Results of the block cipher design contest

The table below contains a summary of the best attacks on the ciphers you designed. 13 of the 17 ciphers were successfully attacked in HW2, and as you can see from the table, many of these attacks have surprisingly low complexities. The lesson seems to be that designing secure, efficient block ciphers is somewhat tricky.

I'm not aware of any attacks on the remaining 4 designs. Of those 4, which are highlighted in bold italics below, 2 appear to be very efficient (I estimate that Borisov's and Johnson's ciphers may be able to encrypt as fast as ≈ 10 cycles/byte), and the other 2 are slower (I predict that Obata's will run at ≈ 60 cycles/byte, and Twohey's at ≈ 120 cycles/byte). For comparison, the AES runs at ≈ 15 cycles/byte.

	Designer	Best attacks, credit		Attack technique
0	***Borisov***	—	[Wagner]	no good attacks known Note: exhaustive keysearch can be sped up by $2\times$, due to minor flaw in key schedule
1	Chen	3 CP	[Johnson]	differential: $(\alpha,0,\ldots,0) \to (?,\ldots,?,0)$ holds with prob. 1 for all but last round; then guess 16 bits of last-round subkey
2	Geels	32 CP	[Raja.-Rao]	attack given in class on $S \circ L \circ S$
3	Harren	2 CP	[Raja.-Rao, Reichardt]	differential: $(\alpha,0,\ldots,0) \to (0,?,\ldots,?)$ holds with prob. $\frac{1}{2}$
4	***Johnson***	—		no attacks known
5	Kissner	30 CP	[Harren]	differential: flip one bit in input; then only 26 of 64 bits are expected to change after 3 rounds
6	Karlof	1 CP	[Reichardt, Shankar, Sorkin]	$E_k(0) = 0$ for all k
7	Li	1 CP 8 CP	[Sorkin] [Shankar]	$E_k(0) = 0$ if $IV = 0$; for $IV = a$, $E_k((a,0,\ldots,0)) = 0$ multiplicative truncated differential: $(-1,-1,0,\ldots,0) \to (-1,?,\ldots,?)$ holds with prob. $\frac{1}{4}$
8	Manapat	2 KP	[Reichardt]	it's linear!
9	***Obata***	—		no attacks known
10	Raja.-Rao	32 CP	[Borisov]	differential: flip an input bit, and only 52 of 64 bits are affected after 6 of 8 rounds; guess ≤ 30 subkey bits in last two rounds
11	Reichardt	256 CP	[Harren]	attack given in class on $S \circ L \circ S$
12	Rhea	2^{32} CP 2^{17} CP	[Chen] [Wagner]	truncated differential: $(0,\alpha) \to (\alpha,0)$ has prob. 2^{-32} improve Chen's attack by choosing plaintexts of form $(0,x)$
13	Shankar	2^{33} KP	[Harren]	find collisions with birthday attack; cipher not reversible
14	Sorkin	2 CP	[Chen]	differential: $(\alpha,0,\ldots,0) \to (?,\ldots,?,0)$ holds with prob. 1 for all but last round; then guess 8 bits of last-round subkey
15	Soto	2^{17} CP	[Raja.-Rao]	truncated differential: $(0,\alpha,0,0) \to (\alpha,0,0,\alpha')$ holds with prob. 2^{-32}
16	***Twohey***	—		no attacks known

Notation: CP = chosen plaintexts, KP = known plaintexts.
The work factor was never much more than the data complexity.

Sample solutions for the March 5th homework

As for the ciphers given in the homework assignment, here is a summary of the best attacks I know of. More detailed solutions follow.

Problem	Best attacks	Attack technique
1 Finite-field cipher	4 KP, ∞ rounds	rational polynomial interpolation
2 SPN: 2×2 S-boxes	2 KP, ∞ rounds	it's linear!
2 SPN: 3×3 S-boxes	193 CP, ∞ rounds, 2.8% of ciphers 2^{40} CP, 30 rounds, 44.0% of ciphers 2^{40} CP, 20 rounds, 99.8% of ciphers	diff. crypt.: 1-round iterative char. of prob. 1 diff. crypt.: 30-round char. of prob. 2^{-31} diff. crypt.: 20-round char. of prob. 2^{-31}
3 Odd architecture	4 KP, ∞ rounds	linear cryptanalysis: parity is preserved
4 Keying perm.'s	2 KP, 2^{80} work, 5 rounds	meet-in-the-middle attack, plus a trick

1 Finite-field ciphers

The problem. Define a round function $R_k(x) = I(x+k)$ where $I(x) = x^{-1}$ is the inversion map; then the block cipher is

$$E_k(x) = R_{k_n}(\cdots(R_{k_1}(x))\cdots).$$

Summary. We can break an arbitrary number of rounds, with about 4 known plaintexts and a small, constant work factor. The analysis follows by expressing the cipher as a ratio of linear polynomials.

The following explanation of the attack appears here thanks to Nikita Borisov.

Analysis. We use the following simple fact:

Claim 1. $E_k(x) = \frac{ax+b}{cx+d}$ *for some* a,b,c,d, *which depend only on the key.*

Proof. This fact can be verified by induction on the number of rounds. After 0 rounds, we have that $x = \frac{1x+0}{0x+1}$, proving the base case. Next suppose that after n rounds we have

$$R_{k_n}(\cdots(R_{k_1}(x))\cdots) = \frac{ax+b}{cx+d}.$$

Then, after $n+1$ rounds, we have

$$\begin{aligned} R_{k_{n+1}}(R_{k_n}(\cdots(R_{k_1}(x))\cdots)) &= \left(\frac{ax+b}{cx+d} + k_{n+1}\right)^{-1} \\ &= \left(\frac{ax+b+k_{n+1}cx+k_{n+1}d}{cx+d}\right)^{-1} \\ &= \frac{cx+d}{(a+k_{n+1}c)x+(b+k_{n+1}d)} \end{aligned}$$

Setting $a' = c, b' = d, c' = a+k_{n+1}c, d' = b+k_{n+1}d$ leaves the result in the desired form. $\square$

Now we are left with the task of cryptanalyzing the algorithm

$$E_{a,b,c,d}(x) = \frac{ax+b}{cx+d}$$

Given a known plaintext x and its matching ciphertext y, we know that $\frac{ax+b}{cx+d} = y$. Cross-multiplying and simplifying yields

$$xa + b - xyc - yd = 0.$$

This is a linear equation in the 4 unknowns a, b, c, d (since x and y are known). If we have 4 known plaintexts, we can obtain 4 such equations, which will in general be sufficient to solve for a, b, c, d.

Note that this gives an efficient attack on a cipher that is provably secure against all differential, linear, or key-recovery attacks. (The former two results come from the homework assignment and from what we showed in class, and the cipher is secure against key-recovery attacks if it uses 5 or more rounds because in this case each key has at least 2^{128} equivalent keys.) Also, the same attack applies even to a generalization of this cipher where $R_{k,k'} = I(k \times x + k')$.

2 Substitution-permutation networks

2.1 2x2 S-boxes

Summary. We can attack any number of rounds of this cipher, using 2 known plaintexts and a small constant amount of work.

The following explanation of the attack borrows from the write-up of Nikita Borisov.

Analysis. The main insight is that the cipher is *always* linear.

Claim 2. *Every bijective 2x2 S-box is affine.*

Proof. Suppose $S(a_0, a_1) = (b_0, b_1)$, and consider just the first bit b_0 of the S-box. We can build the following table of all possibilities for the S-box; each line expresses the first output bit of the S-box in terms of the input bits:

a_0	a_1	b_0					
0	0	0	0	0	1	1	1
0	1	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	1	1	1	0	1	0	0
$b_0 =$		a_0	a_1	$a_0 + a_1$	$a_0 + a_1 + 1$	$a_1 + 1$	$a_0 + 1$

Note that this table is exhaustive, since b_0 has to assume 0 and 1 an equal number of times. Moreover, in each case b_0 is an affine function of a_0 and a_1. The same will be true for b_1, hence every such S-box is affine. □

As a consequence, we can note that the entire cipher is linear.

Claim 3. *The cipher has the form $E_k(x) = M \cdot x + z + k'$ for some fixed 128×128 matrix M, some fixed 128-bit vector z, and some 128-bit subkey k' dependent only on the key k.*

Proof. Consider one round of the cipher, $R_k(x) = P(T(x+k))$. We can write each S-box as an affine function $S(a) = M_S \cdot x + z_S$ for fixed M_S, z_S, hence we can write the entire T transformation as an affine function $T(x) = M_T \cdot x + z_T$ for some fixed 128×128 matrix M_T and some fixed 128-bit vector z_T (these depend only on the choice of S-boxes and not on the key). Also, every bit-permutation is linear, so P can be represented as a 128×128 matrix. We see that there exists M_R, c_R so that $R_k(x) = M_R \cdot (x+k) + z_R = M_R \cdot x + z_R + k'$ where $k' = M_R \cdot k$.

If we iterate many rounds, the result will still be linear, as the composition of linear maps is linear. Consider, say, $R_{k_2}(R_{k_1}(x))$: we have

$$R_{k_2}(R_{k_1}(x)) = M_R \cdot (M_R \cdot x + z_R + k'_1) + z_R + k'_2 = M' \cdot x + z' + k''$$

where $M' = M_R^2$, $z' = M_R \cdot z_R + z_R$, and $k'' = M_R \cdot k'_1 + k'_2$. By induction, a similar result holds for the entire cipher, no matter how many rounds it has. □

Breaking the cipher is now easy. Obtain a known plaintext-ciphertext pair (p,c). Then we can compute an equivalent key k' via $k' = c + M \cdot p + z$, where M, z are as in the statement of the claim above. This reveals the cipher key, which allows us to decrypt any further ciphertexts we may see. For instance, to construct a distinguishing attack we may obtain an additional known plaintext-ciphertext pair (p', c') and then test whether $M \cdot p' + z + k' \stackrel{?}{=} c'$; the latter equality will always hold if it is the real cipher, but will rarely hold if we are given a random permutation. There are also trivial linear characteristics of bias 1 and differential characteristics of probability 1 that could alternately be used in a linear or differential attack to distinguish this cipher from a random permutation.

2.2 3x3 S-boxes

I show, for a large fraction of ciphers designed in this way, many rounds will be breakable with differential cryptanalysis. For instance, about 44% of these ciphers have 30-round differential characteristics with probability 2^{-31} or more. In more detail: for fixed S-boxes $S_1, \ldots, S_{64}$ and permutation P, let $\rho(S_1, \ldots, S_{64}, P)$ denote the maximum number of rounds coverable by a differential characteristic with probability at least 2^{-40}, i.e.,

$$\rho(S_1, \ldots, S_{64}, P) = \max\{r : \exists \Delta, \Delta' \neq 0 \text{ such that } \Pr[\Delta \to \Delta' \text{ for } r \text{ rounds}] \geq 2^{-31}\}.$$

Then I will show that $\Pr[\rho(S_1, \ldots, S_{64}, P) \geq 30] \approx 0.44$, where the probability is taken over the choice of S-boxes and permutation. This indicates that the proposed design algorithm—namely, picking S-boxes and permutations at random—leaves one with a cipher that is not likely to be very secure.

My results are summarized in the following table, which shows the cumulative distribution of ρ:

Rounds (r)	$\Pr[\rho \geq r]$	Attack complexity
∞	0.030	193 CP
63	0.040	$\leq 2^{40}$ CP
30	0.438	$\leq 2^{40}$ CP
25	0.636	$\leq 2^{40}$ CP
20	0.998	$\leq 2^{40}$ CP

For instance, this table shows that for about 44% of the ciphers, we can break 30 rounds or more with at most 2^{40} chosen plaintexts (there exists an 30-round differential characteristic of probability 2^{-31} or greater). For about 3% of the ciphers, we can break any number of rounds with 193 chosen plaintexts (there exists a non-trivial iterative differential characteristic of probability 1).

These values were calculated empirically by generating 10000 ciphers at random and analyzing each one to find the best differential characteristic. I only looked for differential characteristics that have exactly one active S-box in each round, hence the above should be viewed as a lower bound on the number of rounds that can be broken in a differential attack. Moreover, I did not take into account the possibility of using "structures" to bypass the first round nor the possibility of leaving the last 4 rounds uncovered (the one-bit difference can only avalanche to at most a 81-bit difference after 4 rounds, hence right pairs can still be recognized). In practice, it seems reasonable to expect that differential attacks can break at least 5 rounds more than the above table would suggest.

I'll give some intuition for where these numbers might come from. Consider any one-bit difference e_i, and let's ask when there exists an iterative differential characteristic $e_i \to e_i$ with non-zero probability for one round. This happens just when $e_i \xrightarrow{T} e_j$ by the S-boxes and $P(j) = i$. Note that $P^{-1}(i)$ had better refer to a bit position at the output of the same S-box that e_i enters, and this condition is satisfied for $1/64$ of the permutations P. Moreover, when this condition is satisfied, only $(6/7)^4 \approx 0.54$ of the S-boxes do not ever send the difference e_i to e_j with non-zero probability. Of course, when $e_i \to e_j$ does have non-zero probability, its probability will be at least $1/4$. Thus, we see that for at least $0.46 \times 1/64 \approx 0.007$ of the ciphers, we have an iterative one-round differential characteristic with probability $1/4$, and this leads to a 20-round characteristic with probability 2^{-40}. So at least 0.7% of the 20-round ciphers are breakable in this way.

Of course, this vastly underestimates the fraction of breakable ciphers, because it only considers a very narrow class of differential characteristics. If we consider r-round iterative differential characteristics, the fraction q of ciphers with such a characteristic of probability $\geq (1/4)^r$ is roughly

$$q \approx 1 - \left(1 - \frac{p^r}{192}\right)^{3^r} \approx 1 - e^{-(3p)^r/192}, \qquad \text{where } p = 1 - (6/7)^4 \approx 0.46.$$

Note that q is close to 1 when $r \approx 20$, since then $(3p)^r/192 \approx 3.3$ and thus $q \approx 1 - e^{-3.3}$. (In general, we expect q to grow with r, because there are more possible trails for the characteristic to follow as we add more rounds.) This gives some intuitive justification to explain why almost all 20-round ciphers can be broken with differential cryptanalysis using at most 2^{40} chosen plaintexts.

It may be surprising that there is a non-negligible fraction of ciphers breakable for any number of rounds. This, however, can be explained by the following sort of consideration. Consider the differential characteristic $1 \to 1$ for one round, and imagine first that $P(1) = 1$ and $1 \to 1$ holds with probability 1 for the first S-box, S_1. Then we obtain an iterative differential characteristic of probability 1, and the conditions hold for $(4! \times 2^4/8!) \times 1/192 \approx 2^{-14.3}$ of all ciphers designed in this way. More generally, if we consider differential characteristics of the form $e_i \xrightarrow{T} e_j \xrightarrow{P} e_i$, there are 192×3 choices for i, j, and each one gives a $2^{-14.3}$ chance at a probability-one characteristic. All in all, one can estimate that for $1 - (1 - 2^{-14.3})^{192\times 3} \approx 0.028$ of all ciphers, there exists some one-round iterative differential characteristic with probability 1. For this subset of weak ciphers, one can break any number of rounds with just 193 chosen plaintexts (we need to hit all 192 possible input differences once).

3 A slightly different architecture

The problem. Fix a 64-bit nonlinear function f. Define $T(u, v) = (u + f(u + v), v + f(u + v))$, $U(x, y) = (x \lll y, y \lll (x \lll y))$, and $R_k(x) = U(T(x + k))$; then the block cipher is

$$E_k(x) = R_{k_n}(\cdots(R_{k_1}(x))\cdots).$$

Summary. We can distinguish any number of rounds of the cipher with about 4 known plaintexts and a small constant amount of work, using linear cryptanalysis.

The following explanation of the attack borrows from the write-up of Nikita Borisov.

Analysis. Let $\Gamma = 1^{128} = 11\cdots11$. Then $\Gamma \cdot x$ is the parity of x. Consider the linear characteristic $\Gamma \cdot E_k(x) = \Gamma \cdot x$.

Claim 4. *The equation $\Gamma \cdot E_k(x) = \Gamma \cdot x + k'$ holds for all x and all k, where $k' \in \{0,1\}$ depends only on the key k.*

Proof. First, we will show that this holds for one round. The round function is $R_k(x) = U(T(x+k))$. We can see that $\Gamma \cdot U(x) = \Gamma \cdot x$ always holds, as U is simply a bit permutation, which preserves the parity of its input. More surprisingly, $\Gamma \cdot T(x) = \Gamma \cdot x$. To see this, let $x = (u, v)$ where $u, v \in 0, 1^{64}$ and $\Gamma' = 1^{64}$. Then

$$\Gamma \cdot x = \Gamma' \cdot u + \Gamma' \cdot v$$

and

$$\begin{aligned} \Gamma \cdot T(x) &= \Gamma' \cdot (u + f(u+v)) + \Gamma' \cdot (v + f(u+v)) \\ &= \Gamma' \cdot u + \Gamma' \cdot v + \Gamma' \cdot f(u+v) + \Gamma' \cdot f(u+v) \\ &= \Gamma' \cdot u + \Gamma' \cdot v = \Gamma \cdot x \end{aligned}$$

So, we have

$$\Gamma \cdot R_k(x) = \Gamma \cdot U(T(x+k)) = \Gamma \cdot (x+k) = \Gamma \cdot x + \Gamma \cdot k.$$

Thus, for the full cipher, we have

$$\Gamma \cdot E_k(x) = \Gamma \cdot x + \sum_{i=1}^{n} \Gamma \cdot k_i.$$

Taking $k' = \sum_{i=1}^{n} \Gamma \cdot k_i$ yields the claimed result. $\square$

It is now straightforward to break the cipher using linear cryptanalysis. Since the term k' is constant for a fixed choice of subkeys, the linear characteristic $\Gamma \to \Gamma$ has bias 1 for any number of rounds of the cipher. We can verify this bias with 3–4 known plaintexts, which gives a distinguishing attack with large advantage.

Also, this shows that the cipher leaks the parity of plaintexts. Given one known plaintext, one can learn the parity of the decryption of all subsequent ciphertexts you see, and this is arguably not such a good property.

A different attack. A few others noted the existence of a differential attack. The first observation is that, when $x + x' = y + y'$ and $y \equiv y' \equiv x \lll y \equiv x' \lll y' \pmod{64}$, the output difference $U(x, y) + U(x', y')$ is just a rotation of the input difference $(x + x', y + y')$. Also, the differential characteristic $(\Delta, \Delta) \to (\Delta, \Delta)$ passes through the key xor and the T transformation with probability 1. Consequently, we obtain a one-round differential characteristic $(\Delta, \Delta) \to (\Delta', \Delta')$ for the whole round function, and if we choose $\Delta = 0^i 10^{63-i}$ and $\Delta' = 0^j 10^{63-j}$, this characteristic will have probability approximately $59/64 \times 1/64 \approx 2^{-4.24}$. Thus, one can break up to 11 rounds or so in a differential attack using up to 2^{40} chosen plaintexts. Though this attack is considerably weaker than the linear cryptanalytic attack, I still liked it.

A truncated differential attack. There is even a truncated differential attack. Let $S = \{\Delta \in \{0,1\}^{128} : \Gamma \cdot \Delta = 0\}$ be the set of 128-bit differences with even parity. Then the truncated differential $S \to S$ holds with probability 1, i.e., $\Pr[E_k(x) + E_k(x') \in S \mid x + x' \in S] = 1$. This follows as a straightforward consequence of the linear characteristic, since $\Gamma \cdot (E_k(x) + E_k(x')) = \Gamma \cdot E_k(x) + \Gamma \cdot E_k(x') = \Gamma \cdot x + \Gamma \cdot x' = \Gamma \cdot (x + x')$. Actually, I think it's much more natural to think of this as a linear attack instead of a truncated differential attack, but if you like you can always think in the latter terms instead.

4 Keying an unkeyed permutation

The problem. Fix a 128-bit nonlinear bijective function f. If k is a 40-bit string, let $\overline{k}$ denote the result of concatenating k to itself enough times to obtain a 128-bit string. Define the round function $R_k(x) = f(x+\overline{k})$; then the block cipher is

$$E_k(x) = R_{k_n}(\cdots(R_{k_1}(x))\cdots).$$

Summary. Using a meet-in-the-middle attack, we can break 5 rounds of the cipher using about 2^{80} work and 2 known plaintexts.

Analysis of the 4-round reduced cipher. To give the intuition, we first describe a simpler case: how to break a 4-round cipher. We will use a meet-in-the-middle attack. Fix a known plaintext-ciphertext pair (p,c), and define

$$\begin{aligned} g(\kappa_1,\kappa_2) &= f(f(p+\overline{\kappa_1})+\overline{\kappa_2}) \\ h(\kappa_3,\kappa_4) &= f^{-1}(f^{-1}(c)+\overline{\kappa_4})+\overline{\kappa_3} \end{aligned}$$

We will use $\kappa_1,\dots,\kappa_4$ to denote a guess at the real key $k_1,\dots,k_4$. Note that the correct key $k_1,\dots,k_4$ satisfies

$$g(k_1,k_2) = h(k_3,k_4).$$

Hence, we will look for κ satisfying $g(\kappa_1,\kappa_2) = h(\kappa_3,\kappa_4)$, and with high probability this will be the correct key value, i.e., we will have $k = \kappa$.

To find κ satisfying this equation, we use a meet-in-the-middle. For each of the 2^{80} choices of κ_1,κ_2, we compute $g(\kappa_1,\kappa_2)$ and store the tuple $(g(\kappa_1,\kappa_2),\kappa_1,\kappa_2)$ in a sorted list or hashtable keyed on its first element. Then, for each of the 2^{80} possibilities for κ_3,κ_4, we compute $h(\kappa_3,\kappa_4)$ and check whether $h(\kappa_3,\kappa_4)$ appears as a key value in the sorted list or hashtable. All in all, this requires 4×2^{80} computations of f (equivalent to 2^{80} trial encryptions) and 2^{80} space.

Analysis of the full 5 rounds. We'll use the same idea, but we'll exploit a trick to cancel out the effect of k_3. Let (p,c), (p',c') denote two known plaintext-ciphertext pairs. Define

$$\begin{aligned} g'(\kappa_1,\kappa_2) &= f(f(p+\overline{\kappa_1})+\overline{\kappa_2}) + f(f(p'+\overline{\kappa_1})+\overline{\kappa_2}) \\ h'(\kappa_4,\kappa_5) &= f^{-1}(f^{-1}(f^{-1}(c)+\overline{\kappa_5})+\overline{\kappa_4}) + f^{-1}(f^{-1}(f^{-1}(c')+\overline{\kappa_5})+\overline{\kappa_4}) \end{aligned}$$

Note that the correct key $k_1,\dots,k_5$ satisfies

$$g'(k_1,k_2) = h'(k_4,k_5).$$

This works, for the following reason: if we let a,a' denote the intermediate values after two f-applications but before xor-ing k_3, and let b,b' denote the intermediate values after the xor with k_3, we see that $b = a+k_3$, $b' = a'+k_3$, hence $a+a' = b+b'$.

Now we simply use a straightforward meet-in-the-middle attack to find $\kappa_1,\kappa_2,\kappa_4,\kappa_5$ satisfying $g'(\kappa_1,\kappa_2) = h'(\kappa_4,\kappa_5)$. This can be done with 8×2^{80} f-computations (equivalent to about 2^{81} trial encryptions) and 2^{80} space. This computation reveals k_1,k_2,k_4,k_5, and once they are known, k_3 can be recovered in a straightforward process.

There are other optimizations: this can be sped up by a factor of two or more, the data complexity can be reduced by a factor of two, one can reduce the total complexity by using more chosen plaintexts, there are ways to reduce the space usage, and so on. Thus, the attack I give here is not the best possible one. However, there was no need to get into any of that for the purposes of this question.