THE USE OF PGP TO PROVIDE SECURE EMAIL DELIVERY OF CAA RESULTS

Simon Hatton, Alan Boyle, Sue Byrne, and Chris Wooff

The Use of PGP to Provide Secure Email Delivery of CAA Results

Simon Hatton, Chris Wooff, Sue Byrne
Computing Services Department
The University of Liverpool
Brownlow Hill
Liverpool
L69 3BX

Alan Boyle
Department of Earth Sciences
The University of Liverpool
Brownlow Street
Liverpool
L69 3GP

s.c.hatton@liverpool.ac.uk

Abstract

An important component of any assessment procedure is the security of results and authentication of the examinee. Unfortunately the use of regular email for the delivery of CAA (Computer Assisted Assessment) results is not immune from these problems as regular email suffers from a number of potential security flaws.

When an email is sent across the Internet it is transmitted in a readable text format. This means that if an unauthorised user managed to access the message whilst in transit or while stored on an email server then they could easily read the email, or even alter the content of the message. Additionally regular email offers no form of authentication. It is possible for a user to send an email but make it look as though somebody else actually sent it.

To prevent these problems a number of software packages have been developed, one such program is PGP (Pretty Good Privacy). PGP can encrypt and sign an email message before it is sent, therefore providing the following security:

- Prevent unauthorised users reading the message (privacy)
- Proof that the message has not been altered (integrity)
- Confirmation of the origin of the message (authentication)

At the University of Liverpool a JISC (Joint Information Systems Committee) funded pilot project was setup to investigate the use of PGP to provide secure email delivery of CAA results.

Introduction

The Department of Earth Sciences at the University of Liverpool has been running CAAs since 1995. The department predominantly uses the Authorware-based TRIADS (Tripartite Interactive Assessment Delivery System) engine to help in the production of CAAs. The TRIADS engine is extensible as it allows the user to insert their own Authorware code before and after the main assessment. Therefore our aim was to add our own code after an assessment to allow the use of PGP (Pretty Good Privacy) to sign and encrypt a student's answers before they are emailed to the examiner.

What is PGP?

PGP is an email and file security program that was originally developed and released in 1991 by Phil Zimmerman. The reason that PGP was chosen for this project instead of an alternative security software package was because PGP is the most widely used piece of software for sending secure data over the Internet.

How does PGP work?

PGP uses a technique called public-key cryptography to allow the exchange of secure data. In this system each user generates what is known as a public and private key, referred to as a PGP key-pair. These two keys are related, but they are not deducible from each other. It is the use of public and private keys that allows the exchange of secure data.

Once generated, the public and private keys are stored in 2 separate keyring files on the user's computer. The public keyring file stores the user's own public key, and a collection of other people's public keys. The private keyring contains the user's own private key.

The public key can be made available to as many people as possible; it is freely distributable. For users to be able to exchange secure data they must have access to other users' public keys.

The private key, however, is exactly what the name suggests, private, and it should be kept secret by its owner. As added protection the private key is encrypted using a passphrase. Whenever the private key needs to be used, the correct passphrase must be provided. Therefore, if someone were to obtain another person's private key it would be useless to them without the corresponding passphrase.

Encrypting and Decrypting a Message

Encrypting a message involves taking a readable text message, and converting it to an unreadable format. The message can then only be decrypted by the intended recipient, and therefore providing the following security benefit:

- Prevent unauthorised users reading the message (privacy)

For example, if Alice wanted to send an encrypted message to Bob, then Alice must first have access to Bob's public key.

- Alice writes a message for Bob
- PGP encrypts the message using Bob's public key
- The encrypted message is sent to Bob
- Upon arrival PGP decrypts the message using Bob's private key
- As Bob is the only person who knows the passphrase to unlock his private key then he is the only person who will be able to decrypt the message

Figure 1 shows an example of how PGP keys are used to encrypt/decrypt a message.

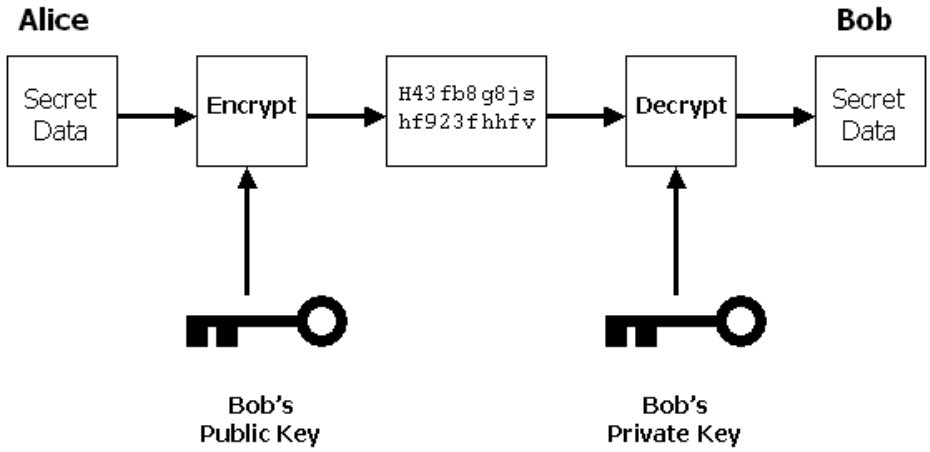

Figure 1 – Encrypting / Decrypting a Message

Signing and Verifying a Message

Signing a message provides the following security benefits:

- Proof that the message has not been altered (integrity)
- Confirmation of the origin of the message (authentication)

For example, if Alice wanted to send a signed message to Bob, then Bob must have access to Alice's public key.

- Alice writes a message for Bob
- PGP runs the message through a mathematical algorithm, and a unique value is generated based upon the message
- PGP encrypts this value using Alice's private key to create a digital signature
- PGP adds the digital signature to the bottom of the message
- The signed message is sent to Bob
- Upon arrival PGP decrypts the signature by using Alice's public key, as Alice is the only person who knows the passphrase to unlock her private key then only she could have created the signature
- Decrypting the signature revels the original value that was generated
- PGP again runs the message through a mathematical algorithm

- If the value generated by PGP this time is the same as the one sent in Alice's digital signature then Bob's knows that the message has not been altered.

Figure 2 shows an example of how PGP keys are used to sign/verify a message.

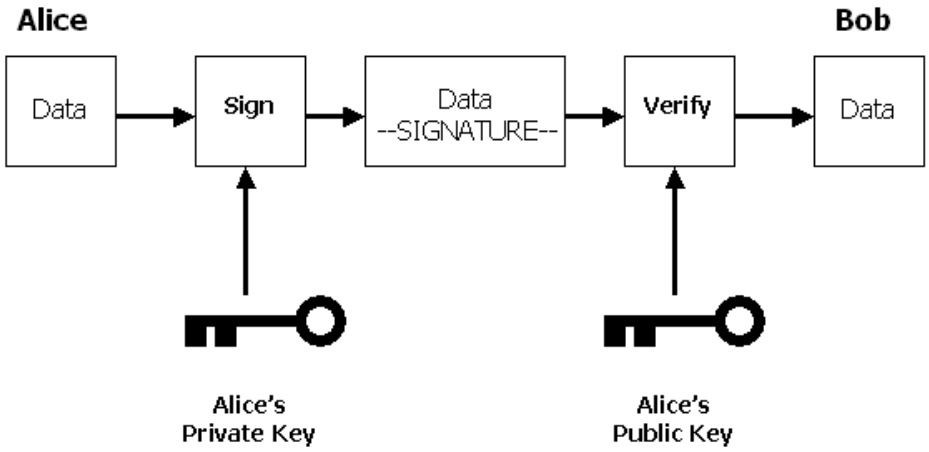

Figure 2 – Signing / Verifying a Message

Signing and Encrypting a Message

Signing and encrypting a message combines the security benefits of both techniques. When sending signed and encrypted messages they are signed first and then encrypted. Upon arrival the message is decrypted first and then the signature is verified.

In order for users to send signed and encrypted messages then they must have all created their own PGP key-pairs, the message sender must have access to the recipients public key, and the recipient must have access to the senders public key.

This concludes the basic introduction into the concepts of PGP, for more information please refer to The International PGP Homepage at http://www.pgpi.org.

Integrating PGP with TRIADS

The planned integration between PGP and TRIADS was broken down into 2 separate sections:

i) As highlighted in the previous section each user requires their own PGP key-pair in order to exchange secure data. Therefore for this project the examiner and the students required a public and private key.

ii) Exactly how would PGP be used from within a TRIADS assessment to ensure that a student's answers are encrypted and signed before they are emailed to the examiner?

Conditions
Any solutions that were achieved for this project had to satisfy the following conditions:

- The CAAs and PGP were run on Microsoft Windows 2000, which is the preferred operating system at the University of Liverpool.
- The students were to have no knowledge of, or control over, the involvement of PGP within the CAA. The encryption and signing of their answers was completely hidden and automatic.

The main reasons for not letting the students know about PGP also happen to make the operation a lot easier to implement. It would be problematic if the students were allowed to create their own PGP key-pairs themselves. For example, if after completing the assessment one of the students had forgotten the passphrase associated with their private key, then this would have presented serious difficulties. In addition, the less the student knows about the actual mechanism of results recording, the better the security.

PGP key-pair for the examiner

In order for the student's results to be encrypted the examiner must have their own PGP key-pair. This was setup by simply installing the PGP software onto the examiner's computer, and then generating a PGP key-pair for the examiner.

PGP key-pair for every student

In order for the student's answers to be signed the students required a PGP key-pair. However as the student's have no knowledge of PGP's involvement then the examiner created their key-pairs beforehand.

Two possible solutions for creating the students' keys were considered:

i) Create 1 global PGP key-pair that would be used to sign each student's results
ii) Create an individual PGP key-pair for each student taking the assessment

Solution i) is a lot easier to implement as only one PGP key-pair will have to be created for all of the students. However, this will reduce the authentication because examiners can then only prove that the answers were signed by one of the students, they cannot prove for certain which student sent it.

Solution ii) offers much better authentication because each student has their own PGP key-pair so the examiner can prove exactly which student sent which answers. This was the method we chose for signing the student's results. However, there were still a couple of problems: how is the correct PGP key-pair linked to the correct student, and how could all of these key-pairs be automatically generated.

To overcome the first problem of ensuring that the correct PGP key-pair is linked to the correct student we decided that the naming system used to

identify each student's key-pair would be based on the current system used to run CAAs at the University of Liverpool.

Currently whenever CAAs are run the students do not login to the PCs being used for the assessment with their own username/password. Instead the examiner uses a list of temporary usernames to login to each PC before the students arrive for the assessment. The usernames are all in a fixed format, such as, scese00, scese01, scese02 etc.

Using temporary usernames guards against the possibility of students storing relevant information under their own username. Also by linking the temporary usernames to seat numbers in the computer room, and getting each student to fill in a registration card before they start an assessment, allows the examiner to know where each student was sat.

Therefore we decided that the name associated with the PGP key-pairs for each of the students would be based on these temporary usernames. In addition the passphrase used to access the private key was the same as the username that identifies that key.

The following list explains the stages that were carried out so that each student will have their own PGP key, and that their results are signed correctly.

- The examiner was given a list of temporary usernames e.g. scese00 to scese99.
- The examiner created a public/private key for each temporary username. The passphrase for each private key was the same as the name associated with the key.
- Before the students arrive for the assessment the examiner logged into each computer with one of the temporary usernames.
- The students then began the assessment
- When each student completed the assessment TRIADS automatically used PGP to sign/encrypt their results using the correct private key, and passphrase. For example if the username scese07 was used by the examiner to logon to a PC, then the private key and passphrase, scese07, would be used to sign the student's answers on that PC.

Having decided upon a method of ensuring that the correct PGP key-pair would be used by the correct student then this brought us onto the next problem of how to generate the keys for every student.

Automatic PGP key-pair generation

The problem with creating a PGP key-pair for every student sitting a CAA is that there could be potentially hundreds of students taking the assessment, and therefore hundreds of PGP key-pairs have to be generated.

PGP comes with a key generation wizard that allows users to generate their own key-pair. However, this program is not suited for creating many key-pairs as the user has to step through a series of questions defining the name

associated with the key, what type of key they want, what their passphrase is, etc. If the examiner had to use the key generation wizard to create every PGP key-pair for all of the temporary usernames then this would be time consuming, and the chances of making a mistake somewhere in the naming of the keys would be high.

Therefore it was obvious that some form of automatic PGP key-pair generation would be extremely helpful. As a result we wrote a C++ program, called *genkey*, where the user only has to specify the first and last usernames that require a key-pair, for example scese00 and scese99. The program then creates a PGP key-pair for each username between scese00 and scese99.

The *genkey* program benefits from the fact that we have associated the temporary usernames with the keys. As the temporary usernames are all of a similar format the program can calculate how many PGP key-pairs are required by just specifying the first and last username.

Emailing answers to the examiner

Having come up with the method for creating the PGP keys for the examiner and all of the students, the second major stage was to work out how PGP would be executed from within a TRIADS assessment. The following list, along with Figure 3, gives an indication of the steps used to ensure that the student's answers were signed and encrypted before they were emailed to the examiner.

a) A student completes the TRIADS assessment
b) TRIADS saves their answers into a temporary file
c) PGP takes the answers file, signs it with the correct student's private key, and encrypts it with the examiners public key. This generates a new file.
d) TRIADS copies the contents of this new file into the body of an email message
e) TRIADS sends the email to the examiner
f) TRIADS deletes the temporary files
g) Upon receipt the examiner uses PGP to decrypt/verify the email to reveal each student's answers.

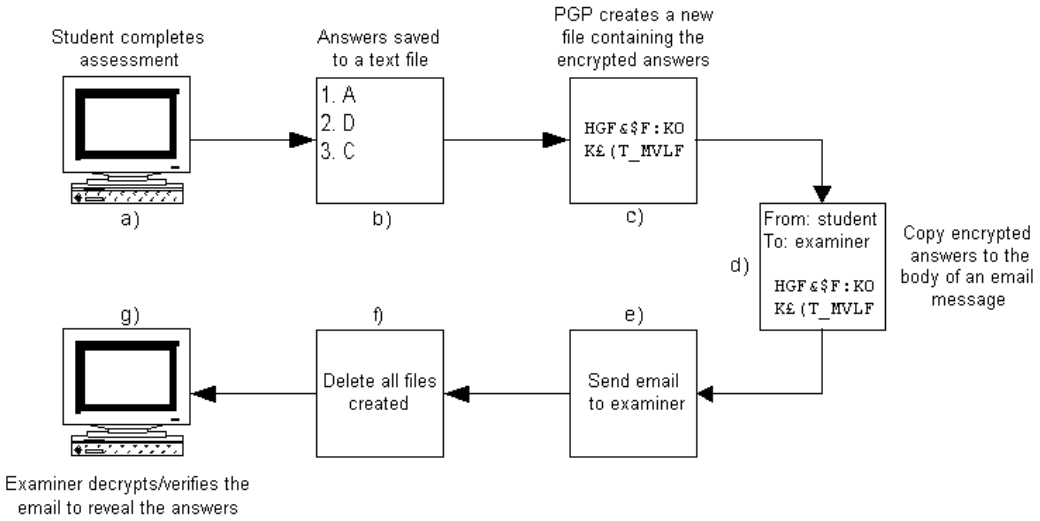

Figure 3 – Sending secure email results to the examiner

Figure 3 shows that PGP is only involved in steps (c) and (g). TRIADS carries out all of the other operations such as saving the answers to a file, emailing the answers to the examiner etc.

In order to run PGP, from within TRIADS, another C++ program was written, called *pgp_triads*. This program took the following input parameters:

- The name of the file containing the student's answers
- The name of the student's private key that will be used to sign the answers file
- The name of the examiner's public key that will be used to encrypt the answers file

Based on these parameters the *pgp_triads* program creates a new file containing the signed and encrypted answers.

Having written the *pgp_triads* program then the final stage was to add the necessary Authorware code at the end of the TRIADS assessment to save the student's answers into a file, launch the *pgp_triads* program with the correct parameters, copy the contents of the file created by *pgp_triads* into the body of an email message, and then send this message to the examiner. Upon receipt the examiner is then able to decrypt the message, and verify the signature to reveal the students answers.

Running PGP with a TRIADS assessment

On the 14th January 2002 the use of PGP to deliver secure email results for a CAA was trialled with a group of 80 students who were taking an exam comprising of 84 questions.

Preparations for the assessment

- PGP was installed on the examiners PC and a PGP key-pair was created.
- The program *genkey* was used to create the temporary PGP keys needed for the students. In total 100 PGP keys were created for the

usernames scese00 to scese99. These keys were only used once for this assessment
- The TRIADS assessment was setup to call the *pgp_triads* program, and then send the encrypted and signed results to the examiner

Running the assessment
- The examiner logged into every computer being used for the assessment, using the temporary usernames, before allowing any students into the examination room
- The students were then admitted to the computer room, and sat the assessment
- When each student completed the assessment the *pgp_triads* program ran completely hidden and automatically. TRIADS then emailed the encrypted and signed answers to the examiner

After the assessment
- The examiner received the encrypted and signed results as email messages from each student
- The examiner was then able to decrypt each message using their private key, and verify the signature using the student's public key. As 80 students took part in the assessment, and there are 3 emails sent from each student, this meant that the examiner had 240 messages to decrypt/verify. The first email simply summarises the score. The second contains all of the individual question scores together with the overall score. The third is a complete transcript of the individual student's responses, which is useful if the student claims to have done better than the computer recorded. For processing marks, the second email is the most appropriate. Decrypting these 80 emails, appending them together, importing them into a spreadsheet, sorting rows, and then deleting all extraneous material took approximately 10 minutes.

Conclusions

The use of regular email for the delivery of CAA results is vulnerable to a number of potential security flaws that exist with email. PGP is a secure email software tool that can solve these problems by encrypting and signing an email message. The advantages of this are:
- Prevent unauthorised users reading the message (privacy)
- Proof that the message has not been altered (integrity)
- Confirmation of the origin of the message (authentication)

At the University of Liverpool we have managed to successfully use PGP to provide secure email delivery of results to a CAA created using the TRIADS engine.

Acknowledgements

CIAD

The Centre for Interactive Assessment Development (CIAD), based at the University of Derby, are responsible for the development and maintenance of TRIADS. We would like to thank them for their help with the development of the TRIADS engine.

JISC

The work for this project was carried out as part of a one-year research grant funded by the Joint Information Systems Committee (JISC), and their support is greatly appreciated.