PREVENTION AND DETECTION OF CERTAIN TYPES OF PLAGIARISM DURING COMPUTERIZED ASSESSMENTS

Levon R. Hayrapetyan, Houston Baptist University

ABSTRACT

Unfortunately, plagiarism is widespread on university campuses across the nation. The advances in the information technology provide even more sophisticated cheating prospects. Although there are many commercially available tools for detecting plagiarism but policing alone is not a comprehensive solution. We should strive to change the overall culture on university campuses in such a way that it discourages academic dishonesty. In this study, we present a tool called Test Guard that has two main features. First, it disables some cheating "techniques" such as copy-paste, insert file, etc. Then it checks the test for several types of plagiarism and generates a report on its findings.

KEYWORDS: academic dishonesty, cheating, plagiarism, computerized assessment

INTRODUCTION

As educators, we should strive to create an environment in academic classrooms that discourages academic dishonesty. Naturally, there are two ways of fighting plagiarism on university campuses – implementing processes to prevent them as much as possible, and employing procedures to detect them if they occur. This study aims to address these both approaches in one single methodology.

Unfortunately, the students' academic misconduct was always part of the higher (and not only) education. The advances in the information technology provide even more advanced and sophisticated cheating prospects. Recent studies on academic dishonesty show an alarming high rate ranging from 52% to 90% (Lester and Dekhoff, 2002; Vowell and Chan, 2004). Diekhoff and his colleagues suggest that the rate of plagiarism have grown from 54.1% in 1984 to 61.28% in 1994, and "it is getting worse" (Diekhoff, LaBeff, Clark, Williams, Francis, and Haines, 1996). In more recent years, Don McCabe conducted a research for the Center for Academic Integrity at Duke University by surveying 50,000 undergraduate students during 2002 to 2005 on 60 university campuses nationwide. The research showed that 70% of surveyed students reported cheating (Hudd, S.H., Apgar, C., Bronson, E., F., and Lee, R.G., (2009)).
The current literature is analyzing the cheating phenomenon from various angles, such as why do students cheat, which type of relationships exist between cheating and demographic factors such as gender, ethnicity, major, Greek membership, how to prevent cheating in different settings, and how to detect cheating if it occurs.

By analyzing the literature on academic integrity, one can notice that educators have directed their efforts mainly towards detection of plagiarism; much less work is done towards prevention of it. This paper is a contribution in filling this gap for a specific yet widely used type of assessment – computerized tests and quizzes. In the paper, we present a tool called *Test Guard*, which has dual functionality. First, it tries to prevent cheating by disabling some cheating "techniques" such as copy-paste, insert file, etc. Then it checks the test for several types of plagiarism and generates a report on its findings. The report is embedded and hidden in the student's test file and is password protected – only the professor can see it.
The remainder of the paper is organized as follows. Section 2 briefly discusses the relevant literature on academic dishonesty. The short description of the functionality of the *Test Guard* is presented in the Section 3. Finally, the Section 4 concludes the paper with discussion, limitations of the proposed methodology and possible directions for future research.

LITERATURE REVIEW

There have been many discussions of academic integrity in resent literature -Rudebock, R. D. (2005), McCabe, D., and Pavela, G. (2004), Chapman, K. J., Davis, R., Toy, D., and Wright, L. (2004), Storch, E. A., and Storch, J. B. (2002), Bernardi, R. A., Metzger, R. L., Bruno, R. G. S., and Hoogkamp, M. A. W. (2004), Carpenter D. D., Harding, T.S., Montgomery, S. M., Steneck, N. H., and Dey, E. (2002), Caruana, A., Ramaseshan, B, and Ewing, M. T. (2000), Coleman, N., and Mahaffey, T. (2000), Hard, S. F., Conway, J. M., and Moran, A. C. (2006), Hardling, T. S., Carpenter, D.D., Finelli, C. J., and Passow, H. J. (2004), Hudd, S.H., et al. (2009), Hardling, T. S. (2000). The topic continues to be one of the hotest topics in the educational research because the severity of academic dishonesty is increasing. Academic dishonesty is analyzed and studied from various angles. For example, McCabe D.L. el al. (2004) have outlined principles of academic integrity; Chapman, K.J., et al. (2004) and Mathison, D.L. (2010) studied academic integrity in the Business School environment. Similar study is done by Passow and his colleagues for engineering students (Passow, Mayhew, Finelli, Harding, and Carpenter (2006); Kidwell, L.A. et al. (2003) and Koljatic, M. et al. (2002) analyzed and compared students' and faculty's perception of dishonest academic conduct. Some researchers study the cheating pattern in their particular universities and provide specific guidelines for preventing and detecting plagiarism (Hudd et al., 2009; Kitahara, Westfall, and Mankelwicz, 2011).

Nonis and Swift analyzed the correlation between academic dishonesty and workplace dishonesty (Nonis, S. A., and Swift, C. A. (2001). Steam studied how the relationship between students and professors effects on academic integrity (Stearns, S. A. (2001). There are many commercially available tools for detecting plagiarism and the TurnItIn is probably the most commonly used among them (http://turnitin.com). An effective procedure for handling cheating on multiple-choice tests was developed by Harpp and his colleagues (1993, 1996). Nath, L., and Lovaglia, M. (2009) developed software, which goes further by precisely estimating the probability that copying of answers took place between two students taking a multiple-choice exam. Joseph Fulda developed a plagiarism detection method using WordPerfect (Fulda, J.S., 2009). Regardless of the wide range of research articles related to the plagiarism, all of them came to the same conclusion – the problem does exist, it is getting worse and we need to come up with policies and procedures, which will help to create an environment of academic honesty, prevent plagiarism as much as possible, and detect it when it occurs.

THE TEST GUARD

As it was mentioned above, the tool is designed to be used in computer-based tests or quizzes. It allows the detection of certain types of cheating, namely when one student submits a work that was completely or partially done by another student on another computer. Here are some possible cheating scenarios:

1. Student A asks his classmate Student B to help with the test. Student B completes the test and sends a copy of it to Student A. Then Student A slightly "personalizes" the file (adds his name, changes some colors or fonts, etc.) and submits the file as his own work.
2. Student A sends test file and requirements to a person who may be located anywhere in the world. This person completes the test and sends the file back to Student A. Then Student A submits the ("personalized") work as his own.
3. Student A gives his password to a person who may be located anywhere in the world. This person logs into the system as Student A and completes the test. Then Students A changes his password.

Because the vast majority of tests require the creation of Microsoft Word documents, the current version of *Test Guard* it designed to prevent and detect plagiarism in case when students should either create a new Word document from scratch, or complete a test file prepared by the professor. The idea is to "beefup" the test file with a special software written in VBA – Visual Basic for Applications (Mansfield, R. (2008), Jacobson, R. (2002), Albright, C. S. (2001)), which will not have any visual effect on the test file at all, i.e., the test file will look like exactly the same before and after "beefing-up". This hidden software is the backbone of the tool – it prevents and detects possible occurrences of plagiarism. The user of the tool, i.e. the professor giving a test or a quiz, does not have to know anything about programming at all - the programming part is completely hidden from the professor (as well as from the students).

The *Test Guard* can be used when a professor is providing a partially completed test file in Microsoft Word (input file) and asking students to finish it. If the nature of the test is such that students should create a file from scratch, then Test Guard can still be easily used by creating a blank Word file and asking students to use it for the test. The *Test Guard* does its job by adding invisible code to the test's input file, which prevents various cheating methods, creates report on computers usage for the test, as well as implements some other security features.

Preparing an input file for the test is very simple task and it takes less than 20 seconds to complete. The professor should open the single-page "*Test Guard.docm*" file, which contains short descriptions of the tool and one-step instruction which tells the user to click on "*No cheating*" button that was added (by *Test Guard*) to the "Quick Start" menu located in the upper left corner of the screen. When the button is clicked, a standard "Open" dialog box appears where the user can select the test file. If there is no initial test file, then *Test Guard* will ask the user to enter a name for a blank test file. As soon as the selection is made, i.e. either an existing test file is selected or a name for a blank test file is provided, the user is asked to enter a password, which will be used to display various statistical information on usage of the test file by a student(s). This is all the professor should do. The input file for the test is now "processed", i.e., "beefed up" with security features and is ready for distributed to students.

As it was mentioned above, one of the most important characteristics of this tool is its ability to prevent certain types of plagiarism. When student opens a test file and starts working on it, various "cheating" techniques are disabled. For example, student cannot copy an information from somewhere (from other file, Internet, etc.) and paste it into his or her document because "paste" feature will simply not work! The same is true for inserting a file into the test document. There are some other security measures implemented in *Test Guard*.

All these measures are good tools to prevent cheating on the test, but unfortunately, some student may still try to find ways to cheat – see some possible scenarios described above. The good news is that the same *Test Guard* can not only prevent, but also detect cheating. The professor does not have to do anything extra to use those detection features – they are already implemented in the test file.

This is how it works: when a professor opens the student's test file for grading, a new line adds to the "Add-Ins" menu item called "*Any Cheating?*" (If there was no "Add-Ins" on the ribbon then it will be added by *Test Guard*). It acts as a toggle switch, which shows or hides a password protected textbox with descriptive information on usage of the test file. It also checks for possible cheating. Table 1 depicts a content of the textbox before the test file is distributed to the students.

Table 1: The content of the textbox before distributing the test to students

	Just Created	
--	Activity Report	--
USERNAME	COMPUTER NAME	TIME
Administrator	HOME-AB29CFC53F	6/9/2010 7:08:39 PM – Professor

This table presents the activity report, which was embedded in the test file by ***Test Guard*** *before distributing the test to students.*

The first entry in this textbox is related to the professor who created the test and contains his or her username, computer name and the test creation time.

When students submit the test files, the professor may check and see if there was any cheating involved. In order to do so, the professor should unhide the embedded (in the test file) statistical information generated by *Test Guard*. This information is password protected and can be retrieved by selecting the "Any Cheating?" command in the "Add-Ins" menu. Here are some possible types of embedded information.

Example 1: Sam Brown Did Not Cheat

Figure 4 below shows the information which the professor retrieved by clicking the "Any Cheating?" command in the "Add-Ins" menu while grading Sam Brown's test file. As one can see, Sam Brown honestly took his test on computer station #1356 and completed it at 8:10 pm. No cheating was detected.

Table 2: Sam Brown Did Not Cheat

No Cheating Was Detected!

-- Activity Report--

USERNAME	COMPUTER NAME	TIME
Administrator	HOME-AB29CFC53F	6/9/2010 7:08:39 PM – Professor
sbrown	DSKT00001356	6/10/2010 7:04:37 PM
sbrown	DSKT00001356	6/10/2010 8:09:56 PM

This table presents the activity report, which was embedded in Sam Brown's test file by ***Test Guard****. Mr. Brown completed his test honestly.*

Example 2: Tom White Did Cheat

Table 3 below shows the activity report embedded in Tom White's test file, which was retrieved by clicking the "Any Cheating?" command in the "Add-Ins" menu. Cheating was detected, because computers' names in the second column are different. This informs the professor that Tom got a test file from a student N. Simon, sitting at station #1345. In this scenario, both students who were engaged in cheating are uniquely identified.

Table 3. Tom White Did Cheat

Cheating!

-- Activity Report---

USERNAME	COMPUTER NAME	TIME
Administrator	HOME-AB29CFC53F	6/9/2010 7:08:39 PM – Professor
nsimon	DSKT00001345	6/10/2010 7:06:55 PM
nsimon	DSKT00001345	6/10/2010 7:55:16 PM
twhite	DSKT00001367	6/10/2010 7:59:35 PM
twhite	DSKT00001367	6/10/2010 8:12:24 PM

This table represents the activity report, which was embedded in Tom White's test file by ***Test Guard****. Cheating was detected.*

Example 3: Nancy Brown Also Cheated

Table 4 below shows the activity report embedded in Nancy Brown's test file, which was retrieved by clicking the "Any Cheating?" command in the "Add-Ins" menu. In the activity report, cheating was detected because computer names in the second column are different. This informs the professor that Nancy sent her test file to Bill Gray, who completed the test and sent it back to Nancy. Then Nancy probably made some cosmetic changes and submitted the file as her own work. In this case, Bill Gray could be located anywhere in the world.

Table 4: Nancy Brown Also Cheated

Cheating!

-- Activity Report---

USERNAME	COMPUTER NAME	TIME
Administrator	HOME-AB29CFC53F	6/9/2010 7:08:39 PM – Professor
nbrown	DSKT00001334	6/10/2010 7:03:12 PM
Bill Gray	Billy	6/10/2010 7:15:19 PM
Bill Gray	Billy	6/10/2010 7:48:32 PM
nbrown	DSKT00001334	6/10/2010 8:01:48 PM

This table represents the activity report, which was embedded in Nancy Brown's test file by ***Test Guard****. Cheating was detected.*

DISCUSSION

Disappointingly, cheating is widespread on almost all university campuses. As educators, we have to create a learning environment where plagiarism is unacceptable. But how to do it? Obviously, it is not enough to limit ourselves with policing students while they are taking a test or a quiz. It seems that one possible solution could be the implementation of an honor code on university campuses, but the literature reports contradicting results. While McCabe and his colleagues found that, the honor code reduces the level of cheating (McCabe, Trevino, and Butterfield, 1999), Vandehey and his colleagues found that "Student awareness of an honor code with broad university support did not reduce student cheating" (Vandehey, M. A., et. al., 2007). As many courses became available online and many professors started giving computerized tests and quizzes, the cheating on such assessments become more sophisticated. Nowadays, students may use various Internet-connected devises in a way that are harder to detect than in the past. Copying a text from the Internet, modifying it and submitting as his or her own work is even not considered as cheating by some students.

The *Test Guard* introduced in this study, is a methodology, which tries to incorporate two aspects of cheating – prevention and detection, in a single tool. The tool has dual functionality. First, it tries to prevent cheating by "disarming" students in terms of disabling various commonly used cheating techniques, such as copy-paste and insert. Second, it implements variety of detection features, which will alarm the professor about the cheating, if it took place. The latter is reaches by collecting various types of information on students' activities during the assessment process. This information is invisibly embedded in the students' test flies and is password protected – only the professor can review them. Based on this information, the professor can not only detect the plagiarism but also figure out which type of cheating did occur. To our knowledge, this approach, i.e., combining the prevention and detection in one integrated tool, is new and there are no competing tools on the market.

We should mention some limitations of this study. First, the tool is limited to computerized assessments using Microsoft Word only and the assessment should take place in the classroom. We deliberately started with Microsoft Word, because it is one of the most commonly used assessment environment in the higher education. The second limitation is more technical – in order to use the tool, the macros in the Microsoft Word should be enabled otherwise the embedded software cannot work. If macros are disabled, then the students should enable them because it is impossible to enable macros programmatically (for security reasons). A computer-savvy student may try to take advantage of this fact and tamper with the functionality of the tool.

The future research may develop in various directions. One direction could be the extension of the tool beyond Microsoft Word. For example, creation of a similar tool for assessments in Microsoft Excel. It could be a useful instrument in assessing student enrolled in classes like "Introduction to computers". Another direction could be making this tool web-based. Yet another direction could be adding mere security features into the tool, which will prevent tampering with it by computer-savvy students.

CONCLUDING COMMENTS

Recent studies on academic dishonesty show an alarming high rate ranging from 52% to 90% (Lester et al., 2002; Vowell et al., 2004). The advances in IT can make it even worse. Thus, it is imperative for us, educators, to be very proactive in developing and implementing various tools for prevention and detection of plagiarism. This study is dedicated to this goal – the integrated methodology is proposed which combines both prevention and detection in a single and easy-to-implement tool.

REFERENCES

Albright, C. S. (2001). VBA for Modelers: Developing Decision Support Systems Using Microsoft® Excel. *Duxbury.*

Bernardi, R. A., Metzger, R. L., Bruno, R. G. S., and Hoogkamp, M. A. W. (2004). Examining the Decision Process of Student Cheating behavior: An Experimental Study. *Journal of Business Ethics, 50(4), 397 – 414.*

Carpenter D. D., Harding, T.S., Montgomery, S. M., Steneck, N. H., and Dey, E. (2002). *Student Perceptions of Institutional and Instructor Based Techniques for Dealing with Academic Dishonesty. The 32nd ASEE/IEEE Frontier in Education Conference, Boston, MA.*

Caruana, A., Ramaseshan, B, and Ewing, M. T. (2000). The Effect of Anomie on Academic Dishonesty Among University Students. *The International Journal of Educational Management, 14(1), 23 – 30.*

Chapman, K. J., Davis, R., Toy, D., and Wright, L. (2004). Academic Integrity in the Business School Environment: I'll Get by with a Little Bit of Help From My Friend. *Journal of Marketing Education, 26(3), 236 – 250.*

Coleman, N., and T. Mahaffey (2000). Business Student Ethics: Selected Predictors of Attitudes toward cheating. *Teaching Business Ethics 4, 121-136.*

Diekhoff, G., M, LaBeff, E., Clark, R.E., Williams, L.E., Francis, B., and Haines, V., J. (1996). College Cheating: 10 Years Later. *Research in Higher Education, 37, 4, 487-502.*

Fulda, J. S. (2009). Using WordPerfect to Detect Plagiarism on Open-Book Essay Tests. *British Journal of Educational Technology. 40, 5, 950 – 955.*

Hudd, S.S., Apgar, C., Bronson, F.E., and Lee, G.R. (2009). Creating a Campus Culture of Integrity: Comparing the Perspectives of Full- and Part – time Faculty. *The Journal of Higher Education, 80, 20,146-177.*

Hard, S. F., Conway, J. M., and Moran, A. C. (2006). Faculty and College Student Beliefs About the Frequency of Student Academic Misconduct. *The Journal of Higher Education, 77(6), 1058 – 1080.*

Harpp, D. N., and Hogan, J. J. (1993). Crime in the Classroom: Detection and Prevention of Cheating on Multiple-Choice Exams. *Journal of Chemical Education, 70(4), 306 – 311.*

Harpp, D. N., Hogan, J. J., and Jennings, J. S. (1996). Crime in the Classroom: Part II. An update. *Journal of Chemical Education, 73(4), 349 – 351.*

Hardling, T. S., Carpenter, D.D., Finelli, C. J., and Passow, H. J. (2004). Does Academic Dishonesty Relate to Unethical Behavior in Professional Practice? An Exploratory Study. *Science and Engineering Ethics, 10(2), 311-324.*

Hardling, T. S. (2000). Cheating: Students Attitudes and Practical Approaches to Dealing with it. *The 30th ASEE/IEEE Frontier in Education Conference, Kansas City, MO.*

Jacobson, R (2002). Microsoft® Excel 2002 Visual Basic for Applications. Step-by-Step. *WA: Microsoft Press.*

Kidwell, L. A., Wozniak, K., and Laurel, J. P. (2003). Student Reports and Faculty Perceptions of Academic Dishonesty. *Teaching Business Ethics, 7(3), 205 – 214.*

Kitahara, R., Wstfall, F, and Mankelwicz, J. (2011). New, Multi-Faceted Hybrid Approach to Ensuring Academic Integrity. *Journal of Academic and Business Ethics, 3, 1 – 12.*

Koljatic, M., and Silva, M. (2002). Comparison of Students' and Faculty's Perception of Occurrence of Dishonest Academic Behaviors. *Psychological Reports, 90, 883 – 888.*

Lester, M. C., and Diekoff, G. M. (2002). A Comparison of Traditional and Internet Cheaters. *Journal of College Student Development, 43, 5, 2 – 7.*

Mansfield, R. (2008). Mastering VBA for Microsoft Office 2007. *Wiley.*

Mathison, D.L. (2010). Cheating In Business Schools, the Millennial Generation, Gender and Racial Diversity: Has the Paradigm Shifted? *American Journal of Business Education*, 3, 11, 111-117.

McCabe, D. L., Trevino, L. K., and Butterfield, K. D. (1999). Academic Integrity in Honor Code and Non-Honor Code Environment. *The Journal of Higher Education, 70, 2, 211 – 234.*

McCabe, D., and Pavela, G. (2004). Ten Principles of Academic Integrity. *Change, 36(3), 10-15.*

McCabe, D. (2005). Levels of Cheating and Plagiarism Remain High, Honor Codes and Modified Codes as Shown to be Effective in Reducing Academic Misconduct. Retrieve

Nath, L., and Lovaglia, M. (2009). Cheating on Multiple-Choice Exams: Monitoring, Assessment, and Optional Assessment. *College Teaching, 57(1), 3 –87.*

Nonis, S. A., and Swift, C. A. (2001). An Examination of the Relationship between Academic Dishonesty and Workplace Dishonesty: A Multicampus Investigation. *Journal of Education for Business, 77(2), 69 – 77.*

Passow, H. J., Mayhew, M. J., Finelli, C. J., Harding, S. T., and Carpenter D. D. (2006). Factors Influencing Engineering Students' Decisions to Cheat by Type of Assessment. *Research in Higher Education, 47(6), 643-684.*

Rudebock, R. D. (2005). When Honesty is the Expected Policy: Utilizing Peer Relations in Classroom Management. *College Teaching, 53(4), 145.*

Stearns, S. A. (2001). The Student-Instructor Relationship's Effect on Academic Integrity. *Ethics & Behavior, 11(3), 275 – 285.*

Storch, E. A., and Storch, J. B. (2002). Fraternities, Sororities, and Academic Dishonesty. *College Student Journal 36(2), 247-252.*

Vandehey, M. A., Diekhoff, G. M., and LaBeff, E. E. (2007) College Cheating: A Twenty-Year Follow-UP and the Addition of an Honor Code. *Journal of College Student Development, 48, 4, 468*

Vowell, R., and Chan, J. (2004). Predicting Academic Misconduct: A Competitive Test of Four Sociological Explanations. *Sociology Inquiry, 74, 2, 226 – 249.*

BIOGRAPHY

Dr. Levon Hayrapetyan is a professor in School of Business at Houston Baptist University (HBU). He earned his BS and MS Degrees in applied mathematics from Yerevan State University, Armenia, and his Ph.D. in theoretical computer science from Kiev State University, Ukraine. Before his tenure at HBU, Dr. Hayrapetyan worked as a research scientist at State University of New York, Health Sciences Center at Syracuse, NY, where he was actively involved in research in the area of computational neuroscience.