

CMEA

CMEA

- ❑ Cellular Message Encryption Algorithm
- ❑ Designed for use with cell phones
 - To protect confidentiality of called number
 - For “control channel”, not the “data channel”
 - Data channel encrypted with ORYX
- ❑ Part of a standard developed by TIA
 - Flaw in cipher discovered in 1997
- ❑ Cipher design process not open
 - In violation of Kerckoffs Principle

CMEA

- ❑ Block cipher
 - 64 bit key
 - Variable block size, typically 2 to 6 bytes
- ❑ CMEA is its own inverse
 - Recall that Enigma is its own inverse
 - Not clear that this is useful for CMEA
- ❑ CMEA uses "Cave Table"
 - A fixed 256-byte lookup table
 - Not a permutation

Cave Table

- Table has 256 bytes:
 - 164 distinct values
 - 97 appear just once
 - 44 occur twice
 - 21 occur three times
 - 2 occur four times
 - Highly non-uniform!

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	d9	23	5f	e6	ca	68	97	b0	7b	f2	0c	34	11	a5	8d	4e
1	0a	46	77	8d	10	9f	5e	62	f1	34	ec	a5	c9	b3	d8	2b
2	59	47	e3	d2	ff	ae	64	ca	15	8b	7d	38	21	bc	96	00
3	49	56	23	15	97	e4	cb	6f	f2	70	3c	88	ba	d1	0d	ae
4	e2	38	ba	44	9f	83	5d	1c	de	ab	c7	65	f1	76	09	20
5	86	bd	0a	f1	3c	a7	29	93	cb	45	5f	e8	10	74	62	de
6	b8	77	80	d1	12	26	ac	6d	e9	cf	f3	54	3a	0b	95	4e
7	b1	30	a4	96	f8	57	49	8e	05	1f	62	7c	c3	2b	da	ed
8	bb	86	0d	7a	97	13	6c	4e	51	30	e5	f2	2f	d8	c4	a9
9	91	76	f0	17	43	38	29	84	a2	db	ef	65	5e	ca	0d	bc
a	e7	fa	d8	81	6f	00	14	42	25	7c	5d	c9	9e	b6	33	ab
b	5a	6f	9b	d9	fe	71	44	c5	37	a2	88	2d	00	b6	13	ec
c	4e	96	a8	5a	b5	d7	c3	8d	3f	f2	ec	04	60	71	1b	29
d	04	79	e3	c7	1b	66	81	4a	25	9d	dc	5f	3e	b0	f8	a2
e	91	34	f6	5c	67	89	73	05	22	aa	cb	ee	bf	18	d0	4d
f	f5	36	ae	01	2f	94	c3	49	8b	bd	58	12	e0	77	6c	da

- For example

$$C[0x6a] = 0xf3$$

CMEA

- Let K_0, K_1, \dots, K_7 be bytes of 64-bit key
- Let C be Cave Table
- For byte x , define (all "+" are mod 256)
$$Q(x) = C[(x \oplus K_0) + K_1] + x$$
$$R(x) = C[(Q(x) \oplus K_2) + K_3] + x$$
$$S(x) = C[(R(x) \oplus K_4) + K_5] + x$$
$$T(x) = C[(S(x) \oplus K_6) + K_7] + x$$
- Table defined by $T(x)$ used in CMEA

CMEA

- We have

$$Q(x) = C[(x \oplus K_0) + K_1] + x$$

$$R(x) = C[(Q(x) \oplus K_2) + K_3] + x$$

$$S(x) = C[(R(x) \oplus K_4) + K_5] + x$$

$$T(x) = C[(S(x) \oplus K_6) + K_7] + x$$

- Note that $T(x) - x$ is in C
 - Same is true of $S(x) - x$, $R(x) - x$, and $Q(x) - x$
- Implies these values are biased
 - These facts used heavily in attacks

CMEA Algorithm

- ❑ Encrypt block of n bytes
- ❑ Uses T table
 - Which uses Cave Table
- ❑ Cipher is its own inverse
 - Same algorithm used for decryption

```
// all arithmetic is mod 256 and “V” is OR
// (c[0],c[1],...,c[n-1]) = output block of ciphertext bytes
1. (p[0],p[1],...,p[n-1]) = input block of plaintext bytes
2. z = 0
3. for i = 0 to n - 1
4.     k = T(z ⊕ i)
5.     p[i] = p[i] + k
6.     z = z + p[i]
7. next i
8. h = ⌊n/2⌋
9. for i = 0 to h - 1
10.    p[i] = p[i] ⊕ (p[n - 1 - i] V 1)
11. next i
12. z = 0
13. for i = 0 to n - 1
14.    k = T(z ⊕ i)
15.    z = z + p[i]
16.    c[i] = p[i] - k
17. next i
```

SCMEA

- The “v 1” in line 10 of CMEA complicates attack
- We define Simplified CMEA (SCMEA) to be same as CMEA, without “v 1”
- That is, replace line 10 of CMEA with:

$$10.' p[i] = p[i] \oplus p[n - 1 - i]$$

SCMEA Chosen Plaintext Attack

- Consider plaintext block of the form
- (*) $(p_0, p_1, p_2) = ((\ell \oplus 1) - T(0), (j \oplus 2) - (\ell \oplus 1) - T(\ell), 0)$
- Corresponding 1st ciphertext byte is
$$c_0 = (\ell \oplus 1 \oplus T(j)) - T(0).$$
- The plan of attack
 - Use chosen plaintext to find putative $T(0)$
 - With more chosen plaintext, can then find putative $T(j)$ for $j=1,2,\dots,255$
- Note: Recover T table, and key is "broken"

SCMEA Chosen Plaintext

- Choose plaintext blocks of the form

$$(p_0, p_1, p_2) = (1 - x_0, 1 - x_0, 0)$$

where x_0 is in the Cave Table

- Suppose we obtain

$$c_0 = (1 \oplus x_0) - x_0 = \begin{cases} 1 & \text{if } x_0 \text{ is even} \\ 255 & \text{if } x_0 \text{ is odd.} \end{cases}$$

- Setting $\ell = j = 0$ in (*) and we see that such an x_0 is consistent with $T(0) = x_0$
- Then we have found a candidate for $T(0)$

SCMEA Chosen Plaintext

- Given candidate $x_0 = T(0)$, choose plaintext
 $(p_0, p_1, p_2) = (1 - x_0, (j \oplus 2) - x_0, 0)$
for each $j = 1, 2, \dots, 255$
- Then from (*) with $l = 0$, we have
 $c_0 = (1 \oplus x_j) - x_0$
and we can solve for x_j
- If it is true that $x_0 = T(0)$ then $x_j = T(j)$

SCMEA Chosen Plaintext

- We can obtain putative $T(0)$ and putative $T(j)$, for $j=0,1,\dots,255$
- How can we know whether this is correct T table?
- Recall, $T(j) - j$ is in Cave Table for all j
- Check whether $x_j - j$ is in Cave Table
 - If it fails for any j , then $T(0)$ incorrect

SCMEA Chosen Plaintext Attack Algorithm

- Use $\ell = j = 0$ in (*) to find putative $T(0)$
- Set $\ell = 0$ in (*) and $j = 1, 2, \dots, 255$ to find putative $T(j)$
- For each putative $T(j)$, check if $T(j) - j$ is in the Cave Table
 - If this fails for any j , then start over
 - If holds for all j , then have found T table

SCMEA Chosen Plaintext

- ❑ How much chosen plaintext needed?
- ❑ Recall: 164 distinct elements in Cave Table
- ❑ Ignoring false alarms...
 - Since $T(0)$ is in Cave Table, need 82 chosen plaintext blocks to find $T(0)$
 - Then 255 more blocks to find T table
 - Total of **337** chosen plaintext blocks
- ❑ Consider false alarms for CMEA attack...

CMEA Chosen Plaintext Attack

□ Similar to SCMEA, if

$$(**) (p_0, p_1, p_2) = ((\ell \oplus 1) - T(0), (j \oplus 2) - (\ell \oplus 1) - T(\ell), 0)$$

then $c_0 = ((\ell \oplus 1 \oplus (T(j) \vee 1)) - T(0)$

and $c_1 = (j \oplus 2) - (\ell \oplus 1) - T(\ell \oplus (T(j) \vee 1)).$

□ A more complex expression for c_2

○ Homework problem

□ As in SCMEA attack, let $\ell = j = 0$

CMEA Chosen Plaintext

- Letting $\ell = j = 0$, we have that plaintext

$$(p_0, p_1, p_2) = (1 - T(0), 1 - T(0), 0)$$

yields ciphertext

$$c_0 = ((1 \oplus (T(0) \vee 1)) - T(0))$$

$$c_1 = 1 - T(T(0) \vee 1)$$

$$c_2 = T(0) - T(((1 \oplus (T(0) \vee 1)) + 1) \oplus 2)$$

- Again, choose plaintext of the form

$$(p_0, p_1, p_2) = (1 - x_0, 1 - x_0, 0)$$

CMEA Chosen Plaintext

- Choose plaintext of the form

$$(p_0, p_1, p_2) = (1 - x_0, 1 - x_0, 0)$$

- Any of these that satisfy

$$c_0 = (1 \oplus (x_0 \vee 1)) - x_0 = \begin{cases} 0 & \text{and } x_0 \text{ is even} \\ 255 & \text{and } x_0 \text{ is odd} \end{cases}$$

are consistent with $x_0 = T(0)$

- Can reduce false alarms by using Cave Table conditions on both c_1 and c_2

CMEA Chosen Plaintext

- Given candidate $x_0 = T(0)$, choose $(p_0, p_1, p_2) = (1 - x_0, (j \oplus 2) - x_0, 0)$ for each $j = 1, 2, \dots, 255$
- Then from (**), with $l = 0$, we have $c_0 = (1 \oplus (T(j) \vee 1)) - T(0)$ and we can solve for $T(j) \vee 1$
- Note: low-order bit of $T(j)$ is unknown

CMEA Chosen Plaintext

- ❑ Attack algorithm
- ❑ Use $\ell = j = 0$ in (**) to find x_0 , putative $T(0)$
- ❑ Set $\ell = 0$ in (**) and $j = 1, 2, \dots, 255$ to find x_j which is putative $T(j) \vee 1$
- ❑ For each x_j , check if $x_j - j$ is in the Cave Table and/or $(x_j \oplus 1) - j$ is in the Cave Table
 - If **both** fail for any j , then x_0 incorrect
 - If one fails,, then have unique putative $T(j)$
 - If neither fails, then 2 choices for $T(j)$

CMEA Chosen Plaintext

- ❑ How to resolve ambiguous $x_j = T(j)$?
 - Both $x_j - j$ and $(x_j \oplus 1) - j$ in Cave Table
- ❑ Create array A of size 256
- ❑ Set $A_i = 0$ if low-order bit of x_i is known
 - And $A_i = 1$ if low-order bit of x_i is ambiguous
- ❑ We can use this array to resolve ambiguous low-order bits

CMEA Chosen Plaintext

- Suppose putative $T(k)$ is ambiguous
- Find t and j with $k = t \oplus (T(j) \vee 1)$ where $A_t = 0$
- Let
$$(p_0, p_1, p_2) = ((t \oplus 1) - T(0), (j \oplus 2) - (t \oplus 1) - T(t), 0)$$
- Encrypting this chosen plaintext yields
$$T(t \oplus (T(j) \vee 1)) = (j \oplus 2) - (t \oplus 1) - c_1$$
- Which implies $T(k) = (j \oplus 2) - (t \oplus 1) - c_1$
- We have resolved ambiguity in $T(k)$

CMEA Chosen Plaintext

- ❑ How much chosen plaintext is required?
- ❑ 82 blocks to find $T(0)$, on average
- ❑ 255 more to recover T table
- ❑ $0.6 \cdot 255 = 153$ to resolve ambiguous
 - 0.6 probability that both are in Cave Table
- ❑ $0.258 \cdot 9 = 2.3$ for incorrect $T(0)$ s
 - 0.258 prob, each takes 9 blocks to resolve
- ❑ Total chosen plaintext blocks: **492.3**

CMEA Chosen Plaintext

- Analytically, have shown that **492.3** chosen plaintexts required
- Empirical results from 10^6 trials very close to predicted results:

Trial	Average to Find $T(0)$	Find $T(j) \vee 1$	Average Ambiguous	Average False Alarms	Total
10^6	81.84	255	152.89	2.43	492.16

CMEA Chosen Plaintext Attack: Bottom Line

- ❑ Recover T table, not the actual key
- ❑ Relies on relationship between plaintext and ciphertext
 - And the special role of $T(0)$
- ❑ Generally not practical
 - Since requires **chosen plaintext**
- ❑ Attack clearly shows CMEA is weak
- ❑ Next, **known plaintext** attack

SCMEA Known Plaintext Attack

- ❑ Similar to chosen plaintext attack
- ❑ Relatively complex attack, 2 phases
- ❑ Primary phase
 - Find $T(0)$ or small number of candidates
- ❑ Secondary phase
 - Determine key
 - Backtracking and meet-in-the-middle

SCMEA Known Plaintext Attack: Primary Phase

- Since $T(0)$ in Cave Table, 164 choices
 - Let v_0, v_1, \dots, v_{163} be Cave Table elements
- For each v_i , make 256×256 table A
 - Initialize $A_{i,j} = 1$ for all i and j
- Assuming $T(0) = v_i$, set $A_{i,j} = 0$ if $T(i) = j$ is "impossible"
- Since $T(j) - j$ is in Cave Table,
 $T(j) \in \{v_0 + j, v_1 + j, \dots, v_{163} + j\}$
- This gives 92 zeros in each row of $A_{i,j}$

SCMEA Known Plaintext Attack: Primary Phase

- Each putative $T(0)$ has A table with 92 zeros and 164 ones per row
- Use known plaintext to insert more 0s
- If $T(0)$ is incorrect, given enough known plaintext, get a contradiction
 - For example, a row of A is all 0
- Ideally, one $T(0)$ survives

SCMEA Known Plaintext Attack: Primary Phase

- How to use known plaintext to insert more 0s into A tables?
- Consider plaintext $P = (p_0, p_1, p_2)$
- SCMEA, ciphertext yields equation
$$\begin{aligned} & ((c_0 + T(0)) \oplus (p_0 + T(0))) - p_2 \\ & = T((p_0 + p_1 + T(0) + T((p_0 + T(0)) \oplus 1)) \oplus 2) \end{aligned}$$
- Given P , c_0 , and putative $T(0)$, we can add 0s to corresponding A table as follows...

SCMEA Known Plaintext Attack: Primary Phase

- We have
$$((c_0 + T(0)) \oplus (p_0 + T(0))) - p_2$$
$$= T((p_0 + p_1 + T(0) + T((p_0 + T(0)) \oplus 1)) \oplus 2)$$
- Suppose $P = (a1,95,71)$ and $c_0 = 04$, in hex
- Consider A table for $T(0) = 34$
- Equation becomes $7c = T((6a + T(d4)) \oplus 2)$
 - Guess $x = T(d4)$ and let $y = (6a + x) \oplus 2$
 - If $A_{y,7c} = 0$, then x impossible, set $A_{d4,x} = 0$
 - Repeat for all choices of $T(d4)$
- Repeat for all known plaintext and iterate

SCMEA Known Plaintext Attack: Primary Phase

- Known plaintext requirement is large
 - About 300 blocks
- But we are not using all available info
- Possible to also use c_1 and c_2
 - See homework problems!

Ciphertext bytes used	c_0 only	c_0 and c_1	c_0, c_1 and c_2
Known plaintext blocks	300	90	60

SCMEA Known Plaintext Attack: Primary Phase

- With enough known plaintext, primary phase yields one (correct) $T(0)$
 - Also uniquely determine some $T(j)$
 - Can use this in secondary phase
- Secondary phase recovers the key
 - Use backtracking or meet-in-the-middle

SCMEA Known Plaintext Attack: Secondary Phase

- Assume we have correct $T(0)$
- Want to determine the key
 - Recall, 64-bits key K_0, K_1, \dots, K_7 where
$$Q(x) = C[(x \oplus K_0) + K_1] + x$$
$$R(x) = C[(Q(x) \oplus K_2) + K_3] + x$$
$$S(x) = C[(R(x) \oplus K_4) + K_5] + x$$
$$T(x) = C[(S(x) \oplus K_6) + K_7] + x$$
 - Here, C is the Cave Table

SCMEA Secondary Phase: Backtracking

- Have $T(x) = C[(S(x) \oplus K_6) + K_7] + x$
- Since $S(x) - x$ is in Cave Table,
 $T(x) = C[((v + x) \oplus K_6) + K_7] + x$
for some v in Cave Table
- Guess (K_6, K_7) and choose some x
- For each $v \in C$, compute
 $y = C[((v + x) \oplus K_6) + K_7] + x$

SCMEA Secondary Phase: Backtracking

- Guess (K_6, K_7) and choose x
- For each $v \in C$, compute
$$y = C[((v + x) \oplus K_6) + K_7] + x$$
- If every choice of v gives $A_{x,y} = 0$, then putative key (K_6, K_7) is incorrect
 - Putative key is not consistent with $T(0)$
- Repeat for each choice of x

SCMEA Secondary Phase: Backtracking

- Repeat for each choice of x
- Reduces number of possible (K_6, K_7)
 - Number of survivors depends on A table
 - The A table depends on known plaintext
- Empirical results:

Known plaintext blocks	50	75	100	150
Partial keys (K_6, K_7)	19800	2002	42	2

SCMEA Secondary Phase: Backtracking

- For each putative (K_6, K_7) consider
$$S(x) = C[(R(x) \oplus K_4) + K_5] + x$$
- For each candidate (K_4, K_5) , compute
$$z = C[((v + x) \oplus K_4) + K_5] + x$$
 and
$$y = C[(z \oplus K_6) + K_7] + x$$
- Then use A table as in (K_6, K_7) case
- Same idea extends to K_3, K_2, K_1, K_0

SCMEA Secondary Phase: Backtracking

- Let n be number of (K_6, K_7) expected
- Then expect about n^4 putative keys
- From previous slide, about 150 known plaintexts yields $2^4 = 16$ putative keys
 - However, with 75 known plaintexts, number of keys is about 2^{44}
 - Exhaustive key search is 2^{63} work

SCMEA Secondary Phase: Meet-in-the-Middle

- ❑ Can do much better than backtracking
- ❑ Practical if 4 or more unique $T(j)$ in A
 - At least 4 rows of A each with a single 1
 - May be practical if at least 4 rows with small number of 1s
- ❑ Meet-in-the-middle empirical results:

Known plaintext blocks	50	75	100	150
Number uniquely determined	3	6	15	45

SCMEA Secondary Phase: Meet-in-the-Middle

- Spse $T(a), T(b), T(c), T(d)$ known from A
- For each (K_0, K_1, K_2, K_3) compute
$$Q(a) = C[(a \oplus K_0) + K_1] + a$$
$$R(a) = C[(Q(a) \oplus K_2) + K_3] + a$$
- And similarly for b, c, d
- Store $(R(a), R(b), R(c), R(d))$ and (K_0, K_1, K_2, K_3) in a row of matrix M
- Then M has 2^{32} rows—sort on R

SCMEA Secondary Phase: Meet-in-the-Middle

- Again, $T(a), T(b), T(c), T(d)$ known from A
- For each (K_4, K_5, K_6, K_7) work backwards from (known) $T(a)$ to find $S(a), R(a)$ satisfying
$$S(a) = C[(R(a) \oplus K_4) + K_5] + a$$
$$T(a) = C[(S(a) \oplus K_6) + K_7] + a$$
- And similarly for b, c, d
- Note that must "invert" Cave Table
- Search for $(R(a), R(b), R(c), R(d))$ in M

SCMEA Secondary Phase: Meet-in-the-Middle

- If match, have met-in-the-middle
- Then we have $(K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7)$ for which $T(a), T(b), T(c), T(d)$ all match their known values
- With 4 known Ts, expect few solutions
 - Work and storage on order of 2^{32}
- It is possible to do better (again)!

SCMEA Secondary Phase: Meet-in-the-Middle

- ❑ Improved version
 - Work of 2^{32} but storage of only 2^{24}
- ❑ Somewhat tricky...
- ❑ Again, $T(a), T(b), T(c), T(d)$ known from A
- ❑ For each (K_0, K_1, K_2) compute
$$a' = (C[(a \oplus K_0) + K_1] + a) \oplus K_2$$
- ❑ And similarly for b', c', d'

SCMEA Secondary Phase: Meet-in-the-Middle

- For each (K_0, K_1, K_2) compute
$$a' = Q(a) \oplus K_2 = (C[(a \oplus K_0) + K_1] + a) \oplus K_2$$
- And similarly for b', c', d'
- Create a table M with each row $(a', b', c', d', K_0, K_1, K_2)$
- Indexed by $(a' - d', b' - d', c' - d')$
- Note that M has 2^{24} rows

SCMEA Secondary Phase: Meet-in-the-Middle

- Given the table M
- For each (K_4, K_5, K_6, K_7) find a'' such that
$$R(a) = C[a''] + a$$
$$S(a) = C[(R(a) \oplus K_4) + K_5] + a$$
$$T(a) = C[(S(a) \oplus K_6) + K_7] + a$$
- And similarly for b'', c'', d''

SCMEA Secondary Phase: Meet-in-the-Middle

- Recall $R(a) = C[(Q(a) \oplus K_2) + K_3] + a$
- Since $a' = Q(a) \oplus K_2$ and $R(a) = C[a''] + a$
- We have $a'' = a' + K_3$
- Then
$$a'' - d'' = (a' + K_3) - (d' + K_3) = a' - d'$$
- And
$$(a'' - d'', b'' - d'', c'' - d'') = (a' - d', b' - d', c' - d')$$

SCMEA Secondary Phase: Meet-in-the-Middle

- Bottom line: $(a'' - d'', b'' - d'', c'' - d'')$ forms index into table M
- If such an entry exists, it matches some $(a' - d', b' - d', c' - d')$
- We have met-in-the-middle!
- Know putative $(K_0, K_1, K_2, K_4, K_5, K_6, K_7)$
- Compute $K_3 = a'' - a'$

SCMEA Secondary Phase: Meet-in-the-Middle

- Note: Some chance of false alarm
 - Must test each putative key by trial decryption
- Note: backtracking and meet-in-the-middle can be combined
 - Use backtracking to find putative keys
 - Use meet-in-the-middle to on the resulting putative keys

CMEA Known Plaintext Attack

- ❑ Almost the same as SCMEA attack
- ❑ More known plaintext required to mark impossible entries in A tables
 - Due to ambiguity on low-order bit
- ❑ Once the A tables have been found, attack is exactly the same as SCMEA

More Secure CMEA?

- ❑ Skewed Cave Table is crucial for attack
- ❑ What if we make Cave Table a perm?
 - Make Cave Table is a key-dependent permutation?
- ❑ Eliminate attacks discussed here
- ❑ But are there other attacks?

CMEA Conclusions

- ❑ Designed to be highly efficient
 - At the expense of some security
- ❑ Cave Table is unusual
- ❑ Attacks use combinatorial algorithms
- ❑ CMEA is a weak block cipher
 - But interesting