WATA
A System for Written Authenticated though Anonymous Exams

Giampaolo Bella, Gianpiero Costantino and Salvatore Riccobene
Dipartimento di Matematica e Informatica, Università di Catania, Catania, Italy

Keywords: Exam, Open Competition, Authentication, Anonymity, Platform independence.

Abstract: University exams or open competitions raise significant security concerns because they are typically driven by strong interests. The candidate and the examiner may want to cheat on each other, or even coerce each other. While candidate authentication would address certain threats, his anonymity would address some other threats, but their conjugation appears to be contradictory. The classical approach to facing this security problem is weak as it poses significant trust upon the examiner. It is the double envelope, with a big envelope containing an anonymous exam sheet and a smaller envelope in turn sealing the candidate's identity. Surprisingly, there appears to be almost no computer-assisted solutions available.
WATA is a system, implemented in a portable software, for Written Authenticated though Anonymous exams. It protects the candidate and the examiner from each other by ensuring that the examiner corrects an authenticated though anonymous exam sheet. The candidate is the sole entity who can establish the link between his identity and his exam sheet, and normally has interest in safeguarding such a link. With various additional functionalities, WATA can be freely downloaded from the Internet for academics to try (Gianpiero, 2009).

1 INTRODUCTION

Taking an exam typically is a sensitive step for a candidate. Success may signify advancing towards a university degree or getting an important job. It is therefore an obvious requirement that both the candidate and the examiner behave honestly in their respective roles.

Nevertheless, various forms of cheating may occur. For example, the candidate may try to send someone else, who is particularly skilled and thus certain to pass, to sit for the exam on his behalf. In practice, the accomplice might be a friend or even an expert who is paid for his illegitimate task. *Authentication* can effectively withstand these threats, and in this context can be read as: the candidate really has the identity he claims on his exam sheet. The obvious enforcement of authentication during an exam is to have the examiner check that the candidate's ID matches what the candidate writes down on his exam sheet.

However, other forms of cheating are possible during an exam. The candidate might explicitly or implicitly influence an examiner who wants to mark the exam sheets fairly but happens to be a friend or relative of the candidate's. The candidate might even coerce the examiner. Also the examiner may want to act dishonestly by favouring or disfavouring an exam sheet for whatever reason, for example because he likes or dislikes the corresponding candidate. Remarkably, these forms of cheating are exacerbated by authentication, because the examiner is entitled to associate each exam sheet to a candidate. *Anonymity* can effectively withstand these threats, and in this context can be read as: the candidate can effectively conceal his identity on the exam sheet. It is clear that authentication and anonymity are, at least at the abstract level, the opposite of one another. Therefore, a security system to thwart cheating during an exam is not trivial to design because it cannot simply enforce one of the two goals, while enforcing both of them appears to be conceptually impossible.

The original motivation to tackle this problem dates back to 2003 for the inception of the *Computer Security* course towards the degree in Computer Science at our university. Not only was it was felt that a computer-assisted solution to the problem would pro-

Bella G., Costantino G. and Riccobene S. (2010).
WATA - A System for Written Authenticated though Anonymous Exams.
In *Proceedings of the 2nd International Conference on Computer Supported Education*, pages 132-137
DOI: 10.5220/0002797401320137
Copyright © SciTePress

vide fairness to both the examiner and the candidate, but it was also taken as a stimulating research challenge. In fact, it was somewhat surprising to find out that no attempts existed at tackling the problem using computers, as there was no public availability of the only related publication (Gray, 2003), which was just about to appear (§2). Our efforts to study the problem from a security perspective produced the first beta version of *WATA*, a system for *Written Authenticated Though Anonymous* exams, which has recently met a stable Java implementation, as we shall see below.

The gist of WATA is to mechanise in a software the classical method of the *double envelope*, which is often used during open competitions. As we detailed below, because this system classically has no computer support, it is easy for a dishonest examiner to subvert it. The main idea is to authenticate each exam sheet conventionally, but to keep it anonymous for the examiner until he terminates his marking. The only real limitation of WATA is against a scenario where the candidate and the examiner collude to subvert the exam. This seems, however, rather extreme, and may as such require dedicated enforcement.

This manuscript begins with a survey of the related work (§2). It then describes WATA (§3), illustrates its interface (§4) and outlines its implementation (§5). Finally, it terminates (§6).

2 RELATED WORK

As mentioned above, only two significant works are related to ours at present. One is the double paper envelope, which is a classical attempt at conjugating authentication and anonymity without computer support. Two envelopes are used for each candidate. A big one, which must be anonymous, will contain his exam sheet. A small one will contain only the candidate personal information. When the candidate finishes his exercises, the examiner puts the exam sheet inside the big envelop, whereas his personal information gets sealed in the small envelop — sealing-wax could be used to make the small envelop more tamperproof. During the marking phase, the examiner will open only the big envelop in order to mark an anonymous exam sheet. Only afterwards, when the examiner will have marked the exam sheet, he will associate the anonymous sheet with the personal information of the candidate by opening the small envelop. This terminates the marking phase, which is authenticated though anonymous only if the examiner is honest.

Clearly, a dishonest examiner could easily violate the anonymity of the exam sheet. If he is not supervised by someone else, or if all members of the examining committee agree, he/they can decide to open the small envelope beforehand, and dishonestly advantage or disadvantage some candidate. Various cheating scenarios could take place, especially if the small envelop is weakly sealed, because anyone in the examining committee could open and close it without any apparent tampering. As mentioned, sealing-wax would be of some help here.

Another related work is a software to submit coursework in an electronically from the student to the examiner while maintaining the student anonymous (Gray, 2003). Based on the client-server paradigm, the system lets the student use a pseudonym to submit his coursework. This process is mechanised through a JAVA (Microsystems, 1991) application, which keeps the identity of the student from the examiner. The latter is therefore deemed to mark the coursework fairly. Only when the course (that is, the actual teaching classes) terminates, will the examiner associate the pseudonym with the real identity. The system has weaknesses, as the author himself says; for example, a student could submit his coursework with his personal information in order to invalidate the anonymity of his work. This may happen either deliberately or indeliberately, that is by mistake. It remains questionable, however, whether this qualifies as a significant weakness in a realistic threat model where the student has interest in conforming to the rules in order to get through the exam.

The goals of this system are rather different from WATA's. Rather than assisting with coursework submission prior to the end of teaching, when anonymity is relaxed, WATA is targeted at exams whose marks must be produced in due time, as is the case of open competitions or final exams of university courses, for example. Another difference is that WATA is a tool for the examiner and, as such, does not need to be distributed.

3 OUR SOLUTION: WATA

In this Section, we advance a system called WATA. It is a system for Written Authenticated Though Anonymous exams, which offers an examiner full computer support to print, mark and notify the marks of exam sheets that are anonymous though still authenticated. Anonymity is relaxed only at the final phase of mark notification, that is after the actual marking phase. WATA is an application software that accomplishes its goals by printing exam sheets with a particular layout.

3.1 Authentication

To support authentication, we design what we name an *authentication token* (Fig.1), which essentially is a form for personal information.

Figure 1: Authentication token.

WATA prints this token on the exam sheet. At the beginning of the exam, the candidate is required to fill it in with his personal information (name, surname, student enrollment number), date and signature.

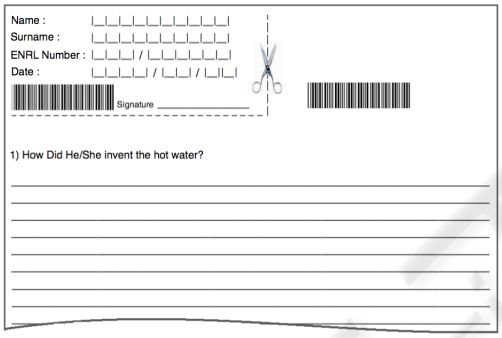

Figure 2: Top of an exam sheet.

The barcode is the most important feature of the token. Because it is unique, and is duplicated on the exam sheet, it associates the token to its sheet, as we shall see below (§3.2).

When the candidate has filled in the token, the examiner authenticates him by asking his identity document. If the personal information as shown on the candidate's ID matches what he has written on the token, the examiner puts a signature on the token to validate it. By doing so, he certifies the association between the candidate and his token. In addition, each token will be reinforced with a stamp of the examiner's organization in order to make it more tamper-proof. The examiner's signature and the organization's stamp are the only non-digital expedients that WATA adopts.

Finally, the candidate is required to cut out the token from the exam sheet and to keep it with him in a safe place. After this, the candidate, who is authenticated, can start answering the questions on an anonymous exam sheet.

3.2 The Barcode

The barcode is printed on the sheet in two different places, one is inside the token and the other one is on the top-right corner of the sheet. It therefore is the only connection between an authentication token, that is, between an identity, and an exam sheet. Of course, no barcode must be printed on more than one exam sheet, otherwise it would become impossible to later associate the candidate to his exam sheet.

WATA supports the reading of a barcode through a standard barcode scanner, so all operations described in the following can be significantly mechanised and speeded up. It is understood that this raises some trust in the examiner, who may in practice, at authentication time, memorise the barcode number corresponding to the candidate he wants to mark disonestly.

To guarantee uniqueness and unpredictability of a barcode, we use a random function that is detailed below (§4.1). During the computation of the number to represent as a barcode, we apply an extra sanity check that the last extracted number has not already been computed. This helps to circumvent any weaknesses of the implemented random function. Since the extracted numbers are stored in a database with the primary key in the number of the barcode, checking whether a number is present is not heavy.

When the generated barcodes are enough for the exam sheets, the system is ready to print out the barcode using the standard barcode codification *cod.39*. This codification covers uppercase letters, digits and special characters such as % but, for the sake of simplicity, WATA only encodes numbers as barcodes.

3.3 Anonymity

An exam sheet is anonymous when an examiner cannot go back during the marking phase to the candidate who completed that exam sheet. The authentication token of WATA declares the owner of the sheet, but without it the sheet remains anonymous. As explained before, the candidate must remove the token from the exam sheet after the examiner validates the token, but prior to submitting the sheet. In our practical use of WATA, the examiner enforced this by declaring the policy that failing to detach the authentication token would automatically invalidate the exam sheet: hardly anyone walked away without his token.

Fig.3 shows the exam sheet without the token. This is what the examiner sees while he is marking the exam sheet. It can be noticed that the sheet is anonymous.

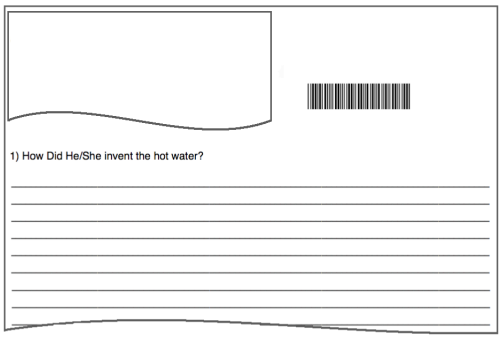

Figure 3: Exam sheet without Authentication Token.

3.4 The Database

The structure of the database underlying WATA is simple, as it consists of three tables: the barcode table, the question table and the mark table. The barcode table is very simple as it stores the generated barcodes. We shall see that it is particularly useful during the notification of marks to the candidates in order to detect fake barcodes the candidate might have built by himself (§4.3). The second table stores the questions for the exam sheets (Table 1).

Table 1: A question table.

QID	Question
1	How did he/she discover the hot water?
2	How far is the sun?
3	What was the best thing before sliced bread?
4	Can crop circles be square?

Each question is associated to a progressive number called "QID" (acronym of question identifier), while the questions belong to the "Question" field. In addition, WATA allows one to manage questions for different subjects, such as different university courses. The examiner merely needs to create one such table for each subject. Creating a table does not require particular skill; it can be done using the dedicated button that WATA provides (§4.4).

Table 2: A mark table.

S-Barcode	Mark	S-ID	Date	ENRL
5070920	Passed	1	2009/07/09	Empty
6483021	Failed	2	2009/07/09	Empty
1111220	Passed	3	2009/07/09	Empty
3628103	Passed	4	2009/07/09	Empty

The database also contains a mark table (Table 2). The examiner marks an exam sheet that is tokenless. So, he only stores the barcode of the sheet in "S-Barcode", the final mark (in "Mark") and the date of the exam in "Date". The examiner has no means to know the enrollment number "ENRL", which is only filled in later, during the notification phase (§4.3). Therefore, the mark table supports full auditing of the candidate's attempts at passing an exam by recording the mark for each date the candidate took an exam.

4 USING WATA

Using WATA is simplified by a friendly user interface (Fig.4). The main features can be accessed through the buttons on the left hand side. The remaining buttons interact with the underlying database.

Figure 4: Interface to WATA 2.0.

The "Add Questions" button initiates a straightforward interaction to expand the database of questions for an exam. The other buttons are explained in the sequel of this Section.

4.1 Print Exam Sheets

To prevent collaboration between candidates during an exam, WATA supports the printing of different questions for each exam sheet. Prior to printing, the questions are shuffled using a random function, which is the same as that used for barcode generation (§3.2). The random function combines the number of seconds elapsed from 1970-01-01 with the output of the standard random function provided by the programming language. The details can be observed from the omissis of the source code, which is detailed separately (§5.1).

Finally, the shuffled questions form a list to print questions from. The examiner can choose how many exam sheets and how many questions per exam sheet to print. In case the number of available questions is insufficient, WATA shuffles the questions again. This feature was added in order to allow an examining committee to print the desired number of exam sheets even if the number of questions available were rather small. Clearly, this would cause lower entropy.

```
// Closing the query to avoid error
DataModule1.getDataModule().getQueryDsQuestions().close();

// Setting the query
DataModule1.getDataModule().getQueryDsQuestions().setQuery(new
    com.borland.dx.
    sql.dataset.QueryDescriptor(DataModule1.getDataModule().dbwata,
        "Select Question
         From "+jCbTable.getSelectedItem()+
       " Order by rand((UNIX_TIMESTAMP()+"+Math.random()+"));"
        , null, true, Load.ALL));

// Opening object to perfom the query
DataModule1.getDataModule().getQueryDsQuestions().open();

// Adding the questions in a vector
do
{
   questions.add(DataModule1.getDataModule().getQueryDsQuestions().getString(0));
   i++;
}
while (DataModule1.getDataModule().getQueryDsQuestions().next() != false &&
      (i < neededQuestions));
```

Figure 5: Omissis of extraction of shuffled questions from a question table.

4.2 Mark Exam Sheets

When the examiner has finished to mark all exam sheets, he is ready to insert the marks in the system. Using the "Mark Exam Sheets" button, the examiner fills in the fields "S-Barcode", "Mark", "Date" inside the mark table seen above (Table 2). This is the only information that the examiner can gather from an anonymous exam sheet. The enrollment number ("ENRL") will be filled in later.

4.3 Notify Exam Sheets

When the examiner has finished marking the anonymous exam sheets and stored the marks in the system, he is ready to notify the marks to the candidates. However, he cannot link each exam sheet to its owner yet. This can be done only through the authentication tokens. So, the examiner asks each candidate to hand his token in. As the examiner scans in a barcode, the system matches it against two tables.

First, the live barcode is searched in the barcode table to prevent potential attempts at submitting a fake token with a fake barcode. Without this check, the candidate might have downloaded our system (Gianpiero, 2009), printed out an exam sheet with the proper layout, a barcode and some plausible questions at home, and answered the questions before the exam date. At the exam, he would have needed to swap the real exam sheet with his fake, terminated, one before being authenticated.

If this first check succeeds, the live barcode is searched in the mark table. When found, the examiner fills its "ENRL" field with the enrollment number read from the token. This is how WATA audits each candidate's attempts at passing an exam. Then, WATA opens up a popup window with the candidate's latest mark and full audit trail. The examiner can then tell the candidate his mark if the trail complies with the exam policy — for example, certain university policies prevent a candidate who fails an exam to try it again at the very next available exam date.

This process shows how the system guarantees anonymity of the candidate until the candidate himself gives the token back. As noted above, this feature protects both the candidate and the examiner from each other, but cannot help towards their collusion.

4.4 Other Features of WATA

WATA also implements the following useful facilities (Fig.4) to interact with the underlying database.

- *Show Table*: it visualizes the contents of a table, such as a question table or a mark table.
- *Create Table*: it creates an empty table, for example to support exams for more than one discipline.
- *Rename Table*: it edits a table name conveniently.
- *Import from File*: it imports questions from a formatted text file, such as those exported from Microsoft Access.

5 IMPLEMENTATION OF WATA

Portability is one of the most important requirements for a software. It should be possible to install the software on different computers regardless of the Operating System. To meet this requirement, we decided to write WATA in JAVA, which allows one to develop multiplatform-software and to run it through the Java Virtual Machine *(JVM)*.

Moreover, WATA uses MySQL (Sun-Microsystem, 1996) as DataBase Management

System *(DMBS)*. The implementation of WATA is lightweight and can therefore run on all current machines. The only requirements are a JAVA version (al least 1.4.2), and a MYSQL version (at least 5.0).

5.1 A Glimpse at the Code

While the full code of WATA can be obtained from the authors on demand, it may be useful to look more closely at a key fragment here.

Fig.5 shows the kernel of the method used to shuffle the questions from the question table (§4.1). In particular, rows 440 to 446 pertain to the SQL query, which shuffles the questions according to a random number taken as a parameter. It is interesting to observe how that random number is computed, as it is also used for barcode generation (§3.2). Our initial attempt to compute the random number was to use the number of seconds elapsed from the date 1970-01-01, which is output by UNIX_TIMESTAMP(). However, it was soon realised that this is insufficient because the query could be launched more than once in the same second and would then yield the same output. So, the output of the Math.random() function is added.

Then, the loop from rows 451 to 455 extracts the questions just shuffled and stores them in a dynamic vector. During the printing phase, WATA takes the questions from that vector and sends them out to the printer.

5.2 History and Future of WATA

The version of WATA described in this paper is WATA 2.0, which supersedes WATA 1.0.

WATA 1.0 was the first stable release. It was only available for Microsoft Windows (Microsoft, 1985), as it was written in Visual Basic and the database support was provided by Microsoft Access. The big drawback of WATA 1.0 was scalability. In fact, when the examiner tried to print a high number of exam sheets, the software became very slow because the algorithm used to shuffle the questions was inefficient. Today, with WATA 2.0, we overcome the scalability and portability limitations of WATA 1.0

As future extensions WATA, could introduce a functionality to separate the questions into difficulty classes. This would enable the system to let the examiner somewhat control the difficulty of the questions printed out on the exam sheets. For example, it would be possible to implement a requirement that each exam sheet contain one question of difficulty "medium" and two questions of difficulty "high".

Another feature that we plan to introduce in the future is some LaTeX(LaTeX, 1985) support. The current version of WATA allows to store only text questions, and so no mathematical symbols are allowed. A LaTeXplugin would offer full support to a large range of symbols.

6 CONCLUSIONS

University exams or public competitions may suffer various forms of cheating. Authentication and anonymity may alternatively help face some of them, but are inherently difficult to use jointly.

We advance a solution called WATA to face these problems in case of written tests. WATA supports the establishment of exam sessions that are authenticated though anonymous. The system relies on an authentication token, which contains the candidate's personal information, and a truly anonymous exam sheet. The authentication token is retained by the candidate who therefore is the sole entity who can protect his own anonymity. The token is mechanically linked to the right exam sheet by scanning a barcode.

WATA has been used in support of the final exam of the Computer Security course at our university ever since 2004. The results are encouraging: the overhead on the lecturer was only negligible, while the students gladly accepted the new system. WATA was described at the beginning of the course. At the end of the course some students felt that WATA had given them more freedom in their interaction with the lecturer, because it had dissipated their concerns to somewhat negatively influence the lecturer's evaluation of their exams.

The latest version of WATA has been made publicly available through the Internet these days (Gianpiero, 2009). WATA currently comes under a Creative Commons (Creative-Commons, 2001) — Attribution Non-Commercial No Derivatives — licence.

REFERENCES

Creative-Commons (2001). Creative commons.

Gianpiero, C. (2009). Wata.

Gray, D. (2003). Anonymous coursework submission. In *CompSysTech '03: Proceedings of the 4th international conference conference on Computer systems and technologies*, pages 556–561, New York, NY, USA. ACM.

LaTeX (1985). Latex project site.

Microsoft (1985). Microsoft windows.

Microsystems, S. (1991). The source for java technology.

Sun-Microsystem, M. (1996). Mysql.