Undergraduate Research in Computer Forensics

Mary DeVito, Ph.D.
East Stroudsburg University
Computer Science Department
200 Prospect Street
East Stroudsburg, PA 18301
570-422-3666
mdevito@esu.edu

Christine Hofmeister, Ph.D.
East Stroudsburg University
Computer Science Department
200 Prospect Street
East Stroudsburg, PA 18301
570-422-3666
chofmeister@esu.edu

Michael Jochen, Ph.D.
East Stroudsburg University
Computer Science Department
200 Prospect Street
East Stroudsburg, PA 18301
570-422-3666
mjochen@esu.edu

N. Paul Schembari, Ph.D.
East Stroudsburg University
Computer Science Department
200 Prospect Street
East Stroudsburg, PA 18301
570-422-3666
nschembari@esu.edu

ABSTRACT
We describe a research program in cybercrime and computer forensics which relies heavily on research by undergraduates. The goals of this program are to complete projects as defined by clients, but also to allow students to experience the frustrations and enjoyment of solving real-world problems. Furthermore, this experiential approach builds a rich educational environment for students where they must “work for a boss” (faculty member) and with “coworkers” (other students) in a group effort. Overall, students have found the experiences to be beneficial, and we typically have more applicants for the positions than are possible to fill. We describe three of the actual projects used and illustrate our lessons learned.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Computer Science Education

General Terms
Security, Human Factors

Keywords
Computer Security, Computer Forensics, Research Experiences for Undergraduates, Education, Internships

1. INTRODUCTION
The National Science Foundation's Research Experiences for Undergraduates (REU) program is well known [1]. This program began in 1986 as an outgrowth of the NSF Undergraduate Research Participation program [2]. The goals and outcomes of REU are in alignment with almost every Bachelor granting college and university. For example, as stated in the program solicitation, providing research experiences to undergraduates is one of the best methods for attracting and retaining students.

In the East Stroudsburg University of Pennsylvania (ESU) Computer Science Department, the recognition that providing research experiences is important for an undergraduate education occurred very early. The department was founded in 1986, and before this while the Computer Science program was still part of the Mathematics Department, faculty and students were performing joint research. That is, undergraduate research has been an active part of the ESU Computer Science curriculum since its inception. In fact, one of the authors who is an ESU Computer Science faculty member, participated in faculty-led research projects as a student while completing her Bachelor's and Master's degrees in the department. It should be noted that while ESU has used REU as a model, we have not received funding from this program.

Since 2006, ESU has received funding for research in cybercrime and computer forensics from the US Departments of Justice and Education. With this funding, multiple students have participated in research projects in the detection of cybercrime such as website defacement, intellectual property theft, fraud, and terrorism. We have also had students research in projects involving digital steganography and watermarking, and data categorization related to evidence handling.

1.1 ESU and Its Computing Programs
ESU is one of the fourteen universities of the Pennsylvania State System of Higher Education [3]. ESU is a National Center for Academic Excellence in Information Assurance Education [4] as designated by the US National Security Agency (NSA) and
Department of Homeland Security, and has held this designation since 2003. The ESU Computer Science Department offers two Bachelor's degrees – one in Computer Science and the other in Computer Security. In fact, the ESU BS in Computer Security was one of the first full Bachelor's degrees (not a concentration or certificate) in computer security in the nation. The department currently enrolls approximately 120 students in the two majors.

The first two years of the ESU Computer Science and Security programs are identical – students complete courses in introductory and assembly programming, data structures, computer organization, and operating systems. In the third and fourth years, students who pursue the degree in Computer Science specialize in software development by completing courses in programming languages and software engineering along with a course in ethical issues and multiple electives. Students who pursue the degree in Computer Security instead complete courses in databases, networking, security engineering, risk analysis, cryptography, network security, and legal issues. Many students who choose to major in one of these areas will actually complete both majors. For both Bachelor degrees, the capstone course is the internship.

1.2 ESU Internship Courses in Computer Science and Security

Students who pursue either the Bachelor's degree in Computer Science or Computer Security are required to complete an internship. Students who major in both areas are required to complete two internships. A single internship consists of 180 hours of work experience, typically at the junior or senior level. The internship is considered an actual course, and students earn 1 credit for each 60 hours completed with a maximum of 12 credits (typically as a “co-op”), and three credits required in each major.

To complete this requirement, students will often find a prospective employer and participate in the employer's internship program. ESU students have completed internships at large corporations such as Merck Pharmaceuticals, at federal facilities such as NSA or Picatinny Arsenal, in state and local government, and at multiple small companies across a variety of industries. A second method in which students have completed their required internship is in the participation of faculty-led research.

Some of the goals of the Computer Security internship “course” are as follows:

- Provide an opportunity to experience the importance and relevance of ideas learned in classroom work.
- Broaden the experience of the student by exposure to and skills in working with new computer systems, new applications or new developments in computer security.
- Give the student the experience of working within a group/team.

Since faculty-led research typically involves software development and/or experimentation, with students working in teams, we can easily see that either working with an employer or on faculty-led research allows students to meet the course goals.

1.3 The Cybercrime and Computer Forensics Institute

Since 2006, ESU has been part of the Cybercrime and Computer Forensics Institute (CCFI), a partnership of ESU, Drexel University, and Rider University. With CCFI funding, ESU has researched in multiple areas including:

- Cyber Fraud Data Analysis
- X-Engine – Advanced Search for Cybercrime Investigations
- Digital Watermarking and the Detection of Digital Steganography

The basic problem of the Cyber Fraud Data Analysis project was provided by the client, an office of the state of New Jersey, to Rider University. The client wanted the ability to collect information about insurance claims in a database, and then analyze these claims for the possibility of fraud. ESU faculty and students have worked on this project in Microsoft Access as requested by the client, as well as SQL Server with a C# front end. Second, the X-Engine is the creation of Dr. William Amadio of Rider University for the categorization of data. The CCFI application to cybercrime has been in the analysis of a large collection of files such as a set of email belonging to one person so that “unimportant” email can be ignored. Also at ESU, we have taken the product which had been built using third party software and MATLAB code and created our own implementation in C++ and C#.

Finally, in the steganography project, ESU students and faculty have performed steganalysis (undergraduate students as well as in Master's theses) for Discrete Cosine Transformation steganography, as well as against video based steganography. These research teams have also created a framework so that new steganography algorithms can be easily tested.

In the sections that follow, we describe our experiences with these projects. Since 2003, we have involved approximately 50 students in our research, and each project has had a group of approximately 5 students per year. At the beginning of each academic year, the faculty meet with select students (usually third-year, fourth-year, and graduate students in the Computer Security major), and we allow the students to choose their projects based on their interests. Also, we will show that the projects vary from being entirely client driven (Cyber Fraud Data Analysis) to more research based (Steganography Detection and Watermarking), to something in between (X-Engine). Hence, it is possible to work with a variety of projects in this experiential learning approach as long as there is strong interest from students and faculty. We conclude with “lessons learned” and suggestions for faculty who are interested in leading similar research programs.

2. CYBER FRAUD DATA ANALYSIS

As is the case in the many levels of government, the state of New Jersey has multiple professionals devoted to the investigation of insurance fraud. In the completion of training with Rider University, some of these NJ professionals began discussions with faculty on the creation of a database which would allow for easier identification of insurance fraud. Through the Cybercrime and Computer Forensics Institute, Rider University faculty outlined the problem to ESU faculty and a partnership was formed to help create a solution. Faculty from both institutions and students from ESU have worked on various parts of the project since 2006.

2.1 Client Meetings and Database Design

In an effort for student education, one of the goals was to have students interact with real customers, where their initial task was to capture user requirements. In preparation for our first client meeting we asked students to read about medical fraud and learn some of the terminology involved in this area of expertise. We then meet with the students so that they could tell us what they
had learned and answer any questions they might have. Although this seems like a trivial task, students must realize that they cannot enter a meeting with a customer unless they are prepared to have an intelligent conversation with them using their terminology. In addition, customers will expect service providers, which the students were, to speak and understand their language. We explained to the students that the first meeting with customers is critical because it will either build confidence in the customer about their ability, or, in a nutshell, the customer will feel that this effort does not have much of a chance at success and they will not want to dedicate their time, a vital requirement for any successful project. In addition we reminded students, as we do in class, that a majority of the work that they will eventually perform at their job will often be in domains (medical, financial, manufacturing, etc.) of which they are not familiar, and that a first step will always be to research that domain and become familiar with their terms, requirements and processes.

During the initial meeting with New Jersey state fraud investigators, the students asked excellent questions concerning the process of health insurance. In addition the students asked how the data was captured, how they tracked evidence and what where the triggers within the data that indicated potential fraud. The potential end users placed major programming constraints on the effort: all programming needed to be done in an old version of Microsoft Office Access and Visual Basic. The version of Access was so old we actually had a problem finding it. They placed these constraints for very good reasons: (1) they were familiar with Access and Basic, and (2) they only had the older version of Access available at their work site and mentioned that there was no discussion of an upgrade.

Based on the information the students gathered from potential users along with sample health insurance forms, they began to design the initial relational database model in Third Normal Form (3NF). The students have a tendency, as they do in class, to want to move directly to programming the problem. Instead we had students enter data into the model to check for anomalies, and any data redundancies that were discovered were removed from the model. The model in a high level form was then brought back to the client for validation. We did have an advantage in this case, in that the end users had some experience in creating their own small databases in Access, and therefore they were very familiar with tables and relationships. This was very helpful; they were able to work through the model with the students and help make changes, especially in the area of gathering evidence, which was much more involved than we originally thought, since this evidence may need to be presented in court. The evidence must be treated carefully since, as is well known, a defense attorney will try invalidate it by asking how evidence was gathered, when was it gathered, etc. Based on the clients' expert knowledge in this area, the evidence portion of the model was redesigned, which involved major changes not just with the evidence tables but also with the relationships from other portions of the model to the evidence. The student experience with the potential users was really fantastic; it was truly a roll-up your sleeves session that took most of a day. After this excellent experience we had an agreed upon model, and the students were ready to code.

2.2 Coding

The implementation requirement of utilizing only an old version of Microsoft Access and Visual Basic became a bigger issue then we had originally thought, but the students worked through it. The students began with the basics, the ability to enter/edit/search data (doctors, medical offices, insurance companies, businesses, insured employees and their family members, addresses, insurance claim forms, evidence, etc.), generate reports (claims by patient, claims by doctor, claims by facility, claims by Current Procedural Terminology (CPT) code, claims by diagnosis, etc.) and data maintenance. Throughout the stages of implementation the students met or conferredenced with the potential customers. The clients tested the system and made helpful suggestions for the prototype to make it easier for end-users.

In 2008, the tool was updated so that more queries and reports could be used to help better detect instances of fraud. A second improvement was the implementation of the system as a Windows application using SQL Server for the database backend and C# forms for the front end. The original Access file was converted to an SQL database and improvement were made including adding relationships, adding new tables, and deleting unused or unnecessary tables. The forms and navigation system was created for the new application using the existing Access implementation as a template. Multiple forms were completed and all navigation was put into place. Furthermore, the team worked with one of the investigative professionals to identify and prioritize fifty issues where upgrading was needed.

Recently, we have worked on making improvements to the C# version of the database. There are 3 limitations to the current system:

1. The system does not integrate “TIN runs” (Tax Identification Number runs). The TIN represents the payee of the claim form, and the TIN runs involve the input of data files that contain multiple insurance claim forms from the insurance companies. Each claim would have to be parsed and correctly placed into multiple tables and created with foreign key and primary key relationships.

2. The system does not contain database triggers that would search the data for potential fraud and notify the user. Currently the user can generate reports that he/she would then have to visually analyze for fraud.

3. The system does not integrate complicated car accident insurance claim forms.

These are the improvements that we will try to accomplish with our students during the current funding cycle, as time permits.

2.3 Applications to the Classroom

The process described above is exactly the one that we now use in our Database Systems class. Representatives of non-profit organizations come to the classroom five times throughout the semester to discuss (1) the requirements, (2) the initial database model, (3) the final database model, (4) a prototype implementation, and (5) the final implementation, respectively. The students work in teams throughout the semester, and at the end of the semester the non-profit chooses the database developed by the student team that best fits their needs. The students have developed database systems for nonprofit groups like Pocono Services for Families and Children, The SHELBY Project, and Women’s Resources of Monroe County. These joint efforts have turned out to be a win-win situation for everyone involved. The students implement “real” systems and the non-profit organizations receive software solutions that they would not have been able to afford.
3. X-ENGINE – ADVANCED SEARCH FOR CYBERCRIME INVESTIGATIONS

The X-Engine system is designed to support cybercrime investigators by automatically searching large volumes of text documents in order to identify the key concepts contained in the documents. This concept searching differs from keyword searching in several important ways.

First, for a keyword search, the set of keywords or search terms describe what is being sought, so they must be known before the search. For the X-Engine’s search, the concepts are not known before performing the search. Instead the goal of the search is to identify the main concepts contained in the documents.

The second difference is that keyword searches cannot detect homographs such as “lead,” which could mean either the metal or the verb. The X-Engine’s search forms concepts based on words that co-occur. Typically the two meanings of “lead” will appear in conjunction with different sets of words, so the X-Engine will form different concepts for the different meanings.

Finally, the X-Engine can even form different concepts for a single meaning of a word. An example is if the word “car” appears in the context of a “car show” and also appears in the context of “fencing stolen cars”.

The concept of the X-Engine came from Dr. William Amadio of Rider University. He saw that non-negative matrix factorization (NMF) was being used to extract features from large datasets. Berry and Browne report on its use in analyzing 1.5 million email messages obtained by the Federal Energy Regulatory Commission during the Enron investigation. Examples of concepts they found are listed in Table 1 [5].

<table>
<thead>
<tr>
<th>Topic Description</th>
<th>Dominant Terms in Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enron collapse</td>
<td>partnership, fastow, shares, sec, stock, shareholder, investors, equity, lay</td>
</tr>
<tr>
<td>Fantasy football</td>
<td>game, wr, qb, play, rb, season, injury, updated, fantasy, image</td>
</tr>
<tr>
<td>Enron collapse</td>
<td>dow, debt, reserved, wall, copyright, jones, cents, analysts, reuters, spokesman</td>
</tr>
<tr>
<td>Texas Longhorn football newsletter</td>
<td>UT, orange, longhorn, texas, true, truorange, recruiting, oklahoma, defensive</td>
</tr>
</tbody>
</table>

The results of the NMF are a set of concepts found in the documents. Each concept is numbered, and it contains the set of words that co-occur, such as the ones shown in the right column of Table 1. The topic descriptions are assigned by a person.

In the context of cybercrime investigation, the investigator would run the X-Engine, specifying the documents to be searched. Next the investigator would look at the results and identify the topics of interest. The documents that do not contain any topics of interest can then be eliminated from consideration.

Next the investigator may wish to peruse the remaining documents. This process could be repeated using only the documents of interest, in order to refine the topics. When using NMF the investigator specifies the number of topics to be identified, and this parameter can be changed for each run.

3.1 The X-Engine Design

The X-Engine design is decomposed into three main parts: the initial parsing system, the mathematical engine that processes the information gathered from the parser, and a reporting module.

The parser takes the input set of documents and produces an internal representation of the documents and words in matrix form. The rows in the matrix come from the set of words used in all documents. Only the root (stem) of each word is used, and generic words such as “maybe” and “meanwhile” are ignored. Also ignored are the words that occur most frequently and least frequently.

The columns of the matrix are the documents, so each matrix entry represents the “strength” of that word in that document. If the word does not occur in that document, the entry is 0. The higher the value of the entry, the more times it occurs in the document. Note that these entries are not a strict count of the number of occurrences; the values are normalized.

Once the matrix has been prepared, the NMF algorithm produces a factorization of the original matrix. The original matrix, $V$, is expected to be quite large (for a large set of documents). If there are $n$ words and $m$ documents, its dimensions are $n \times m$. Recall that the number of topics to be identified is configured by the investigator. Normally we want this number, $k$, to be much smaller than either $n$ or $m$. The NMF produces two new matrices, $W$ and $H$, such that $V \approx W \times H$. Note that the dimensions of $W$ are $n \times k$, and the dimensions of $H$ are $k \times m$, so both $W$ and $H$ are much smaller than $V$. The matrix $W$ identifies the words in each of the $k$ topics or concepts, and the $H$ matrix identifies which document contains which of the $k$ topics. All of the NMF algorithms are approximation algorithms, and the factorization is repeated either a set number of times or until the solution converges within a small error value.

Finally, the report module displays the results. The results are prepared in Excel-compatible format. They are meant to be used for testing the X-Engine and for input to a separate browsing system. This second system allows the investigator to provide topic descriptions, filter the set of documents using the topics, and examine the use of a topic in a document. This browsing system is developed at Rider University and is not part of the ESU X-Engine system.

3.2 X-Engine Development History

The project was designed as a multi-year project with a different team of students each year. At the start of the project Dr. Amadio provided a MATLAB prototype of the X-Engine algorithm including the use of a stemming algorithm. Since 2006 the teams of students have completed the following development activities:

- Gather the system requirements and record them in a requirements specification document, including screen shots, error conditions, and a definition of the full scope of the project.
- Prepare test data for checking correctness.
- Develop a GUI for configuring and running the system.
- Determine the appropriate container classes to use in the implementation.
- Investigate and incorporate an existing stemming algorithm [6].
- Implement in C++ a simple approach for removing the most-frequently and least-frequently used words in the dataset.
The students had varying levels of familiarity with software engineering concepts, so these topics were explained as the need arose. Visual Studio was used as the development platform. The first few groups used C++ as the implementation language, which is the language of instruction in several of their required courses. Later groups wished to use the internship experience to help them learn C#, which is not covered in their required courses. The faculty identified areas where students needed guidance and constructed lectures on those topics, which included matrix mathematics and NMF principles.

The students had varying levels of familiarity with software engineering concepts, so these topics were explained as the need arose. Visual Studio was used as the development platform. The first few groups used C++ as the implementation language, which is the language of instruction in several of their required courses. Later groups wished to use the internship experience to help them learn C#, which is not covered in their required courses. The first few groups of students handled their own configuration management, and then all forensics projects were put under Subversion [9].

Each of the three projects described in this document meet our internship course outcomes outlined in Section 1.2 above. As an illustration of this fact, the outcomes of the internship course and the student experiences with the X-Engine project are correlated below.

**Outcome 1:** Provide an opportunity to experience the importance and relevance of ideas learned in classroom work.
- Gain experience working on a larger system.
- Understand and maintain prior code (except the first group of students).
- Produce a robust specification.
- Provide adequate user documentation.

**Outcome 2:** Broaden the experience of the student by exposure to and skills in working with new computer systems, new applications or new developments in computer security.
- Understand how matrix factorization can identify features in a dataset.
- Learn NMF algorithms.
- Understand some natural language processing (parsing, stemming, cleaning of word set).
- Learn C# (later groups of students).
- Develop a form-based GUI.
- Identify and implement features of a good user interface.
- Gain experience handling large datasets.
- Apply targeted performance optimizations.

**Outcome 3:** Give the student the experience of working within a group/team.
- Work with other students on the X-Engine.

Clearly, this project is giving students a great experience in research.

### 4. DIGITAL WATERMARKING AND DETECTION OF DIGITAL STEGANOGRAPHY

Student workers in the Digital Watermarking and Detection of Digital Steganography project were exposed to the research process through the design and implementation of systems to conduct digital steganography and digital steganalysis. The research goals of this project are three-fold:

1. Design a system to embed a digital watermark that is robust to common methods of attack
2. Design a system to detect the presence of hidden content in otherwise innocuous cover media
3. Evaluate the effectiveness of the two system implementations.

This project also has the following educational goals for student participants:

1. Familiarize the students with the research process
2. Introduce students to state of the art techniques for information hiding in digital content
3. Introduce students to state of the art techniques for detection techniques to find hidden content
4. Strengthen student teamwork and programming skills in a group environment.

#### 4.1 Background on Steganography and Steganalysis

Steganography is a form of information hiding [11]. Loosely translated from the Greek, steganography means “covered writing”. By utilizing steganographic techniques, a sender and receiver can exchange a secret message with a very low probability of detection that the communication even took place. One of the earliest recorded uses of steganography is that of Histiaeus from the time of the ancient Greeks [12]. Histiaeus communicated covertly with Aristagoras to mount a revolt against the Persians. The communication took place by a message in a
tattoo. Histiaeus shaved the head of his most trusted slave and tattooed a message on it. After letting the slave’s hair grow back, the message was essentially hidden from view. The slave was sent to Aristogoras. Aristogoras was instructed to shave the slave’s head to read the message in the tattoo.

While this example is a rather low-tech form of communication, modern-day steganography typically manipulates statistical randomness or noise in cover media, like digital images, audio, video, or other electronic content. For example, in Least Significant Bit (LSB) steganography on digital imagery, the least significant bit for pixels in the image can be replaced with each bit of the message that the sender desires to communicate. The message recipient can reconstruct the message by reassembling the LSBs of the image into the original bit-stream for the message.

Modern day applications for steganography include covert communication and intellectual property protection (IPP). To achieve the goal of IPP, we can embed a robust digital watermark as a means to assert copyright ownership or we can embed a unique serial number to accomplish a goal of traitor tracing. To facilitate claims of copyright ownership, a digital watermark must be robust against many forms of attack. For example, the watermark must be difficult to remove from the cover media and must be difficult to modify in any way (i.e., overwrite or alter in any way). Digital watermarking can differ from steganography in that the watermark may or may not be detectable in the cover media. The main idea behind watermarking is to embed some mark within some file or media such that the presence of the mark supports one’s claim to ownership of the original file or media.

Traitor tracing typically employs a more traditional steganographic approach, in that the embedded content is hidden or obscured from casual observance. The goal of traitor tracing is to identify individuals that are releasing content that they are not authorized to release. For example, company A sells a version of some software to company B. Before company A delivers the software to company B, company A embeds a serial number that is unique to the version of software sold to company B. If company A finds any other versions of the software with the same serial number that is not in the custody of company B (e.g., available for sale/download from a third party website), then company A can suspect company B is improperly distributing the software.

Steganalysis attempts to find, and if possible reveal, hidden steganographic content [13]. Steganalysis is complicated by the fact that there exists many different steganographic methods for information hiding. As such, much current research attempts to find new approaches for steganalytic methods.

4.2 Student Work in Steganography and Steganalysis

Student researchers began their investigations by reading current research in the areas of steganography, watermarking, and steganalysis for a variety of content formats (e.g., digital picture files and video files). The students then began working in small groups to design and implement tools to perform various tasks for the project: generate steganographic content and conduct steganalysis and other techniques. The students were given the opportunity to identify which portions of the project held the most interest for them. This information was used to assign individual responsibilities within the project.

When possible, more senior students were utilized to facilitate the development of a mentor relationship between more advanced students and those students with less experience. In one particular instance, a master’s student served as a group leader/mentor for undergraduate workers on a steganalysis project.

Two main projects were formed: a project on still image steganography and watermarking, and a project on video steganography and steganalysis. For the students working on the still image project, the goals are to develop a modular tool that will support various embedding techniques and various content forms. Students began by implementing simple embedding algorithms (e.g., LSB), and are currently working on more complicated embedding schemes (to include keyed embedding and other techniques). The development process for the embedding tool is being used to provide students with a practical knowledge of the relevant issues, techniques, and background required to explore the development of a new digital watermarking technique. During the implementation process, the students are encouraged to perform incremental improvements to existing techniques and observe the resulting effect.

Students working on the video steganalysis project are attempting to identify hidden content in full motion video files. To investigate this problem the team extended an existing embedding technique that manipulates motion vectors that describe differences between frames of video content [14]. The extensions to the original algorithm include adding a finer granularity of control over how aggressive the embedding operation is, and some subtleties in how the motion vectors are manipulated to encode the secret content. After video test content files are created with the new tool, the students are conducting steganalysis by utilizing a Support Vector Machine (SVM) to examine various feature sets for the video. These feature sets are combined and input to the SVM to determine the likelihood that the video contains steganographic content. Feature sets are being considered among both spatial and temporal domains as incremental extensions to existing techniques [15].

4.3 Observations

Students received additional readings and group discussion on a number of topics to be able to understand and utilize current techniques for information hiding and detection. For example, knowledge of Discrete Cosine Transformations was required in order to analyze and manipulate JPEG images.

The students were given the opportunity to choose the implementation language for each system. Since the majority of students were most familiar with Java, both systems are implemented in the Java language. Additional libraries were imported to enable image/video analysis and manipulation. For example, the FFmpeg [16] library enabled the video steganalysis project to work directly on mpeg video files.

Each steganography project is in the process of generating evaluation data. The video steganalysis project is attempting to identify an optimal set of features to facilitate efficient identification of steganographic content. The still image steganography and watermarking project is exploring features that would be robust against known methods of steganalysis.
5. CONCLUSION

When faculty are able to provide research experiences to their students, in this case undergraduates, all parties benefit. The students have the ability to apply knowledge gained in the classroom, or work through other results that they have not yet learned, and work on current unanswered questions. Many students and educators feel that this type of learning is more beneficial [17]. The faculty also gain a greater understanding of their research topic or might expand their knowledge to broader topics.

Because of these benefits, faculty may have an interest is starting such a program, or improving a program they already have in place. The following are our suggestions and lessons learned based on our experiences.

(1) The faculty should decide on the ultimate goal of their program. Is the program being designed to advance faculty research or is it to improve student education? The answer to this question may make the design and effort quite different.

(2) Student preparation is very important. For our projects, all students needed a review of the literature, and we also used client meetings where appropriate. We sometimes had students read articles on their own and sometimes had lecture sessions where literature was summarized. Finding adequate sources may be a problem, especially for undergraduates. However, when sources good for undergraduate reading are not available, a faculty member could create such sources, and this could lead to publications.

(3) Both client-based and purely research-based problems (or something in between) can be used as the underlying project. Our examples above fit into each of these categories. The benefit of a client-based project is that the students will find this a true “real world problem” which is attractive for them. One negative of such a project is that the client may have a timeline or certain expectations that the faculty member or students do not like. Of course, with a research based project, the faculty member is typically the client and hence controls all aspects.

(4) If a client-based project seems a better fit, non-profit groups may be possible clients. Because they are receiving a “finished” product for free, and have worked with donations in the past, their expectations may also be a bit lower. Also, since many universities now place a priority on service-based learning, a computing or security solution for a non-profit would meet this priority.

(5) If possible, give student assignments based on interest. In our multiple iterations, we usually poll students to see what projects may interest them. Also, specific tasks of each project are usually assigned based on interest. This has been possible because we have found funding for multiple projects.

(6) If desired, the project can be treated as a “real job”. That is, the faculty can be thought of as employers while the students are employees. This relationship gives the students a simulated work experience which may be a goal.

(7) Expect slow results from students. We must keep in mind that we’re working with undergraduates with different motivations, multiple pressures, who are just learning to start their careers, can be overwhelmed by the scope of projects, etc. Students will produce results slowly, but over time deliverables will be met. Also, it is possible that students will lose interest or succumb to other pressures and end up not participating in the experience.

(8) One challenge has been the continuation of projects through the years as students graduate. We have found that we also need to temper our expectations when a new set of students enters the program. The students need time to understand the project, and we must also resist their temptation to start from scratch.

Overall, our undergraduate research program has proven quite successful. Faculty have had the chance to work on interesting problems and have enjoyed working with students. As students complete their internships, we provide a survey to gauge their opinions, and the data indicates that they are also very satisfied with these research experiences. We highly recommend educational security programs to incorporate such projects for their students.

6. ACKNOWLEDGMENTS

The projects described in this document have been supported by Grants 2010-DD-BX-K039 and 2008-DD-BX-K013 awarded by the National Institute of Justice, by Grant 2006-DD-BX-0471 awarded by the Bureau of Justice Assistance, and by Grant P16Z080055 awarded by the US Department of Education. Points of view or opinions in this document are those of the authors and do not represent the official position or policies of the United States Department of Justice or the US Department of Education.

7. REFERENCES


