

Editorial

This is the first issue of **Login Lite** for 2001.

Login aims to present articles, reviews, practical classroom activities and correspondence from members of ECAWA, and the wider educational-computing community, and also to promote discussion regarding issues of concern to educators.

To these ends, we encourage input and contributions from teachers, policy-makers and technical experts in the various aspects of the applications of information technologies across the K-12 curricula, and at the post-secondary level, particularly with regard to teacher education.

ECAWA makes considerable input into discussions regarding policies being developed and considered at the various political and educational system levels in Western Australia.

ECAWA also offers its members affordable and very available practical and professional support in their use of information technologies in a wide range of learning environments.

Each year the voluntary ECAWA Committee members facilitate a number of conferences and professional development programs for members, and the Committee encourages new teachers to become members.

Regularly-updated information on ECAWA's conferences and professional development programs is published on the ECAWA website, found at <http://www.ecawa.asn.au>.

Information on the ECAWA Committee, and current membership details, are found later in this journal.

On behalf of the ECAWA Committee, I wish all members (both Individual, and Institutional), a productive and happy year, and the Committee looks forward to having you participate in some way in our programs and ventures throughout this year.

Kim Lego

EDITOR

Reflections on the value of the ECAWA Conference

Vicki Healey

Carine Primary School (2000)

Vicki was a first-time delegate in last year's annual ECAWA Conference, held in Mandurah.

Vicki's comments reflect a summary of the values of the conference to practising teachers, newly-appointed graduates, and those wishing to explore issues of current technologies in the classroom and their broad applications across the K-12 curriculum.

Shortly, the Committee will be starting the long processes of planning this year's Conference: updates on the progress will be regularly published on the ECAWA website at <http://www.ecawa.asn.au>.

What was on offer at the 2000 Conference?

A smorgasbord of information was on offer for individual groups, with a sprinkling of "good medicine" from keynote speakers for everyone.

Inspirational thoughts from the Keynote Speakers and ECWA (Past) President

- We are now confronted with evidence that in the workplace Information Technology is necessary Teachers have been left unprepared Skills need to be mastered IT is no longer an option Teachers have a moral obligation to achieve competency Professional Development (PD) is needed. *Bhavneet Singh (ECWA Past President)*
- It's going to happen IT will be cheaper and faster We'll need to work smarter and be more diverse Change happens and it upsets a lot of people. *Janine Bowes (Leader in area of Vocational Education)*
- Teachers may not need specific IT skills but need to know how to integrate IT use in the curriculum. *Ken Price (Churchill Fellow)*
- Kids know more - that's OK Be inspired, not threatened. *Jen Elwin (Middle School teacher from South Australia)*
- Software needs to be looked at for its pedagogical (teaching) practice It needs to support inquiry learning It needs to develop a learning environment that goes beyond "point and click". (*Assoc. Prof. Sandra Wills – builder of the First Fleet Web Resource*)

Ideas picked up from talking to teachers and from attending workshops

- An "open door" policy can exist where students can go to other classes or the lab to use computers if they are not already in use.

| Date | What I Did | Problem | Solution |
|------|------------|---------|----------|
| | | | |

- Children have a computer log book to record daily learning experiences:
- Be forgiving of technology - always have a plan B and C.
- For Internet users, "education is not a treasure hunt". Teachers need to be heavily involved. Check out Rosemary Horton's Trinity College Resource Centre Homepage - <http://www.students.trinity.wa.edu.au/library/>
She has spent hours hunting for Internet sites that fit with the WA Curriculum for students and teachers.
- Ethics of 'net copying: "Just because you can do it, you don't have to do it."
- Researching on the net? Have a word processing document (e.g. **Word**) open at the same time. Copy/paste relevant information.
- When word processing a report, children could use the highlight icon and highlight the work in, say, pink, and their own work in, say, yellow. The teacher can see the percentage copied and suggest that instead of 75% copied and 25% their own work, that next time the student try for a 50/50 split, then a 25/75 split and so on.
- Plagiarism needs to be discussed as part of ethical obligations.
- Plan to limit copying and pasting by changing the assignment to cater for this. E.g. Instead of getting them to report on "The Bee", get them to do "A Day in the Life of a Bee". Web Quests have lot of these ideas.
- Sending students to "work on the computer" without any structure or accountability won't necessarily improve their learning. The facility of technology makes it easy to look busy without achieving anything of value. Don't assume students are learning anything (if they are) on the computer unless what the outcomes are have been recorded in advance. (*Dr Jim Mullaney*)
- Email - "without boundaries that are clearly described and consequences for inappropriate use, giving students access to email can create big problems." (*Dr Jim Mullaney*)
- Good uses for email - collaborative projects (like recording observations for a world wide research on clouds) and mentors (ask a professional question - like ask the zookeeper what the polar bears had for breakfast). Look up buddy schools all over the world through *Epals*. (If doing research on natural disasters, ask for any schools that may have been in the area of a tornado, disaster or flood in the last year or so. Children can then talk to other children about their experiences).
- Make email messages short. Check children's hand-drafted messages first.
- *PowerPoint™*. It's all about the message. Keep it quiet. Sounds? What does they add? Nothing? Then don't use it. Clipart? Only if needed. If graphics are needed, it is best if students create their own. If they add to the presentation, then yes, put them in.
- And finally from Dr Jim Mullaney:

"Pedagogical methods have to change. Nothing changes just because you have computers in your classroom."

Principles of encryption for secure email and e-commerce

Kim Lego

Many people currently use email and the internet for all sorts of transactions.

Most users are aware that many internet transactions are able to be intercepted by unauthorised people.

This article provides a brief explanation of the processes involved in encryption of secure internet transactions, such as those used whenever e-shopping and e-banking is done.

Sources for some of the material in this article are:

- Part II in a series on encryption by Lachlan O'Dea, published in *Cyclops* (the quarterly journal from Computer Associates, producers of VET™) Issue 35 2000/01, and
- The BBC TV series *The Science of Secrecy – Going Public*, by Simon Singh, broadcast in January 2001 on ABC TV.

There are many different kinds of cryptographic ciphers around, and these are widely used for the passing of messages and other data around the world. They all work by performing carefully chosen mathematical operations on *plaintext* (i.e. text as written in any human language) to produce *ciphertext* (text that has been altered in some way using a complex code) that cannot be read without some extra information about how the encryption was performed. Fortunately we don't need to understand any of the mathematics behind encryption to be able to use it!

Cryptographic ciphers (also known as *cryptographic algorithms*) can be divided into two basic categories: *symmetric*, and *asymmetric*.

Symmetric Encryption

This form of encryption basically involves using a secret piece of information called a "key" to encrypt data. Once the data is encrypted using a single particular key, it cannot be decrypted without that same key.

This is the traditional method of sending "secret" messages or "codes" to others, and has been used for thousands of years in various forms. To send a message in this way, symmetric encryption is used, whereby each party to the message (the *sender* and the *receiver*) needed to have exactly the same *key* (i.e. means of encoding and decoding the message). The key had, in some way, to be transferred from sender to receiver via a *different* channel (often involving complex subterfuge). With the same key as was used to transmit the message, the receiver could decipher (or *decrypt*) it. This kind of encryption is called "symmetrical", because the *same key* is used both to encrypt and decrypt the information.

This was the means by which the German armed forces in World War II enciphered their war correspondence using the *Enigma* machine – although the encryption algorithm was very clever (using 4^{26} possible combinations for each coded message, with the key being regularly changed), code books containing lists of the keys necessary to decipher the coded messages had to be sent to all recipients of messages, and by capturing code-books in raids on some German Navy ships, the Allied forces managed to read German messages.

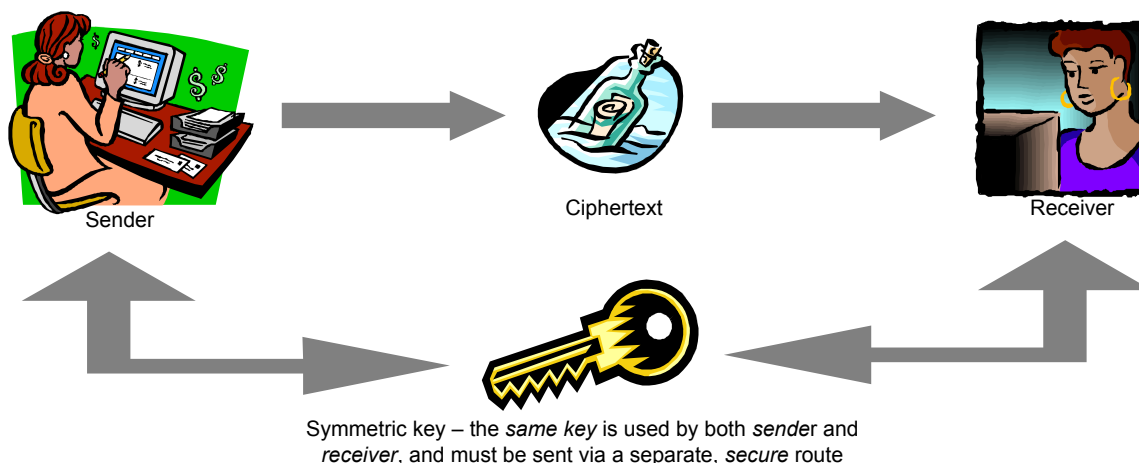


Figure 1: Symmetric Encryption (not used for internet transactions)

Asymmetric Encryption

Symmetric encryption was the only system of encryption possible up until the mid 1970s, when two teams of mathematicians based in Britain (Ellis and Cox), and at Stanford University in USA (Diffie and Helman) respectively, independently devised a means of using *two* keys, and thus invented the concept of *asymmetric* encryption, whereby one key (the “public” key) could be sent *with* the enciphered message, and the other key (the “private” key) remained with the *receiver* of the message, to be able to decipher the message sent.

The idea is rather simple: the person to whom the message is to be sent (the *receiver*) has a form of key called the *public* key, which can be sent to any person wishing to send a message to them via the internet. The *receiver* also has a special “partner” key (the *private* key) which *must remain a secret to the receiver*.

The *sender* uses this public key to initially encipher the message, which is then sent to the receiver.

The *receiver* then combines the encoded message (generated using her/his *public* key) with their secret *private* key to decipher the message.

The two keys are mathematically related to each other using two converse mathematical equations. Information that is encrypted with the first (*public*) key earlier sent to the *sender* by the *receiver*, can only be decrypted with the second (*private*) key, held by the *receiver*.

Cox devised **Cox’s Equation** for encryption, which stated:

$$C = M^e \pmod{N}$$

where **C** represented the character which had been *enciphered* (the *ciphertext* character), **M** represented the *unenciphered* (or *plaintext*) character, and **N** represented the *public* key (which was the product of two large prime numbers), which were in turn kept by the *receiver* as the *private* key). The two prime numbers used to determine the public key are calculated based upon the multiple of two randomly-chosen, very large *prime* numbers at the *receiver’s* end.

It is important at this point to note that it is extremely improbable mathematically to be able to determine the actual *values* of the two large primes used to generate the *public* key (using factorisation), and herein lies the strength of asymmetric encryption.

The exponent **e** is also a prime number.

As an example, to send the plaintext character **X** (ASCII **88**) in this way, assume that the two primes “chosen” by the *receiver* as her/his *private* key were 11 and 17, and the exponent “chosen” was 13. The *receiver* would have previously sent the *sender* the *public* key (of **187**, or 11 x 17, in this example):

Encryption (by *sender*):

$$C = 88^{13} \pmod{187}$$

i.e. C = 165, where C is the *ciphertext* for **X** sent

So the *ciphertext* sent in this case would be % (as the ASCII code consists of 128 characters, 165 - 128 = 37. ASCII 37 is %).

The *receiver* of the ciphertext “knows” that the *public* key is **187** as it was used to create the ciphertext, and also is the holder of the *private* key (in this case, of 11 and 17). Using the converse of Cox’s Equation, the receiver can decipher the message:

Decryption (by *receiver*):

$$M = C^{1/e} \pmod{(p-1)(q-1) \pmod{N}}$$

where $p = 11$, $q = 17$ and $e = 13$ in this case.

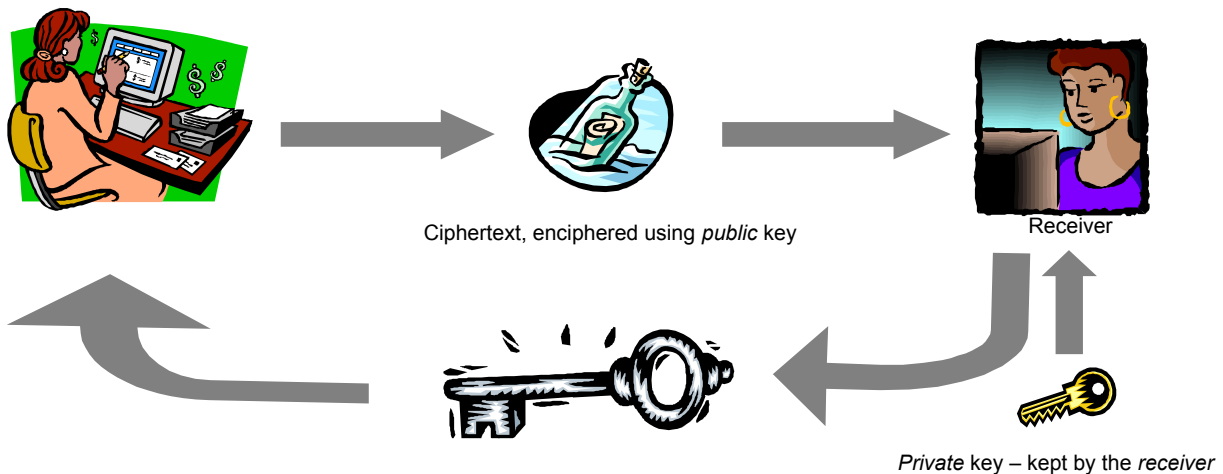
$$\text{i.e. } M = 165^{1/13} \pmod{(11 - 1)(17 - 1) \pmod{187}}$$

Thus,

$$M = 88 \text{ (which is } \mathbf{X} \text{ in } \textit{plaintext})$$

It must be pointed out here that the use of two such “small” primes as 11 and 17 could easily be determined by factorisation of 187 using an algorithm which repeatedly finds and tests multiples of two primes.

In reality, the primes are much larger numbers (now often based on values of 2^{128} – the so-called *128-bit key*), and thus the multiple of the numbers “chosen” are almost impossible to factorise.



Asymmetric encryption is interesting because it is the basis for providing "Public Key Cryptography". It is by using public key cryptography that we can achieve confidentiality, authenticity and integrity on the Internet.

Figure 2: Asymmetric (Public Key) Encryption

Public Key Cryptography

The concepts of PKC were devised independently by:

- Ellis and Cox, of the British Secret Service in 1973 (but kept secret until they published their findings in 1997), and
- Diffie and Helman, of Stanford University in 1975.

As a result of Diffie and Helman publishing their findings in a **Scientific American** article in 1977, USA mathematicians Rivest, Shamir and Adleman devised the **RSA** encryption algorithms, now widely used as the basis for internet encryption (estimated to be at least 500 million users worldwide at this time).

By using *symmetric* encryption alone, private messages that can only be read by people with the secret key can be sent. However, this approach quickly runs into trouble in many situations. The classic example is with email.

If email is used to send a person some encrypted data, the receiver will need to somehow know what the key is before they will be able to read it. The key cannot just be emailed as well - that would defeat the whole point of encrypting it in the first place. Another form of "secure channel" to send the key to the receiver is needed. The public telephone system is pretty safe and could be used to give them the key. However, this quickly becomes impractical if there is a need to communicate with several people.

Public key cryptography addresses this problem by removing the need for the separate secure channel. It is based on *asymmetric* encryption with two keys. One key is "public" and the other "private". A person's public key can be made freely available to anyone, by any means. The private key must not be revealed to anyone else, or the whole system falls over.

Once a message has been encrypted with a public key, it cannot be decrypted without the corresponding private key (held only by the *receiver*). Only the receiver will be able to read it, as they have the appropriate private key.

Public key cryptography also allows for checking authenticity and integrity of a message by using a *digital signature*. A security code - or *digital signature* - (such as a Cyclic Redundancy Check value) is calculated by the sender's system based upon the original plaintext message. The public key is then used to encrypt the sender's plaintext message and embedded digital signature. The receiver can decrypt the ciphertext using their private key in the manner described previously. The *security code* (or *digital signature*) is then compared to that embedded in the deciphered message. If the digital signatures match, then the message has not been changed.

Problems With Public Key Cryptography

So, public key cryptography sounds great. We get confidentiality, authenticity and integrity all over a totally public system such as the Internet. There is a catch, however. The big problem is that of managing the public keys.

A public key can be freely distributed without any risks. However, *using* a public key is a different matter. Assume that you wanted to send an encrypted email message to a colleague called Bob. You might receive an email from Bob saying:

"Hi, Bob here. I've attached my public key, so send me your encrypted email message".

You can use the attached public key to encrypt your message, but how can you be sure that the public key is actually Bob's key? That email might have been sent by Alice, pretending to be Bob. If the public key actually belongs to Alice, then she will be able to decrypt your messages, as she has the corresponding private key.

The basic problem is how to establish *trust* in the use of a public key. In other words, how does a *sender* really know who has the private key for a given public key.

There are various approaches to this trust problem that are in current daily use. One of the commonly used is the "web of trust", which is used by **PGP** ("Pretty Good Privacy").

Web Of Trust

A web of trust works by establishing a few trusted keys and using them to establish the trust of other keys. The mechanism used to establish trust in a *public key* is to sign the key with a digital signature (see above).

To start the process going, a means of *verifying the owner* of a key is needed. A common method would be to have a friend or colleague hand you their key on a floppy disk. You can then be certain who the key belongs to. If you are certain that a key is legitimate, you sign the key using your own private key. By signing the key you are saying "I hereby certify that this key is OK".

The web of trust is established by checking keys to see who has signed them. If a person receives a new public key, and it is signed by someone whose key they know and trust, then the new key can be trusted as well. Extending this, any keys signed by the new key can also be trusted. In this way the web of trust can be increased over time.

The web of trust has the advantage that it is fairly simple to use. Anyone can start a web of trust without much difficulty. However, problems can occur when dealing with lots of people. There are various *keyservers* on the Internet that can help with these problems.