or cheaters. Game providers should provide prompt responses to complaints from players. Otherwise, the enthusiasms of players will be hurt.

On the other hand, game providers can also use this complain-response channel to make players security-aware. For example, providers may disseminate their own definition of cheating behaviours, or distribute security alerts and possible cheatings through this channel.

7. Logging and audit trail

Logging and audit trail provide not only good protection against insider cheating, but also a unique solution for dealing with some cheats. *Scoring cheat* is a good example. Good Go players always evaluate their own situation, and even estimate the final result while their games are in progress. Some of them can quickly recognise a *scoring cheat* happened to them. Unfortunately, they cannot change their results by themselves, and what they can do is to complain to the game operator. Logging each game as a session record can help guarantee fairness for honest players.

Different from logs used for insider cheating prevention, it is not necessary to permanently store session logs in CD-ROMs, and instead, they may be stored in hard disks and removed after a short time.

8. Post-detection mechanisms

Appropriate post-detection mechanisms are needed when cheatings are detected and cheaters identified. Cheaters should be punished by disciplinary means, and victim damage unfairly caused by cheating should be restored. A checkpoint mechanism can be used for this recovery.

III. AVAILABILITY ISSUE IN GAME HOSTING

Many online games heavily rely on a farm of servers to host game services. If the servers are flooded, players will not be able to play the online games at all. So the service availability of game servers is also a critical issue for online games. Network denial of service, especially distributed denial of service (DDoS) attack is a severe threat for game hosting.

A DDoS attack exploits a number of subverted machines as attack bots to launch a large coordinated packet flood at a target. Since many bots flood the victim at the same time, the traffic is huge and more than the target can cope with, and because it comes from many different sources, it can be very difficult to stop. Unfortunately, most suggested countermeasures would not work, since they ignored the fact that the DDoS threat is at heart a manifestation of what economists call the "tragedy of the commons": while everyone may have an interest in protecting a shared resource (Internet security), individuals have a stronger motive to cheat (connecting insecure computers).

XenoService can effectively defend DDoS attacks (Yan, 2000b), and it is also applicable to game hosting. Technically, a DDoS attack might jam either network connection or local operating system of a victim, or both. The XenoService uses XenoServers, which provide quality of service guaranteed resource management to prevent themselves being jammed by DoS attacks, and was developed at Cambridge University for distributed hosting of latency- and bandwidth-critical network services. In the case of game hosting, the XenoService is a distributed network of XenoServers that host game services, and respond to a DDoS attack on any one service by replicating it rapidly and widely. In this way, a game service that comes under an attack can within a few seconds acquire more network connectivity than Microsoft, so that it can absorb a packet flood and continue serving.

On the other hand, it also helps to mitigate network DoS attacks if server-end game software is designed to drop non-game packets by distinguishing them from game packets.

IV. PRACTICAL SECURITY ENGINEERING ISSUES

Though the bug patching approach seems to be a mainstream security method and work somehow, it is a choice out of no choice, and it is a direct by-product out of ignorance of security in the system design stage, which leads system security to an endless loop of security bug exploiting and then patching. In most cases, security is not a feature that can be patched later on, and it should be considered at the very first stage of the system design. Some game designers started to understand this principle from suffering (Greenhill, 1997), and others, unfortunately, do not yet.

Although "security" may mean th ings greatly different from one system to another, a common security engineering approach (Anderson, 2001), which was summarised from various applications, is helpful for game developers, operators or both to achieve good online game security. Understanding of potential security threats in a system is the first step of the security engineering. Game providers should know what types of threats they might have, and what level of skills, tools and determination the attacker might have. Then, the following steps are to define appropriate security policies based on the threat models, and then design specific protection mechanisms to implement these policies. Many security failures occur because either the wrong things are protected, or the right things protected in a wrong way.

Anderson observed that security engineering would most likely benefit from an open source approach (Anderson, 1999). Nonetheless, for those game developers who do not want to adapt this approach, we recommend an alternative method: game security can be regarded as a type of risk management, and high-risk threats could be taken cared with high priority.

Moreover, security, system performance and player convenience sometimes may lead to conflicting requirements. A good balance of these three aspects is also an important issue that should be handled properly.

V. CONCLUSION

The emergence of online games fundamentally changed the security requirement for computer games. In the new context, copy protection is not, at least not the only, security issue any more. Though online games, on the other hand, are commonly regarded as one of distributed E-Commerce applications, they have their own unique security challenges. In this paper, we discussed various security issues in online games. Especially, we tried to categorise various online cheats that have happened or might happen, and looked into some cheating details. We also proposed a systematic framework for generic cheating prevention and management. Due to the space limitation, we omitted many other security issues that online games are highly relevant, such as liability, anonymity and privacy protection.

Online game security is one of those areas, where domain (game) specialists are not security experts, and security specialists are not familiar with complicated domain knowledge that may appear to be easy though. Little serious security research has been done, though many interesting security challenges are there. This is the first step of our research in online game security. It is incomplete, but calls for a better cooperation between two fields. The comprehensive details will be worked out to satisfy the requirements of online game industry.

VI. ACKNOWLEDGEMENT

Ross Anderson inspired the first author to look into online game security, and provided valuable comments. The authors are also grateful to George Danezis for his proofreading and useful comments.

VII. REFERENCES

Anderson, R. (1999), "How to Cheat at the Lottery", Proc. Of Annual Computer Security Applications Conference, 1999.

Anderson, R. (2001), Security Engineering, Wiley, New York.

Ben-Or, M., Goldreich, O., Micali, S. and Rivest, R. (1990), "*A fair protocol for contract signing"*, IEEE Trans. on Information Theory 36, pp. 40-46.

Chosun Ilbo (2001), "Rayegard system developer is punished", news article published on June 27, 2001. Available http://www.chosun.com/w21data/html/news/200106/200106270431.html.

Greenhill, R. (1997), "*Diablo, and Online Multiplayer Game's Future",* GamesDomain Review. Available http://www.gamesdomain.com/gdreview/depart/jun97/diablo.html.

Hankyoreh (2001), "Online cheating is ubiquitous", news article published on May 9, 2001. Available http://www.hani.co.kr/section-005100025/2001/05/005100025200105091907004.html.

Pritchard, M. (2001), "How to Hurt the Hackers: The Scoop on Internet Cheating and How You Can Combat It", Information Security Bulletin, February.

Yan, J.J., Blackwell, A., Anderson, R. and Grant, A. (2000a), "The Memorability and Security of Passwords -- Some Empirical Results", Technical Report No. 500, Computer Laboratory, University of Cambridge. Available http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/tr500.pdf

Yan, J.J., Early S. and Anderson R. (2000b), "The XenoService - A Distributed Defeat for Distributed Denial of Service", Proc. of Information Survivability Workshop 2000, IEEE computer society, Boston, USA. Also available at http://www.cl.cam.ac.uk/ftp/users/rja14/xeno.pdf

Yan, J.J. (2001), "A Note on Proactive Password Checking", ACM New Security Paradigms Workshop, New Mexico, USA, September.