COLLIN GROS
02/26/20

CS 478

CH 1

1.a. IF ALICE NEEDS TO ALERT BOB THAT TRUDY IS ABOUT TO ATTACK, CONFIDENTIALITY MIGHT NOT MATTER AS MUCH AS INTEGRITY. BOB NEEDS TO BE ABLE TO READ THE MESSAGE, AND DOESN'T CARE WHO SEES IT.

b. INCREASING AVAILABILITY MIGHT WEAKEN CONFIDENTIALITY. ASSUME ALICE HAS A VIDEO SHARING WEBSITE. IF SHE WANTS TO INCREASE HER AVAILABILITY TO ALLOW PEOPLE OF EVERY COUNTRY TO ACCESS IT, SHE VIOLATES CONFIDENTIALITY THAT OTHER COUNTRIES (SUCH AS CHINA) ENFORCE, DUE TO CENSORSHIP LAWS.

2. (A.a.) C=KITLKE (TABLE 2.1)

e	h	i	k	l	r	s	t
000	001	010	011	100	101	110	111

P = thrill k=?

K I T L K E

011 010 111 100 011 000 = C

(+) 100 011 010 110 111 100 = K

111 001 101 010 100 100 = P

(A.b.) P=tiller

K I T L K E

011 010 111 100 011 000 = C

100 011 010 110 111 100 = K

111 001 101 010 100 100 = P

3. (29.a.) 40 BITS → 2^{40} KEYS TOTAL, OR $\frac{2^{40}}{2} = 2^{39}$ KEYS ON AVERAGE

(29.b.) TRUDY CAN CHECK WHETHER A DECRYPTION CONTAINS ANY VALID WORD IN THE ENGLISH LANGUAGE (IF P HAD ENGLISH WORDS) BY USING A DICTIONARY AND HASH TABLE.

IF THERE ARE MULTIPLE DECRYPTIONS WITH A SINGLE WORD, THE DECRYPTION MOST LIKELY CORRECT IS THE ONE WITH THE MAX AMOUNT OF ENGLISH WORDS.

(24.c) THE NUMBER OF WORDS IS COUNTED EVERY KEY. THERE WILL BE OVERHEAD DEPENDING ON THE DATA STRUCTURES USED.

(24.d.) THERE MAY BE A FEW FALSE ALARMS, HOWEVER, ONE MAY EASILY TELL IF ONE MAKES SYMANTIC SENSE.

4.a. □ □ □ □ □ □ □ □

↑

0–9, a–z, A–Z

$10+26+26=62$

KEY SPACE $= 8^{62}$

b. $2^x = 8^{62}$

$x \log 2 = 62 \log 8$

$x = \frac{62 \log 8}{\log 2} = 186$ → KEYSPACE $= 2^{186}$

c. 2^{40} KEYS/SEC, 2^{186} KEYS, $= \frac{2^{186}}{2^{40}}$ SECONDS $= 2^{146}$ SECONDS

5.a. ASCII BITS USED:

0–9 → 0000 — 1001 (4 USED)

a–z → 0110 0001 → 0111 1010 (7 USED)

A–Z → 0100 0001 – 0101 1010 (7 USED)

LEAVES 6 FOR ENCRYPTION

→ 8^6 POSSIBLE KEYS ADDED

→ KEYSPACE $= 8^{68}$ ← $2^x = 8^{68}$ $x = \frac{68 \log 8}{\log 2} = 204$ $= 2^{204}$

b. 2^{40} KEYS/SEC, 2^{204} KEYS $= \frac{2^{204}}{2^{40}}$ SECONDS $= 2^{164}$ SECONDS

CH 3

6. (2.a.) THE KEYSTREAM IS GENERATED USING STEPPING OF REGISTERS (SHORT REGISTERS). THESE WILL EVENTUALLY REPEAT A KEY STREAM BECAUSE OF THEIR SYSTEMATIC SHIFTING. KEY SPACE IS ALSO ONLY 2^{64}.

(2.b.) TRUDY WILL BE ABLE TO USE AN ATTACK ON A REPEATING CIPHER BY FOLLOWING THE ALGORITHM TO BREAK IT, MUCH LIKE THE REPEATING ONETIME PAD PROBLEM. RC4 HAS ISSUES WITH A REPEATING KEYSTREAM IF THE FIRST 256 BYTES OF THE FIRST KEY STREAMS ARE NOT DISCARDED.

7. (3.a.) KEYSTREAM ⊕ P = C → KEYSTREAM = P ⊕ C

TRUDY KNOWS THESE

7. (3.b.) $P \oplus K = C$

$P' \oplus K = C'$

	L	I	M	E	
	00	01	10	11	= P
$\oplus$	01	11	11	11	= K
	01	10	01	00	= C
	I	M	I	L	= C

	M	I	L	E	
	10	01	00	11	= P′
$\oplus$	01	11	11	11	= K
	11	10	11	00	= C′
	E	M	E	L	= C′

BOB DECRYPTS AND GETS MILE

8. (11.a.) A FEISTEL CIPHER IS A "GENERAL DESIGN PRINCIPAL, NOT A CIPHER."

P IS SPLIT TO L AND R HALVES

$P = (L_0, R_0)$

FOR EACH ROUND $i = 1, 2, \dots, n$ (K_i IS SUBKEY FOR ROUND i) (KEY SCHEDULE ALG.)

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

$C = (L_n, R_n)$

(11.b.) YES, DES IS A FEISTEL CIPHER.

(11.c.) NO, AES IS NOT A FEISTEL CIPHER.

(11.d.) TEA IS "ALMOST" A FEISTEL CIPHER BECAUSE IT USES ADDITION AND SUBTRACTION INSTEAD of XOR.

9. (16.) 2DES as $C = D(E(P,K_1)K_2)$? C_i

2DES IS NORMALLY $C_3 = E(E(P,K_1), K)$

ENCRYPT P ∀ K		DECRYPT C_3 ∀ K	
K_1	C_1	K_1	C_1
K_2	C_2	K_2	C_2
⋮		⋮	
K_n	C_n	K_n	C_n

FIND = C, GIVES YOU K_1 K_2

ENCRYPT P ∀ K	
K_0	C_{i_0}
K_1	C_{i_1}
⋮	
$K_{2^{56}}$	$C_{i_{2^{56}}}$

ENCRYPT C_3 ∀ K	
K_0	C_{i_0}
K_1	C_{i_1}
⋮	
$K_{2^{56}}$	$C_{i_{2^{56}}}$

K_1 FIND WHERE $C_{i_n} = C_{i_n}$ K_2

NOW FOR 2DES AS $C_3 = D(\underbrace{E(P,K_1)}_{C_i}, K_2)$

(MEET IN THE MIDDLE)

10. (22.a.) YES, THE IV MUST BE RANDOM.

(22.b) IF SELECTED IN SEQUENCE (IV):

DISADVANTAGES	ADVANTAGES
FIRST 2 ENCRYPTIONS DON'T REALLY MATTER	LESS COMPUTING TIME

11. (25.a.) $C_0 = IV \oplus E(P_0, K)$, $C_1 = C_0 \oplus E(P_1, k)$, $C_2 = C_1 \oplus E(P_2, k)$

$P_0 = D(IV \oplus C_0, k)$, $P_1 = D(C_0 \oplus C_1, k)$, $P_2 = D(C_1 \oplus C_2, k)$

(25.b.) 1. TRUDY CAN SEE A PATTERN IN THE CIPHERTEXTS LIKE IN ECB MODE

2. TRUDY CAN BRUTE-FORCE EASIER IF IDENTICAL INPUT/OUTPUT PAIRS ARE FOUND (SINCE IV IS NOT RANDOM, THIS IS MORE LIKELY)

12. (31.a.) WHEN CBC IS USED, TRUDY CAN SEE THAT 2 CIPHERTEXTS ARE THE SAME, MAKING IT EASIER TO UNDERSTAND THE CIPHER TEXT SEQUENCES AND EASIER TO DISCOVER THE PLAINTEXT.

(31.b.) WHEN CTR IS USED, THE INITIAL VALUE OF THE IV DOES NOT MATTER, COMPARED TO IF THE IV IS EVER RE-USED (WHICH WOULD ALLOW TRUDY TO DO A CHOSEN PLAINTEXT ATTACK)

(31.c.) IN CBC MODE, TRUDY CAN ONLY BE ABLE TO COMPARE CIPHERTEXTS AND DETERMINE IF THEY COME FROM THE SAME PLAINTEXT, WHEN IV IS NOT RANDOM.
HOWEVER, IN CTR MODE, THE IV AND COUNTER ACT AS A ONE-TIME PAD. IF IV IS KEPT THE SAME, JUST LIKE REUSING A ONE-TIME PAD, SECURITY IS LOST. THEREFORE, CTR IS LESS SECURE AND CBC IS MORE SECURE.

13 (36.c.) NO, ANY CHANGE IN A CIPHERTEXT BLOCK BEFORE THE LAST TWO WILL NOT SHOW UP IN DECRYPTION. CBC MODE SELF-CORRECTS ITSELF IN DECRYPTION, AS EACH DECRYPTED BLOCK ONLY DEPENDS ON THE PREVIOUS 2 CIPHERTEXTS.

14 (41.b.) MEET IN THE MIDDLE ATTACK...

2^8 ENCRYPTIONS FOR LEFT TABLE

$C = E(E(P, K_1), K_2)$ FOR EXAMPLE...

ENCRYPT P ∀K GIVES YOU A LIST OF CIPHERTEXTS, C_1

DECRYPT C ∀K GIVES YOU ANOTHER LIST OF CIPHERTEXTS, C_2

$C_1 \cap C_2$ – GET K_1 & K_2 FROM MATCHED KEY PAIRS FROM CIPHERTEXT LIST INTERSECTIONS...

$\downarrow$ K_1 $\downarrow$ K_2

REPEAT INTERSECTIONS FOR EVERY PAIR OF ENCRYPTIONS!

NEWLY ALICE'S ENCRYPTION IS BROKEN IN 2^8 WORK, OR 2^7 IF TABLE WAS ALREADY COMPUTED.

15. $i = 0$ (GENERATED FROM A PROGRAM I WROTE IN C)

$L_i = 0101$

$R_i = 0101$

$i = 1$

$L_i = 0101$

$R_i = 0100$

$i = 2$

$L_i = 0100$

$R_i = 0000$

$i = 3$

$L_i = 0100$

$R_i = 1010$

$C = 01001010$

```
collin@collin-debian:~/Documents/school/cs/hw1-1$ ./a.out
plaintext: 10110101

ROUND: 0
Li: 0101
Ri: 0101

ROUND: 1
Li: 0101
Ri: 0100

ROUND: 2
Li: 0100
Ri: 0100

ROUND: 3
Li: 0100
Ri: 1010
ciphertext: 01001010
collin@collin-debian:~/Documents/school/cs/hw1-1$
```