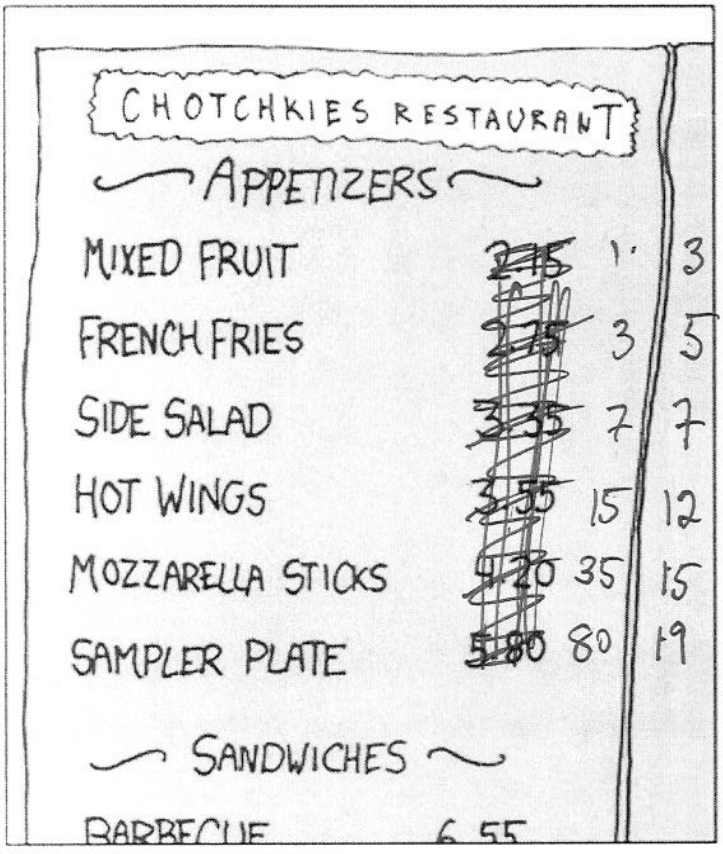
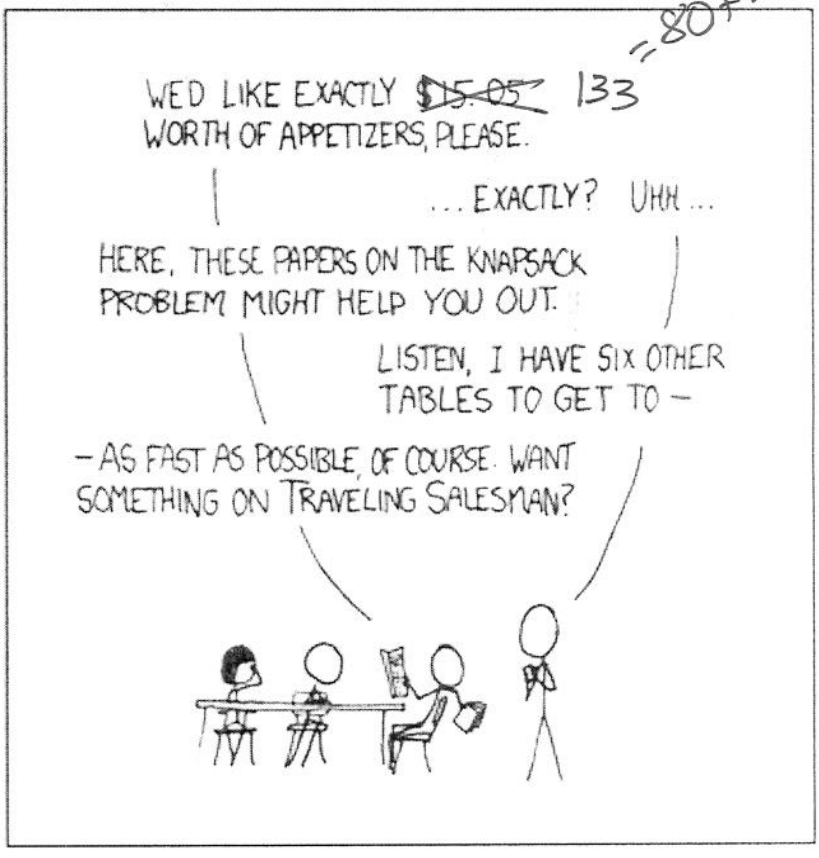
80+35+15+3
=
+18
80+35
=
80+53

MY HOBBY:

EMBEDDING NP-COMPLETE PROBLEMS IN RESTAURANT ORDERS





1+2+4+8+11

1+3+7+15+35+80

Random Superincreasing Sequence

Fix $n \geq 1$ and $k > 1$. Then

1. Let s_1 be a random number between 1 and k.
2. For i from 2 to n, let

$$s_i = s_1 + s_2 + \cdots + s_{i-1} + m_i,$$

where m_i is a random number between 1 and k

Merkle-Hellman Knapsack Cryptosystem

1. Choose a superincreasing sequence

a and $s_1, s_2, \ldots, s_n$. ← secret

2. Choose p to be a large prime such that

$$p > s_1 + s_2 + \cdots + s_n.$$

M → convert to binary $x_1 x_2 \cdots x_n$

3. Let a be a random number between 1 and $p-1$ and publicly announce

$$t_i := as_i \mod p.$$ ← public

Encryption Process: To encode a message $(x_1, x_2, \ldots, x_n)$ (made of bits of 0 and 1), one sends the single number

$$C := \sum_{i=1}^{n} x_i t_i. \pmod p$$

Encryption Process: To decode, we need only solve the subset sum problem for

$$M := a^{-1}C \mod p.$$

$$= \sum_{i=1}^{n} x_i s_i$$

Merkle-Hellman: Example

Let ↓ secret key

$$\{3, 5, 12, 21, 43\} \quad p = 89 \quad a = 15$$

Therefore, the T sequence is given by: $\frac{12}{180}$

$$\{45, 75, 2, 48, 22\} \pmod{89}$$

↑ public key

Encode 01101 by:

$$\begin{aligned} C &= 0 \cdot 45 + 1 \cdot 75 + 1 \cdot 2 + 0 \cdot 48 + 1 \cdot 22 \\ &= 10 \mod 89 \end{aligned}$$

To decode, since $a^{-1} = 6 \mod 89$, we have that

$$\begin{aligned} M = a^{-1}C = 60 &= 17 + 43 \\ &= 5 + 12 + 43 \\ &= 0 \cdot 3 + 1 \cdot 5 + 1 \cdot 12 + 0 \cdot 21 + 1 \cdot 43 \end{aligned}$$

↓

0 1 1 0 1

$89 = 5 \cdot 15 + 14$

$15 = 1 \cdot 14 + 1$

$1 = 15 - 14$

$= 15 - (89 - 5 \cdot 15)$

$= 6 \cdot 15 - 89$

$6 \cdot 15 \equiv 1 \pmod{89}$

$15^{-1} \equiv 6 \pmod{89}$

E Bit-Selection Table

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Permutation P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Selection Function S_1

middle 4 digits

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

first & last input bits

00 = 0
01 = 1
10 = 2
11 = 3

Example: 110100

- Use first and last digit as row index: 10 (base 2) = 2
- Use middle four digits as column index: 1010 (base 2) = 10
- The number 9 appears in row 2, column 10
- 9 = 1001 (base 2)

$S_1(110100) = 1001 = 2^3 + 2^0 = 9$

$10 = 2^1$

$1010 = 2^3 + 2^1 = 10_{\text{base } 10}$

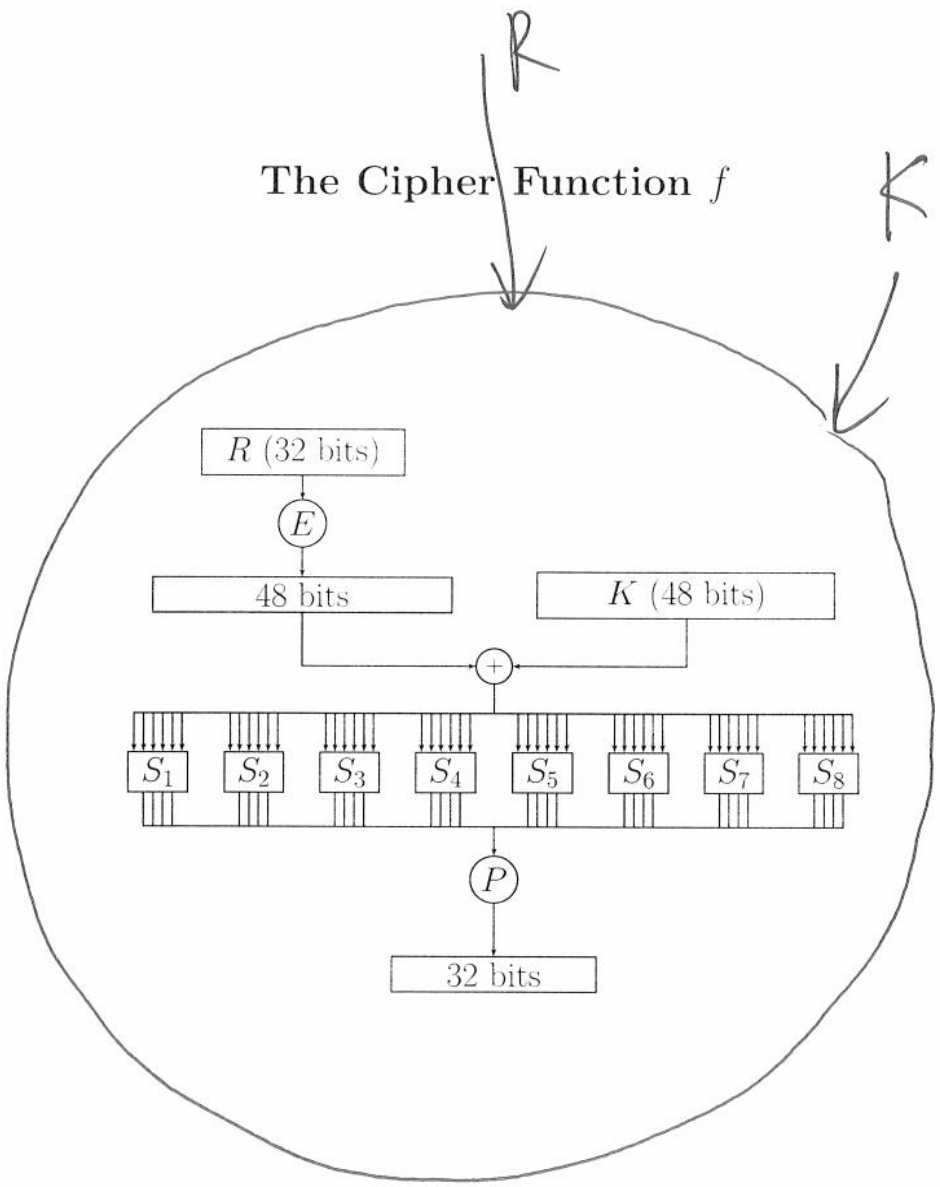
Initial Permutation

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

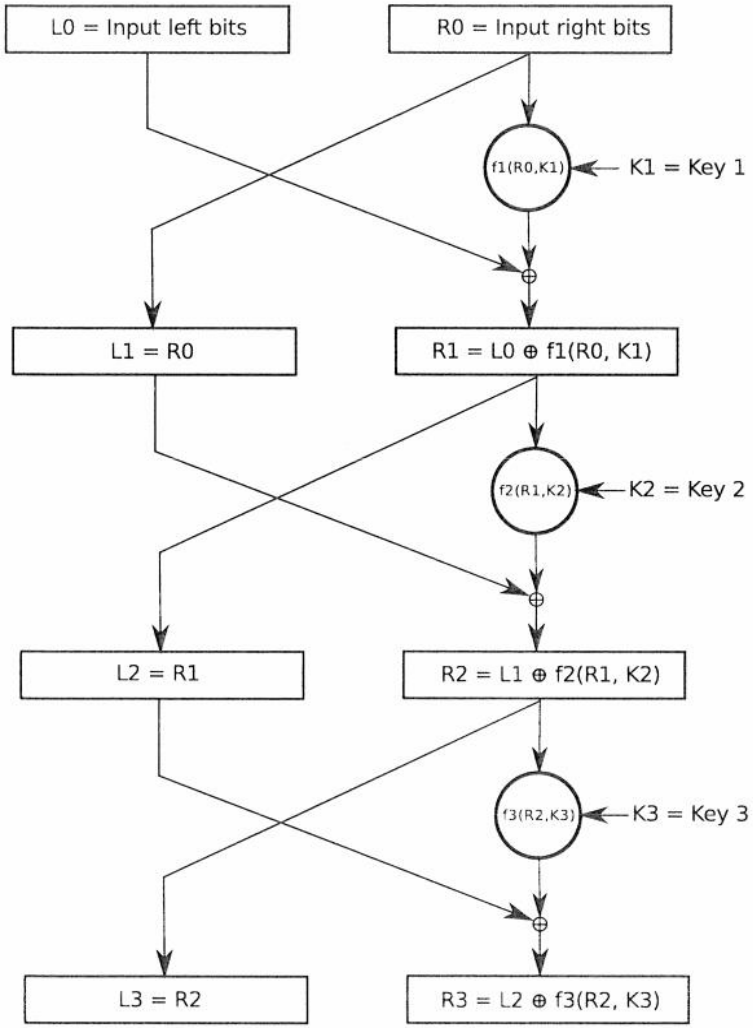
Inverse Permutation

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

The Cipher Function f



Feistel cypher repeated



If you repeat a couple of layers of this cypher, then all bits get scrambled quite well and with a good choice of functions and keys this cipher provides a good level of security.

This basic cipher is the building block of most modern ciphers (e.g. DES and Blowfish). Some variations involve splitting the numbers of left and right bits differently

Example of bit operations:

```
 1011 0001 0101 1111
⊕0111 1101 0001 0101
 1100 1100 0100 1010
```

Basic operation

⊕ XOR

Binary code for values:

0 = 0000	8 = 1000
1 = 0001	9 = 1001
2 = 0010	10 = 1010
3 = 0011	11 = 1011
4 = 0100	12 = 1100
5 = 0101	13 = 1101
6 = 0110	14 = 1110
7 = 0111	15 = 1111

$2^1+2^0 = 3$

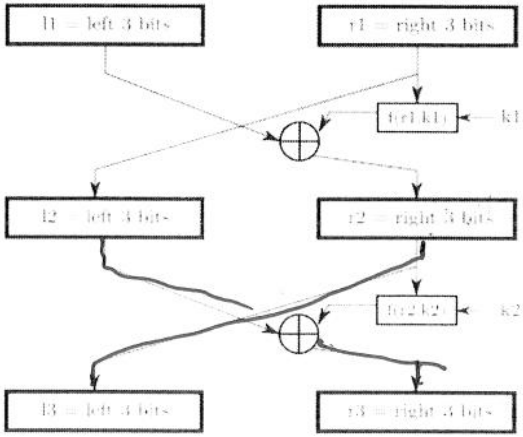
$11 = 1011 = 2^3+2^1+2^0$

$13 = 1101 = 2^3+2^2+2^0$

B 001011 a 100101 t 111000

0	000000	0	1	000001	1	2	000010	2	3	000011	3	4	000100	4	5	000101	5
6	000110	6	7	000111	7	8	001000	8	9	001001	9	A	001010	10	B	001011	11
C	001100	12	D	001101	13	E	001110	14	F	001111	15	G	010000	16	H	010001	17
I	010010	18	J	010011	19	K	010100	20	L	010101	21	M	010110	22	N	010111	23
O	011000	24	P	011001	25	Q	011010	26	R	011011	27	S	011100	28	T	011101	29
U	011110	30	V	011111	31	W	100000	32	X	100001	33	Y	100010	34	Z	100011	35
.	100100	36	a	100101	37	b	100110	38	c	100111	39	d	101000	40	e	101001	41
f	101010	42	g	101011	43	h	101100	44	i	101101	45	j	101110	46	k	101111	47
ℓ	110000	48	m	110001	49	n	110010	50	o	110011	51	p	110100	52	q	110101	53
r	110110	54	s	110111	55	t	111000	56	u	111001	57	v	111010	58	w	111011	59
x	111100	60	y	111101	61	z	111110	62	,	111111	63						

A Feistel cipher is used according to the following diagram encrypting each letter of a message separately. The left and right 3 bits correspond to a number between 0 and 7. The function $f(r,k) = (r+k)^2 + 3r + k \ (mod\ 8)$ is used in the Feistel cipher with $k_1 = 3$ and $k_2 = 0$ to encrypt a three letter word. Find the binary representations of $r_1, \ell_1, f(r_1,k_1), r_2, \ell_2, f(r_2,k_2)$ and the plaintext for the cyphertext *FoQ* (these correspond to l_3 and r_3 in the diagram).



B
First letter
ℓ_1: 001 r_1: 011 $f(r_1,k_1)$: 000 $001 = l_3$
ℓ_2: 011 r_2: 001 $f(r_2,k_2)$: 100 $111 = r_3$ F

a
Second letter
ℓ_1: 100 r_1: 101 $f(r_1,k_1)$: 010 $110 = l_3$
ℓ_2: 101 r_2: 110 $f(r_2,k_2)$: 110 $011 = r_3$ O

t
Third letter
ℓ_1: 111 r_1: 000 $f(r_1,k_1)$: 100 $011 = l_3$
ℓ_2: 000 r_2: 011 $f(r_2,k_2)$: 010 $010 = r_3$ Q