Black box Neuro-Identifier for RSA and AES Cryptosystems

Hamza A. Ali *Computer Engineering Dept.,*
Namareq Odey *Electrical Engineering Dept.,*
College of Engineering, University of Basrah, Iraq.

Abstract:

Black box identification system is a feature of neural networks; hence such system could be utilized for cryptanalysis approach to construct a neuro-identifier. The most widely used and hard to break cryptosystems on the web nowadays are AES and RSA cryptosystems. This paper reports some efforts to incorporate the black box techniques and neuro-identifier for cryptanalysis of these two systems. Theoretical study of a black box model has manifested considerable success in the identification of encryption/decryption keys search.

Keywords: Computer Security, Artificial Neural Networks, Cryptanalysis, Neuro-Identifier.

1. Introduction

A black box can be considered as a device, system, or object with known or specified performance characteristics but unknown or unspecified constituents and means of operation. In cryptography the black box can be used as an approach for system identification when no information is available about the system except its input and output. Cryptographic methods, both symmetric and anti-symmetric are the widely used mean for data protection on computer and communication networks [1]. The strength of any cryptographic systems is either inherently due to the difficulty in the inverting encryption transformation as in the symmetric systems or due the computational difficulty of some parameters as in the case of asymmetric systems. Both systems require key(s) and procedure(s) in order to perform. Therefore, to break any system, the key(s) and procedure(s) have to be identified, hence the problem turned to be an identification problem [2].

An artificial neural network (ANN) is a computing system made up of a number of simple but highly interconnected processing elements. Each element has the capability of processing information by its dynamic state response to external inputs that mimic the operation of biological neuron. Neural networks can be used to represent a dynamical system, where the mathematical characterization of which is unknown. All that is available as a set of labeled input –output data generated by the system at discrete instant of time at some uniform rate ,the neural model operates under the influence of an algorithm that controls

the necessary adjustments to the synaptic weights of the neurons. ANN is an adaptive learning device that can acquire a generalization property through approximation of input/output mapping. Due to this property, ANN lends itself easily as a black box identification system [3-6].

In this paper, an artificial neuro-identifier is studied to be suggested to be used in the search for emulation and cryptanalysis of strong cryptosystem such as RSA and AES. These two systems are describe briefly first in section 2, then the proposed model is described in details in section 3. Results are list and discussed in section 4 and finally the work is concluded in section 5.

2. Cryptographic system

A message M to be encrypted known as plaintext is transformed into a cyphertext C by a function that is parameterized by a secret key K. C is transmitted over an unsecure channel to be received at the receiver who recovers the message M by decrypting C using the same secret key. This process is called secret key or symmetric cryptosystem and shown in fig 1 [7]. The key is usually agreed upon through a key generation and distribution trusted third party. The main problem associated with this system is key distribution system. To avoid this problem, an alternative cryptosystem known as asymmetric or public key cryptosystem is implemented, where two keys are used one is public KU used for encryption and the other is private KR used for decryption, as shown in fig 2. There are too many available systems in practical used of both types; however, the most widely used systems on the internet of the former type are DES and AES while that of the latter type is RSA. AES and RSA are considered in this paper and will be briefly described below.

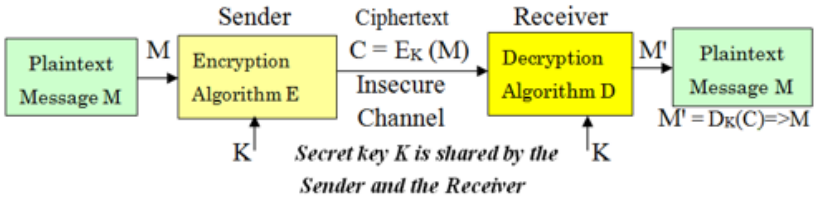


Fig 1. Symmetric Cryptographic system.

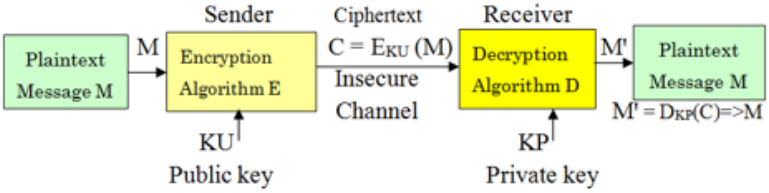


Fig 1. Asymmetric Cryptographic system

2.1 RSA Cryptosystem

One of the oldest public key cryptosystem and still considered as the most powerful and widely used over the internet is the RSA cryptographic system [15], which is proposed in 1977 by Rivest, Shamir and Aldeman. Its security rests on the hard computation problem which is based on the difficulty of factoring large integers. Several other systems were also developed and in use for various applications, such as discrete algorithm, elliptic curve, etc. [1]. RSA system simply involves two key sets one is private KR= {d, n} and the other is public KU= {e, n}, where n, e and d are integers. In this system, every message is divided into blocks of certain length. Each block becomes a sequence of bits representing an integer number M. Encryption and the decryption procedures are modular exponentiations operations according to equations 1 and 2, respectively.

$$C = M^e \bmod n \quad \textbf{(1)}$$

$$M = C^d \bmod n \quad \textbf{(2)}$$

Where M **is the plaintext message** $(1 \le M < n)$, C **is ciphertext. The keys generation algorithm for n, e, and d is shown in fig 3.**

Key generation algorithm
Select p , q (p and q both large prime integers, p≠q)
Calculate n=p*q
Calculate Φ(n)=(p-1)(q-1)
Select e such that gcd(Φ(n),e)=1,1<e<Φ(n);
Calculate d, where d Ξ(1/e)mod Ø(n)
Public key: KU= {e,n}
Private key: KR= {d,n}

Fig 3. The key generation process.

2.2 Advanced Encryption Standard (AES)

AES was announced by National Institute of Standards and Technology (NIST) on November 26, 2001, it is also called Rijndael. This cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum.

AES operates on a 4×4 column-major order matrix of bytes, termed the *state*. Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the

original plaintext using the same encryption key. These processes are illustrated in fig 4.

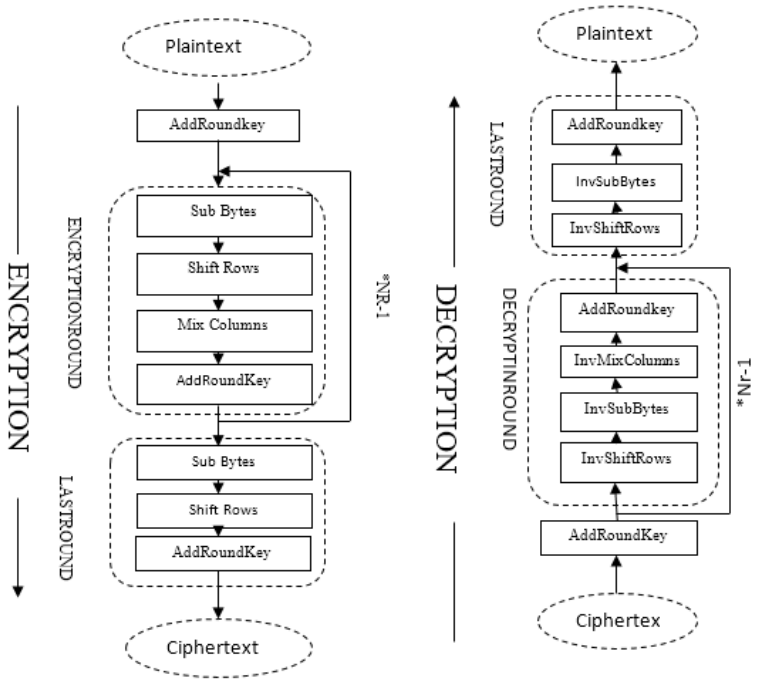


Fig 4. Block diagram of encryption and decryption for AES cryptosystem.

3.1 Neuro Identifier

No golden rule is followed to select the architecture and learning algorithm of neural model, hence it can be made through trial and error. Then once the neural network model is chosen and system input–output data are available, learning can begin to produce the approximate values of synaptic weights. Finding or approximating the intended unknown function using neural network modeling is referred to as neuro identifier. A typical neuro identifier (NID) circuit is shown in fig 2, which is basically a multilayer feed forward (MLFF) neural network. It consists of an input layer followed by nonlinear hidden layer and an output layer. It must be stated that more hidden layers may be implemented for more complex problems.

The optimal size of neural network is the one that uses as little neurons as possible while fulfilling the preset criteria in order to reduce the hardware architecture [8, 9]. It is stated that linear identifiers are not capable of identifying nonlinear systems. Hybrid identifiers can identify nonlinear systems but not complex ones [3, 10].

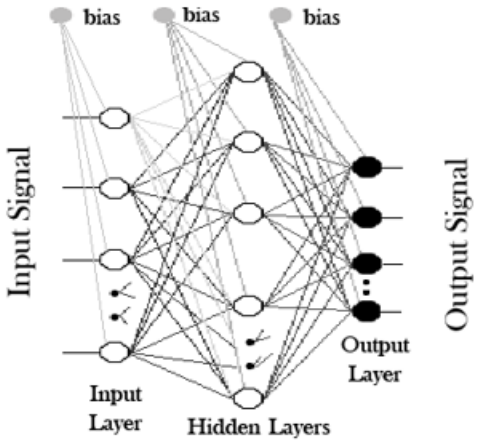


Fig 2. MLFF Neuro Identifier

2.2 NN Training algorithm

Using NID in cryptanalysis: Cryptographic system takes a plaintext and a key as input then produces a ciphertext as output. The proposed NID may be used to identify cryptographic systems in two activity schemes; either for encryption or for decryption. Both of these activities may be simulated in the following two approaches, namely emulation and cryptanalysis

1-Emulation approach:

- **a- Encryption cipher :input data TP,TK and desired output data TC**
- **b- Decryption cipher :input data TC, TK and desired output data TP**

2-Cryptanalysis approach:

Input data: TC, TP desired output data: TK

Emulation means constructing a neural network that can be trained by using plaintext-ciphertext pairs first. Then when it converges to minimum error, it can be used to produce plaintext when given ciphertext only. Such capability has proven the success of black box NID. The cryptanalysis approach means the production of the cryptographic key by providing plaintext-ciphertext to the system.

The training and testing data were obtained using applying the target system under investigation which are in our case RSA and AES systems. The obtained data were split into two lots one for training and the other for testing.

3. Training algorithm

The back propagation algorithm is widely used for NN training. It is not only more general than analytical derivations which handle only the case of special network topologies, but also much easier to follow. It is gradient descent algorithm which can be efficiently implemented in computing systems in which only local information can be transported through the network. Although it is easy to implement but it requires a lot of training data [17].

When the network is a single hidden layer one it is easy to find the optimum solution in selecting the learning rate but when it has more than one hidden layer, it gets complex [11, 13]. For the NN of fig 2, the input to j^{th} hidden neuron $S_j(k)$ is

$$S_j(k) = \sum_{i=1}^{m} W_{ji}.X_i(k) \quad , 1 \le l \le b \qquad \text{..... (1)}$$

Where X_i is the output of i^{th} neuron of the input layer and b is the number of inputs to the j^{th} hidden neuron at the k^{th} epoch. The output of the j^{th} hidden layer $O_j(k)$ is given by

$$O_j(k) = f(S_j(k)) \qquad \text{..... (2)}$$

Where f represents the nonlinear activation function. Therefore, the output of the k^{th} neuron will be

$$y_{NK}(k+1) = \sum V_{jk}(k) \cdot O_j(k) \ , 1 \le K \le c \qquad \text{..... (3)}$$

Where c is the number of network output neurons and W_{ij} & V_{jk} are the weights for the first and hidden layers, respectively, given by the simple gradient principle of the sum squared error of the network output $E(k)$ given by

$$E(k) = (\frac{1}{2}) \cdot \sum_{K=1}^{c} (y_{dK}(k) - y_{NK}(k))^2 \qquad \text{..... (4)}$$

$$W_{ij}(k+1) = W_{ij}(k) + \zeta_W \cdot f'(S_j(k)) \cdot \sum V_{jk}(y_{dk} - y_{Nk}) \cdot X_i(k) \qquad \text{... (5)}$$

$$V_{jK}(k+1) = V_{jK}(k) + \zeta_V \cdot (y_{dk} - y_{Nk}) \cdot O_j(k) \qquad \text{..... (6)}$$

Where m is the number of hidden neurons, ζ_W & ζ_V are the rates for the first weight layers and the hidden layer, respectively, and y_{dk} is the desired k^{th} output

4. Implementation and Results:

In this study the neuro-identifier is implemented for both AES and RSA cryptosystems. The stopping criterion in the training process is taken to be the preset number of iteration (900) which is decided experimentally. Once the training process is terminated, all obtained network weights are saved to be used for the simulation later. The results for these two systems are listed in the following.

4.1 AES cryptosystem

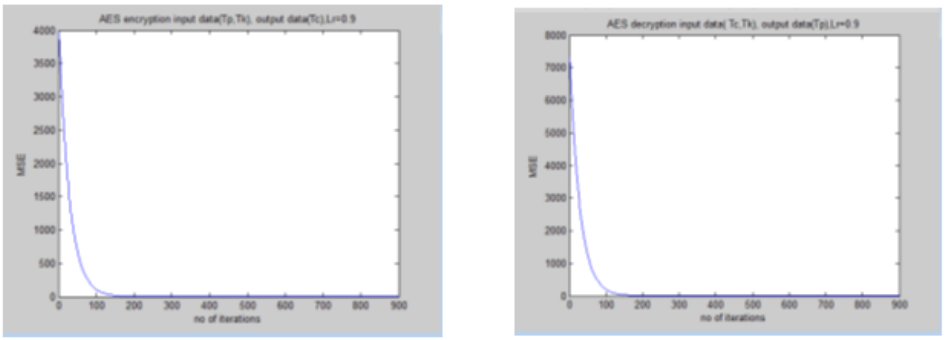
In case of AES 128 bit algorithm is used with key [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]. The AES cryptosystem is run first in order to obtain some plaintext/ciphertext pairs as data for training the neuro-identifier in emulation mode (encryption and decryption). The plaintext is encrypted by the AES system which gives the output as hexadecimal then it is converted to decimal to be used for the FFNN training using aback propagation technique. The obtained Plaintext/Ciphertext pairs are decimal numbers as shown in appendix A. These obtained results gave 100% accuracy, after training it 900 times as shown in table 1 for emulation and table 2 for cryptanalysis. The mean square error MSE curves for emulation (encryption and decryption) are calculated and plotted in figure 5, and cryptanalysis is plotted in figure 6.

Table 1. Creation of emulation model in AES cipher

Method	Mode	Trail sets	NN size	No. Epoch	Execution time (sec)
AES	Encryption	184	22	900	38
	Decryption	184	22	900	34

Table 2. Creation of cryptosystem model

Method	Trail set	NN size	No. Epoch	Execution time(sec)
AES	184	22	900	49



(a) For AES Encryption (b) For AES Decryption

Fig 5. MSE error curves for AES emulation

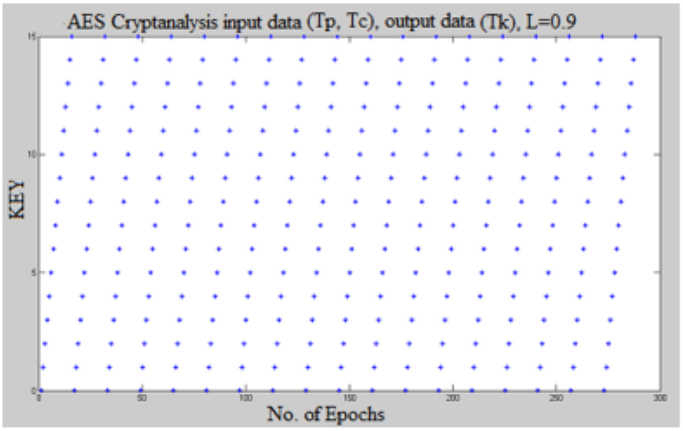


Fig. 6. Cryptanalysis for AES Cipher

4.2 RSA Cryptosystem

Different prime numbers are tried for RSA algorithm but the results listed here are taken for RSA system for p=211 and q=227. Calculating N, choosing e=227 then finding d by using the algorithm shown in figure 3, the public and

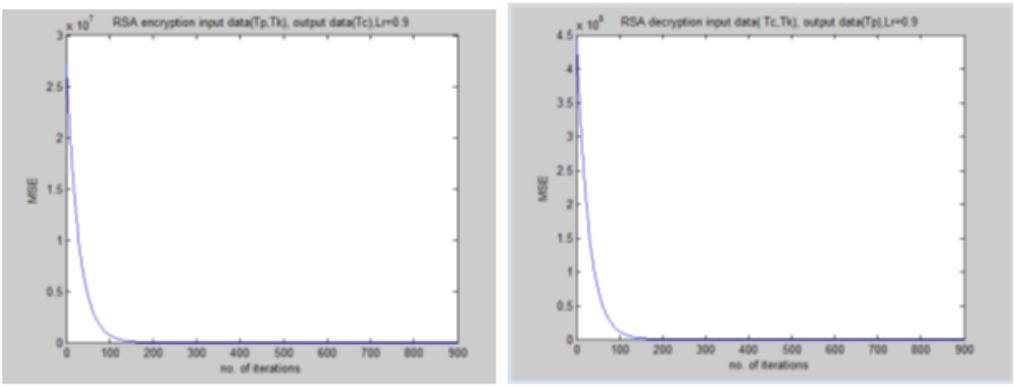
private keys are found to be {257,47897} and {23453,47897}, respectively. The block size in the implemented RSA encryption/decryption algorithm is two characters; therefore, if the plaintext message has odd number of characters, an extra character "x" is added to make it even. The obtained Plaintext/Ciphertext pairs are decimal numbers as shown in appendix B. The obtained results gave 100% accuracy, after training it 900 times as shown in table 3 for emulation and table 4 for cryptanalysis. The mean square error MSE curves for emulation (encryption and decryption) are calculated and plotted in figure 7, and cryptanalysis is plotted in figure 8.

Table 3. Creation of emulation model in RSA cipher

Method	Mode	Trail set	NN size	No. Epoch	Execution time(sec)
RSA	Encry	134	22	900	22
	Decry	134	22	900	23

Table 4. Creation of cryptosystem model in RSA Cipher

Method	Trail set	NN size	No. Epoch	Execution time(sec)
RSA	134	22	900	23



(a) For RSA Encryption (b) For RSA Decryption

Fig 7. MSE error curves for RSA emulation Cipher

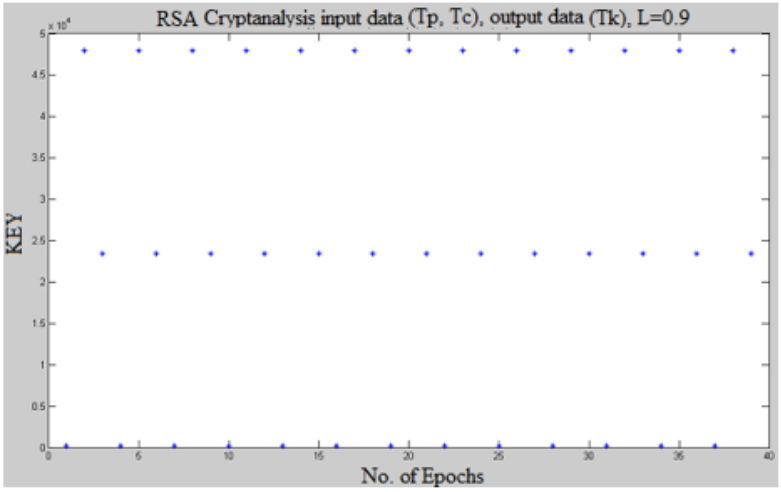


Fig 8. Cryptanalysis for RSA Cipher

5. Conclusions

Neural networks proved to be suitable for black-box NID for constructing Neuro-model for cipher system. Wide range of NN architecture can be made available for the neuro-identifier construction. So many parameters are involved in NN design; such as number of layers, number of neurons in each layer, initial conditions, learning rate, momentum term, weights, activation function type, etc. The choice of these parameters may affect the system accuracy and convergence which might inevitably cause convergence to wrong local minima. Besides, there is no systematic mean to select NN parameter values, therefore, most common they are chosen by practical trial and process.

Due to the generalization feature of NNs, the proposed NID technique can be used to identify many cipher systems and build their equivalence from the input-output data pairs. This can be achieved only by the proper choice of network parameters.

The used back propagation method for training the NID gave fast convergence, stable and 100% accuracy for emulation and cryptanalysis tests. However, care must be taken as back propagation training technique is prone to local minima and sensitive to initial conditions. Other training algorithms such as Lavenberg-Marquardt LM and genetic Algorithms GA may be tried.

References

[1] Schnier, B., "Applied Cryptography, Protocols, Algorithm, and Source Codes in C", John Willey and Sons, 1996.

[2] Lannart, L., "System Identification: Theory for the User", Prentice Hall, 1987.

[3] Alallayah, K., M. Amin, W.A. El-Wahed and A. Alhamami, "Attack and Construction of Simulator for Some of Cipher Systems using Neuro-

Identifier", The International Arab Journal of Information Technology IAJIT, Vol. 7, No. 4. Oct 2010, PP 365-372.

[4] Alallayah K., M. Amin, W.A. El-Wahed and A. Alhamami, "Applying Neural Networks for Simplified Data Encryption Standard (SDES) Cipher System Cryptanalysis", The International Arab Journal of Information Technology IAJIT, Vol. 9, No. 2. March 2012, PP 163-169.

[5] Romariz, A., "Neural Network Applied to Nonlinear Modelling", available at:http://www.ene.unb.br/romariz/, last visited 2012.

[6] Waleed, A.J. R. and H. A. Ali, "Generalization Aspect of Neural Networks on Upgrading Assimilation structure into Accommodating Scheme", Journal of Computer Science, Vol. 5, No. 3, 2009, PP 177-183.

[7] Lester, A. and N. Jonas, "Some Aspects of Neural Net and RelatedModel Structures for Nonlinear System Identification", Journal of Computer Science, Vol.6, No. 1, 2010, PP29-35.

[8] Karayiannis, B. and A. Venetsanopoulos, "Artificial Neural Neural Networks, Learning Algorithms, Performance Evaluationand Applications", Kluwer Academic Publishers, 1995.

[9] Mahmood, K. and A. Wassim, "Nonlinear System Identification using Neural Networks", Computer Journal of NCC, Vol 8, No 37, 2000, PP39-41.

[10] Tanomaru, J., "Comparative Study of Two Neural Network Approaches for Nonlinear Identification", Proceeding of International Symposium on Speech, Image Processing and Neural Networks, Hong Kong, 2002, PP487-490.

[11] Hagan, M. T. and M. B. Menhaj, "Training feed forward network with the Marquardt algorithm," IEEE Trans. on Neural Net., vol. 5, no. 6, pp.989-993, 1994.

[12] Battiti, R., "First- and second-order methods for learning between steepest descent and Newton's method," Neural Computation, vol. 4, no. 2, pp. 141-166, 1992.

[13] Hagan M., Demuth H. and Beale M.: Neural network design. PWS Publishing Co., Boston, 1996.

[14] Amir Abolfazl Suratgar, Mohammad Bagher Tavakoli, and Abbas Hoseinabadi, "Modified Levenberg-Marquardt Method for Neural Networks Training", World Academy of Science, Engineering and Technology, Vol 6, 2005, PP46-48.

[15] Rivest, R., A. Shamir and L. Adlemann, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Commun. ACM, 21, 1978, 120–126.

[16] Stalling, W., "Introduction to Cryptography and Network Security", 4rd Ed., Prentice Hall of India, 2006.

Appendix A: AES data

Plaintext:

Themessagestobeencryptedknownastheplaintextaretransformedbyafunctionthatisparameterizedbyakeytheoutputoftheencryptionprocessknownasacipherttheartofbreakingcipherscalledcryptanalysis

Cipher in hex:

46 8f 83 59 59 98 6d 83 59 46 76 22 83 83 5e 83 5b 18 a7 f4 46 83 86 46 83 8b 3b 5e 7b a9 5e 98 59 46 83 f4
08 98 88 5e 46 83 0c 46 98 18 83 46 18 98 5e 59 93 7b 18 1a 8b 22 a7 98 93 8a 5e 5b 46 88 7b 5e
46 8f 98 46 88 59 f4 98 18 98 1a 83 46 83 18 88 a3 83 8b 22 a7 98 3b 83 a7 46 8f 83 7b 8a 46 7b 93
46 8f 83 83 5e 5b 18 a7 f4 46 88 7b 5e f4 18 7b 5b 83 59 59 3b 5e 7b a9 5e 98 59 98 5b 88 f4 8f 83 46 83
0c 46 46 8f 83 46 18 46 76 93 22 18 83 98 3b 88 5e 6d 5b 88 f4 8f 83 18 59 5b 98 08 08 83 8b 5b 18
a7 f4 46 98 5e 98 08 a7 59 88 59

Cipher in dec:

75 143 131 2 131 89 89 152 109 131 89 75 129 34 131 131 94 91 24 167 244 75 131 139 54 94 129 169 94
152 89 75 143 131 244 8 152 136 94 75 131 12 75 152 24 131 75 24 94 147 129 24 2 131 139 34 167 152
147 138 94 91 74 136 129 94 75 143 152 75 136 89 244 151 24 151 2 131 75 131 24 136 136 131 139 34
167 152 54 131 167 75 143 131 129 138 75 244 138 75 129 147 75 143 131 131 94 91 24 167 244 75 136
129 94 244 24 129 91 131 89 89 54 94 129 196 94 152 89 152 91 136 244 143 131 24 75 131 12 75 75 143
131 152 24 75 129 147 34 24 131 152 54 136 94 109 91 136 244 143 131 24 89 91 152 8 8 131 139 91 24
167 244 75 152 94 152 8 167 89 136 89

Plaintext in dec:

116 104 101 109 101 115 115 97 103 101 115 116 111 98 101 101 110 99 114 121 112 116 101 100 107 110
111 119 110 97 115 116 104 101 112 108 97 105 110 116 101 120 116 97 114 101 116 114 97 110 115 102 111
114 109 101 100 98 121 97 102 117 110 99 116 105 111 110 116 104 97 116 105 115 112 97 114 97 109 101 116 101
114 105 112 101 100 98 121 97 107 101 121 116 104 101 111 117 116 112 117 116 112 117 116 111 102 116 104 101
101 110 99 114 121 112 116 105 111 110 112 114 111 99 101 115 115 107 110 111 119 110 97 115 97 99 105 112 104
101 114 116 101 120 116 116 104 101 97 114 116 111 102 98 114 101 97 107 105 110 103 99 105 112 104 101 114
115 99 97 108 108 101 100 99 114 121 112 116 97 110 97 108 121 115 105 115

Appendix B: RSA Cryptosystem Data

Plaintext:

Theonlycatchisthatweneedalgorithmsthatindeedsatisfyallbeenthreerequirementsduetothepotentialadvantagesofpublickeycryptographymanyresearchersarehardatworkandsomealgorithmshavealreadybeenpuplishedtheideahereisthatsomeoneownesalargenumberofobjectseachwithdifferentweight

Plaintext in dec:

29800 25967 28268 31075 24948 25448 26995 29800 24948 30565 28261 25956 24940 26479
29289 9800 28019 29800 24948 26990 25701 25956 29537 29801 29542 31073 27756 25189
25966 29800 29285 25970 25969 30057 29285 28005 28276 29540 30053 29807 29800 25968
28532 25966 29801 24940 24932 30305 28276 24935 25971 28518 28789 25196 26979 27493
31075 29305 28788 28519 29281 28776 31085 24942 31090 25971 25953 29283 26725 29299
24946 25960 24946 25697 29815 28530 27489 28260 29551 28005 24940 26479 29289 29800
28019 26721 30309 24940 29285 24932 31074 25957 28272 30064 27753 29544 25956 29800
25961 25701 24936 25970 25961 29556 26721 29811 28525 25967 28261 28535 28261 29537
27745 29287 25966 30061 25189 29295 26223 25194 25955 29811 25953 25448 30569 29800
25705 26214 25970 25966 29815 25961 26472 29816

Ciphertext in dec.:

44545 41341 19714 22067 36852 7670 28178 44545 36852 21772 30921 21419 9352 2477 46830 44545 24268
44545 36852 29240 9353 21419 14240 39595 30432 6585 21657 38345 16132 44545 19837 11560 10839 29122
19837 19289 7855 3587 8199 39712 44545 4206 1148 16132 39595 9352 11133 46940 7855 2541 9373
32240 9379 18159 3202 23152 22067 23984 18066 12399 25336 21676 2664 24684 3978 9373 32494 24970 8476
36542 47037 23935 47037 19447 42984 4185 31248 42848 20070 19289 9352 2477 46830 44545 24268 1073 4015
9352 19837 11133 14512 18973 36244 37644 23001 9056 21419 44545 41173 9353 29988 11560 41173
41012 1073 19861 21745 41341 30921 32632 30921 14240 31595 28628 16132 41987 38345 47525 22175
31480 15029 19861 32494 7670 18813 44545 7798 45925 11560 16132 42984 41173 23145 7396