

# Combating Shill Bidding in Real Time: Prevention, Detection and Response

Kazi Mamun<sup>1</sup>

<sup>1</sup>Software Systems Engineering, University of Regina, Regina, Canada

Correspondence: Kazi Mamun, Software Systems Engineering, University of Regina, Regina, SK., S4S 0A2, Canada. Tel: 1-306-581-6975. E-mail: mamun20k@uregina.ca

Received: February 26, 2015

Accepted: March 29, 2014

Online Published: April 26, 2015

doi:10.5539/cis.v8n2p24

URL: <http://dx.doi.org/10.5539/cis.v8n2p24>

## Abstract

Human cheating has been a barrier to establishing trust among e-commerce users, throughout the last two decades. In particular, in online auctions, since all the transactions occur among anonymous users, trust is difficult to establish and maintain. Shill bidding happens when bidders bid exclusively to inflate (in forward auctions) or deflate (in reverse auctions) prices in online auctions. At present, shill bidding is the most severe and persistent form of cheating in online auctions, but still there are only a few or no established techniques for shill defense at run-time. In this paper, I evaluate the strengths and weaknesses of existing approaches to combating shill bidding. I also propose the ShillFree1 auction system to secure and protect auction systems from shill bidders for both forward and reverse auctions. More precisely, by using a variety of bidding behavior and user history, proposed auction system prevents, monitors and detects shill activities in real time. Moreover, to detect shilling thoroughly I propose IP tracking techniques. The system also takes necessary actions against shill activities at run-time. The experimental results demonstrate that, by prevention, detection and response mechanisms, the proposed auction system keeps the auction users secured from shill bidding and therefore establishes trust among online auction users.

**Keywords:** online auction, shill bidding, trust management, run-time monitoring, e-commerce

## 1. Introduction

Among all online crimes, auction frauds are concurrently one of the most reported, about 35.7% in 2007 (IC3, 2007), and the top five in 2011 (IC3, 2011). The Internet Crime and Complaint Center (IC3) received over 200,000 complaints of auction related frauds in 2007, and more than 40,000 in 2011 (SecurePuter, 2008; IC3 2011). IC3 classifies auction frauds into six categories: misrepresentation of products, non-delivery of products, triangulation, fee staking, selling of black-market products, multiple bidding, and shill bidding (Jenamani et al. 2007). Shill bidding refers to artificial price inflating in case of forward auctions (Trevathan & Read, 2005) and price deflating in reverse auctions in order to generate an interest for the auctioned item. A case study on shill bidding demonstrates that in 2008 auction users have lost about 250 million dollars because of shilling (Cohen P., 2009). Shill bidding is the hardest to detect among all online auction frauds.

Researchers have been working on combating shill bidding in forward auctions (Trevathan & Read, Xu et al., Patel et al., Dong et al. and Wang et al.). However, at present all existing auction houses, and most approaches proposed by researchers have no functionality that detects shill bidding in live auctions and do not take any action until a report is made by an auction user. Since the damage occurs during the auction, it is preferable to detect and stop shilling in real time, rather than detect it afterwards. If shilling is not detected in real time, by the end of the auction, the winning seller/buyer has already been cheated.

In this paper, I first evaluate existing solutions for shill detection in online auctions, then identify common patterns and approaches of shill bidders and illustrate them through real auction examples. Subsequently, I present the ShillFree1 auction system, which can protect users from shill bidders in both forward and reverse auctions. The ShillFree1 auction system is secure, trustworthy and easy to modify as new patterns of shilling are detected. To manage the users, the ShillFree1 auction system generates and maintains user profiles based on their used period and behavior in previous auctions. It also controls users' behavior during auctions through user limits and authorization of different user requests. The ShillFree1 auction system monitors the bidding process during auctions, detects shilling attempts, and responds in real time while the auction is still running. To ensure

skill free auctioning, my auction system tracks and examines the behaviors and IPs of the bidders at run time. To the best of my knowledge, no existing and only very few proposed auction system detect shilling based on the behavior and signature of users. Next, I report on an experiment involving 10 concurrent auctions where the participating users have predefined roles. This experiment demonstrates that the SkillFree1 system is able to detect all four skill bidders who are shilling in seven auctions. I conclude this paper with a discussion of possible future work.

## 2. Related Works

Considering the increasing number of problems with skill bidding in online auctions, some researchers have proposed methods of combating it. Travathan and Read (2005) present an algorithm to determine the degree of shilling being performed by a designated user. The algorithm observes shilling patterns of a user over a series of auctions and provides a skill score for the user. However, the algorithm works only for a single suspected user rather than a whole auction. Travathan and Read (2007) give an improved algorithm that detects collusive skill bidding where multiple skill bidders form a group and outbid each other's bids for shilling. The algorithm provides a collusion score based on bid sequences of alternating bid, alternating auction and hybrid strategies. However, the algorithm considers only a limited number of bid sequences where only two skill bidders are colluding, and when multiple auctions are taken in consideration. In case of different auction intervals the algorithm will not work in real time.

Patel et al. (2007) introduce a skill monitoring system for agent-based online auctions. They propose using role-based access control mechanisms to control skill bidders. They design a model with a security agent that monitors auction transactions for skill bidding. The proposed auction system considers forward auctions, where they consider user feedback for role assignment while user feedback can be falsified. The skill monitoring system monitors users instead of auctions. The authors consider a few shilling patterns, whereas there are other patterns which are also used.

Xu and Cheng (2007) propose an approach to detect skill bidding by verifying bidding behaviors in concurrent auctions. The authors gather bidding data from two concurrent auctions compare and develop a toolkit to detect skill suspects. The system is applicable only for two concurrent auctions with similar items. Moreover the approach is not applicable in real time. Xu et al. (2008) present a framework for Agent-based Trust Management (ATM) in online auctions. The ATM framework consists of agents for monitoring, analysis and security. A key idea is that, different agents are used to handle trust issues w.r.t. users' role assignment. The author uses a few bidding patterns and 'reputation scores' or 'feedbacks' of other users to determine skill bidding, which can be easily falsified. Moreover, some severe techniques like duplicate identity shilling and group shilling are not addressed in this paper. Xu et al. (2009) analyze some shilling patterns and propose a Dynamic Auction Model (DAM) for skill detection in real time. DAM requires an estimated sale price and a reserve price. The system is not useful for common products in online auctions such as used items, art effects and antiques, because it is difficult to estimate sale prices. Moreover, in the presence of auction snipers (Travathan et al., 2011) and other bidding agents (Jay & Jay, 2006), skill bidders no longer need to bid a high price close to the reserve price to do skill bidding, instead they can outbid general users with a small amount instantly until a certain desired price is reached.

Dong et al. (2009) propose using knowledge-based Dempster-Shafer (DS) theory to secure online auctions from shilling. They introduce a two-step model, where in the first step they propose to use the previously introduced skill detection technique of concurrent auctions (Xu and Cheng, 2007). In the second step they propose to use DS theory to verify the detected results of shilling in the first step for more accuracy. Their proposed DS theory based model checker performs skill verification only when the auction ends, which is not in real time when the shilling is happening. Although the authors claim that the technique can detect shilling in real time, their proposed model checker performs a check on the bidding behaviors only when the auction ends (Dong et al., 2009). If this analysis is correct, the model is not able to perform skill verification in real time. Dong et al. (2012) study the relationship between final prices of online auctions and skill activities and propose a system to identify skill bidding based on the difference between the final auction price and the expected auction price. Their approach is restricted to auctions where the expected price can be estimated and once again this system is only applicable when an auction ends.

The approach I present in this paper is an extension of our previous work for combating skill bidding in online auction systems (Mamun & Sadaoui, 2013). Our previous work consists only of the software architectural design, no implementation and evaluation have been carried out. In this paper, I improved agents and algorithms behind the SkillFree1 auction system which features skill prevention, detection and response. I implemented the entire

system with Jadex, which is an agent-based simulation platform. I also evaluate the feasibility of my approach on a large data set of users and live auctions.

Since shill bidding is a common and well-known problem for online auctions, some popular auction houses warn their users about the possibility of shilling on their websites. However, to the best of my knowledge, no auction system takes any action against shill bidding in real time. Table 1 shows the detection techniques used by some popular auction sites and the disclosed actions taken against shill bidding.

Table 1. Shill Detection Techniques by Popular Auctions

<b>Auction Site Name</b>	<b>Shill Bidding Detection Technique</b>	<b>Response for offline Auctions</b>
eBay	If reported by other users	Limit buying and selling privileges and suspending user account
uBid	Unknown	Unknown
eBid	If suspected by eBid	Hold placed on user account
Bidz	If evidence found by Bidz	Unknown

### 3. Shill Bidding

Shill bidding takes place when a seller in a forward auction wants to sell his item for more, or a buyer in a reverse auction wants to buy the item for less than the usual price (Mamun & Sadaoui, 2013).

#### 3.1 Common Patterns of Shill Bidding

Shill bidders use different approaches to achieve their goal of shilling. By examining real auction data, we can find some common patterns of their approaches of shilling. They include:

**P<sub>1</sub>**: to increase or decrease the price, a shill bidder continually bids to outbid his own bid even when he is the top bidder in the auction (Mamun & Sadaoui, 2013).

**P<sub>2</sub>**: a shill bidder bids within a short interval of time to outbid his own bid or others' to give more time to the other potential bidders (Mamun & Sadaoui, 2013).

**P<sub>3</sub>**: a shill bidder makes an unnecessarily large price change to increase or decrease the price rapidly (Mamun & Sadaoui, 2013).

**P<sub>4</sub>**: a shill bidder bids more in the beginning of the auction to make sure that other bidders get more time to bid.

**P<sub>5</sub>**: a shill bidder bids more times on average than other bidders.

**P<sub>6</sub>**: a shill bidder asks another bidder to bid on the same item (Wang et al., 2007).

**P<sub>7</sub>**: a shill bidder establishes a bidding ring composed of multiple sellers or buyers bidding on the buyer's or seller's item, with or without the direct involvement of the buyer or the seller (Wang et al., 2007).

**P<sub>8</sub>**: a group of shill bidders may form a bidding ring composed of multiple sellers or buyers bidding on each other's items (Wang et al., 2007).

**P<sub>9</sub>**: two or more shill bidders work together in the same auction to inflate or deflate the price, which is also known as collusive shill bidding (Trevathan & Read, 2007).

**P<sub>10</sub>**: a shill bidder bids exclusively only on one or few users' items.

**P<sub>11</sub>**: a shill bidder creates multiple identities and bids on his own item using a single computer with the same IP address (Mamun & Sadaoui, 2013).

**P<sub>12</sub>**: a shill bidder bids multiple times on the same item while changing both his identity and IP address of his computer.

#### 3.2 Concrete Examples of Shill Bidding

Here, I analyze some concrete examples of shilling from eBay, a popular online auction house where signs of shill bidding can often be seen (Cohen P. 2009, Spaced 2013).

Example 1: This forward auction shows shill bidding of pattern P<sub>1</sub>. In Table 2, we can see that the auction started with a bid of \$50 and after three days, there was only one more bid, which increased the price by just \$2. Then, user s\*\*\*e bids three consecutive times within a minute and increased the price by \$13. In the end, this user did not win the auctioned item, because user t\*\*\*o bought it. It is likely that user t\*\*\*o paid a higher price than necessary because of shill bidding. If we examine the bid history and profile of the user s\*\*\*e in Table 3, we can notice some other features about shilling. User s\*\*\*e has only seven feedbacks, which means he is a new user in eBay, and in the last 30 days he bid on only this item.

Table 2. Bidding History of an Auction Displaying Pattern  $P_1$ 

Bidder	Bid Amount	Bid Time
t***o	US \$67.00	Feb-29-12 03:25:57 PST
s***e	US \$65.00	Feb-29-12 02:49:11 PST
s***e	US \$60.00	Feb-29-12 02:48:47 PST
s***e	US \$55.00	Feb-29-12 02:48:20 PST
t***a	US \$52.00	Feb-29-12 02:16:15 PST
w***a	US \$50.00	Feb-26-12 22:42:15 PST

Table 3. Bidding Profile of a Suspected Shill Bidder Displaying Patterns  $P_6$  and  $P_{10}$ 

Bidder Information	Data
Bidder ID	s***e(7)
Bids on this item	3
Total bids	3
Items bid on	1
Bid (%) with this seller	100%
Item description	#M Mpro120 LED
Feedback	100%

**Example 2:** If we examine the bidding prices for a Sony Vaio laptop as shown in Table 4, we see strong evidence of shilling. The laptop had 126 bids, which represents too many bids compared to the other products in the same category. The starting price of the laptop was \$22.57. After few hours, user j\*\*\*e started bidding at \$24 and kept bidding, sometimes outbidding others and usually himself until the product reached the price of \$289 (Pattern  $P_1$  &  $P_2$ ). The history of user j\*\*\*e (cf. Table 5) indicates that he bid 248 times on 44 items, but 41% of his bids are particularly for this seller and more than 90% of his bids are for only four users' items (Pattern  $P_{10}$ ).

Table 4. Bidding History of the Auction Displaying Patterns  $P_1$  and  $P_2$ 

Bidder	Bid Amount	Bid Time
i***m	US \$305.00	Apr-12-13 07:30:53 PDT
h***a	US \$ 300.00	Apr-12-13 07:30:52 PDT
j***e	US \$289.00	Apr-11-13 01:41:05 PDT
j***e	US \$286.00	Apr-11-13 01:40:58 PDT
j***e	US \$282.00	Apr-11-13 01:40:52 PDT
j***e	US \$280.00	Apr-11-13 01:40:45 PDT
(Continuing...)	(Continuing...)	(Continuing...)
j***e	US \$27.00	Apr-07-13 20:12:50 PDT
j***e	US \$25.00	Apr-07-13 20:12:41 PDT
j***e	US \$24.00	Apr-07-13 20:12:33 PDT
n***c	US \$23.00	Apr-07-13 07:45:40 PDT
Starting Price	US \$22.57	Apr-07-13 07:30:55 PDT

Table 5. Bidding Profile of a Suspected Shill Bidder Displaying Pattern  $P_{10}$ 

Bidder Information	Data
Bidder ID	j***e(45)
Bids on this item	103
Total bids	248
Items bid on	44
Bid (%) with this seller	41%
Item description	Sony Vaio
Feedback	100%

**Example 3:** The auction shown in table 6 started with a price of only \$1 (cf. Table 6). Then user n\*\*\*a increased the price to \$201, even though when there were no other bidders (pattern  $P_3$ ). Then user i\*\*\*j inflated the price to \$3000 (pattern  $P_3$  and  $P_6$ ) and then to \$3900 (pattern  $P_1$ ). In the end, he stopped bidding and lost the auction for \$50. This behavior suggests that he intentionally bid just to inflate the price to benefit the seller of this

auction. Also we can see from Table 7, that user i\*\*\*j has the profile of a shill bidder. He has only 12 feedbacks and has only two bids in the last 30 days, both of which are for one seller’s item. Apparently, this user is only bidding on this particular auction to increase the price.

Table 6. Bidding History of the Auction Displaying Pattern P<sub>3</sub>

Bidder	Bid Amount	Bid Time
s***o	AU \$3950.00	17:29:53 AEST
i***j	AU \$3900.00	21:48:36 AEST
s***o	AU \$3550.00	14:44:54 AEST
i***j	AU \$3000.00	16:13:39 AEST
n***a	AU \$201.00	16:25:20 AEST
n***a	AU \$176.00	16:25:10 AEST
n***a	AU \$156.00	16:24:53 AEST
n***a	AU \$68.00	21:54:57AEST
e***m	AU \$1.00	22:39:57 AEST

Table 7. Bidding Profile of a Suspected Shill Bidder Displaying Pattern P<sub>6</sub>

Bidder Information	Data
Bidder ID	i***j(12)
Bids on this item	2
Total bids	2
Item bids on	2
Bid (%) with this seller	100%
Item description	c1880s
Feedback	100%

#### 4. The ShillFree1 Auction System

The ShillFree1 auction system was designed to meet the main goal of combating shill bidding. To fulfill the goal I designed a three-layer architecture of the ShillFree1 auction system, which is depicted in Figure 1. The GUI layer presents information of all auctions and users. For registration and sign in, every user has to be approved by the business layer. The data layer keeps the history of users and auctions. The business layer processes the users' actions performed in the GUI, monitors, detects and takes necessary actions against shill bidding at run time. The architecture is implemented as a multi-agent system, where each agent, based on a set of beliefs, desires and intensions (Rao & Georgeff, 1995), solves a particular problem. All agents cooperate in order to achieve the ultimate goal of protecting the running auctions from shilling. The system conducts both forward and reverse auctions according to the English protocol (Wang et al., 2001).

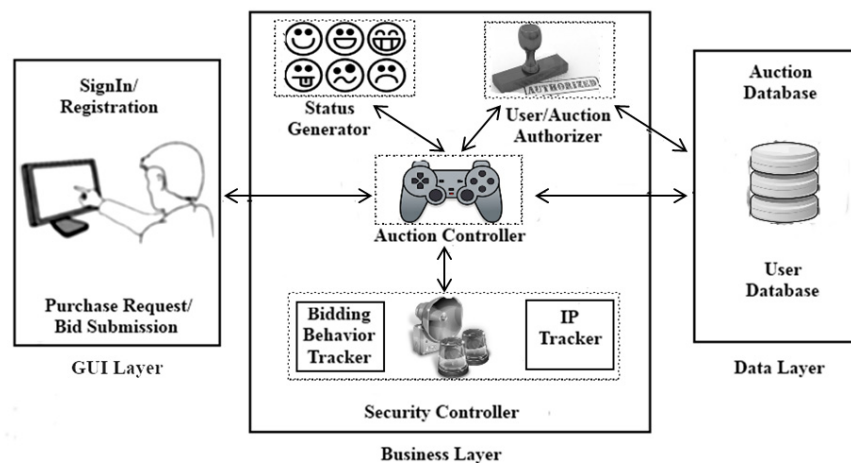


Figure 1. Software Architecture for Online Auctions

#### 4.1 Status Generator Agent

Researchers have proposed giving ratings and states to users based on their shilling activities (Travathan et al 2007, Patel et al. 2007). The Status Generator agent assigns a range of statuses to users based on three metrics: Used Period (UP), Used Manner (UM) and Shilling Attempts (ShilAtts). UP indicates the number of days since a user joined my auction system, UM the number of selling and buying activities of a user, and ShilAtts the number of shilling attempts made by a user in a certain period. In my system, a user is assigned one of the six statuses given in Table 8. To increase the safety of a live auction, the status of every participant is shown besides all his submitted bids. For instance, MostReliableUser is allocated to a user who is registered in the system for more than  $n$  days, and performed more than  $m$  actions, and has performed at most  $v$  number of skill attempts. UnReliableUser status is assigned to a user who is registered in the system for more than  $n$  days, or performed more than  $m$  actions, and has performed at least  $y$  and at most  $z$  skill attempts, where  $z$  is greater than  $y$ .

Table 8. User Status and Selling/Buying Limit

User Status	Equations	User Selling/Buying Limit (in dollars)
<i>NewUser</i>	$(UP < n \text{ or } UM < m) \text{ and } ShilAtts \leq v$	$l_4$
<i>MostReliableUser</i>	$(UP \geq n \text{ and } UM \geq m) \text{ and } ShilAtts \leq v$	$l_1$
<i>ReliableUser</i>	$(UP \geq n \text{ or } UM \geq m) \text{ and } v < ShilAtts \leq x$	$l_2$
<i>AverageReliableUser</i>	$(UP \geq n \text{ or } UM \geq m) \text{ and } x < ShilAtts \leq y$	$l_3$
<i>UnReliableUser</i>	$(UP \geq n \text{ or } UM \geq m) \text{ and } y < ShilAtts \leq z$	$l_5$
<i>MostUnReliableUser</i>	$(UP > n \text{ and } UM < m \text{ and } ShilAtts > v) \text{ or } ShilAtts > z$	$l_6$

We may note that the values of  $n$ ,  $m$ ,  $v$ ,  $x$ ,  $y$  and  $z$  are flexible thresholds, which can be determined by the auctioneer according to their auction policies. In my system, I consider  $n$  as 30 days,  $m$  as 10 auctions. For the number of skill attempts, I assign  $v$ ,  $x$ ,  $y$  and  $z$  as 3, 5, 10 and 20 respectively. For example, a user who has been registered in my auction house for 200 days, has bid in 15 auctions, and has done 4 skill attempts will get 'ReliableUser' status. In the future, if the same user has 15 skill attempts, he will get 'UnReliableUser' status.

The Status Generator also assigns selling and buying limits for users based on their statuses which are also flexible thresholds (cf. Table 8). In my system I assign  $l_1$ ,  $l_2$ ,  $l_3$ ,  $l_4$ ,  $l_5$ , and  $l_6$  as unlimited, \$10,000, \$5,000, \$1,000, \$500 and \$100 respectively. For instance, when a *NewUser* joins in my system, he will get an initial limit of \$1,000 for selling or buying any products. He cannot sell or buy items for more than this amount. Periodically, the limit of a user will increase or decrease according to his updated status in order to increase the trust in my auction.

#### 4.2 Authorizer Agent

When any user tries to access any auction for bidding or requests a product for buying, he has to obtain authorization from this agent. For example, to approve a reverse, Authorizer checks whether the buyer's asking price is less than or equal to his permitted limit or not. To authorize a seller's bidding price, Authorizer first verifies that the seller's bid is less than or equal to his allowed limit or not. After that, it also inquires whether the bid is less than the asking price or not. If all these conditions are satisfied, Authorizer approves the submitted bid.

The Authorizer agent handles a large number of users' registrations, auction requests and bids to restrict suspicious user activities in my auction system.

#### 4.3 Security Controller Agent

Security Controller performs the most important role to maintain my auction system secured. It is composed of two sub-agents: Bidding Behavior Tracker and IP Tracker.

##### 4.3.1 Bidding Behavior Tracker

According to L. Rousseau (2001). Time and date information are not useful alone as they do not disclose what happened at that time. We need a pattern of timestamp instead of a clock service to monitor and detect shill bidding in live auctions. To this end, I analyze bidding behaviors in different time patterns and come up with some approaches to detect shilling. The following detection mechanisms are employed by the Bidding Behavior Tracker during the auction running time to calculate a shill score, ranging from 0 to 5 (cf. Algorithm 1).

Multiple Bid Detection: a shill bidder  $U$  often uses multiple bids ( $OutBidOwn_U$ ) to outbid his own bids (see pattern  $P_1$ ). I utilize a threshold of 2 consecutive bids for a maximum allowable multiple bids of a user:

$$OutBidOwn_U \geq 3 \quad (1)$$

Outbidding Time Detection: to give more time to normal bidders, a skill bidder bids shortly or right after being outbid compared to a normal bidder (see Pattern P<sub>2</sub>). Therefore, the average outbidding time of a user (AverOutBidTime<sub>U</sub>) may not exceed of 5 minutes incase of shilling:

$$AverOutBidTime_U \leq 5 \text{ minutes} \quad (2)$$

Outbidding Price Detection: a skill bidder outbids any bid by increasing (forward auction) or decreasing (reverse auction) the price by more than required minimum bid in order to be the top bidder (see Pattern P<sub>3</sub>) but his main goal is to not to win the auction. So, the AverBidIncrease<sub>U</sub> may exceed 10%.

$$AverBidIncrease_U > 10\% \quad (3)$$

Bidding Time Detection: a skill bidder usually bids more at the beginning of the auction (see pattern P<sub>4</sub>). The auction interval can be divided into two halves, first 50% of the elapsed time in auction interval and second 50% of the elapsed time in auction interval. Skill bidders number of bids in the first interval (BidFreqFirHalf<sub>U</sub>) is usually more than the second interval (BidFreqSecHalf<sub>U</sub>):

$$BidFreqFirHalf_U > BidFreqSecHalf_U \quad (4)$$

High Bidding Frequency Detection: a skill bidder also have more number of bids than other bidders (see Pattern P<sub>5</sub>). Their bidding frequency (BidFreq) may exceed more than half of the total bids in the auction:

$$BidFreq_U > (TotalBidsInAuction/2) \quad (5)$$

Algorithm 1: *Bidding Behavior Tracker*

---

**Inputs:** TotalBidsInAuction, BidFreq<sub>U</sub>, BidFreqFirHalf<sub>U</sub>, BidFreqSecHalf<sub>U</sub>, AverBidIncrease<sub>U</sub>, OutBidOwn<sub>U</sub>, AverOutBidTime<sub>U</sub>; **Integer**

**Outputs:** ShilStatus, ShilScore: **Integer**

**Sources:** User Database, Auction Database

```

{  ShilScore = 0;
   if (OutBidOwn_U ≥ 3)           then ShilScore = ShilScore + 1; // Pattern P1
   if (AverOutBidTime_U ≤ 5)     then ShilScore = ShilScore + 1; // Pattern P2
   if (AverBidIncrease_U > 10)   then ShilScore = ShilScore + 1; // Pattern P3
   if (BidFreqFirHalf_U > BidFreqSecHalf_U) then ShilScore = ShilScore + 1; // Pattern P4
   if (BidFreq_U > (TotalBidsInAuction/2)) then ShilScore = ShilScore + 1; //Pattern P5
   return ShilScore;
   if (ShilScore ≥ 3) then ShilStatus = 1;
   else ShilStatus = 0;
   return ShilStatus;
}

```

---

#### 4.3.2 IP Tracker

A wide range of bidding behavior detection is employed by my Security Controller Agent. However, skill behavior of duplicate identities and of users having no previous bidding history is always difficult to detect. This agent determines if any user is faking an IP address, or using multiple identities with the same IP address, or bidding on his own auction. If suspected, IP Tracker reports to Security Controller immediately (by returning a binary result: 1 if suspected; 0 otherwise). IP Tracker will utilize the shilling patterns P<sub>6</sub>, P<sub>7</sub>, P<sub>8</sub>, P<sub>9</sub>, P<sub>10</sub>, P<sub>11</sub> and P<sub>12</sub> to detect shill attempts.

#### 4.3.3 Shil Detection by Security Controller Agent

The Security Controller agent determines the shill attempts for a user according to both trackers' reports as shown in Table 9. So, if any of the sub-components provides a positive result of shilling in a running auction, Security Controller considers it as a shill attempt. To ensure trust, every live auction should be automatically monitored and examined at least three times during the auction running time (the frequency can be increased or decreased by the auctioneer). For example, in my auction house the security agent is required to compute the shill attempts for all the bidders as follows: first after 10% of the auction time has elapsed, second in the middle of the auction period, third after 90% of the auction interval has elapsed. In this way, the security agent ensures reliable results of shill detection. A user may unintentionally show one or two shill patterns. So the Bidding

Behavior Tracker considers a user as a skill bidder only when the skill score is more than 2. On the other hand, there might be some cases when a user may try to manipulate my system and use only one or two patterns. In such cases, IP Tracker will give a positive result, so that the user can be detected as a skill bidder.

Table 9. Skill Detection

Skill Score of Bidding Behavior Tracker	IP Tracker Score	Combined Result	Shill Attempts
(0-2) → 0	0 or 1	(0,0)	0
(3-5) → 1		(0,1)	1
		(1,0)	1
		(1,1)	1

4.4 Auction Controller Agent

Auction Controller works as the brain of the whole system. Any task related to a user’s account, an auction and database is done through Auction Controller to keep the system centralized and convenient. This agent is the medium between the three layers, so that it simultaneously handles run-time actions of the other agents.

4.4.1 Functionalities of Auction Controller

The Auction Controller communicates with the other agents and performs several specific tasks.

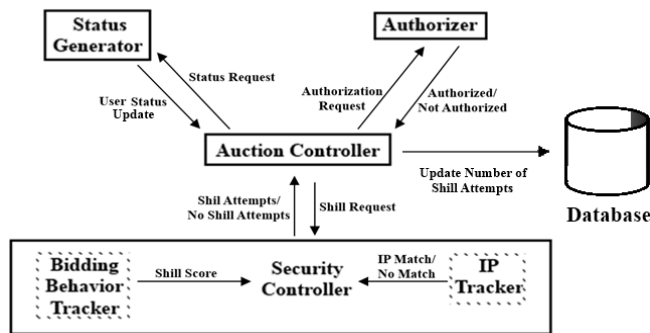


Figure 2. Functionalities of Auction Controller

When a registered user wants to sell or buy a product and submits auction or procurement data, the Auction Controller sends an authorization request to the Authorizer agent, which verifies input data and accepts or rejects the user's request. This agent also requests a status update from the Status Generator every time when Security Controller detects any shill attempts. The Security Controller computes shill attempts along with the skill scores from the Bidding Behavior and IP Trackers periodically (see Section 4.2.2). On the other hand, shilling might be suspected and reported by an auction user as well. In such situation, the Auction Controller may request the Security Controller to perform a shill check on the suspected running auction. Auction Controller also updates the User Database after any shill attempts made by any user.

4.4.2 Responses of Auction Controller

After every shill attempt in a running auction, the Auction Controller agent takes actions against the user and/or the auction according to the updated status of the shilling user (cf. Table 10).

Table 10. Actions against shilling

Result of Security Controller	Updated Status of a Shilling User	Actions of Auction Controller
Shill Attempt	<i>NewUser</i>	Action 1 and Action 3
1	<i>MostReliableUser</i>	Action 1 and Action 3
	<i>ReliableUser</i>	Action 1 and Action 4
	<i>AverageReliableUser</i>	Action 1 and Action 4
	<i>UnReliableUser</i>	Action 2 and Action 5
	<i>MostUnReliableUser</i>	Action 2 and Action 5

**Action 1:** Auction Controller pauses the running auction temporarily in case of any shilling activity.



**Action 2:** Auction Controller stops the running auction permanently in case of a determined shilling activity.

**Action 3:** Auction Controller gives a warning to the shilling user to act more reliably in the auction.

**Action 4:** Auction Controller decreases the shilling user's selling and buying limit by 10% (cf. Table 10) for every shill attempts.

**Action 5:** Auction Controller suspends the shill bidders' account temporarily for one month or permanently.

## 5. Implementation, Experiment and Results

I have implemented the ShillFree1 Auction System (cf. Figure 3) with the agent-based simulation platform Jadex by Java and XML (Braubach et al. 2004). I have used Jadex for platform architecture, core services and message transport mechanisms based on the FIPA specifications. I have created four agents in the business layer to perform specific tasks including registration, sign-in, auctioning, bidding, shill detection and etc. Each agent has a set of beliefs and facts to realize the current environment and status of the auction system. Moreover, they have specific goals that are sets of desires by which they can make appropriate decisions. Furthermore, they use messages to communicate with each other and plans by which they perform their assigned tasks. I have programmed six Agent Definition Files (ADFs) by XML and nine Java files to incorporate the plans of the agents. For the integrated development environment I use Eclipse IDE.

### 5.1 Experiment Data

I build three datasets for the experiment: users' history (cf. Table 11), auction information (cf. Table 12) and auction bids (cf. Table 13). Table 11 presents some information about the 10 registered users who have been authorized to bid on items. This table includes users with different statuses, used period, auction attendance and shill attempts. Table 12 consists of 10 English auctions initiated by different sellers and having various numbers of total bids. Table 13 exposes the bidders' data of 3 auctions (i.e. 20 rows of input data) out of 10 auctions (a total of 50 rows of input data) that are required by the Security Controller agent (TB represents *TotalBidsInAuction*, BF *BidFreq<sub>U</sub>*, BFFH *BidFreqFirHalf<sub>U</sub>*, BFSH *BidFreqSecHalf<sub>U</sub>*, ABI *AverBidIncrease<sub>U</sub>*, OBO *OutBidOwn<sub>U</sub>* and AOBT *AverOutBidTime<sub>U</sub>*).

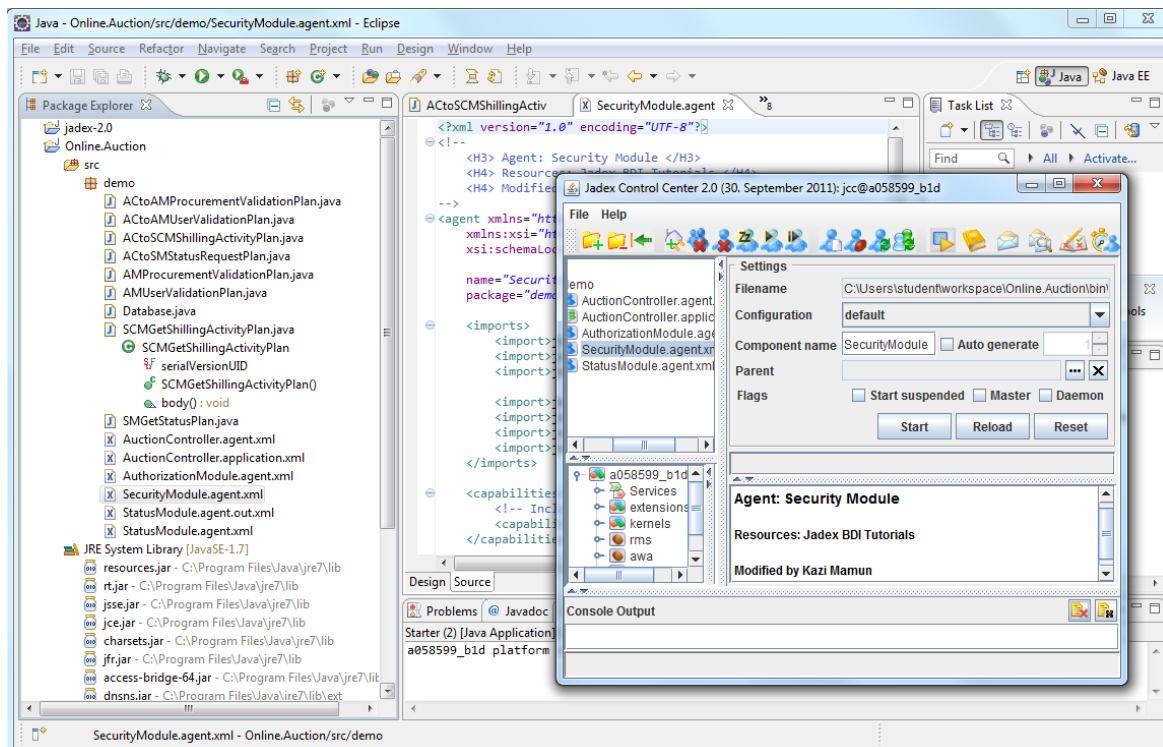


Figure 3. Implemented Auction System in Jadex

Table 11. Users' History

User ID	User Status	Used Period (in days)	Attended Auctions	Shill Attempts
U001	<i>NewUser</i>	1	1	0
U002	<i>MostReliableUser</i>	512	31	0
U003	<i>MostReliableUser</i>	321	29	2
U004	<i>AverageReliableUser</i>	227	50	7
U005	<i>UnReliableUser</i>	467	56	13
U006	<i>MostUnReliableUser</i>	89	30	25
U007	<i>ReliableUser</i>	69	23	4
U008	<i>NewUser</i>	20	4	1
U009	<i>MostReliableUser</i>	211	59	0
U010	<i>MostReliableUser</i>	324	45	0

Table 12. Auction Information

Auction ID	Seller ID	Total Bids
A001	U010	122
A002	U008	56
A003	U004	135
A004	U008	109
A005	U008	68
A006	U010	89
A007	U010	80
A008	U004	177
A009	U005	22
A010	U005	8

Table 13. Auction Bids

Auction ID	Bidder	TB	BF	BFFH	BFSH	ABI	OBO	AOBT (in minutes)
A001	U001	122	20	15	5	2	3	2
A001	U002	122	2	1	1	0	13	62
A001	U005	122	60	50	10	20	5	2
A001	U006	122	1	0	1	2	0	30
A001	U007	122	1	0	1	5	0	60
A001	U008	122	3	1	2	3	0	36
A001	U009	122	35	15	20	4	5	3
A002	U003	56	40	35	5	20	25	4
A002	U007	56	1	0	1	2	0	46
A002	U009	56	5	1	4	4	0	30
A002	U005	56	1	0	1	5	0	31
A002	U010	56	1	1	0	3	0	58
A002	U001	56	1	1	0	2	0	5
A002	U002	56	1	1	0	1	0	36
A002	U003	56	6	2	4	3	1	2
A003	U006	135	112	100	12	12	90	5
A003	U010	135	1	0	1	2	0	300
A003	U007	135	1	1	0	3	0	60
A003	U005	135	18	15	3	10	10	1
A003	U003	135	3	1	2	3	1	32

## 5.2 Results

During the auction, Auction Controller can request the Security Controller to check Shill Attempts for all bidding users. Security Controller agent determines the shilling status (whether a shill bidder or not) and score of a particular user in a particular auction. At this time, my Security Controller does not include results from the IP Tracker.

Auction Controller performs shill checks for all the users in all the 10 auctions. Table 14 shows the shilling activity results of bidders for three auctions (out of 10 running auctions) which contains the 10 users' 20 results (out of 50 results).

Table 14. Shilling Results

Auction ID	Bidding User ID	Bidding Behavior	Shill Score
A001	U001	SB	3
A001	U002	NSB	1
A001	U005	SB	4
A001	U006	NSB	0
A001	U007	NSB	0
A001	U008	NSB	0
A001	U009	NSB	2
A002	U003	NSB	1
A002	U007	NSB	0
A002	U009	NSB	0
A002	U005	NSB	0
A002	U010	NSB	1
A002	U001	NSB	1
A002	U002	NSB	1
A002	U003	NSB	1
A003	U006	SB	4
A003	U010	NSB	0
A003	U007	NSB	1
A003	U005	SB	3
A003	U003	NSB	1

One user may do shill bidding in one auction but not in others, and some users may show shilling in several auctions. Also some legitimate users may coincidentally show some shilling resulting in a positive shill despite the fact that no intentional shilling has occurred. To determine the shilling activity and vulnerability of an auction the system computes the shill scores, number of shill attempts and average shill scores of the users. I have conducted experiments on the 10 auctions with the 10 bidding users. Figure 4 shows the final results, where user U004 shows highest shilling activity as suggested by the highest total shill score (21 in five auctions) with the highest average and number of shill attempts, whereas U009 and U010 attended most auctions but can be considered as reliable as they have no shill attempts. In 10 suspected auctions, 4 out of 10 users are detected as shill bidders.

My auction system has a policy of taking actions depending on the user status; Table 15 gives the different consequences after shill attempts made by four users in 10 running auctions. For example, U004's status and selling and buying limit has been changed and the auctions are also paused temporarily. On the other hand, *UnReliableUser* U005 has been suspended temporarily for one month from the auction system and also the auctions he has attended are stopped.

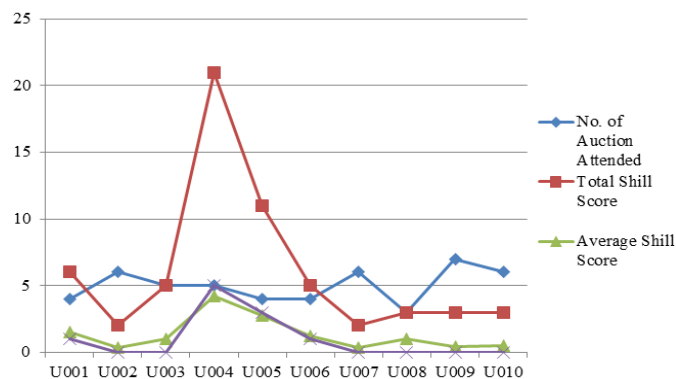


Figure 4. Final Results

Table 15. Actions taken by Auction Controller after shill attempts detected

User ID	User's Previous Status	User's New Status	Actions Taken in Live Auctions
U001	<i>NewUser</i>	<i>NewUser</i>	Paused the auction A001 and warned the shilling user
U004	<i>AverageReliableUser</i>	<i>UnReliableUser</i>	Changed status, limit and paused auctions A004, A005, A006, and A009
U005	<i>UnReliableUser</i>	<i>UnReliableUser</i>	Stopped auctions A001, A003, A008 and shilling user is suspended temporarily for one month
U006	<i>MostUnReliableUser</i>	<i>MostUnReliableUser</i>	Shilling user is suspended permanently

#### 4. Conclusion and Future Work

Fraudulent activities like shill bidding are damaging the reputation of online auctions, and have already become a serious problem in e-commerce in terms of security and trust. In this paper, I presented an auction system to secure online auctions from shill bidding at run-time. My three-layer architecture is composed of a GUI layer, a business layer and a data layer. The business layer contains four different agents to ensure shill-proof mechanisms in auction systems, in this layer the 'Auction Controller' agent keeps the system centralized and protects from shill bidding based on user status, proper authorization and shill reports from the 'Security Controller'. My auction system is secure from manipulation by shill bidders. My proposed system also takes necessary actions at run-time against any users who are detected as performing shill bidding.

For future work, to determine the shilling activity more accurately, I will implement the IP Tracker sub-agent of the Security Controller agent based on the two tracking techniques IP traceback (Snoreren et al., 2002) and DHCP origin traceback (Majumdar et al. 2011). To the best of my knowledge, no existing auction systems have considered rewarding the users for good behaviors. So, another future direction of this work is to implement an automated reward system for good behaviors.

#### Acknowledgments

I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support of this research through the Vanier Canada Graduate Scholarship.

#### References

- Braubach, L., Pokahr, A., & Lamersdorf, W. (2004). Jadex: A Short Overview. In Net.ObjectDays, AgentExpo.
- Cohen, P. (2009). Shill Bidding on eBay: a case study. *ECommerce Bytes*. Retrieved September 9, 2013, from <http://www.ecommercebytes.com/forums/vbulletin/showthread.php?t=22659>
- Dong, F. S., Shatz, M., & Xu, H. (2009). Inference of Online Auction Shills Using Dempster-Shafer Theory. *Proceedings of the 6th International Conference on Information Technology: New Generations*.
- Dong, F., Shatz, S. M., Xu, H., & Majumdar, D. (2012). Price comparison: A reliable approach to identifying shill bidding in online auctions? *Electronic Commerce Research and Applications*, 11(2), 171-179.
- IC3 (Internet Crime Complaint Center) (2007). Internet Crime Report. *Bureau of Justice Assistance*. Retrieved September 9, 2013, from [http://www.ic3.gov/media/annualreport/2007\\_ic3report.pdf](http://www.ic3.gov/media/annualreport/2007_ic3report.pdf)
- IC3 (Internet Crime Complaint Center) (2011). Internet Crime Report. *Bureau of Justice Assistance*. Retrieved September 9, 2013, from [http://www.ic3.gov/media/annualreport/2011\\_IC3Report.pdf](http://www.ic3.gov/media/annualreport/2011_IC3Report.pdf)
- Jenamani, M., Zhong, Y., & Bhargava, B. (2007). Cheating in online auction – Towards explaining the popularity of English auction. *Electronic Commerce Research and Applications*, 6(1), 53-62. <http://dx.doi.org/10.1016/j.elerap.2005.12.002>
- Majumdar, S, Kulkarni D., & Ravishankar C. V. (2011). DHCP origin traceback. Springer-Verlag, Berlin, Heidelberg, 394-406. [http://dx.doi.org/10.1007/978-3-642-17679-1\\_35](http://dx.doi.org/10.1007/978-3-642-17679-1_35)
- Mamum, K., & Sadaoui, S. (2013). Combating Shill Bidding in Online Auctions. *Proceedings of International Conference on Information Society*, Toronto, IEEE press.
- Patel, R., Xu, H., & Goel, A. (2007). Real-Time Trust Management in Agent Based Online Auction Systems. In *Proceedings of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE'07)*, Boston, USA.

- Rao A. S., & Georgeff, M. P. (1995). BDI-agents: From Theory to Practice. *In Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, San Francisco.
- Rousseau, L. (2001). Secure time in a portable device. *Proceedings of 3rd Gemplus Developer Conference*, France.
- SecurePuter. (2008). Do Not Fall Victim to Internet Auction Fraud. Internet and Web safety. Retrieved September 9, 2013, from <http://www.secureputer.com/series-do-not-fall-victim-to-internet-auction-fraud>
- Snoreren, A. C., Partridge, C., Sanchez, L. A. C., Jones, E., Tchakountio, Schwartz, F., Kent, B. S. T., & Strayer, W. T. (2002). Single-packet IP traceback". *IEEE/ACM Trans. Netw*, 10(6), 721–734. <http://dx.doi.org/10.1109/TNET.2002.804827>
- Spaced, O. (2013). Shill Bidding on eBay. *CollectSPACE*, 2005. Retrieved October 2, 2013, from <http://www.collectspace.com/ubb/Forum3/HTML/002958.html>
- Trevathan, J., & Read, W. (2005). Detecting Shill Bidding in Online English Auctions. *Technical Report*.
- Trevathan, J., & Read, W. (2007). Detecting Collusive Shill Bidding. *In Proceedings of the Fourth International Conference on Information Technology: New Generations (ITNG 2007)*, Las Vegas, Nevada, USA.
- Trevathan, J., & Read, W. (2007). Investigating shill bidding behavior involving colluding bidders. *Journal of Computers*, 2, 63-75. <http://dx.doi.org/10.4304/jcp.2.10.63-75>
- Trevathan, J., Read, W., Lee, Y. J., & Atkinson, I. (2011). Targeting the strategies of a bid sniper. *44th Hawaii International Conference on System Sciences (HICSS)*, IEEE. <http://dx.doi.org/10.1109/HICSS.2011.396>
- Wang, W., Hidvegi, Z., & Whinston, A. B. (2001). Shill Bidding in English Auctions. Working paper, Center for Research on E-Commerce, McComb School of Business, University of Texas, Austin.
- Xu, H., & Cheng, Y. T. (2007). Model checking bidding behaviors in Internet concurrent auctions. *International Journal of Computer Systems Science & Engineering*, 22(4), 179-191.
- Xu, H., Bates, C. K., & Shatz, S. M. (2009). Real-Time Model Checking for Shill Detection in Live Online Auctions. *Software Engineering Research and Practice*.
- Xu, H., Shatz, S. M., & Bates, C. K. (2008). A Framework for Agent-Based Trust Management in Online Auctions. *In Proceedings of the 5th International Conference on Information Technology: New Generations (ITNG 2008)*, Las Vegas, NV, USA.

### Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).