INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

An overview to Integer factorization and RSA in Cryptography

[1]Sonal Sarnaik, Dinesh Gadekar[2], Umesh Gaikwad[3]

[1]Assistant Prof . Marathwada Institute of Technology
Aurangabad.
sonalsarnaik141@gmail.com,

[2]Third year MCA, Marathwada Institute of Technology,
Aurangabad.
dinesh.gadekar@gmail.com

[3]Third year MCA, Marathwada Institute of Technology,
Aurangabad.
gaikwadumeshb@gmail.com

Abstract: *Cryptography is the practice and study of techniques for secure communication. Two parties who want to communicate with each other in secure way, uses cryptography to provide confidentiality to the data. Data encryption and decryption is done through use of key. If same key is used to encrypt and decrypt data then it is called as symmetric key cryptography and if different keys are used then it is called as asymmetric key cryptography. Asymmetric key based cryptographic algorithms are also called as public key algorithms. Public key algorithms can further be classified on the basis of integer factorization, discrete logarithm and elliptical curve. In this paper we have discussed integer factoring based algorithm – RSA and different integer factorization algorithm such as Pollard rho as low exponent attack, Euler phi attack etc. Using Pollard rho and Pollard p-1 we have demonstrated algorithm and Pollard p-1 algorithm with its implementation. There are different attacks on RSA such the Euler phi attack on RSA*

Keywords: *RSA, Integer Factorization, Euler phi attack.*

1. INTRODUCTION

We are leaving the era of wireless communication. With wireless communication lot's of insecurities comes which can be as demolishing business, collapsing country, destroying personal images, financial losses etc. This gives us the definition and essence of security. Security can be achieved by hiding the data or converting it into some unreadable form. This conversion is accomplished by cryptography. Cryptography is the study of mathematical science which is used to convert the data in some incomprehensible form which gives security to the data, i.e. cryptography is the art of secret writing [9][11][19][23]. By using cryptography sender and receiver can communicate with each other without knowing data to third person. The process of converting original data (plaintext) to some incomprehensible data (cipher text) is called Encryption process and incomprehensible data to original data is called Decryption. This conversion of data is achieved by applying key to original data and vice versa to read the original data. If the same key is used at sender and receiver for encryption and decryption process then the key is called as Symmetric key cryptography and if different key is used for encryption and decryption process them key is called as Asymmetric key cryptography[7][11][19][23][27].

In symmetric key cryptography same key is used for encryption and decryption process.[26]. Main drawback of this type of cryptography is to share a secrete key, i.e Key Distribution Problem. The general working of symmetric key cryptography is shown in below figure (fig.1.1) .In this sender and receiver can share their data securely by applying key to it. This key must be secret and share through some secure medium. Algorithm as well key will be same for encryption process (at sender) and Decryption process (at receiver)[9][7]. Types of symmetric key cryptography can be divided in stream ciphers and block ciphers. Stream cipher encrypts digits (bytes) of message at a time. Examples of stream ciphers are FISH, Helix, ISAAC, MUGI, SEAL, SOBER etc. Block cipher encrypts bits of message at a time. Examples of block ciphers are AES[16], DES[17], IDEA, RC5, BLOWFISH [7-8] [10] [11] [19] [21], [23], [27].

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

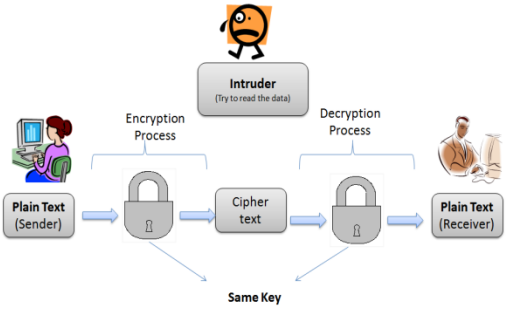


Figure 1: Symmetric key cryptography

2. ASYMMETRIC KEY CRYPTOGRAPHY

Asymmetric key cryptography overcomes the major drawback of symmetric key cryptography i.e sharing key through secure medium [26]. In asymmetric key cryptography sender and receiver both uses different keys for encryption and decryption process [21]. The general working of asymmetric key cryptography is shown in below figure (fig 2.1). Key for encryption is publically declared hence called as public key and key for decryption process is private only known to receiver is called as private key. It is very difficult for any intruder to read the message even when encryption key is publically declared and to find decryption key is very difficult as it is only known to receiver. It is also called as Public Key Cryptography [7][9][17][19][27].

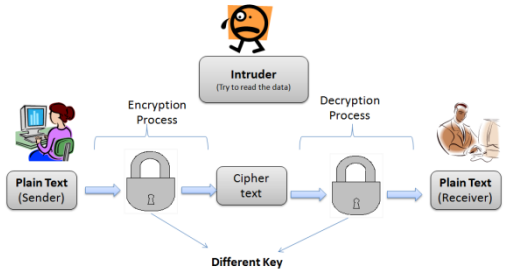


Figure 2: Asymmetric key cryptography

Diffie Hellman proposed a new algorithm "Diffie-Hellman key exchange algorithm" in 1976 which was the base of public key algorithm. By studying various research papers we categorize asymmetric key cryptography in three groups. Integer factorization, discrete logarithm and Elliptical curve. The examples of integer factorization is RSA algorithm, the example discrete logarithm is Tahar and Elgamal algorithm, new era of cryptography is started from Elliptical curve cryptography. In this paper we focused on the Integer Factorization [7][9-10][17][23].

3. MATHEMATICAL BACKGROUND

To study RSA algorithms and integer factorization algorithms, one should know some mathematical concepts. Such as Eular phi function and modular multiplicative inverse [7][9][17][19][23][27].

3.1. Euler Phi Function

Euler phi function is also called as Euler's totient function [19]. This function is used to find the relationship with the real numbers. It is denoted by **φ(*n*)** [9][15][17-18][23][27]. There are three forms through which it is calculated.

i. when *n* is a prime number

$\varphi(n) = n-1$ ----------(3.1.1)

ii. when *m* and *n* are co-prime(GCD(m , n) =1)

$\varphi(m*n) = \varphi(m)*\varphi(n)$. ------- (3.1.2)

iii. If the prime factorization of n is given by,

$n = p_1^{e}{}_1 * \ldots * p_n^{e}{}_n$,

then

$\varphi(n) = n*(1-1/p_1)*\ldots\ldots(1-1/p_n)$ -------- (3.1.3)

1.2. Modular multiplicative inverse

In modular arithmetic modular multiplicative inverse of an integer a modulo m is an integer a^{-1} such that[9][11][17][18][23],

$$a * a^{-1} \equiv 1 \bmod m$$ ------- (3.2.1)

4. RSA

RSA algorithm was introduced by Ron Rivest , Adi Shamir, and Leonard Adleman In 1976. RSA is motivated by the published works of Diffie and Hellman developed in 1976 .[2][3][4][9][19][27].RSA algorithm is used for public key algorithm and for digital signature. It uses two types of key, public key and private key. Public key is used to encrypt data; this key is publically declared and known to all. Private Key is used to decrypt the data be receiver, this key is private and no one else can use this key. The security of the RSA algorithm is that it is mathematically infeasible to factor sufficiently large integers [5][6][9][10][13][14] [17][18][20] [22][23][24][28].

Algorithm:-

Alice	**Publically Declared Values**	**Bob**

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Step 1:	Alice will choose two large prime number **p** and **q** Find **n= p*q.** calculate **ω(n)=(p-1) * (q-1)**		
Step 2:	Choose **e** such that it satisfies following condition. **i. Gcd(e, ω(n))=1** **ii. Max(p,q)** **iii. e must be prime no**		
Step 3:	Find **d** such that it satisfies:- **i. e. d ≡ 1 mod ω(n)** **ii. d>log$_2$(n)** **iii. Gcd(d, ω(n))=1**	**n** and **e**	If M is message to send, then cipher text is created by, **Step 4:** **C=M^e mod n**
		Cipher text is send to Alice	
Step 5	Orignal message is derived by , **M=C ^{d} mod n**		

Table 1: Stepwise representation of RSA algorithm

Where **e** is the encryption key which is publically declared, **d** is decryption key which is private, M is the Original message and C is the Cipher text generated.

Example:

Alice

Step 1: p=103,q=113

n= p*q

n=103*113

n=11639

ω(n)=(p-1)*(q-1)

=(103-1) *(113-1)

=11424

Step 2: If we choose e=5563 then it should satisfy.

i. **GCD(e, ω(n))=1**
GCD(5563,11424)=1 ---Condition satisfy

ii. **Max(p,q)**
Max(103,113)=113
so, **e**>113 i.e **e=5563** ---Condition satisfy

iii. **5563 is a prime number** ---Condition satisfy

Step 3: Find **d** such that it satisfies:-

i. **e. d ≡ 1 mod ω(n)**

Will try possibility for d from 1 till it satisfies the equation.

e	**e. d mod ω(n)≡ 1 mod ω(n)**	**Status**
1	1*5563 mod 11424 =5563	Cannot use
2	2*5563 mod 11424 =11126	Cannot use
3	3*5563 mod 11424 =5265	Cannot use
4	4*5563 mod 11424 =10828	Cannot use
5	5*5563 mod 11424 =4967	Cannot use
6 . . .	. . .	Cannot use . .
115	115*5563 mod 11424=1	**Can be use**

Table 2: Stepwise process to choose **"e".**

ii. **e>log$_2$(n)**
115>13.50667949 -----Condition satisfy

iii. **Gcd(e, ω(n))=1**
GCD(115,11639)=1 ------Condition satisfy
So, **e=5563 and n=11639**

5. INTEGER FACTORIZATION

In number theory of cryptography, Integer factorization is the process, in which composite number breakdown into non-trivial divisors (called as factors), and by multiplying these two non-trivial factors one get given composite number[9][10][13][14][17][18][20][27][28][29].

Example: - n= 21,
Factors=7 and 3.
If 7 * 3 ,we get 21.

In RSA a large composite number is used which breakdowns into two prime numbers.

Example:- n=11639 is composite number
n=103 * 113

Security of RSA is completely depends on these two factors, i.e if n is larger then it will be difficult to find factors. There

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

are many algorithms which are used to find non-trivial factors of composite numbers such as trial division method, Pollard rho algorithm, Pollard p-1 algorithm, Quadratic sieve algorithm etc[13][27]. In this paper we have discussed about pollard rho algorithm and pollard p-1 algorithm its implementation using MatLab.

5.1. Pollard's rho algorithm:

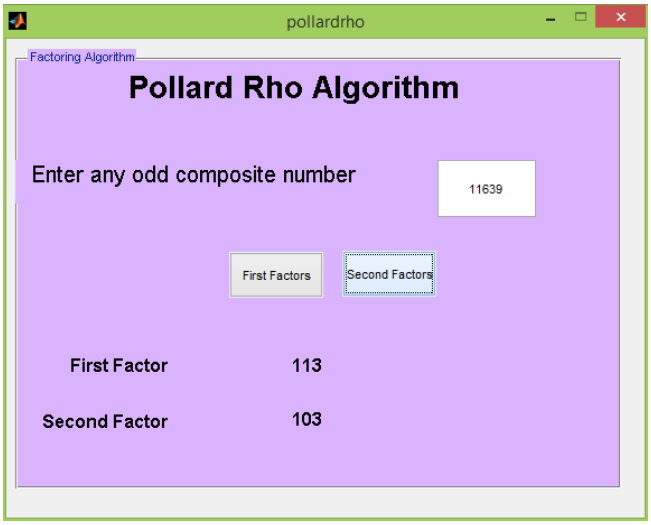
Pollard's rho algorithm is a special purpose factoring algorithm for finding small non- trivial factors of a composite integer. It uses polynomial function f with integer co-efficient, i.e. $f(x) = x^2 + c$, here value of c starts with **c=1** and **x= a** or **b** . if value of d=n then value of c must be incremented to get the factors.n must be an odd composite number[9][13][17][20][27].

INPUT: a composite integer n that is not a prime power.
OUTPUT: a non-trivial factor d of n.
1. Set a←2, b←2.
2. For i = 1, 2, . . . do the following:
2.1 Compute $a \leftarrow a^2 + 1 \bmod n$, $b \leftarrow b^2 + 1 \bmod n$, $b \leftarrow b^2 + 1 \bmod n$.
2.2 Compute d = gcd(a − b, n).
2.3 If 1 < d < n then return(d) and terminate with success.

Table 3: Pollard Rho Algorithm

Above algorithm is implemented using MatLab and it shows the demonstration to find factors of n=11639.Two factors are 103 and 113 which are prime numbers.

Output of Pollard Rho Algorithm




```
.Pollard Rho algorithm
% --- Executes on button press in b1.
function b1_Callback(hObject, eventdata, handles)
% hObject    handle to b1 (see GCBO)
% handles    structure with handles and user data (see GUIDATA)
n=Str2num(get(handles.t1,'String'));
int max a;
int max b;
int max d2;
int mac c2;
a=2;
b=2;
for i=1:100
   a=a.^2;
   a=(mod((a+1),n));
   b=b.^2;
   b=(mod((b+1),n));
   b=b.^2;
   b=(mod((b+1),n));
   c=a-b;
   d = gcd(c, n);
if (1<d && d < n)
   d2=(n/d);
   set(handles.st1,'String',d);
   break;
else
 if (d==n)
for c=1:n
for i = 1:100
   a=a.^2;
   a=(mod((a+c),n));
   b=b.^2;
   b=(mod((b+c),n));
   b=b.^2;
   b=(mod((b+c),n));
   c2=a-b;
   d = gcd(c2, n);
if (1<d && d<n)
    d2 = n/d;
    set(handles.st1,'String',d);
    break;
end;
end;
end;
   break;
end;
end;
end;
 % --- Executes on button press in b2.
function b2_Callback(hObject, eventdata, handles)
% hObject    handle to b2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

n=Str2num(get(handles.t1,'String'));
d=Str2num(get(handles.st1,'String'));
d1=n/d;
set(handles.st2,'String',d1);
```

5.2. Pollard p-1 algorithm:

Pollard p-1 algorithm is a special purpose factoring algorithm that can be used to find any prime factors p of a composite integer n for which (p-1) is smooth with respective to some relatively small bound B. This bound can

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

be set up to $\sqrt{n}$ .If no factors are found then b must be increased [9][17][20][27].

B-smooth: let B be a positive integer ,an integer n is said to be B smooth or smooth with respect to a bound B ,if all its prime factors are <=b. for example, 20= 2^2*5 ,so 5 is smooth.

INPUT: a composite integer n that is not a prime power.
OUTPUT: a non-trivial factor d of n.
1. Select a smoothness bound B.
2. Select a random integer a, 2 ≤ a ≤ n − 1, and compute d = gcd(a, n). If d ≥ 2 then return(d).
3. For each prime q ≤ B do the following:
   3.1 Compute l = $\left\lfloor \frac{\ln n}{\ln q} \right\rfloor$
   3.2 Compute a←a^{ql} mod n
4. Compute d = gcd(a − 1, n).
5. If d = 1 or d = n, then terminate the algorithm with failure. Otherwise, return(d).

Table 3: Pollard P-1 Algorithm

Above algorithm is implemented by using MatLab and its demonstration to find factors of n=11639 is shown below. Two factors are found i.e. 103 and 113 which are prime numbers [9], [27].

5.3. Eular Phi Attack

As it is said that security of RSA is depend on the factors of n. It n is small then it is very easy for find its two factors but if n is sufficiently large then it makes impossible to find factors. There are different attacks on RSA such as Euler phi attack, Common modulas attack, Large and small exponent attack[1][3][4][5][9][12][17][20][25].By using factoring algorithms, We have demonstrates Euler phi attack in which pollard p-1 algorithm is used to find two factors (**b** and **b1**) from publically declared **n** [27].

n=b * b1

By using these two factors euler phi is calculated and then by using publically declared **e** , **d** is calculated[12][17].

ω(n)=(b-1) *(b1-1) ……by using (3.1.1) equation
e. (d) ≡ 1 mod ω(n) ……by using (3.2.1) equation

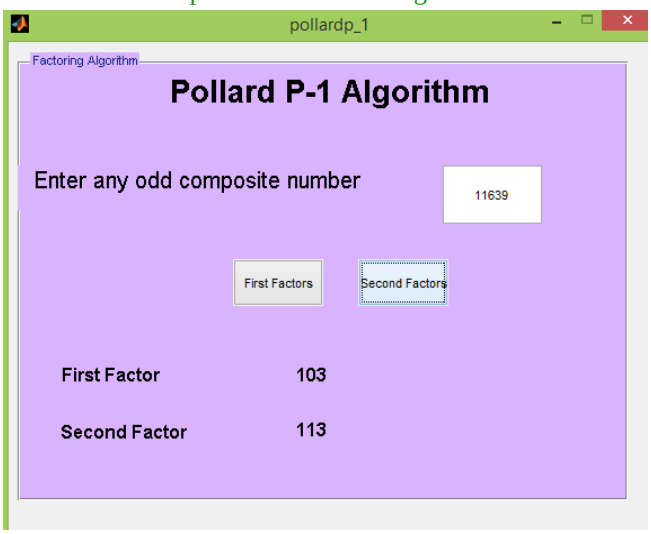
Pollard P-1 algorithm

```
% --- Executes on button press in b1.

function b1_Callback(hObject, eventdata, handles)

% hObject handle to b1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

n=Str2num(get(handles.t1,'String'));
int max a;
int max q;
int max l;
int max d;
b=1000;
a=2;
q=2;
for a=2:n-1
  d = gcd(a, n);
if (d >= 2)
  d1=(n/d);

set(handles.st1,'String',d);
  break;
else
for q=2:b
if(isprime(q)==1)
  l =(log(n)/log(q));
  l=floor(l);
  a =a.^(q*l);
  a =mod(a,n);
  a =a-1;
  d=gcd(a, n);
  d1=(n/d);

set(handles.st1,'String',d);
  break;
else
  break;
end;
  break;
end;
end;
end;

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

n=Str2num(get(handles.t1,'String'));
d=Str2num(get(handles.st1,'String'));
d1=n/d;
set(handles.st2,'String',d1);
```

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Output of Pollard P-1 Algorithm-



Out of Eular Phi Attack-

6. CONCLUSION

Integer factorization play's a very important role in cryptography.RSA algorithm which is used widely in various areas for security purpose uses integer factorization as base As per our knowledge we have demonstrated integer factorization algorithm to find factors and also shown how these algorithms are used to attack on RSA by using Euler phi function.

Eular Phi attack by using Pollard p-1 algorithm

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
n=Str2num(get(handles.t1,'String'));
e=Str2num(get(handles.t2,'String'));
int max a;
int max q;
int max l;
int max d;
a=2;
q=2;
b=1000;
for a=2:n-1
  d = gcd(a, n);
  if (d >= 2)
  d =d;
  d1 =(n/d);
  break;
  else
for q=2:b
  if(isprime(q)==1)
  l =(log(n)/log(q));
  l=floor(l);
  a =a.^(q*l);
  a =mod(a,n);
  a =a-1;
  d=gcd(a, n);
  d1=(n/d);
  break;
end;
  break;
end;
end;
end;
phi=(d-1)*(d1-1);
for i=1:phi
c=e*i;
dinv=mod(c,phi);
con=mod(1,phi);
if(dinv==con)
set(handles.t3,'String',i);
break;
end;
end;
```

References

[1] Adamu Abubakar, Shehu Jabaka, Bello Idrith Tijjani, "CRYPTANALYTIC ATTACKS ON RIVEST, SHAMIR, AND ADLEMAN (RSA) CRYPTOSYSTEM: ISSUES AND CHALLENGES".

[2] B. Padmavati, S. Ranjitha Kumari, "A survey on performance Analysis of DES, AES and RSA algorithm along with LSB substitution technique"2013.

[3] B Hawana "An overview and cryptographic challenges of RSA", IJERMT, 2013.

[4] Daniel M. Gordan,"A survey of fast exponentiation methods" , 1997.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

[5] Michael J. Wiener, "CRYPTANALYSIS OF SHORT RSA SECRET EXPONENTS", 1989 August 3.
[6] Robert S. Boyer and J Strother Moore, "PROOF CHECKING THE RSA PUBLIC KEY ENCRYPTION ALGORITHM". September 1982.
[7] Shafi Goldwasser, Mihir Bellare July 2008,"Lecture Notes on Cryptography".
[8] Bruce Schneier,"The Blowfish Encryption Algorithm", Dr. Dobb's Journal of Software Tools, pp. 4, 38, 40, 98, 99, 1994.
[9] Bruice Schineier, "Applied Cryptography".
[10] Chitra Desai," A Novel Approach for Digital Signature Scheme Based on Solving Two Hard Problems" IJCMSA: Vol. 6, No. 3-4, July-December 2012, pp. 95– 100.
[11] Chitra G. Desai, Rupali Bhakkad, Sonal Sarnaik, "Identifying Quadratic Residuity Using Legendre-Jacobi Symbol".
[12] Dan Boneh, "Twenty Years of Attacks on the RSA Cryptosystem"
[13] David. G. Messerschmitt, "RSA Encryption", 1999.
[14] Dr. Peter Kramer, Instructor ,"ENCRYPTION AND DECRYPTION WITH RSA ALGORITHM MATHEMATICS AND THE COMPUTER", November 8, 1999
[15] James Robinson, Fermat's Little Theorem, "Elaboration On History Of Fremat's Theorem And Implications Of Euler's Generalization By Means Of The Totient Theorem".
[16] Joan Daemen and Vincent Rijmen," Rijndael for AES"., AES Candidate Conference, pp. 343–348, 2000.
[17] Jonathan Katz and Yehuda Lindell, "Introduction to Modern Cryptography".
[18] Lindsay N. Childs "A Concrete Introduction to Higher Algebra", ISBN: 978-0-387-98999-0 (Print) 978-1-4419-8702-0 (Online)
[19] Luca Trevisan, "Cryptography", Lecture Notes from CS276, Spring 2009, Stanford University.
[20] Menezes, Alfred J; van Oorschot, Paul C.; Vanstone, Scott A. (2001),"Handbook of Applied Cryptography",[Online].
[21] Monika Agrawal, Pradeep Mishra, "A Comparative Survey on Symmetric Key Encryption Techniques".
[22] Rajan. S. Jamgekar, Geeta Shantanu joshi,"File encryption and Decryption using RSA".
[23] Rivest, R., A. Shamir and L. Adleman, "A Method for Obtaining Digital Signature and Publickey CryptosystemCommunications"ACM.21:120-126.1978
[24] Sangeeta Patel and Partha Prittam Nayak, "A NOVEL METHOD OF ENCRYPTION USING MODIFIED RSA ALGORITHM AND CHINESE REMAINDER THEOREM",
[25] Satish N. Chalurkar, Nilesh Khochare, B. B. Meshram "Survey on Modular Attack on RSA Algorithm".
[26] Sonal Sarnaik, Nilesh Jaibhai, Rutuja Sontakke, "Traditional Cryptography: A Mathematical overview".
[27] Stinson, Douglas Robert (2006),"Cryptography: Theory and Practice (3rd ed.)", London: CRC Press.
[28] Whitfield Diffie and Martin E. Hellman,"New Directions in Cryptography", IEEE Transactions on Information Theory, 22(6):644-654, 1976.
[29] Yingpu Deng and Yanbin Pan,"An Algorithm For Factoring Integers".