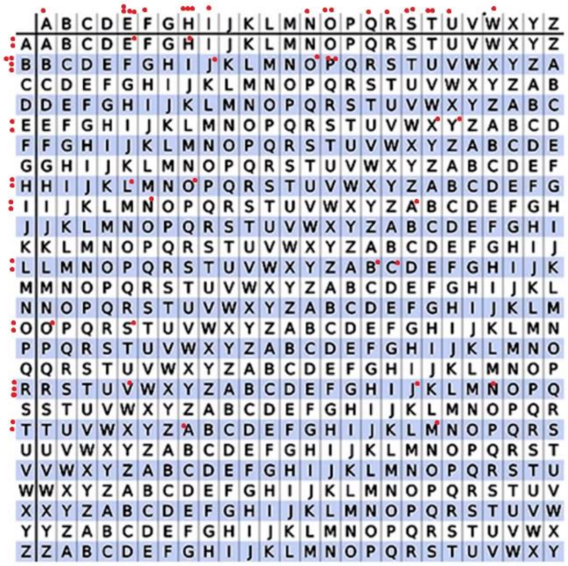
Joshua Main-Smith
September 13, 2020

Module 2 Assignment

1. **(10 points) Encrypt the message "Who is the queen of hearts?" using the Vigenère cipher with keyword rabbithole.**



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Using the top columns for the plaintext and the left columns for the key, you get a ciphertext of:
NHPJAMOSBYVEOPNALOCXJ

2. **(10 points) Suppose someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?**

A quick example of this: key = 1011, bit string = 0100, XOR(key, bit string) = 1111
Now, assuming the key is the same and using the result from above as the new bit string (1111), then XOR(key, bit string) = 0100, which is the value for the original bit string.
The reason it works out like this is because if XOR(K,X) = X^{-1}, then XOR(K,X^{-1}) = X.
Although, one flaw with this scheme is with no authentication mechanism put into place an attack could eavesdrop between sender and receiver and manipulate the message in transit without either part ever knowing it was changed.

3. **(20 points) In this problem, we will compare the security services that are provided by digital signatures (DS) and message authentication codes (MAC). We assume Oscar is able to observe all messages sent from Alice to Bob and vice versa. Oscar has no knowledge of any keys but the public one in case of DS. State whether and how (i) DS and (ii) MAC protect against each attack. The value auth(x) is computed with a DS or a MAC algorithm, respectively.**

a. **(Message integrity) Alice sends a message x = "Transfer $1000 to Mark" in the clear and also sends auth(x) to Bob. Oscar intercepts the message and replaces "Mark" with "Oscar." Will Bob detect this?**
Yes, Bob would detect this with either DS or MAC
(i) DS: The signature is produced by a cryptographic hash using this and Alice's private key as input to the digital signature. When Bob receives the message, the signature is verified by calculating the hash value, then using Alice's public key to verify that the message is valid. Altering the message would alter the hash value, leading to an unverified signature.

(ii) MAC: The MAC is calculated by using Alice's secret key and message as input to the MAC algorithm. When the message arrives to Bob, he will calculate the code using the same secret key and compare that with the code that was appended. If the message was altered, Bob would know since Oscar doesn't know the secret key.

b. **(Replay) Alice sends a message x="Transfer $1000 to Oscar" in the clear and also sends auth(x) to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?**
Both MAC and DS are subject to replay attacks since neither implements the use of random numbers in its algorithm.

c. **(Sender authentication with cheating third party) Oscar claims that he sent some message x with a valid auth(x) to Bob but Alice claims the same. Can Bob clear the question in either case?**
So, as I interpreted the question: Bob received a message x (with auth(x)). Oscar is claiming he sent message x, while Alice is also claiming she is the one that sent message X. Does Bob know who's telling the truth?

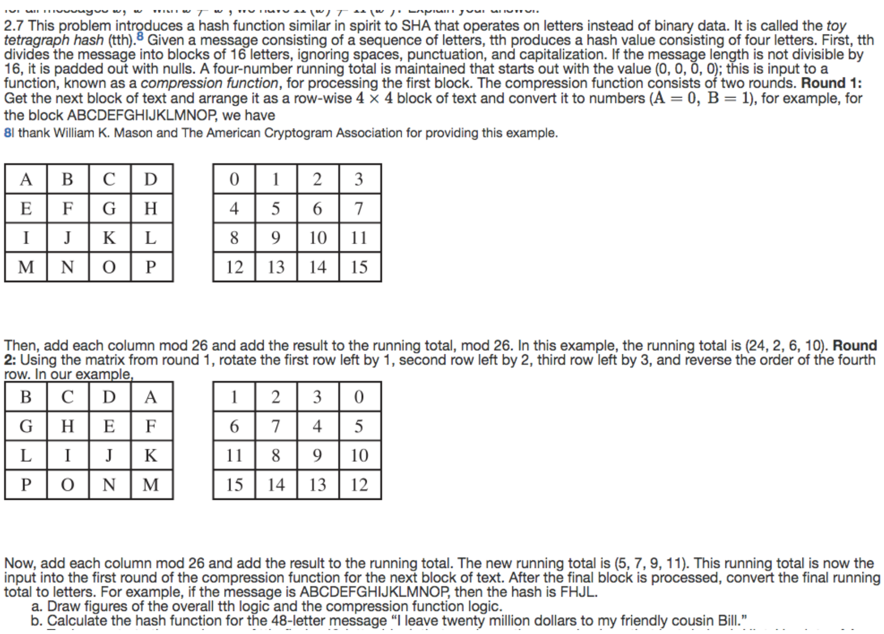
(i) DS: Given Alice sent the message, if Bob were to try and decode the message with Oscar's public key it wouldn't work (assuming Bob knows what public key belongs to Oscar and which to Alice). And, since Oscar doesn't know anyone else's private key, there's no way he could have been an imposter.
If Bob doesn't know who's public key belongs to who, this could be solved by going through a certified authority. If Alice were to get her keys verified by a CA, Bob could clear the question.

(ii) MAC: Since this process involves sharing private keys between two individuals and, assuming that Oscar doesn't know the secret key (the above condition seemed to only be applied to DS), then Bob wouldn't have been able to decode the message without sharing a secret key with Oscar.
Otherwise, if Bob didn't know the name behind the secret key he shares with a nameless person, there isn't any way to know who's telling the truth without a CA.

d. **(Authentication with Bob cheating) Bob claims that he received a message x with a valid signature auth(x) from Alice (e.g., “Transfer $1000 from Alice to Bob”) but Alice claims she has never sent it. Can Alice clear this question in either case?**
(i) DS: Alice could ask Bob to send the same message back, calculating its hash and encrypting it with Alice’s public key. When Alice receives the message, she can calculate the hash and decrypt it with her private key. If the signature matches what Bob has, then she sent the message.

(ii) MAC: Bob should be sure if the message came from Alice or not assuming they’re the only ones that know about the secret key. Since they’re both in possession of the same secret key, Alice wouldn’t be able to clear the question outside of possibly a CA.

4.



2.7 This problem introduces a hash function similar in spirit to SHA that operates on letters instead of binary data. It is called the *toy tetragraph hash* (tth).[8] Given a message consisting of a sequence of letters, tth produces a hash value consisting of four letters. First, tth divides the message into blocks of 16 letters, ignoring spaces, punctuation, and capitalization. If the message length is not divisible by 16, it is padded out with nulls. A four-number running total is maintained that starts out with the value (0, 0, 0, 0); this is input to a function, known as a *compression function*, for processing the first block. The compression function consists of two rounds. **Round 1:** Get the next block of text and arrange it as a row-wise 4×4 block of text and convert it to numbers ($A = 0,\ B = 1$), for example, for the block ABCDEFGHIJKLMNOP, we have

[8] I thank William K. Mason and The American Cryptogram Association for providing this example.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Then, add each column mod 26 and add the result to the running total, mod 26. In this example, the running total is (24, 2, 6, 10). **Round 2:** Using the matrix from round 1, rotate the first row left by 1, second row left by 2, third row left by 3, and reverse the order of the fourth row. In our example,

B	C	D	A
G	H	E	F
L	I	J	K
P	O	N	M

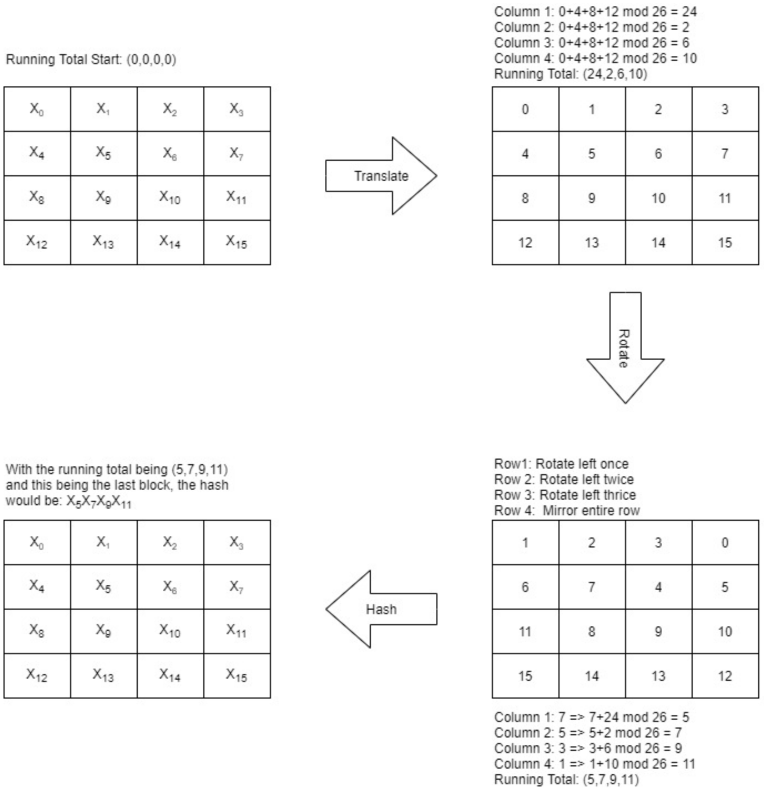
1	2	3	0
6	7	4	5
11	8	9	10
15	14	13	12

Now, add each column mod 26 and add the result to the running total. The new running total is (5, 7, 9, 11). This running total is now the input into the first round of the compression function for the next block of text. After the final block is processed, convert the final running total to letters. For example, if the message is ABCDEFGHIJKLMNOP, then the hash is FHJL.

a. Draw figures of the overall tth logic and the compression function logic.

b. Calculate the hash function for the 48-letter message "I leave twenty million dollars to my friendly cousin Bill."

a)



Running Total Start: (0,0,0,0)

X_0	X_1	X_2	X_3
X_4	X_5	X_6	X_7
X_8	X_9	X_{10}	X_{11}
X_{12}	X_{13}	X_{14}	X_{15}

Column 1: 0+4+8+12 mod 26 = 24
Column 2: 0+4+8+12 mod 26 = 2
Column 3: 0+4+8+12 mod 26 = 6
Column 4: 0+4+8+12 mod 26 = 10
Running Total: (24,2,6,10)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Row1: Rotate left once
Row 2: Rotate left twice
Row 3: Rotate left thrice
Row 4: Mirror entire row

1	2	3	0
6	7	4	5
11	8	9	10
15	14	13	12

Column 1: 7 => 7+24 mod 26 = 5
Column 2: 5 => 5+2 mod 26 = 7
Column 3: 3 => 3+6 mod 26 = 9
Column 4: 1 => 1+10 mod 26 = 11
Running Total: (5,7,9,11)

With the running total being (5,7,9,11) and this being the last block, the hash would be: $X_5X_7X_9X_{11}$

X_0	X_1	X_2	X_3
X_4	X_5	X_6	X_7
X_8	X_9	X_{10}	X_{11}
X_{12}	X_{13}	X_{14}	X_{15}

b)

The same algorithm was used as described in 2.7, including mod 26 being used for the running totals. Each block contains a 4x4 table of the message with its numeric equivalent, followed by a rotation of both the message and numeric equivalent (showing the running total of the numeric equivalent above each 4x4 block).

Block 1

Running Total: (24,2,6,10)

I	L	E	A
V	E	T	W
E	N	T	Y
M	I	L	L

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Running Total: (5,7,9,11)

L	E	A	I
T	W	V	E
Y	E	N	T
L	L	I	M

1	2	3	0
6	7	4	5
11	8	9	10
15	14	13	12

Block 2

Running Total: (15,21,1,7)

I	O	N	D
O	L	L	A
R	S	T	O
M	Y	F	R

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

Running Total: (8,12,16,20)

O	N	D	I
L	A	O	L
O	R	S	T
R	F	Y	M

17	18	19	16
22	23	20	21
27	24	25	26
31	30	29	28

Block 3

Running Total: (4,12,20,2)

I	E	N	D
L	Y	C	O
U	S	I	N
B	I	L	L

32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47

Running Total: (9,15,21,1)

E	N	D	I
C	O	L	Y
N	U	S	I
L	L	I	B

33	34	35	32
38	39	36	37
43	40	41	42
47	46	45	44

Hash: NLLL

Block 3 Again

The assignment on Canvas had “Bob”, but on the screenshot from the book it was “Bill”. I’m providing both for completeness

Running Total: (4,12,20,2)

I	E	N	D
L	Y	C	O
U	S	I	N
B	O	B	NULL

32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47

Running Total: (9,15,21,1)

E	N	D	I
C	O	L	Y
N	U	S	I
NULL	B	O	B

33	34	35	32
38	39	36	37
43	40	41	42
47	46	45	44

Hash is still NLLL

I thought it would be cool to do one where all the characters had a chance to be hashed (mod 48). When I ran through that, I got OTIM