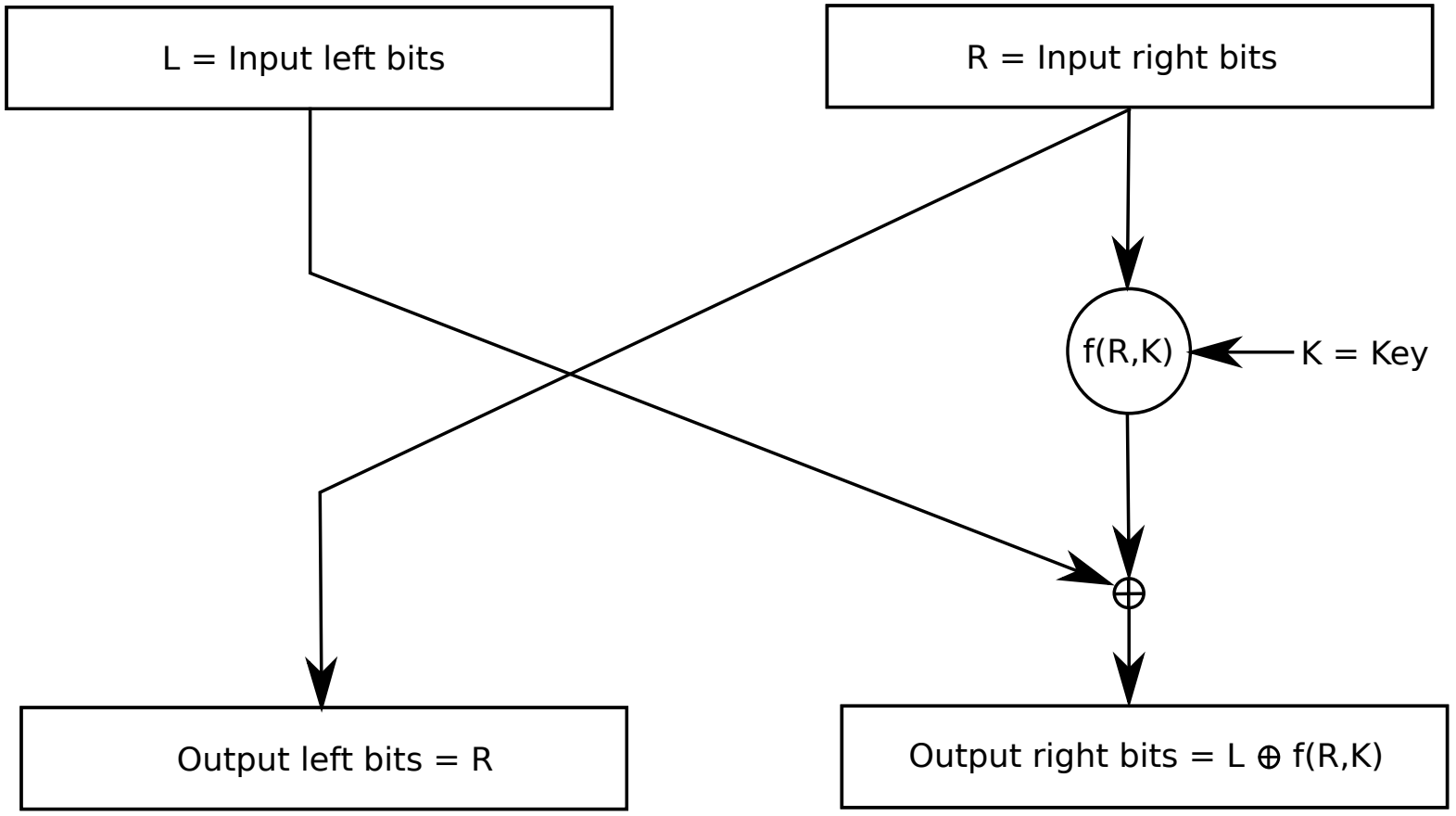
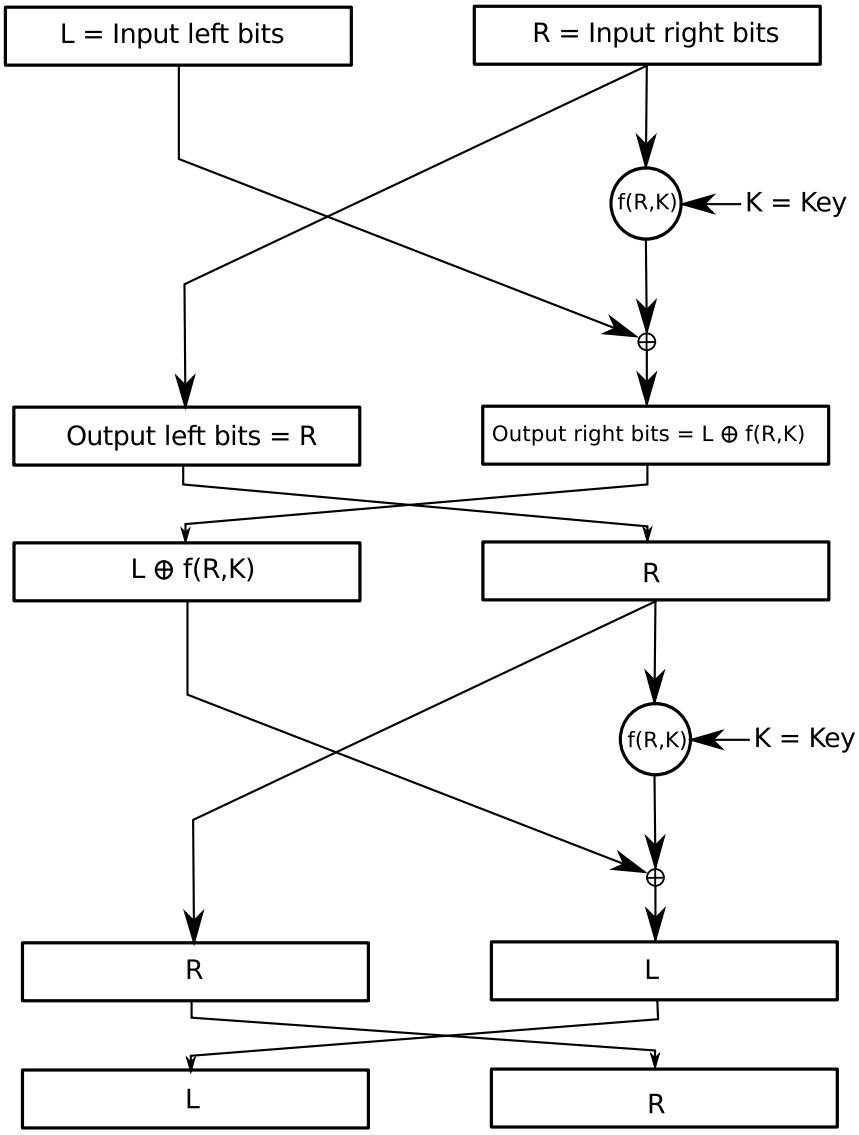
Feistel cypher



Given input in binary form, break it into LEFT and RIGHT bits (usually in half)

The right bits are transferred over to the left bits for output and the left bits are XORed with some function of the right bits and the key.

Decrypt Feistel cypher



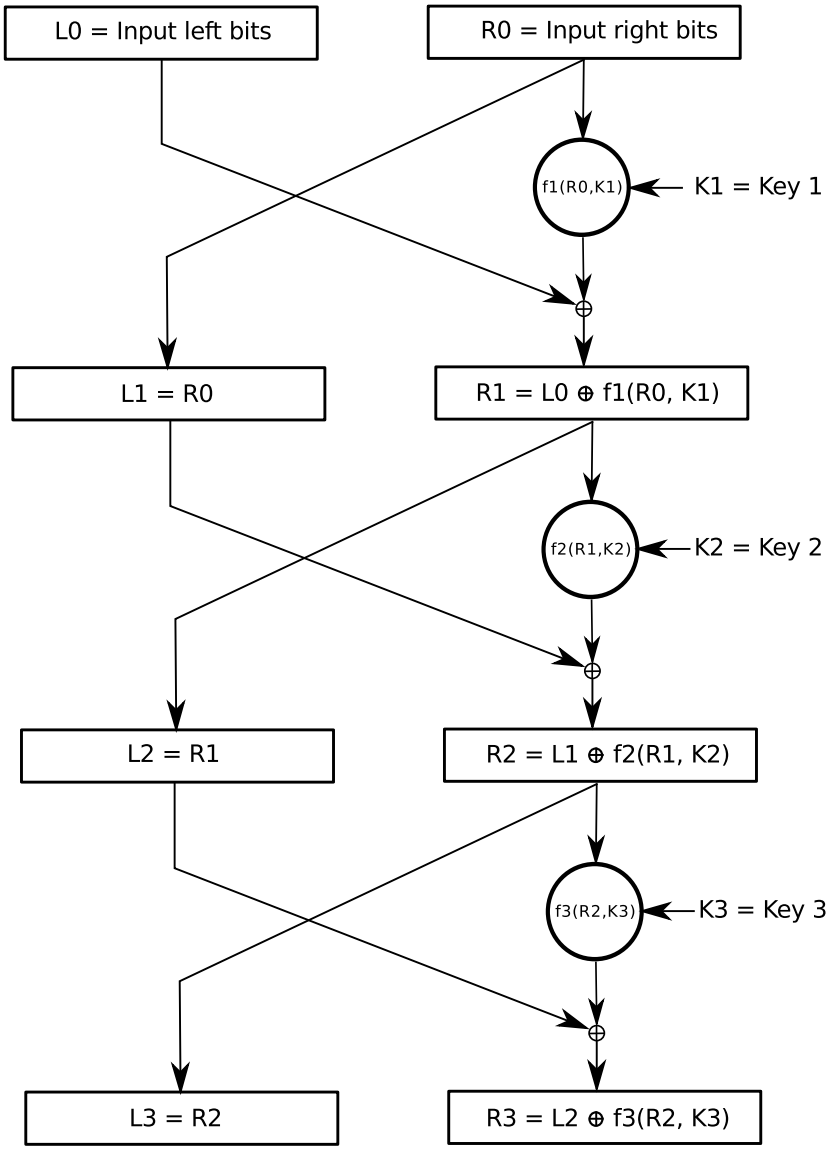
To encrypt use key K and only 'scramble' left bits and exchange them with right bits

To decrypt swap the input and the output and use the EXACT same key

One advantage of Feistel is that the reverse of the procedure is doing the same thing a second time with the output left and right bits switched

The main advantage of the Feistel cypher is speed. Bit operations and the functions f(R,K) can easily be built into chips and the time cost for encrypting and decrypting is minimal. This is an especially important feature if you want to communicate large amounts of data such as voice or video.

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:

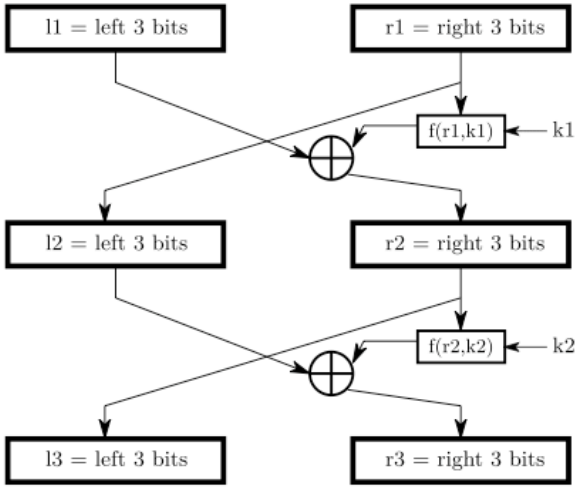
		Basic operation
	1011 0001 0101 1111	
⊕	0111 1101 0001 0101	⊕ XOR
	1100 1100 0100 1010	

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

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).



First letter

ℓ_1: _____ r_1: _____ $f(r_1,k_1)$: _____

ℓ_2: _____ r_2: _____ $f(r_2,k_2)$: _____

Second letter

ℓ_1: _____ r_1: _____ $f(r_1,k_1)$: _____

ℓ_2: _____ r_2: _____ $f(r_2,k_2)$: _____

Third letter

ℓ_1: _____ r_1: _____ $f(r_1,k_1)$: _____

ℓ_2: _____ r_2: _____ $f(r_2,k_2)$: _____