CSCI 3403 **Erika Bailon**
Homework 2 **Fall 2018, CU-Boulder**

1. Given the security levels TOPSECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED and categories A, B, and C, specify what type of access (read, write, both, or neither) is allowed in each of the following situations:

   (a) Paul cleared for TOPSECRET,{A,C} and a document classified SECRET,{B,C}
   -Paul cannot read and cannot write because Document {B,C} is not a subset of {A,C}

   (b) Anna, cleared for CONFIDENTIAL,{C} and a document classified CONFIDENTIAL,{B}
   -Anna cannot read and cannot write on document B because the document is not a subset of C.

   (c) Jessie cleared for SECRET,{C} and a document classified CONFIDENTIAL,{C}
   -Jessie can only read document C because he CAN read down but CANNOT write down.

   (d) Sammie cleared for TOPSECRET,{A,C} and a document classified CONFIDENTIAL,{A}
   -Sammie can only read document A because he CAN read down but CANNOT write down.

   (e) Robin UNCLASSIFIED and a document classified CONFIDENTIAL,{B}
   -Robin can write document B because he CAN write up but CANNOT read up.

2. **Problem 2.1**. Suppose someone suggests the following way to confirm that the two of you are both in possesion 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?

   The question is confusing. There is no flaw in this scheme, is actually a PERFECT scheme because the people doing this scheme will obtain the key of these two people! so the scheme is genius!!!
   However, if the question refers more to what is the PROBLEM with creating a random string and returning the XOR result so the person with the same key does an XOR with that string and their key and also returns a string, then we are in serious trouble

since the people in the channel will have BOTH strings that were the result of the XOR with the key. Having this two string will allow them to do XOR on those string and obtain the key! So the security of the key is now broken and everyone who knows those two string for the results of XORing them with the key, now will have the key.

For example:

```
Paul's key ->       1 1 1 1 0 1 1      XOR
Random string ->    0 0 1 0 1 0 1
                   ----------------
Paul's result ->    1 1 0 1 1 1 0

Robin's key ->      1 1 1 1 0 1 1       XOR
Paul's result  ->   1 1 0 1 1 1 0
                   ----------------
Robin's result ->   0 0 1 0 1 0 1

Paul's result ->    1 1 0 1 1 1 0       XOR
Robin's result  ->  0 0 1 0 1 0 1
                   ----------------
KEY!!!!!!!! ->      1 1 1 1 0 1 1
```

The malicious people obtained the key out of the two string Paul and Robin sent!

3. **Problem 2.2. Part a**. This problem uses a real-world example of a symmetric cipher, from an old U.S. Special Forces manual (public domain). The document, filename *SpecialForces.pdf* is available at box.com/CompSec4e.

   (a) Using the two keys (memory words) *cryptographic* and *network security* encrypt the following message:

   Be at the third pillar from the left outside the lyceum theatre tonight at seven. If you are distrustful bring two friends

Creating the table with the first memory word:

2	10	13	8	12	7	4	11	1	9	5	6	3
c	r	y	p	t	o	g	r	a	p	h	i	c
b	e	a	t	t	h	e	t	h	i	r	d	p
i	l	l	a	r	f	r	o	m	t	h	e	l
e	f	t	o	u	t	s	i	d	e	t	h	e
l	y	c	e	u	n	t	h	e	a	t	r	e
t	o	n	i	g	h	t	a	t	s	e	v	e
n	i	f	y	o	u	a	r	e	d	i	s	t
r	u	s	t	f	u	l	b	r	i	n	g	t
w	o	f	r	i	e	n	d	s	x	x	x	x

5-letter groups:

hmdet ersbi eltnr wplee ettxe rstta
lnrht teinx dehrv sgxhf tmhuu etaoe
iytri teasd ixelf yoiuo toiha rbdtr
uugof ialtc nfsfx

Creating the table with the second memory word:

6	2	11	14	6	8	5	10	3	1	13	9	4	12	15
n	e	t	w	o	r	k	s	e	c	u	r	i	t	y
h	m	d	e	t	e	r	s	b	i	e	l	t	n	r
w	p	l	e	e	e	t	t	x	e	r	s	t	t	a
l	n	r	h	t	t	e	i	n	x	d	e	h	r	v
s	g	x	h	f	t	m	h	u	u	e	t	a	o	e
i	y	t	r	i	t	e	a	s	d	i	x	e	l	f
y	o	i	u	o	t	o	i	h	a	r	b	d	t	r
u	u	g	o	f	i	a	l	t	c	n	f	s	f	x

5-letter groups:

iexud	acmph	gyoub	xnush	tttha	edsrt
emeoa	hwlsi	yutet	fiofe	etttt	ilset
xbfst	ihail	dlrxt	igntr	oltfe	rdeir
neehh	ruora	vefrx			

4. **Problem 2.3**. Consider a very simple symmetric block encryption algorithm, in which 64-bits blocks of plaintext are encrypted using a 128-bit key. Encryption is defined as

$$C = (P \oplus K_0) \boxplus K_1$$

where C = ciphertext; K = secret key; K_0 = leftmost 64 bits of K; K_1 = rightmost 64 bits of K, $\oplus$ = bitwise exclusive or; and $\boxplus$ = addition $mod 2^{64}$.

(a) Show the decryption equation. That is, show the equation for P as a function of C, K_1 and K_2
We have $C = (P \oplus K_0) \boxplus K_1$. To decrypt we need first the additive inverse of K_1, which is $(\boxminus K_1)$.
Obtaining $C \boxminus K_1 = (P \oplus K_0)$. Then to get P we use the inverse of XOR which is still XOR so then we get $(C \boxminus K_1) \oplus K_0 = P$.
Therefore the decryption equation is:
$P = (C \boxminus K_1) \oplus K_0$

(b) Suppose an adversary has access to two sets of plaintext and their corresponding ciphertexts and wishes to determine K. We have the two equations

$$C = (P \oplus K_0) \boxplus K_1 \ ; \ C' = (P' \oplus K_0) \boxplus K_1$$

First derive an equation in one unknown. It is possible to proceed further to solve for K_0 ?
We have two equations with two unknowns so we can use the elimination method.
$(C = (P \oplus K_0) \boxplus K_1)$ $-(C' = (P' \oplus K_0) \boxplus K_1)$
... which is:
$(P \oplus K_0) \boxplus K_1 \boxminus ((P' \oplus K_0) \boxplus K_1)$
$= (P \oplus K_0) \boxplus K_1 \boxminus (P' \oplus K_0) \boxminus K_1$
$= (P \oplus K_0) \boxminus (P' \oplus K_0)$
Problem! $\boxminus$ and $\oplus$ do not have the properties of distribution or associativity, we cannot reduce this formula or associate it or distribute. THEREFORE it is not possible to proceed further to solve for K_0.