CPSC/PMAT 418 — Introduction to Cryptography
ASSIGNMENT 2

SET: Friday, Oct. 7, 2016 DUE: Friday, Oct. 28, 2016 at 11:55 PM

*** * * Please read the entire assignment carefully! * * ***

Problems 1-4 are required for both CPSC 418 and PMAT 418. Problem 5 is required for PMAT 418 only, but CPSC 418 students may do this problem for extra credit. Problem 6 is required for CPSC 418 only, but PMAT 418 students may do this problem for extra credit. Problem 7 is a bonus problem for everyone.

Bonus Credit Policy

- Bonus credit will only be awarded for perfect or near perfect solutions.
- The maximum number of bonus marks awarded is 25.

Group work is *not* permitted on any of the problems. If you consult with other students, make sure that the answers presented are written in your own words. Be sure to show all your work.

Your answers to the written problems must be *typeset* and submitted in PDF format. A LaTeX class file, template and sample assignment can be downloaded from the "latex" page on the course website. There is also a link to that page from the "assignments" page. Place the class file in the same directory as you assignment LaTeX source file. Be sure to use the appropriate customized assignment template (CPSC or PMAT); do *not* use the generic assigment class that is part of your LaTeX distribution.

Submission Instructions: Please follow these instructions carefully as assignments not conforming to the submission procedure will be penalized.

- Be sure that your name, student number, homework number and course number (CPSC 418 or PMAT 418) appear on the top of the first page of your assignment. Enter this information to the LaTeX template in the appropriate lines.
- Your solution must must be submitted via D2L through the course page for CPSC/PMAT 418 by the specified deadline. **Late submissions will be severely penalized.**
- Your written solutions to **Problems 1-5 and 7** must be submitted in one single file in PDF format. If you did any programming for any of these problems, your computer code may be submitted in one or more separate files, but your written solutions must include accompanying explanations and documentation.
- For **Problem 6**, submit the written part of your solution as a separate README file. This file may be in TXT format and need not be done in LaTeX. Do *not* include the written portion of the programming problem in the PDF file containing your written answers to the other problems.

Written Problems for CPSC 418 and PMAT 418

Problem 1 (Conditional entropy, 10 marks)**.**

Let X and Y be two random variables. Recall that the joint probability $p(x, y)$ is the probability that $p(X = x)$ and $p(Y = y)$; it satisfies $p(x, y) = p(x|y)p(y)$. The *conditional entropy* or *equivocation* of X given Y is defined to be

$$H(X|Y) = \sum_y \sum_x p(x, y) \log_2 \left(\frac{1}{p(x|y)} \right) = \sum_y p(y) \sum_x p(x|y) \log_2 \left(\frac{1}{p(x|y)} \right) ,$$

where the sums $\sum_x$ and $\sum_y$ run over all outcomes of X and of Y, respectively, such that $p(x|y) > 0$. Informally, the equivocation $H(X|Y)$ measures the uncertainty about X given Y. (Shannon measured the security of a cipher in terms of the key equivocation $H(K|C)$, i.e. the amount of information about a key K that is not revealed by a given ciphertext C.)

(a) (4 marks) Consider a plaintext space $\mathcal{M} = \{M_1, M_2, M_3, M_4\}$, with corresponding ciphertext space $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$. Suppose that each plaintext and each ciphertext is equally likely, i.e. $p(M_i) = p(C_j) = 1/4$ for $1 \leq i, j \leq 4$. Now suppose that each ciphertext C_j narrows down the choice of corresponding plaintext M_i to two of the four possibilities as follows:

C_1: M_1 or M_2
C_2: M_3 or M_4
C_3: M_2 or M_3
C_4: M_1 or M_4

Compute $H(\mathcal{M}|\mathcal{C})$.

(b) (4 marks) Suppose a cryptosystem provides perfect secrecy, and that $p(M) > 0$ for all $M \in \mathcal{M}$. Prove that $H(\mathcal{M}|\mathcal{C}) = H(\mathcal{M})$.

(c) (2 marks) Does the example of part (a) provide perfect secrecy? Explain your answer?

Problem 2 (Binary polynomial arithmetic, 20 marks)**.**

In this problem, we consider different types of modular arithmetic on polynomial with coefficient in GF(2), the set $\{0, 1\}$ with arithmetic modulo 2.

(a) Recall that a polynomial is *irreducible* if it does not have a factorization into polynomials of smaller positive degree, and *reducible* otherwise.

(i) (2 marks) List all the polynomials of degree 3 with coefficients in GF(2) = $\{0, 1\}$ in lexicographical order.

(ii) (3 marks) List all the reducible polynomials of degree 3 with coefficients in GF(2). For each of these polynomials, provide a proof of reducibility.

(iii) (3 marks) List all the irreducible polynomials of degree 3 with coefficients in GF(2). For each of these polynomials, provide a proof of irreducibility.

(b) Consider the finite field GF(2^4) obtained via arithmetic modulo the irreducible polynomial $p(x) = x^4 + x + 1$.

(i) (2 marks) Compute the product of $f(x)g(x)$ in GF(2^4) where $f(x) = x^2 + 1$ and $g(x) = x^3 + x^2 + 1$.

(iii) (2 marks) Find the inverse of $f(x) = x$ in GF(2^4), i.e. the polynomial $g(x)$ such that $f(x)fg(x) = 1$ in GF(2^4).

Hint: Be smart about this. If you use the fact that $p(x) = 0$ in GF(2^4) in a clever way, you don't need to resort to the extended Euclidean algorithm for polynomials.

(d) Recall that the MixColumns operation in AES performs arithmetic on 4-byte vectors using the polynomial $M(y) = y^4 + 1$; remember that for this type of arithmetic, we have $y^4 = 1$.

(i) (2 marks) Formally prove that in this arithmetic, multiplication of any 4-byte vector by y is a circular left shift of the vector by one byte.

(ii) (2 marks) Prove that in this arithmetic, $y^i = y^j$ for any integer $i \geq 0$, where $j \equiv i \pmod 4$ with $0 \leq j \leq 3$.

(iii) (4 marks) Formally prove that multiplication of any 4-byte vector by y^i ($i \geq 0$) is a circular left shift of the vector by j bytes, where $j \equiv i \pmod 4$ with $0 \leq j \leq 3$.

Problem 3. (Arithmetic with the constant polynomial of MixColumns in AES, 17 marks)
Recall that the MixColumns operation in AES uses the polynomial

$$c(y) = (03)y^3 + (01)y^2 + (01)y + (02) \ ,$$

where the coefficients of $c(y)$ are bytes written in hexadecimal (i.e. base 16) notation. Arithmetic involving this polynomial requires the computation of products involving the bytes (01), (02) and (03) in the AES field GF(2^8).

(a) (3 marks) Write the values (01), (02), (03) as their respective polynomial representatives $c_1(x)$, $c_2(x)$ and $c_3(x)$.

(b) Let $b = (b_7\, b_6 \cdots b_1\, b_0)$ be any byte.

(i) (2 marks) Let $d = (01)b$ be the product of the bytes (01) and b in the AES field $GF(2^8)$ (with arithmetic modulo $m(x) = x^8 + x^4 + x^3 + x + 1$), and write $d = (d_7\, d_6 \cdots d_1\, d_0)$. Provide exact equations for the bits d_i, $0 \le i \le 7$, in terms of the bits b_i of b.

(ii) (3 marks) Provide analogous equations as in part (i) for the byte product $d = (02)b$.

(iii) (3 marks) Provide analogous equations as in part (i) for the byte product $d = (03)b$.

(c) The MixColumns operation computes products of the form $s(y)c(y) \mod y^4 + 1$ where $c(y) = (03)y^3 + (01)y^2 + (01)y + (02)$ and $s(y) = s_3y^3 + s_2y^2 + s_1y + s_0$ is a polynomial whose coefficients are bytes. Recall that $y^4 = 1$ in this type of arithmetic.

(ii) (4 marks) Let $s(y) = s_3y^3 + s_2y^2 + s_1y + s_0$ be a polynomial whose coefficients are bytes. Compute $t(x) \equiv s(x)c(x) \mod y^4 + 1$. The result should be a polynomial of the form $t(x) = t_3y^3 + t_2y^2 + t_1y + t_0$ where t_3, t_2, t_1, t_0 are bytes. Provide exact equations for the bytes t_i, $0 \le i \le 3$, in terms of the bytes s_i. The equations should consist of sums of byte products of the form $01s_i$, $02s_i$, $03s_i$, $0 \le i \le 3$. You need NOT compute these individual byte products as you did in part (b).

(iii) (2 marks) Write your solution for part (ii) in matrix form; i.e. give a 4×4 matrix C such that

$$\begin{pmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \end{pmatrix} = C \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

Note that this yields the matrix representation of MixColumns presented (without proof) in class.

Problem 4 (Error propagation in block cipher modes, 12 marks)**.**
Error propagation is often an important consideration when choosing a mode of operation in practice. In this problem, you will analyze the error propagation properties of an arbitrary block cipher in various such modes; note that these properties are independent of the cipher used.

(a) Suppose Alice wants to send a sequence of message blocks $M_0, M_1, M_2, \ldots$ to Bob, encrypted to ciphertext blocks $C_0, C_1, C_2, \ldots$ using some fixed key K. Assume that an error occurs during transmission of a particular block of ciphertext C_i. Justifying all your answers, explain which plaintext blocks are affected upon decryption of this (faulty) ciphertext block C_i by Bob when the cipher is used in

(i) (2 marks) ECB mode?

(ii) (2 marks) CBC mode?

(iii) (2 marks) OFB mode?

(iv) (2 marks) CFB mode with one register?

(v) (2 marks) CTR mode?

(b) (2 marks) Suppose now that an error occurs in a particular plaintext block M_i before Alice encrypts it and sends the corresponding ciphertext C_i to Bob. Upon decryption of C_i, which plaintext blocks M_j are affected when using the cipher in CBC mode?

Written Problems for PMAT 418 only

Problem 5 (Shannon's Theorem and cryptanalysis of linear ciphers, 41 marks)**.**

(a) (Shannon's Theorem)

In this part of the problem, you will prove the following celebrated result:

> **Shannon's Theorem**. Consider a cryptosystem with plaintext space $\mathcal{P}$, ciphertext space $\mathcal{C}$, key space $\mathcal{K}$, and suppose that $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$. Then the system provides perfect secrecy if and only if
> (S1) every key is chosen with equal probability $1/|\mathcal{K}|$ and
> (S2) for every message $M \in \mathcal{M}$ and ciphertext $C \in \mathcal{C}$, there exists *exactly* one key $K \in \mathcal{K}$ that encrypts M to C, i.e. $E_K(M) = C$.

Note that this theorem provides a one-line proof that the one-time pad provides perfect secrecy if keys are chosen with equal likelihood, which is what you proved in Problem 5 (a) of Assignment 1.

Throughout part (a) of this problem, consider a cryptosystem with plaintext space $\mathcal{P}$, ciphertext space $\mathcal{C}$, key space $\mathcal{K}$, and encryption functions $E_K : \mathcal{M} \to \mathcal{C}$ with corresponding decryption functions D_K ($K \in \mathcal{K}$). We also assume that $p(C) > 0$ for all $C \in \mathcal{C}$.

Parts (i) and (ii) below establish some general results about cryptosystems providing perfect secrecy.

(i) (2 marks) Assume first that the system provides perfect secrecy. Use the definition of $p(K)$ to formally prove that for every $M \in \mathcal{M}$ and $C \in \mathcal{C}$, there exists at least one key $K \in \mathcal{K}$ that encrypts M to C, i.e. $E_K(M) = C$.

(ii) (3 marks) Continue to assume first that the system provides perfect secrecy. Conclude from part (i) that for any $M \in \mathcal{M}$, the map $\Phi_M : \mathcal{K} \to \mathcal{C}$ via $K \mapsto E_K(M)$ is surjective. Conclude further that $|\mathcal{M}| \leq |\mathcal{C}| \leq |\mathcal{K}|$.

The next four parts establish Shannon's Theorem. So assume for the remainder of this problem that $|\mathcal{P}| = |\mathcal{C}| = |\mathcal{K}|$.

(iii) (3 marks) Suppose first that the system provides perfect secrecy. Prove that the map Φ_M of part (ii) is a bijection for any $M \in \mathcal{M}$. Conclude that (S2) holds.

(iv) (3 marks) Continue to assume that the system provides perfect secrecy. Fix any ciphertext C_0, so $p(C_0) = p(C_0|M)$ for all $M \in \mathcal{M}$. Prove that $p(K) = p(C_0)$ for all $K \in \mathcal{K}$. Conclude that (S1) holds.

(v) (3 marks) Suppose now that (S1) and (S2) hold, and let $C \in \mathcal{C}$. Prove that the elements $D_K(C)$, $K \in \mathcal{K}$, are distinct and collectively form all of $\mathcal{M}$. Conclude that

$$\sum_{K \in \mathcal{K}} p(D_K(C)) = 1 \ .$$

(vi) (3 marks) Continue to assume that (S1) and (S2) hold. Prove that $p(C|M) = 1/|\mathcal{K}|$ and $p(C) = 1/|\mathcal{K}|$ for all $C \in \mathcal{C}$ and $M \in \mathcal{M}$. Conclude that the system provides perfect secrecy.

(b) (Cryptanalysis of a class of linear ciphers)

For this part, let $\mathbb{F}_2 = \{0, 1\}$ (alternatively denoted as GF(2)) be the field of 2 elements with the usual arithmetic modulo 2. For any $n \in \mathbb{N}$, let $\mathrm{GL}_n(\mathbb{F}_2)$ denote the set of invertible $n \times n$ matrices with zeros and ones as entries. Calculations involving such matrices work exactly the same as the familiar linear algebra you learned in first year, except that arithmetic using real numbers is replaced by arithmetic modulo 2.

Fix $n \in \mathbb{N}$ and consider the class of linear cryptosystems with $\mathcal{M} = \mathcal{C} = \{0, 1\}^n$, $\mathcal{K} = \mathrm{GL}_n(\mathbb{F}_2)$, and for all plaintexts $\vec{m}$ (interpreted as n-bit column vectors with entries 0 and 1), encryption under a key matrix K is

$$E_K(\vec{m}) = K\vec{m} \ . \tag{1}$$

Then for all ciphertexts $\vec{c}$, decryption under K is obviously

$$D_K(\vec{c}) = K^{-1}\vec{c} \, .$$

(i) (3 marks) Prove that a transposition cipher that operates on bit strings of length n is a a special case of a linear cipher as given in (1) whose key matrices are *permutation matrices*, i.e. matrices with exactly one one and $n - 1$ zeros in each row and in each column.

(ii) (2 marks) Explain how a cryptanalyst Eve can mount a chosen plaintext attack on a cipher of the form (1). The goal of this attack is to chose one or more plaintexts, obtain their encryptions under one unknown key matrix K, and derive K. How should Eve choose her plaintexts, and how many does she need to choose in order to be successful?

The set $\mathbb{F}_2^n = \{0, 1\}^n$ consisting of all n-bit vectors with entries 0 or 1 is an n-dimensional vector space over $\mathbb{F}_2$ (with the usual canonical basis, for example). So linear combinations of vectors in $\mathbb{F}_2^n$ have coefficients 0 and 1.

(iii) (3 marks) Let $\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_i$ be any collection of i linearly independent vectors in $\mathbb{F}_2^n$ for some i with $1 \leq i \leq n$. Prove that there are 2^i vectors $\vec{m}_{i+1} \in \mathbb{F}_2^n$ such that the vectors $\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_i, \vec{m}_{i+1}$ are linearly dependent.

(iv) (2 marks) With the notation of part (iii), how many vectors $\vec{m}_{i+1} \in \mathbb{F}_2^n$ are there so that the vectors $\vec{m}_1, \vec{m}_2, \ldots, \vec{m}_i, \vec{m}_{i+1}$ are linearly independent?

(v) (4 marks) Prove that the number of sets consisting of n linearly independent vectors in $\mathbb{F}_2^n$ is

$$\frac{1}{n!} \prod_{i=0}^{n-1} (2^n - 2^i) \ .$$

(vi) (4 marks) Prove that the the probability of randomly choosing n linearly independent vectors in $\mathbb{F}_2^n$ is

$$P_n = \frac{\displaystyle\prod_{i=0}^{n-1} (2^n - 2^i)}{\displaystyle\prod_{i=0}^{n-1} (2^n - i)} \ .$$

(vii) (2 marks) Suppose $n = 4$ and we choose 4 vectors at random in $\mathbb{F}_2^4$. What is the probability that these 4 vectors are linearly dependent?

(viii) (4 marks) Part (ii) considered a *chosen* plaintext attack, whereas we now consider the scenario of mounting a *known* plaintext attack on a linear cipher as given in (1). Given a set of known plaintext/ciphertex pairs where all the plaintexts were encrypted

to their respective corresponding ciphertexts using the same unknown key matrix K, the goal of this attack is to uncover K.

Assume that ciphertexts in $\mathbb{F}_2^n$ – and hence also plaintexts, since encryptions are bijections — are randomly and independently distributed. Explain how Eve can use multiple attempts at a known plaintext attack to find a key matrix. What is the minimal number of attempts to guarantee Eve a chance of success of least 99 percent when $n = 4$?

Programming Problem for CPSC 418 only

Problem 6 (Secure file transfer, 41 marks)**.**

Your solution to this problem must be implemented in `Java`. Make sure to use good coding practices. You may use whatever development platform you like, but *make sure that the final version compiles and runs on the platform specified below*. Programs that do not compile will *not* be graded. All software you need to complete the programming problem is available on the Linux servers specified below, as well as the Computer Science departmental workstations.

CPSC 418 students: the TA will compile and test your programs on one of the Computer Science department Linux servers using the latest version of `javac/java` installed.

PMAT 418 students: the TA will compile and test your programs on the University Linux server `msf1.ucalgary.ca` using the latest version of `javac/java` installed.

You are to implement a secure file transfer application using a simple client/server socket-based program by which a client sends an authenticated and encrypted file to a server. Specifically, the client program is to transfer a user-specified file to the server while providing confidentiality and data integrity. To provide confidentiality, all protocol messages are to be encrypted using AES-128-CBC, with a shared 128-bit session key. HMAC-SHA-1 is to be used to provide data integrity for all protocol messages. Recall from class that the best practice is to encrypt both the messages and the means for authentication.

Both client and server should prompt the user for a shared key at startup, to be used as the seed in a pseudorandom number generator as in Assignment 1. The client program should also prompt the user for source and destination file names. The destination file name, length of the source file in bytes, and the source file contents (encrypted and integrity-protected) should then be transferred to the server. The server should decrypt the file, check the integrity of the file content, and write the file content to the destination file specified by the client. The server should then send an acknowledgement to the client reporting success if the file was correctly decrypted, i.e. passed the integrity check, and written to the destination file; otherwise it should report failure. The client should exit after receiving the acknowledgement.

You must prevent the following types of attack against an adversary who doesn't know the shared key:

- (Confidentiality) The adversary must not be able to find (any information about) the file content.
- (Data Integrity) The adversary must not be able to change the file content without being detected.

Use the cryptographic primitives provided in the Java Cryptography Architecture. You may use your own solution program from Problem 6 of Assignment 1 inside your client/server application, or the model solution available on D2L.

Your client and server programs must use sockets to communicate. For a tutorial on sockets in `Java`, see

`http://java.sun.com/docs/books/tutorial/networking/sockets/`

See also the JSSE link provided on the "references" course page.

You must use the following three `Java` programs, provided on the assignments page, as the basis of your implementation:

(1) File `Server.java` containing the implementation of the server application. This program accepts one command-line argument:
   - `port` — port number for socket connection

   This argument and the server's IP-address must be echoed upon startup.

(2) File `Client.java` containing the implementation of the client program. This program accepts two command-line arguments:
   - `IP` — IP address of server
   - `port` — port number for socket connection

   These parameters must be echoed upon startup.

(3) File `ServerThread.java` containing a class for the thread to deal with clients who connect to the Server

These applications set up a socket connection between the client and server, and the server echoes any keyboard input obtained by the client. You are to modify these programs to realize the secure file transfer application described above. Be sure to include in your solution any additional files or classes that you feel are necessary to produce a well thought out modular design.

The programs should also be modified to allow a command line argument that defines whether debugging is on or off for the whole program. When enabled, your program should echo all protocol messages sent and received in such a way that it is clearly identified which program is echoing the message. This output must be properly formatted and easily readable. For instance, the call for the Server program would be

```
java Server port debug ...
```

The main function of `Server` should check if the second argument in `args[]` is present. If it is equal to "debug", then set a debug flag in the program. All protocol messages are then put in an `if` statement that checks if the debug flag is set, and if so, the messages are printed. This way, it is easy to turn protocol message echoing on or off.

Be sure to include in your solution any additional files or classes that you feel are necessary to produce a well thought out modular design.

Submit a separate description of your implementation, to be handed in with the rest of your written work. This description must include:

- A list of the files you have submitted that pertain to the problem, and a short description of each file.
- A description of how to compile and test your program (we prefer makefiles), and (for CPSC 418) the name of the department server on which you tested your code. Again, programs that do not compile will *not* be graded.
- A list of what is implemented in the event that you are submitting a partial solution, or a statement that the problem is solved in full.
- A list of what is not implemented in the event that you are submitting a partial solution, or a statement that the problem is solved in full.
- A list of known bugs, or a statement that there are no known bugs.
- A written description of your file transfer protocol including:
  - a precise description of all protocol messages, their format, and how they are parsed upon receipt,

– a description of how encryption and data integrity are employed (i.e. which fields are protected, where data integrity is placed, etc.).
– a description of how you prevented the attacks on confidentiality and data integrity mentioned above.

To assist the TA in marking, please include your name and the name of the file at the beginning of each file you submit.

Bonus Problem for Everyone

Problem 7 (Mixed Vigenère cipher cryptanalysis, 10 marks)**.**

Decrypt the following ciphertext that was encrypted using a *mixed* Vigenère cipher. Show all your work; this includes source code if you used programming. Answers without satisfactory explanation and documentation of how they were obtained will receive *no* credit. Neither will answers obtained by simply running Vigenère decryption from an online crypto applet website on the ciphertext.

A text file containing the ciphertext can be downloaded from the "assignments" page.

Note that this problem is a good deal harder than the bonus problem on Assignment 1 which asked you to cryptanalyze an ordinary Vigenère cipher.

Hint: The key word has length 5.

```
MNPBP NAWGB XJIWU IWZZF GJIWR ILYYO EJPOE YTTVO CJFAE RJYXX
XAJKW ZXSWV WJTAO MLOIE WRFBD CHDXM XNPXG RDYUE IXPBF NATPE
GNYNV RYNWX QTTZF NJIYO SHIHW XMNGH RANGY CSZPE REYYO EJIWR
QAZVS GLOIE VOWGX NDJWB XLYME RJIVM VATBD YYJYY KSPDD LTPAR
NTZZF GATMG ZCGTO WMPXI RATMG ZCGCO JYZPE LNQHM SGYZF GAWXG
EENMG IBPBM WESXV XIBXS ATQHE GOYNR IPGQX ATSIS QYNWK IWQZE
XFQAX CYLWG QYQYS BERGG RDOAJ ZKQAE GYNXS TCQBS XXJVI KGQYW
WDBYO GJIWS WMQYC DYNID CHMGI IJYAO SDGLF GNPDD PLPQF MWQLO
EHGXM GYTZD MNJYF EJAVG WXIWM ZCDXM XASQM VTGWX QTTHE XJIXH
XGRGG JEVWF GJPNG IKPXI GJPWV ZXJEG ICSQE LAEKF GYVGY CSQMV
ZMPXS MNPEF AGPHH XYSHW XPSOD JYGLS ZXYAO CYYDV IXSLZ IXPBH
QENWM ZXJHW XMNVM OXQTF PPSVK IWMGK ZJQOV ZBNVR ZXRPO GYDPE
XFWGI XDRWG XWOQQ XSRVS QKNWR WDPYO BDYDS ATWXV GMQYM BACWH
ZDOAW XWPWF ALEZD MNZZO DEPEO NDNXS TDQVX GNYTD CHQQG ZBZXR
AETBS QYWXG BTZPW WDCGO RNQAX VAEEE GEMWL SDOAE GDRVS QOYNR
ZDPKF GYTPG SHCWX MNPJD ALZGW WDNVQ QJCVQ KLPQF CSSNX KYJPD
VJIWL ZWZWI RYNBM VTGWH WSPAE RNOBY KLOAE GDRXM MNPBS SBVGK
PYCWI RTSXI ZHPGK ZBVGG RAWOE FYQNS NJIWG XCQBM IRPHW WXCPE
AAGQD LAWGY MNOBV ZMAGK WJZPE ZXZVA SYQYC BNOAE RATPE AYQKW
XSVGG ZXYHW XWMNQ WJRXM ZWFBM WASUD PTZXG NKNGS QYTVM ZDPTE
CBFAR MTYAK IWDWK XYJEF LFMXI WKFOF MENKF GYJVI KWFEL NKOAP
OLQBS WMEGY ZWPHO IRFKW MNPXG MTTHE QYNYR ZDPYF MUCYY CJPQS
QYTGY CSTWG YYJPD VATOF SHIHE ANPBR AAZKW XSIVM IVPYW ZXCGK
BNOHE GNOYS XSWWV PORVS QJIWU WXAKV ZCEGY ZWPHW XANHD GJPGK
MNPBV WHIHV NBFAI NSPXV
```