Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Beardsmore	Student ID:	15504319
Other name(s):	Connor		
Unit name:	Fundamental Concepts of Cryptography	Unit ID:	ISEC2000
Lecturer / unit coordinator:	Wan-Quan Liu	Tutor:	Antoni Liang
Date of submission:	19/05/17	Which assignment?	2 (Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: ____________ Date of signature: 19/05/17

(By submitting this form, you indicate that you agree with all the above text.)

FCC200 Report
RSA Cryptosystem Implementation

Connor Beardsmore - 15504319

Curtin University
Science and Engineering
Perth, Australia
May 2017

Binary Modular Exponentiation

```
/**************************************************************************
* FILE: RSA.java
* AUTHOR: Connor Beardsmore - 15504319
* UNIT: FCC200
* PURPOSE: Performs RSA public-key encryption or decryption on a given file
*   LAST MOD: 01/05/17
*   REQUIRES: NONE
**************************************************************************/

import java.util.*;
import java.io.*;

public class RSA
{
//---------------------------------------------------------------------------

    public static void main( String[] args )
    {

    }

//---------------------------------------------------------------------------
    //NAME:
    //IMPORT:
    //EXPORT:
    //PURPOSE:

    public static int modularExpo( int a, int b, int n )
    {
        return 0;
    }

//---------------------------------------------------------------------------
}
```

use the code to print out the value given.

RSA Implementation

Original Test File

```
\subsection{AFS Algebras}
###&&&
The Iris dataset is used as an illustrative example for AFS algebras through
this paper. It has 150 samples which are evenly distributed in three
classes and 4 features of sepal length($f_1$), sepal
width($f_2$), petal length($f_3$), and petal width($f_4$). Let a
pattern $x=(x_{1},x_{2},x_{3},x_{4})$, where $x_{i}$ is the $i$th
feature value of $x$. The following three linguist fuzzy rules have been obtained for Class 1 to build the


\subsection{Shannon��s Entropy}
Let $X$ be a discrete random variable with a finite set containing $N$ symbols
$x_{0}, x_{1}, \ldots, x_{N}$. If an output $x_{j}$ occurs with probability $p(x_{j})$, then the
amount of information associated with the known occurrence of the output $x_{j}$ is defined as
\begin{equation}
I(x_{j}) = -log_{2} p(x_{j})
\end{equation}
Based on this, the concept of Shannon��s entropy is defined as follows:
)))))~~~~~
```

Figure 1: RSA Plaintext

Encrypted Test File

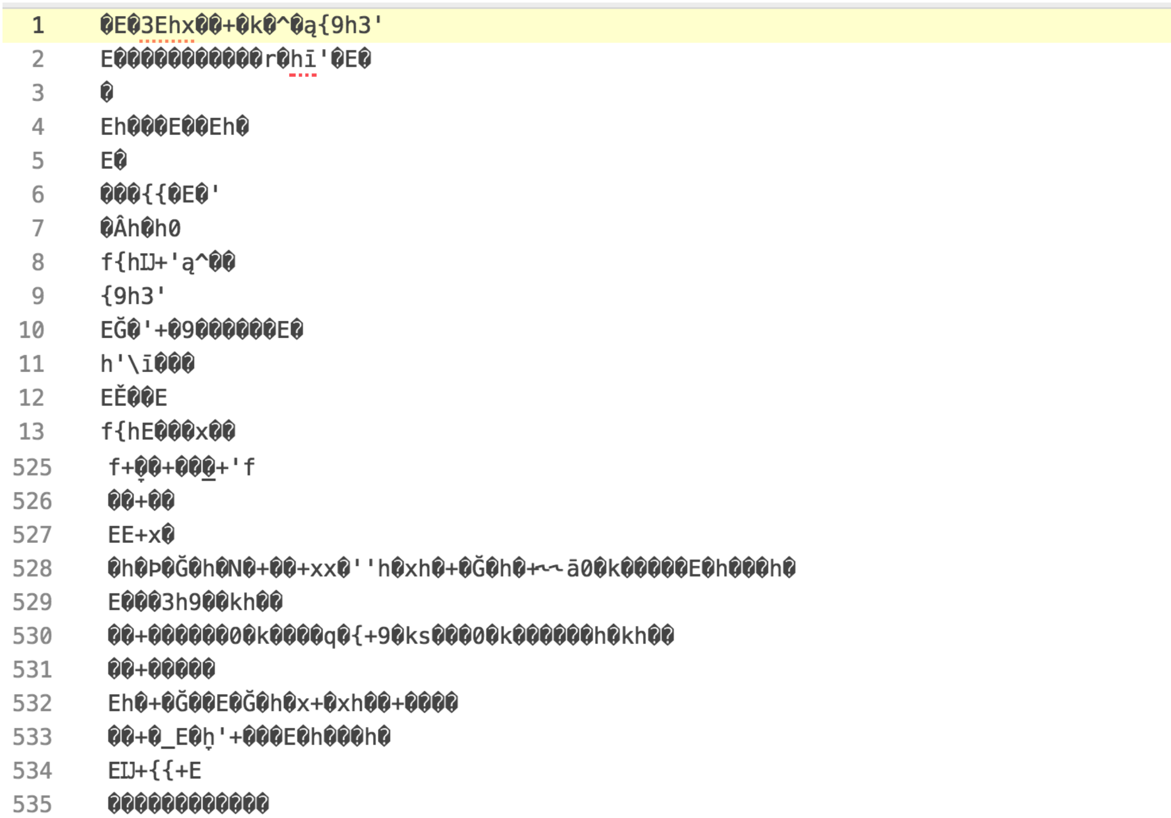
```
�E�3Ehx��+�k�^�ą{9h3'
E�����������r�hī'�E�
�
Eh���E��Eh�
E�
���{{�E�'
�Âh�h0
f{hD+'ą^��
{9h3'
EĞ�'+�9�������E�
h'\ī���
EĚ��E
f{hE���x��
f+��+���+'f
��+��
EE+x�
�h�Þ�Ğ�h�N�+��+xx�''h�xh�+�Ğ�h�+~~ā0�k�����E�h���h�
E���3h9��kh��
��+�������0�k����q�{+9�ks���0�k������h�kh��
��+�����
Eh�+�Ğ��E�Ğ�h�x+�xh��+����
��+�_E�h'+���E�h���h�
ED+{{+E
������������
```

Figure 2: RSA Ciphertext

Recovered Test File

```
\subsection{AFS Algebras}
###&&&
The Iris dataset is used as an illustrative example for AFS algebras through
this paper. It has 150 samples which are evenly distributed in three
classes and 4 features of sepal length($f_1$), sepal
width($f_2$), petal length($f_3$), and petal width($f_4$). Let a
pattern $x=(x_{1},x_{2},x_{3},x_{4})$, where $x_{i}$ is the $i$th
feature value of $x$. The following three linguist fuzzy rules have been obtained for Class 1 to build the


\subsection{Shannon��s Entropy}
Let $X$ be a discrete random variable with a finite set containing $N$ symbols
$x_{0}, x_{1}, \ldots, x_{N}$. If an output $x_{j}$ occurs with probability $p(x_{j})$, then the
amount of information associated with the known occurrence of the output $x_{j}$ is defined as
\begin{equation}
I(x_{j}) = -log_{2} p(x_{j})
\end{equation}
Based on this, the concept of Shannon��s entropy is defined as follows:
)))))~~~~~
```

Figure 3: RSA Recovered Plaintext

Additional Questions

Signature Forgery

lalasdlasksjdksjs

Birthday Attack

lalasdlasksjdksjs

RSA Source Code

RSA.java

```
/***************************************************************************
* FILE: RSA.java
* AUTHOR: Connor Beardsmore - 15504319
* UNIT: FCC200
* PURPOSE: Performs RSA public-key encryption or decryption on a given file
*   LAST MOD: 01/05/17
*   REQUIRES: NONE
***************************************************************************/

import java.util.*;
import java.io.*;

public class RSA
{
//----------------------------------------------------------------------------

    public static void main( String[] args )
    {

    }

//----------------------------------------------------------------------------
    //NAME:
    //IMPORT:
    //EXPORT:
    //PURPOSE:

    public static int modularExpo( int a, int b, int n )
    {
        return 0;
    }

//----------------------------------------------------------------------------
}
```

References

Liu, Wan-Quan. 2017. *Lecture 3: Coding*. Curtin University.

Schneier, Bruce. 1996. *Applied Cryptography*. 5th ed. John Wiley & Sons Inc.

Stallings, William. 2011. *Cryptography and Network Security: Principles and Practice*. 5th ed. Prentice Hall.

Wiegand, Heiko, Thomas & Schwarz. 2011. "Source Coding: Part I of Fundamentals of Source and Video Coding". *Foundations and Trends in Signal Processing* 4.