

December 2001

# Can E-Cheating be Prevented? An Approach to Detect Plagiarism in Computer Skills Courses

James Coakley  
*Oregon State University*

Craig Tyran  
*Oregon State University*

Follow this and additional works at: <http://aisel.aisnet.org/amcis2001>

---

## Recommended Citation

Coakley, James and Tyran, Craig, "Can E-Cheating be Prevented? An Approach to Detect Plagiarism in Computer Skills Courses" (2001). *AMCIS 2001 Proceedings*. 241.  
<http://aisel.aisnet.org/amcis2001/241>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# CAN E-CHEATING BE PREVENTED? AN APPROACH TO DETECT PLAGIARISM IN COMPUTER SKILLS COURSES

**James R. Coakley**

Department of Accounting, Finance, and  
Information Management  
College of Business  
Oregon State University  
coakley@bus.orst.edu

**Craig K. Tyran**

Department of Accounting, Finance, and  
Information Management  
College of Business  
Oregon State University  
tyran@bus.orst.edu

## Abstract

*Articles in the public media, as well as academic journals have reported disturbing research findings about the cheating behavior of students. For faculty that teach in the area of information systems, the news regarding cheating is even more sobering, as studies indicate that cheating is most likely to occur in the disciplines of business, engineering and science (McCabe and Trevino 1996; Newstead et al. 1996). The objective of this manuscript is to describe an approach that we implemented in our information systems program to deal with the challenge of detecting plagiarism in student work involving computer skills. Specifically, we discuss a software program that was developed to detect plagiarism in spreadsheet and database exam assignments. We also offer an illustration of how the approach is used in practice and provide an assessment of our experiences with the approach.*

**Keywords:** Academic honesty, cheating, computer skills education, information systems education, plagiarism

## Introduction

*A new epidemic of fraud is sweeping through our schools*

Cover headline for *US News and World Report* cover story entitled “The Cheating Game” (November 22, 1999)

As highlighted by the US News and World Report cover story cited above, the topic of cheating in schools has received much attention recently. Articles in the public media, as well as academic journals have reported disturbing research findings about the cheating behavior of students. For example, a recent study that surveyed more than 6000 students at 31 campuses found that 70% of students surveyed admitted to one or more instances of cheating on a test (McCabe and Trevino 1996). It appears that many of these students are “repeat offenders” of test cheating, as 38% of the respondents in the study reported that they had been involved with four or more instances of explicit test cheating. Other studies have also found high rates of cheating (e.g., Davis et al. 1992). It seems likely that cheating is going to continue to be a problem for some time to come, as there are many reasons that may compel a college student to be dishonest. These reasons include competition for grades, insufficient study time, heavy workload, insufficient surveillance during testing, and pressure from parents (Baird 1980).

For faculty that teach in the area of information systems the news regarding cheating is even more sobering, as studies indicate that cheating is most likely to occur in the disciplines of business, engineering and science (McCabe and Trevino 1996; Newstead et al. 1996). Based on these types of research findings – along with their personal experiences – many faculty members in the area of information systems (IS) are understandably concerned about the threat of cheating. Unfortunately, it is possible that things may get worse for some faculty before they get better. As with many areas of the university, IS faculty are starting to teach distance education courses or on-line versions of their courses. Since the interaction between the teacher and the students may be more limited in the electronic learning environment and it can be easy to cheat in an environment involving electronic deliverables, distance-learning settings offer the potential for increased opportunities to cheat (Kennedy et al. 2000).

With regard to information systems education, one of the areas in which cheating may occur is the area of computer skills. Examples of computer skills include spreadsheet design, database development, and computer programming. With regard to these types of skills, a common type of cheating problem is plagiarism (Joy and Luck 1999). Plagiarism is defined as the act taking ideas, writings, etc. from another and passing them off as one's own (Webster's 1983). Plagiarism can be a simple act to perform in a computer skills course since a student can simply copy (or cut and paste) portions of the software from a classmate (or another source) into their own submission deliverable. We refer to this type of cheating as "E-Cheating." While it can be easy for a student to E-Cheat, it can be very difficult for a teacher to tell the difference between honest work and plagiarized work. For example, introductory computer skills courses where all students are assigned to address the same problem using the same data and the same problem specifications can present a situation in which it is very difficult for an instructor to detect -- or prove -- that cheating has occurred, as the solutions that the students create should be very similar. For instance, assignments that require a student to create specific formula features for a spreadsheet or specific tables and queries for a database are likely to result in very similar solutions across students.

The objective of this manuscript is to describe an approach that we implemented in our information systems program to deal with the challenge of detecting plagiarism in student work involving computer skills. Specifically, we discuss a software program that was developed to detect plagiarism in spreadsheet and database exam assignments. In the next section, we provide a brief background on the topic of approaches to detect plagiarism for computer skills courses. We then discuss the approach that we have developed and provide an illustration of how the approach is used in practice. We finish with an assessment of our experiences with the approach and concluding remarks.

## **Approaches to Detect Plagiarism**

As noted earlier, in the case of courses involving computer skills, the teacher is presented with difficult challenges with regard to cheating. What techniques and approaches may a teacher of computer skills employ to minimize cheating? In this section, we draw from the literature and our own experiences to discuss detection-based approaches that may be used to control and minimize cheating in computer skills courses.

The threat of detection can provide a strong deterrent to cheating. If a student believes that a teacher can identify—and prove—an incidence of cheating, then a student is less likely to cheat. In the area of computer skills coursework, much of the published research on detecting plagiarism relates to programming assignments (e.g., Joy and Luck 1999). This research has focused on techniques to compare programs with respect to similar attributes (e.g., size of program, counts of operators) or program structure. Research in this area has yielded techniques that are reported to be very effective for identifying plagiarism problems with computer programs (e.g., Whale 1990). These techniques can play a key role in detecting cheating problems in programming courses and are used by many college instructors. Outside of the programming arena, however, our review of the literature has not found much discussion of methods developed to detect plagiarism for other types of computer skills applications. In particular, we could not find literature studies that report on ways to deal with plagiarism for two types of computer skills that are frequently taught in college business programs: spreadsheet skills and database skills. To address this issue, we have developed an approach—referred to as the "Key Detector" approach—for detecting plagiarism in student work involving spreadsheets and databases. This approach is discussed in the next section.

## **The Key Detector Approach**

The Key Detector approach is based on a coding technique in which each student is provided with a unique data set that is used to complete a computer skills assignment. Additionally, procedural controls are implemented when testing and evaluating the students. In this section, we provide background and specifics regarding the approach.

### ***Background***

In our computer skills courses involving spreadsheet and database software, we use Microsoft Excel and Access as the course software. We conduct supervised, hands-on examination assignments as the primary means to test the Excel and Access skills of each student. (While students are also evaluated based on skill-building homework assignments, the examination assignments carry the most weight toward the overall course grade.) Even though the hands-on examinations are supervised in a laboratory setting, it still would be possible for students to share their work files during the exam (e.g., students could send e-mail

attachments to one another or share files using our school's network file server). Hence, we needed to develop an approach that would make it possible to ensure the integrity of our examination process and detect cases of plagiarism.

### ***Using an Identifier Key to Create Unique Data Sets***

The computer skills examinations that we administer require students to work with data sets that are provided to the student. For example, for an exam involving spreadsheet skills, at the beginning of the exam we would provide students with the data that would need to be used in the spreadsheet. Our approach to detect plagiarism on computer skills tests of this nature is to provide each student with a unique, identifiable data set for the evaluation exam. While the data that each student gets is drawn from the same master data set, each individual student data set is slightly modified from the master set by planting several identifier keys inside the data set. The identifier keys are unique to each student and are based on an algorithm involving a student's last name, first and middle initials, and birth date. The way the identifier keys are included in the data makes it very difficult to find them. For example, portions of the identifier keys can be embedded as part of specified numbers or character strings inside the student's data set. Once the data sets for each student are generated, they are distributed to the students just prior to the test. When the testing period is over, each student's test deliverable is collected electronically (the deliverable is an electronic file such as a spreadsheet file). When the deliverables are evaluated, the data in each file is checked to ensure that the identifier keys in the student's submitted data set match the student's unique identifier key. If a student's submission includes an identifier key for another student, then there is strong evidence that a plagiarism event has occurred.

### ***Automating the Process***

The process described above would be very cumbersome if performed manually, as it would take much time to a) create and distribute the unique student data sets, and b) collect and evaluate the students' work. An automated program has been developed to make the process manageable. The automated program was developed using the Microsoft Access database platform and the Visual Basic for Applications (VBA) programming language. The master data set for an examination exercise is stored and managed within an Access database. VBA program routines have been created to generate a different data set for each student and seed the unique identifier into the student's data. In addition, a VBA program has been prepared to automatically distribute a file containing each student's data set to by means of a direct file copy to the student's network storage area on the College's computer network. The student then works on the examination. At the end of the allotted exam time, the completed test deliverable files are automatically "harvested" (i.e., copied directly from the student network area) into the teacher's network folder by another VBA routine. After the files are harvested, a VBA program evaluates the deliverable file and also verifies the unique identifier key seeded into the data set. The program provides a listing of all student submissions that include an unverified identifier key (and are thus candidates for plagiarism). The automated process can be easily run using a standard office PC and the process goes very quickly. For a class size of 150 students, the amount of time required to do the process steps discussed above is less than five minutes for the entire class.

## **The Key Detector Approach in Practice: PC Productivity Software Course**

In this section, to illustrate how the Key Detector approach is used in practice, we discuss an implementation example involving our freshman-level "PC productivity software" course. In the freshman-level course, the evaluation assignment requires the student to integrate word processing and spreadsheet skills. Each student is provided an initial Excel spreadsheet with customer and product information, and a comma-delimited text file that must be imported into the spreadsheet. There are numerous versions of the initial spreadsheet. At the beginning of each term, we seed the version number into the spreadsheet data making each spreadsheet distinctly different from those used in previous terms. The VBA routine in the Access database seeds the student identifier into a table containing sales data.

The students take the exam during the scheduled class time in our computer lab facility and the identity of all students is verified at the beginning of the examination time period. After the automated program prepares the data table for each student, the student's data table is exported as a comma-delimited text file directly into the student's folder on the College computer network, and an email message is sent to the student with exam instructions and one of the initial spreadsheets containing customer/product data. The student then begins the exam, which involves the development of a sales analysis spreadsheet model in Excel based on the imported data set, as well as the creation of charts that display the projected sales performance. Additionally, a Word document must be set up to mail merge using the customer information in the spreadsheet as the data source. The Word document is formatted as a sales report, with a dynamic link to the sales performance chart in the Excel spreadsheet. As the students work on their exam, they save their files in a specific folder on their network storage space.

At the end of the exam period, the files are harvested and every completed exam is automatically checked for the following:

- The version number of the spreadsheet must match the version sent to the student.
- The identifier extracted from the sales data must match the student's unique identifier.
- The Word document must link to the completed spreadsheet—both the data source for the mail merge and the dynamic link for the chart.
- The Word document is checked to see that it was a) created during the exam period (not before and not after), and b) created by the student registered for the exam (we check on this by checking the "Properties" of the Word document for the login id of the user who created the word document).

It is very difficult for a non-expert to cheat on this exam. Since the initial data files are seeded with the identifier keys, students cannot submit work generated by their classmates. Also, it is not possible to use a spreadsheet from a previous term, since the version of the exam varies from term to term. Additionally, the Word document that is part of the deliverable cannot simply be copied from another source. While students may be able to "cut and paste" the content of the document, they cannot cut and paste links to the current spreadsheet. They would need to re-establish the link to the data source for the mail merge, and replace the existing chart with the one in the current spreadsheet.

## Assessment of the Key Detector Approach

We are in our third year of using this approach for our course that involves Microsoft Excel spreadsheet skills and we are in our second year of using the approach for our course that involves Microsoft Access skills. Based on our experiences, the approach has proven to be an effective way to address the problem of potential cheating on computer skills exams. While we identified several cases of cheating during the early phases of implementing this approach, identified plagiarism incidents are now very rare. It appears that the students understand that cheating on the exams will be detected and thus students do not even attempt to do so. While it is difficult to say with certainty, to our knowledge no students have been successful in cheating on the computer skills exams since the approach was implemented. (We should note that it is theoretically possible for a student to cheat with the Key Detector approach, but a student would need to have skills and knowledge that far surpass the skills required to pass the exams.) Since implementing this approach, we have found that our students have tended to work harder to learn their computer skills prior to the exams since they know that they will need to demonstrate their skills during a hands-on examination that is fair and equitable to all.

In addition to effectiveness, the Key Detector approach has also been an efficient way to address the problem of potential cheating on computer skills exams. Although it took significant time in the early stages to develop the software to automate the approach described in this manuscript, now that the program has been developed, it only takes a few minutes to implement the approach for each exam.

Looking toward the future, we intend to continue using the Key Detector approach in all of our computer skills courses involving the Microsoft Office suite. In addition, we have already started to apply this approach to non-Microsoft platforms as well, as we have recently implemented the approach for a database course that involves the Oracle database platform. On another front, the success of the Key Detector approach has been encouraging in that it will enable us to more comfortably move forward with our College's proposal to offer our introductory IS courses using a distance learning format. Without having the detection safeguards afforded by the Key Detector system, we would be reluctant to conduct computer skills assessment exams using the distance format since the potential for cheating in a distance learning format is high.

## Conclusions

*In my experience the most common excuse given by a student caught cheating is that other students were cheating and that the teacher didn't seem to care, at least not enough to do anything to prevent or stop cheating. (p. 98)*

From *Teaching Tips: A Guidebook for the Beginning College Teacher*, by Wilbert McKeachie (1986)

As suggested by the foregoing quote from Wilbert McKeachie's classic book on college teaching, a teacher can play an important role in dealing with the problem of cheating. However, it is not enough for a teacher to simply have the desire to reduce cheating. It is also important for a teacher to have a game plan for minimizing the opportunities for cheating. By making it more difficult

to cheat, an instructor may help to motivate students to study and learn the material instead of trying to “beat the system.” Also, the teacher will provide a “level playing field” for all students in which the students are evaluated based on what they know about the course material. In this manuscript, we have provided a description of an approach that we have developed to address a problem that we encountered with cheating in our computer skills courses. Based on our experiences, the approach has been an effective way to curb cheating on computer skills exam assignments. Students do not cheat on these examinations since they know that it is virtually impossible to cheat – and they know that the teachers care.

## References

- Baird, J.S. “Current trends in college cheating,” *Psychology in the Schools* (17:4), October 1980, pp. 515-522.
- Davis, S.F., Grover, C.A., Becker, A.H., and McGregor, L.N. “Academic dishonesty: Prevalence, determinants, techniques, and punishments,” *Teaching in Psychology* (19:1), February 1992, pp. 16-20.
- Joy, M. and M. Luck. “Plagiarism in programming assignments,” *IEEE Transactions on Education* (42:2), May 1999, pp. 129-133.
- Kennedy, K., Nowak, S., Raghuraman, R. Thomas, J. and Davis, S.F. “Academic dishonesty and distance learning: Student and faculty views,” *College Student Journal* (33:2), June 2000, pp. 309-314.
- Kleiner, C. and Lord, M. “The cheating game,” *US News and World Report* (127:20), November 22, 1999, pp. 54-63.
- McCabe, D.L., Trevino, L. “What we know about cheating in college,” *Change* (28:1), January/February 1996, pp. 28-33.
- McKeachie, W.J. *Teaching tips: A guidebook for the beginning college teacher*, DC Heath and Company, Lexington, KY, 1986.
- Newstead, S., Franklyn-Stokes, A., Armstead, P. “Individual differences in student cheating,” *Journal of Educational Psychology* (88:2), June 1996, pp. 229-241.
- Webster's New Twentieth Century Dictionary*, Second Edition, Simon & Schuster, New York, NY, 1983.
- Whale, G. “Identification of program similarity in large populations,” *The Computer Journal* (33:2), 1990, pp. 140-146.