Proceedings of the International Conference on
Information Technologies (InfoTech-2007),
September 21-23, 2007, Bulgaria
vol. 1

e-LEARNING: A CASE STUDY OF CHOR-RIVEST CRYPTOSYSTEM IN MAPLE[1]

A. Queiruga Dios, L. Hernández Encinas, and J. Espinosa García

Dpt. Information Processing and Coding, Applied Physics Institute, CSIC
{araceli, luis, javier.espinosa}@iec.csic.es
Spain

Abstract: A new experience teaching programming and numerical methods to engineering students, using Maple to implement the Chor-Rivest cryptosystem, is shown. The aim is to give the students a better understanding of cryptography by using Maple software. In this paper we discuss our attempts to structure an on-line course that required the students participation, interest, and evaluation by means of a new environment.

Key words: e-Learning, Cryptography, Chor-Rivest cryptosystem.

1. INTRODUCTION

The Information and Communication Technologies (ICTs) have changed from being considered as a mere object of use towards an instrument of support in the educational innovation (Salinas, 2004). They affect to different aspects in relation to traditional education, like the change in the role of the teacher, who has changed from a simple transmitter of knowledge to be a mediator in the construction of the knowledge of the students; the role of the student has changed as the traditional educative models do not adjust to the processes of learning by means of the use of the ICT (Pérez, 2002). Finally, it is important to take into account that the ICTs do not require of the invention of new methodologies, but it requires a modification in the strategies for the continuous learning of the student (Mason, 1998).

[1] This work is supported by Ministerio de Educación y Ciencia (Spain), under grants SEG2004-02418 and MTM2005-00173, and Consejería de Educación y Cultura, Junta de Castilla y León under grant SA110A06.

Within the ICTs, one of the fields of greater projection in the future and greater impact is Cryptography. As it is known, this science is closely related to the Mathematics and Computer Science. Its aim is the preservation of information integrity, including confidentiality, and authentication. The objective of the Cryptography is to provide safe communications on insecure channels, i.e., to allow several people send messages by means of a channel that can be intercepted by a third person (mail or e-mail, telephone, fax, etc.), so only the authorized receiver can read the messages (Fúster *et al.*, 2004, Menezes *et al.*, 1997).

The great importance of Cryptography in our days is due to the proliferation of personal computers and the facility in the access to Internet. These facilities have cause serious problems of security, like virus, spam, phising, publication of confidential information, etc. All of it makes necessary that students and future professionals are conscious of the dangers that surf the Internet without safety measures supposes.

In this paper, we present an e-learning educational method to learn about cryptography and how to implement the Chor-Rivest cryptosystem. The rest of the paper is organized as follows: In section 2, we comment the background needed to follow this course, i.e., the mathematical tools and the basic Maple commands. The Maple procedures to implement the cryptosystem are presented in section 3. Finally, the methods to evaluate this course will be stated in section 4.

1. BACKGROUND

The goal of this course is that students could implement functions and procedures involved in Cryptography. In fact they will study in detail all mathematical problems, algorithms, and cryptographical concepts in order to understand how to encrypt and decrypt messages, with the Chor-Rivest cryptosystem. In Public Key Cryptography (PKC) two keys are used: The encryption (or public) key is used by the sender to encrypt a message, and the decryption (or private) key is kept in secret by the receiver and it allows him to decrypt the message. The security of these schemes is based on the computational intractability of some Number Theory problems.

2.1. Basic concepts of Mathematics and Cryptography

In the first part the students will learn some mathematical and cryptographic items, i.e., the background needed to follow this course. In this way, they will study the arithmetic of Galois Fields, $GF(p^m)$, several cryptographical concepts: plaintext, ciphertext, key, cryptosystem, etc., and the most important mathematical problems related to cryptography: factorization, discrete logarithm, and knapsack problems.

In particular, the Chor-Rivest cryptosystem was proposed in 1985 (Chor, 1985; Chor and Rivest, 1988). It is based on the arithmetic of finite fields, it needs to compute discrete logarithms in order to generate the keys of each user, and its security is based on the knapsack problem (not on the discrete logarithm problem!).

The *Discrete Logarithm Problem* (DLP) can be defined as follows: Given a generator α of a cyclic finite group of order n, G, and an element $\beta\in G$, to find the integer x, $1<x<n-1$, such that $\beta=\alpha^x$. This problem is considered to be difficult because the best known algorithm for solving it has a subexponential expected running time (Stinson, 2002).

The *Knapsack Problem* (KP) was originated from the economic world: How to optimize the transport of some goods with an economical and size given value if the transport has a limited size. The special case of this problem used in Cryptography is as follows: Maximize

$$\sum_{i=1}^{n} v_i w_i \qquad (1)$$

subject to the restrictions

$$\sum_{i=1}^{n} w_i x_i \le s, \quad x_i \ge 0, \qquad (2)$$

where x_i, w_i, v_i (for $i=1,\dots,n$), and s are known integers, and x_i and s are positive.

2.1. Basic concepts of Maple

The second part of the course is addressed to show to the students how to work with the Maple software in order to implement the Chor-Rivest cryptosystem (Hernández *et al.*, 2000). Maple, is a comprehensive environment for teaching and applying mathematics. It contains thousands of math procedures and it permits to define procedures by using the Maple programming language. Maple contains several packages to help professors to teach and students to understand mathematical concepts. A package is a collection of routines that are collected together, and provides a range of functionality for solving problems in some well-defined problem domain.

After knowing the Maple syntax, the students will practice with the main commands related to Galois fields, KP and DLP. For example, the mod(e,m) operator evaluates the value of e modulo the integer m. The ifactors(n) function returns the complete integer factorization of the integer n. The GF(q,h,f) command returns a table of functions and constants for doing arithmetic in the finite field of q^h elements: $GF(q^h)=GF(q)[T]/(f(T))$, where $f(T)$ is an irreducible monic polynomial of degree h over the integers modulo q. If f is not specified, Maple uses a random one. The Powmod(a,n,f,x) function computes the remainder of a^n in $GF(q^h)$. Finally, if $u=[u_1,\dots,u_n]$ is a list of integers and $m=[m_1,\dots,m_n]$ a list of moduli, pairwise relatively prime, the function chrem(u,m) solves the Chinese Remainder Theorem, i.e., computes the unique positive integer a, $0 < a < M$, such that $a=u_1 \bmod m_1$, $a=u_2 \bmod m_2$, ..., $a=u_n \bmod m_n$.

As it was mentioned, to generate the system parameters, it is necessary to compute discrete logarithms over $GF(q^h)$. To do this, one can use the Baby-step Giant-step and the Pohlig-Hellman algorithms (Pohlig and Hellman, 1978). The input of the PohligHellman procedure is a generator, α, of $G=GF(q^h)^*$ and an element, β, of the group. The output is $\log_\alpha\beta$. The following procedure is designed to compute discrete logarithms in $GF(q^h)$, where q, h and $f=f(T)$ are known.

```
>PohligHellman := proc(beta,alpha)
  local NN, FF, RR, DD, ii, PP, EE, XX, QQ, EX, GG, LL, AA, jj, GG1, BB, OUT:
  NN:=(q^h)-1: FF:=convert(factorset(NN),list): RR:=nops(FF): DD:=ifactors(NN)[2]:
  for ii from 1 to RR do PP[ii]:=DD[ii,1]:EE[ii]:=DD[ii,2]: end do:
  for ii from 1 to RR do
    XX[ii]:=0: QQ:=PP[ii]: EX:=EE[ii]: GG:=1: LL[-1]:=0: AA:=Powmod(alpha,NN/QQ,f,T) mod q:
    for jj from 0 to EX-1 do
      GG:=Powmod(GG*Powmod(alpha,LL[jj-1]*QQ^(jj-1),f,T) mod q,1,f,T) mod q:
      GG1:=Powmod(GG,-1,f,T) mod q:
      BB:=Powmod(Powmod(beta*GG1,1,f,T) mod q, NN/(QQ^(jj+1)),f,T) mod q:
      LL[jj]:=BabyGiant(BB,AA,QQ^EX): XX[ii]:=XX[ii]+LL[jj]*QQ^jj:
    end do:
  end do:
  OUT:=chrem([seq(XX[ii],ii=1..RR)],[seq(PP[ii]^EE[ii],ii=1..RR)]):
  return(OUT):
end:
```

To execute this algorithm, it is necessary to implement, in a similar way, the Baby-step Giant-step algorithm. This implementation will be suggested to students to improve the knowledge of implementations in Maple. Both procedures, with the number theory package, with(numtheory), could be saved in a file to be loaded when it be necessary.

3. CHOR-RIVEST CRYPTOSYSTEM

3.1. Keys generation

The parameters of the cryptosystem are chosen as follows (for more details, see (Chor, 1985, Chor and Rivest 1988):

1. Let q be a prime and let $h \leq q$ be an integer so that the DLP can be efficiently solved in the finite field $GF(q^h)$. This property is crucial because the user has to compute discrete logarithms in $GF(q^h)$ to determine his keys (it is known that the DLP can be efficiently solved if the order of the multiplicative group considered, $G=GF(q^h)^*$, factorizes as a product of small prime factors).
2. A random element $T \in GF(q^h)$ is chosen such that T is algebraic of degree h over $GF(q)$. The student will take into account that the elements in $GF(q^h)$ are polynomials of degree $\leq h-1$ with coefficients in $GF(q)$, and the operations are done modulo q and $f(T)$.

3. Pick a generator $g\in G$. To determine such generator, the student can choose a random element $g\in G$ until it verifies $g^{(q^\wedge h\text{-}1)/z}\neq 1$ for all prime divisors, z, of q^h-1. Remember that q an h are chosen so that q^h-1 has small prime factors, hence the above property is easy to verify.
4. The discrete logarithms, $a_i=\log_g(T+\alpha_i)$, $\forall\,\alpha_i\in GF(q)$, are computed, and the values $b_i=a_{\pi(i)}$ are obtained by using a random permutation, $\pi:\{0,\ldots,q-1\}\rightarrow\{0,\ldots,q-1\}$.
5. A random noise, $0\leq r\leq q^h-2$, is added to obtain the public knapsack: $c_i\equiv(b_i+r)\ \mathrm{mod}(q^h-1)$, $i=0,1,\ldots,q-1$.
6. The public key of the user is $(c_0,c_1,\ldots,c_{q-1})$, and the private key is (T,g,π,r).

These keys (public and private keys) could be saved in a file by each student, and send them to the course tutor as part of the evaluation:

```
>save f, r, pii, g, "PrivateKey.prk":
  PublicKey:=[seq(c[i],i=0..q-1)]: save PublicKey, "PublicKey.puk":
```

In addition, the public key, like the teacher one, will be public. So, it will be make public for all the students in the web course.

3.2. Encryption process

In this on-line course, the students will learn how to encrypt messages with this cryptosystem. A message, M, is a vector with length q and weigh h: $M=(m_0,\ldots,m_{q-1})$, $m_\mathrm{i}\in\{0,1\}$, $i=0,\ldots,q-1$. The sender encrypts that binary message as follows:

$$E=\sum_{i=0}^{q-1} m_i\cdot c_i(\mathrm{mod}(q^h-1)) \qquad (3)$$

The procedure to transform binary messages without constraints into binary vectors of length q and weight h, and its inverse procedure, will be explained to the students in order to ask them both procedures. Follows an example of the first one.

```
>Transformation := proc(int::integer,qq::integer,hh::integer) local NN, QQ, HH, LL, ii:
  NN:=int: QQ:=qq: HH:=hh: LL:=array(1..QQ):
    for ii from 1 to QQ do
      if NN >= binomial(QQ-ii,HH) then LL[ii]:=1: NN:=NN-binomial(QQ-ii,HH): HH:=HH-1:
      else LL[ii]:=0:
      end if:
  end do:
return([seq(LL[ii],ii=1..QQ)]):
end:
```

The second one, Itranformation, will be implemented by students. Moreover, students will encrypt a particular message (we will called it Text), that will be establish during the course, and the ciphertext will be sent to the tutor. As this is a public key cryptosystem, messages will be encrypt with the teacher public key. In the encryption process, the student, S, restarts Maple and loads needed procedures and the public key of the teacher. S writes down the text of his message, computes the length and the number of blocks of the message, transforms and divides the message into blocks of the same length, and transforms it in blocks of length q and weight h. A method to encrypt a message by using this cryptosystem could be the following:

```
>ltext:=nops(Explode(Text)); lengt:=floor(log[2](binomial(q,h)));
 lblock := floor(lengt/8); lmess:=lblock*ceil(ltext/lblock);
 nblocks:=lmess/lblock; mascii:=map(Ord,Explode(Text)):
 for i from 1 to lmess do
   if (i<=ltext) then masciic[i]:=mascii[i]: else masciic[i]:=0: end if:
 end do:
>for i from 1 to nblocks do
 mess256[i]:=(convert([seq(masciic[j],j=(i-1)*lblock+1..i*lblock)],base,256,10)):
 messblock[i]:=sum(mess256[i]['j']*10^('j'-1),'j'=1..nops(mess256[i])):
 M[i]:=Transformation(messblock[i],q,h):
end do:
```

Finally, S saves the ciphertext in a file to be sent to the tutor.

3.3. Decryption process

To decrypt the encrypted message, E, the receiver executes the following steps:

1. Computes $s' \equiv E - hr \bmod (q^h - 1)$.
2. Determines the polynomial of degree $h-1$, $Q(T) = g^{s'} \bmod (f(T))$.
3. Computes the h roots $a_{\pi(i)}$ of $f(T)+Q(T)$ over $GF(q)$:

$$f(T) + Q(T) = \prod_{i \in I} \left(T + \alpha_{\pi(i)}\right) \qquad (4)$$

4. Applies π^{-1} in order to recover the coordinates of the original message M having the bit 1.

The teacher will send to each student a message, EncryptedMessage, encrypted with student's public key, and they must be able to decrypt it, using his private key.

To recover the original message, each student restarts Maple and loads some Maple procedures, the parameters of the system generated by himself, his private key, and the encrypted message. Then, he computes the polynomials of degree $h-1$ over $GF(q^h)$, $Q_i(T)$, and determines the h roots of $f(T)+Q_i(T)$, by factoring those polynomials. Later, the student applies the inverse permutation to recover the coordinates of the original message, obtains the partial messages and computes the integer numbers corresponding to each partial message of length q and weight h.

```
>nblocks:=nops(EncryptedMessage); alias(alpha = RootOf(f)):
 for j from 1 to nblocks do
   sprime[j]:=EncryptedMessage[j]-h*r mod ((q^h)-1):Q[j]:=Powmod(g,sprime[j],f,T) mod q:
 end do:
>parc:=[]: R:=[]:
 for j from 1 to nblocks do
   pol[j]:=Factor(f+Q[j],alpha) mod q:sol[j]:=[msolve(pol[j]=0,q)]:
   for i from 1 to h do parc:=[op(parc),-rhs(op(sol[j][i])) mod q]: end do:
   if member(0,parc) then parc:=Rotate(parc,1): end if:
```

```
   R:=[op(R),parc]: parc:=[]:
  end do:
>Pos:=[]: psc:=[]:
  for j from 1 to nblocks do
   for i from 1 to h do member(R[j][i],pii,'pos'): psc:=sort([op(psc),pos]): end do:
   Pos:=[op(Pos),psc]: psc:=[]:
  end do:
>for j from 1 to nblocks do m[j]:=array(1..q):
   for i from 1 to h do m[j][Pos[j,i]]:=1: end do:
   for i from 1 to q do if m[j][i]<>1 then m[j][i]:=0: end if: end do:
  end do:
>for j from 1 to nblocks do Me[j]:=ITransformation(convert(m[j],list), q, h): end do:
  Dm:=[seq(Me[j],j=1..nblocks)]:
```

Finally, the student decrypts the message and obtains the original message.

```
>for i from 1 to nblocks do
   Mes:=(convert(Dm[i],base,256)): Message[i]:=map(Char,Mes):
  end do:
  RecoveredText:=Implode(Flatten([seq(Message[i],i=1..nblocks)],1));
```

4. EVALUATION

We have designed an on-line course related to Cryptography, in order to facilitate concepts and contents to engineering students that should favor and develop habits of good practice in telecommunications, thus improving Internet security.

As the main aim of this course is to improve the security in communications, it will be evaluated how students acquire the tools and basic knowledge to make it possible. This course tries to cover both mentioned issues: on-line education achieved for all students in all places, and the tools used for security. These tools are Maple commands and procedures for the implementation of the Chor-Rivest cryptosystem over finite fields. The evaluation will include:

1. Formative evaluation: focuses on improvement the security and cryptography knowledge while the course is in progress.
2. Summative evaluation: focuses on results or outcomes. Conduct upon completion of a program.

To make both possible, students will have the on-line course website (see Figure 1) with all the theory, papers and links related to the topic of the subject, they will e-mail all the questions, suggestions, or whatever they need to make this e-learning possible. Moreover, they will have access to electronic chat room, and forums to make possible an on-line participation whenever they could.

As we mentioned in previous sections, we will propose some exercises to have the feedback of the course and be sure that students reach the mentioned aims.

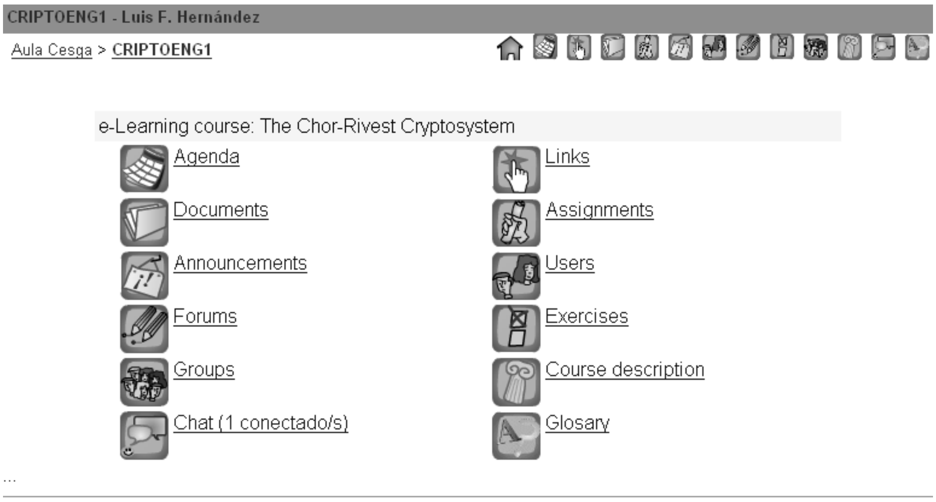

Figure 1. An example of Web page for the course

REFERENCES

Chor, B. (1985). Two issues in public key cryptography. RSA bit security and a new knapsack type system, *The MIT Press*, Cambridge, MS.

Chor, B., Rivest, R.L. (1988), A knapsack-type public key cryptosystem based on arithmetic in finite fields, IEEE Trans. Inform. Theory **34**, 5, 901-909.

Cover, T.M. (1973). Enumerative source encoding, *IEEE Trans. Inform. Theory* **19**, 73-77.

Fúster Sabater, A., de la Guía Martínez, D., Hernández Encinas, L., Montoya Vitini, F., Muñoz Masqué, J. (2004). *Técnicas criptográficas de protección de datos*, RA-MA, 3rd ed., Madrid.

Hernández Encinas, L., Muñoz Masqué, J., Queiruga Dios, A. (2006). Maple implementation of the Chor-Rivest cryptosystem, *Lect. Notes Comput. Sci.* **3992**, 438-445.

Mason, R. (1998). Models of online courses, *ALN Magazine* **2**, 2.

Menezes, A., van Oorschot, P., Vanstone, S. (1997). *Handbook of applied cryptography*, CRC Press, Boca Raton, FL.

Pérez i Garcías, A. (2002). Nuevas estrategias didácticas en entornos digitales para la enseñanza superior, *En Didáctica y tecnología educativa para una univesidad en un mundo digital* (J. Salinas y A. Batista), Universidad de Panamá, Imprenta universitaria.

Pohlig, R.C., Hellman, M.E. (1978). An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, *IEEE Trans. Inform. Theory* **24**, 106-110.

Salinas, J. (2004). Innovación docente y uso de las TIC en la enseñanza universitaria, *Revista de Universidad y Sociedad del Conocimiento (RUSC)* **1**, 1.

Stinson, D.R. (2002), *Cryptography: Theory and practice*, 2nd ed., Chapman & Hall/CRC, Boca Raton, FL.