Document Encryption Through Asymmetric RSA Cryptography

1st I Made Ari Dwi Suta Atmaja
Electrical Engineering
Politeknik Negeri Bali
Badung Indonesia
arisuta@pnb.ac.id

2nd I Nyoman Gede Arya Astawa
Electrical Engineering
Politeknik Negeri Bali
Badung Indonesia
arya_kmg@pnb.ac.id

3rd Ni Wayan Wisswani
Electrical Engineering
Politeknik Negeri Bali
Badung Indonesia
wisswani@pnb.ac.id

4th I Made Riyan Adi Nugroho
Electrical Engineering
Politeknik Negeri Bali
Badung Indonesia
maderiyan@pnb.ac.id

5th Putu Wijaya Sunu
Mechanical Engineering
Politeknik Negeri Bali
Badung Indonesia
wijayasunu@pnb.ac.id

6th I Komang Wiratama
Electrical Engineering
Politeknik Negeri Bali
Badung Indonesia
wiratama.komang@pnb.ac.id

2020 International Conference on Applied Science and Technology (iCAST) | 978-1-7281-9567-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/iCAST51016.2020.9557723

***Abstract*—With advances in technology like today, documents can be sent digitally via the internet media. An important problem faced in sending digital documents is that often documents sent can be accessed by parties who do not have the authority over these documents. The solution to this problem is to secure digital documents before transmission. One of the methods to secure data is cryptography. Cryptography with asymmetric keys is the strongest data security technique to use. One of the most widely used asymmetric cryptography is the RSA (Rivest-Shamir-Adleman) algorithm. The type of document that is encrypted is the most commonly attached document when sent e-mails. The document types are .docx, .pptx, .xlsx, .pdf, .jpg and .mp4. In the encryption process, a public key and a private key will be generated which can be sent separately by sending encrypted digital documents. The decryption process for digital documents is carried out from the receiving end of the document using a private key generated in the encryption process. The encryption result has a size larger than the original file size because it has been encoded in another form according to the RSA algorithm. The longer and bigger the input size, the longer it will take required for encryption.**

***Keywords--Cryptography, Encryption, Decryption, Document*.**

I. INTRODUCTION

The advancement of information technology today has provided many benefits in everyday life, both for individuals and organizations. Technological advances are characterized by easy and fast access to information. Each individual can exchange information in seconds even though the distance is quite far. This convenience is of course accompanied by challenges, namely the security of information exchanged. The easier access to information is, the less secure it will be. Information security includes 3 main aspects, namely: confidentiality, data integrity, and availability [1]. Confidentiality is related to the assurance that information can only be accessed by those who have authority over the information. Data integrity relates to information received by authorized recipients that is intact and free from changes by unauthorized parties. Availability, namely the assurance of system services for authorized parties.

The study of data security is cryptography. According to Rodriguez-Henriquez, cryptography is a discipline that studies mathematical techniques related to information security, such as providing security services in the form of confidentiality, data integrity, authentication, and non-repudiation (cannot be denied) [2]. Until now, various cryptographic algorithms has founded to secure data. Cryptographic algorithms have been classified into 2 based on the key, namely the symmetric key algorithm and the asymmetric key. The symmetric key only uses a secret key that is the same between the sender of the message and the recipient of the message. The message is encrypted (encoded) and decrypted (decoded) with a secret key so that both the sender and receiver will share a secret key. In an asymmetric key, the sender and receiver use different keys [3]. If a message is confidential and only has the right to be known by the recipient, then the recipient will give the public key that has been generated from the private key to the sender. The sender then encrypts the information with that public key. When the recipient receives an encrypted message, the recipient will decrypt it using his private key. The use of this asymmetric key is very widely used today because the recipient does not need to give the secret key to other parties so that only the recipient knows the key.

Nowadays, the use of asymmetric keys is becoming more and more common. Various asymmetric key algorithms have been widely known, among which the most widely used is Rivest-Shamir-Adleman (RSA). Various studies on the RSA algorithm have been carried out, including Chandel and Patel conducting a literature review to encrypt image data, and it was found that RSA is good for doing it [4]. Parkin 2003 conducted a study to use RSA as a digital signature in e-commerce transactions [5]. Shen in 2009 carried out an object-based implementation to accelerate the RSA algorithm [6].

In this study, the implementation of RSA was carried out in document form information. This algorithm is implemented to be able to secure data in documents so that they are safe from unauthorized.

II. RESEARCH METHOD

The data in this study were document text files in the .docx, .xlsx, .pdf format, .ppt, .jpg and also .mp4. The document file will be tested with the built application, they will decode with the RSA encryption algorithm. The results of each encryption will be saved into a text file with text

978-1-7281-9567-4/20/$31.00 ©2020 IEEE

format. When the end-user receives the file and will do the decryption, the opposite of the encryption algorithm used previously. For each encrypted text file, the encryption time will be calculated based on the length of the key used and the file size formed after the encryption process.

In the system design process, it is necessary to make the system process flow itself. The document security system application created has an architecture as shown in Figure 1 below:

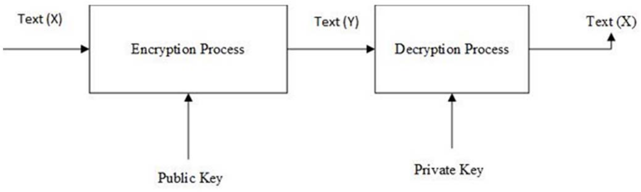


Fig. 1. Asymmetric Key Encryption Application Architecture

In Figure 1, there are 2 users, namely A as the sender and B as the recipient of the message. A has previously been told the private key B. The X text document data which will be referred to as plaintext is input into the system. The system will encrypt X using the RSA algorithm to produce a private key. The encryption result is a ciphertext named Y. B as the recipient will decrypt Y using an existing private key and obtain a text X which has the same content as the plaintext sent by A

For more specific, the process of encryption and decryption of documents is described in the following flowchart:

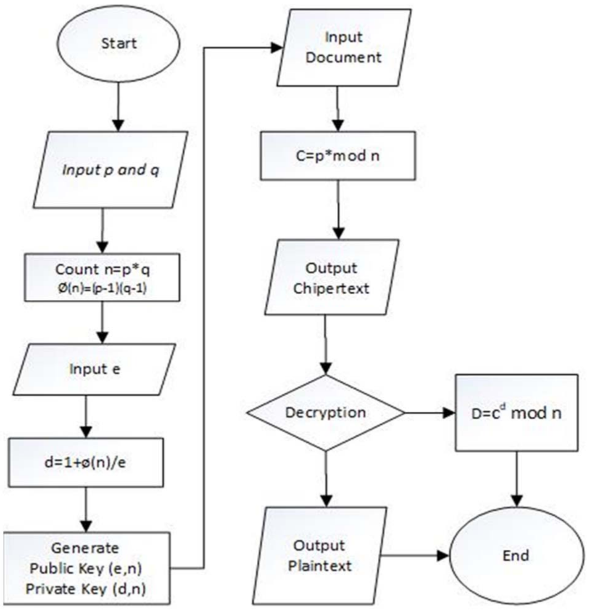


Fig. 2. Flowchart of RSA Asymmetric Key Encryption Application

The process of encryption and decryption of documents following the flow chart in Figure 2 is explained as follows:

1. Key Forming Process:

a. Choose two prime numbers p and q, (try p> q)

b. Calculate n = p x q

c. Calculate Φ (n) = (p-1) x (q-1)

d. Choose a public key that is relatively prime with Φ (n)

e. Calculate the private key with SK = 1 + Φ (n) / PK

2. Encryption Process:

a. Change the plaintext into ASCII code

b. ASCII characters are stored in blocks of bytes.

c. Multiply each block to get the ciphertext with the formula: C = p ^ e mod n

3. Decryption Process:

a. Change the plaintext into ASCII code

b. ASCII characters are stored in blocks of bytes.

c. Multiply each block to get the plaintext with the formula: P = c ^ d mod n

The data used are 6 types of files with different formats and for file size, there are no restrictions.

TABLE I. TYPES OF TEST DOCUMENTS

No	File Type	File Size (KB/MB)	RSA	
			Encryption Time (s)	Decryption Time (s)
1	File 1.docx	Size 1		
2	File 2.pptx	Size 2		
3	File 3.xlsx	Size 3		
4	File 4.pdf	Size 4		
5	File 5.jpg	Size 5		
6	File 6.mp4	Size 6		

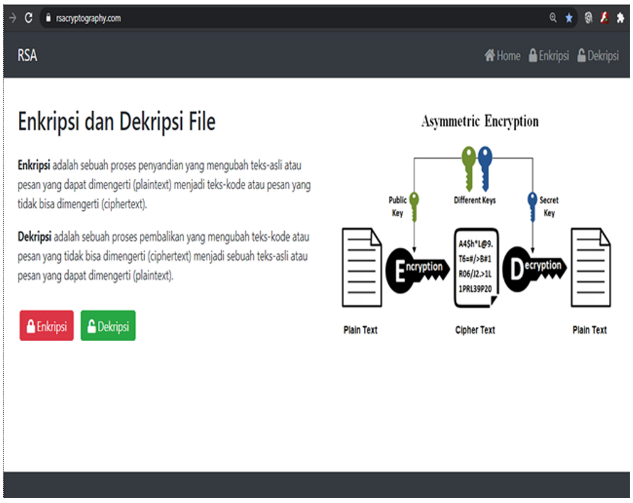
Table 1 above shows the file types that will be used for testing. In the testing process, the time spent in the encryption and decryption process will be calculated for each type of document. System testing will be carried out using several types of documents and videos that are most often transmitted through the internet. The file formats such as *.docx, *.pptx, *.xlsx, *.pdf, *.jpg and *.mp4. This decryption encryption application goal that the document has security so it can't be accessed by unauthorized people.

III. RESULT AND DISCUSSION

The results obtained from this study is a document encryption application with the RSA method. This application was built using the programming language used in building this information system is PHP using the Code Igniter Framework where the storage process is carried out directly into the user's computer internal storage. The results of the research are as follows:

A. Result

The results of this application have been implemented and can be accessed online through the page: https://rsacryptography.com. The following is a view of the document encryption application based on the RSA Asymmetric method, on this page, there are 2 main menus, namely Encryption, and Decryption.



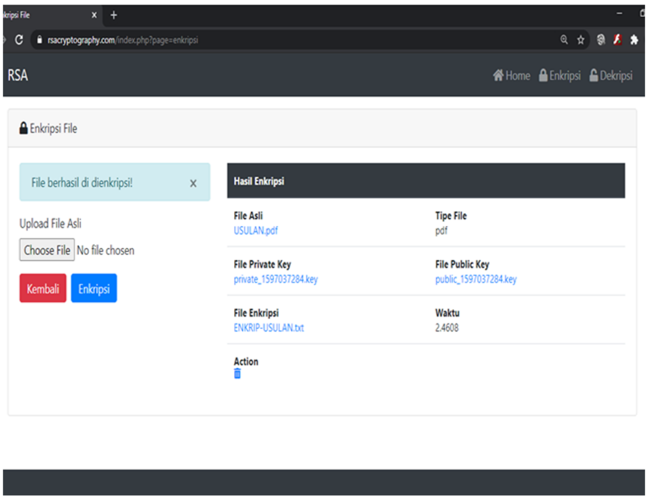
Enkripsi dan Dekripsi File

Enkripsi adalah sebuah proses penyandian yang mengubah teks-asli atau pesan yang dapat dimengerti (plaintext) menjadi teks-kode atau pesan yang tidak bisa dimengerti (ciphertext).

Dekripsi adalah sebuah proses pembalikan yang mengubah teks-kode atau pesan yang tidak bisa dimengerti (ciphertext) menjadi sebuah teks-asli atau pesan yang dapat dimengerti (plaintext).

Fig. 3. Main Page of RSA Asymmetric Encryption Application

For menu 1, namely the encryption process, the document to be encrypted is input into the system and then the encryption process will be carried out. After the encryption process is running, the system will generate information from the document that has been encrypted in the form of the original file name, file type, the resulting private and public key, the name of the encryption file, and the time it takes in the encryption process. As seen in Figure 3 below:



File berhasil di dienkripsi!

Upload File Asli

Choose File No file chosen

Hasil Enkripsi

File Asli: USULAN.pdf

Tipe File: pdf

File Private Key: private_1597037284.key

File Public Key: public_1597037284.key

File Enkripsi: ENKRIP-USULAN.txt

Waktu: 2.4608

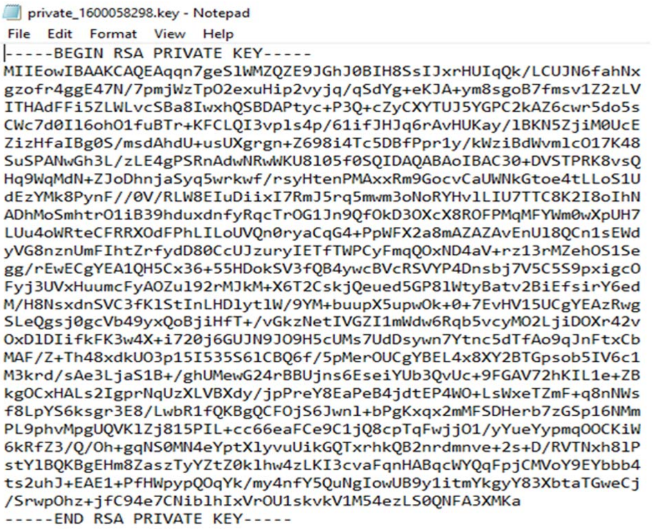
Action

Fig. 4. RSA Asymmetric Key Document Encryption Process

All encrypted files are saved in .txt format. This format is the easiest file format to transmit or send on the internet. After the encryption process is carried out, the document and private key can be downloaded for further use in the decryption process. Encryption files that have been downloaded will be stored directly into the internal storage of the computer used by the user.

The encryption-decryption process produces 2 keys. namely Public Key and Private Key. The Public Key can be known by others. While the private key can only be known by the recipient of the encryption file, where later the private key will be used to decrypt the received document. The private key is generated differently for each encryption process. So each key will not be the same as one another. Private key files can also be downloaded directly and stored in the internal storage of the computer

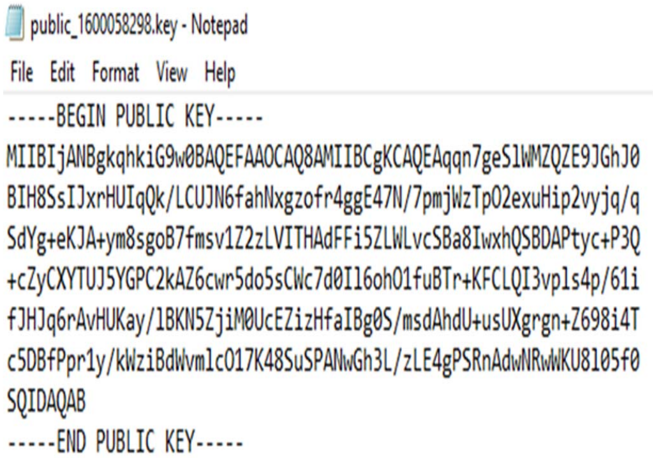
An example of the results of the private key generated from the encryption process is shown in Figure 5 below :



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAqqn7geS1WMZQZE9JGhJ0BIH8SsIJxrHUIqQk/LCUJN6fahNx
gzofr4ggE47N/7pmjWzTpO2exuHip2vyjq/qSdYg+eKJA+ym8sgoB7fmsv1Z2zLV
ITHAdFFi5ZLWLvcSBa8IwxhQSBDAPtyc+P3Q+cZyCXYTUJ5YGPC2kAZ6cwr5do5s
CWc7d0Il6oh01fuBTr+KFCLQI3vpls4p/61ifJHJq6rAvHUKay/1BKN5ZjiM0UcE
ZizHfaIBg0S/msdAhdU+usUXgrgn+Z698i4Tc5DBfPpr1y/kWziBdWvmlcO17K48
SuSPANwGh3L/zLE4gPSRnAdwNRwWKU8105f0SQIDAQABAoIBAC30+DVSTPRK8vsQ
Hq9WqMdN+ZJoDhnjaSyq5wrkwf/rsyHtenPMAxxRm9GocvCaUWNkGtoe4tLLoS1U
dEzYMk8PynF//0V/RLW8EIuDiixI7RmJ5rq5mwm3oNoRYHvlLIU7TTC8K2I8oIhN
ADhMoSmhtrO1iB39hduxdnfyRqcTrOG1Jn9QfOkD3OXcX8ROFPMqMFYWm0wXpUH7
LUu4oWRteCFRRXOdFPhLILoUVQn0ryaCqG4+PpWFX2a8mAZAZAvEnU18QCn1sEWd
yVG8nznUmFIhtZrfydD80CcUJzuryIETfTWPCyFmqQOxND4aV+rz13rMZehOS1Se
gg/rEwECgYEA1QH5Cx36+55HDokSV3fQB4ywcBVcRSVYP4Dnsbj7V5C5S9pxigcO
Fyj3UVxHuumcFyAOZu192rMJkM+X6T2CskjQeued5GP81WtyBatv2BiEfsirY6ed
M/H8NsxdnSVC3fK1StInLHDlytlW/9YM+buupX5upwOk+0+7EvHV15UCgYEAzRwg
SLeQgsj0gcVb49yxQoBjiHfT+/vGkzNetIVGZI1mWdw6Rqb5vcyMO2LjiDOXr42v
OxD1DIifkFK3w4X+i720j6GUJN9JO9H5cUMs7UdDsywn7Ytnc5dTfAo9qJnFtxCb
MAF/Z+Th48xdkUO3p15I535S61CBQ6f/5pMerOUCgYBEL4x8XY2BTGpsob5IV6c1
M3krd/sAe3LjaS1B+/ghUMewG24rBBUjns6EseiYUb3QvUc+9FGAV72hKIL1e+ZB
kgOCxHALs2IgprNqUzXLVBXdy/jpPreY8EaPeB4jdtEP4WO+LsWxeTZmF+q8nNWs
f8LpYS6ksgr3E8/LwbR1fQKBgQCFOjS6Jwnl+bPgKxqx2mMFSDHerb7zGSp16NMm
PL9phvMpgUQVKlZj815PIL+cc66eaFCe9C1jQ8cpTqFwjjO1/yYueYypmqOOCKiW
6kRfZ3/Q/Oh+gqNS0MN4eYptXlyvuUikGQTxrhkQB2nrdmnve+2s+D/RVTNxh81P
stYlBQKBgEHm8ZaszTyYZtZ0klhw4zLKI3cvaFqnHABqcWYQqFpjCMVoY9EYbbb4
ts2uhJ+EAE1+PfHWpypQOqYk/my4nfY5QuNgIowUB9y1itmYkgyY83XbtaTGweCj
/SrwpOhz+jfC94e7CNiblhIxVrOU1skvkV1M54ezLS0QNFA3XMKa
-----END RSA PRIVATE KEY-----
```

Fig. 5. Private Key of RSA Asymmetric Encryption Application

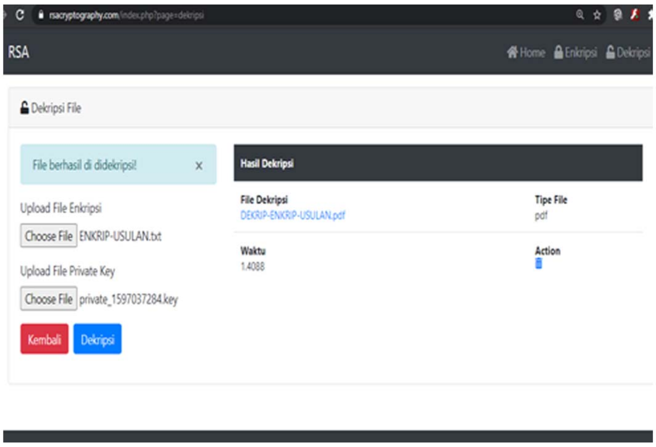
Likewise, the generated public key will be different for each time the encryption process is performed. For an example of the results of the public key generated from the encryption process is shown in Figure 6 below :



```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqqn7geS1WMZQZE9JGhJ0
BIH8SsIJxrHUIqQk/LCUJN6fahNxgzofr4ggE47N/7pmjWzTpO2exuHip2vyjq/q
SdYg+eKJA+ym8sgoB7fmsv1Z2zLVITHAdFFi5ZLWLvcSBa8IwxhQSBDAPtyc+P3Q
+cZyCXYTUJ5YGPC2kAZ6cwr5do5sCWc7d0Il6oh01fuBTr+KFCLQI3vpls4p/61i
fJHJq6rAvHUKay/1BKN5ZjiM0UcEZizHfaIBg0S/msdAhdU+usUXgrgn+Z698i4T
c5DBfPpr1y/kWziBdWvmlcO17K48SuSPANwGh3L/zLE4gPSRnAdwNRwWKU8105f0
SQIDAQAB
-----END PUBLIC KEY-----
```

Fig. 6. Private Key of RSA Asymmetric Encryption Application

In the decryption process, the private key plays a very important role so that the document can be successfully decrypted back into the original document. Each file decryption process uses his private key. in other words each document has its private key, so only the corresponding private key can decrypt the document itself. The decryption process is shown in Figure 7 below:



rsacryptography.com/index.php?page=dekripsi

Dekripsi File

File berhasil di didekripsi!

Upload File Enkripsi: ENKRIP-USULAN.txt

Upload File Private Key: private_1597037284.key

Hasil Dekripsi

File Dekripsi	Tipe File
DEKRIP-ENKRIP-USULAN.pdf	pdf

Waktu	Action
1.4088	

Fig. 7. RSA Asymmetric Key Document Decryption Process

B. Result Evaluation Process

The test carried out is the description test of the 6 types of documents that are most commonly transmitted via internet media. All document types have been successfully encrypted and decrypted. When the encryption file is opened, the original information has been encoded according to the RSA encryption result. The insured message information is shown in Figure 8 below:

Fig. 8. Contents of RSA Encrypted Document Files

All types of documents are tested by obtaining the results as shown in Table II below:

TABLE II. RESULTS OF TESTING ALL TYPES OF DOCUMENTS

No	File Type	File Size (KB)	RSA		Status
			Encryption Time (s)	Decryption Time (s)	
1	Enkripsi.docx	284	0.2808	0.2386	Succeed
2	JSA.pptx	535	0.5034	0.4411	Succeed
3	Rab.xlsx	102	0.1018	0.1131	Succeed
4	Usulan.pdf	1.662	1.5733	1.3877	Succeed
5	Flowchart.jpg	23	0.0274	0.0343	Succeed
6	Video.mp4	1.878	1.7106	1.5132	Succeed

From the testing that has been completed, all types of documents running properly encrypted and decrypted. From the test results, it is also seen that the larger the document size will affect the time in the encryption and decryption process. In this system, there are no restrictions on the size of the documents to be encrypted but in general, in the process of sending documents through the internet, each application has different restrictions. So it is still recommended that the size of the document file to be encrypted can adjust to the application that will be used to transmit the encrypted results. Decryption and encryption with a shorter time is necessary so that the process becomes effective and efficient

IV. CONCLUSION

The conclusion of this research is application software can perform process encryption, decryption, and verification with success, thus providing security that is an aspect of confidentiality and data authentication. In all processes handled by the RSA algorithm, the key size is directly proportional to the processing time/speed. The average Encryption time is faster compared to the decryption time. The encryption result has a size larger than the original file size because it has been encoded in another form according to the RSA algorithm. The longer and bigger the input size, the longer it will take required for encryption.

ACKNOWLEDGMENTS

The authors would like to thank the department of research and community service center of Politeknik Negeri Bali and the Ministry of Research and Technology of Higher Education of the Republic of Indonesia for the financing of this research.

REFERENCES

[1] Stalling, W. 2011. Cryptography and Network Security. Prentice-Hall: New York.

[2] Dwi Liestyowati 2020. Public Key Cryptography. Journal of Physics: Conference Series 1477 052062

[3] Rodriguez-Henriquez, F.; Saqib, N.A; Díaz Pérez, A; Koc, C.K 2007. Cryptography Algorithms on Reconfigurable Hardware. Springer.

[4] Chandel, GS, Patel, P. 2013. A Review: Image Encryption with RSA and RGB Randomized Histograms. International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, 4397-4401

[5] Park, JM. 2003. Constructing Fair-exchange Protocols for E-commerce via Distributed Computation of RSA Signatures. Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing: New York. 172-181

[6] Shen, G, Liu, B, Zheng, X. 2009. Research on Fast Implementation of RSA with Java. Proceedings of the 2009 International Symposium on Web Information Systems and Applications: Nanchang. 186-189