Required homework (Feb 19; due Mar 5)

Instructions. Turn your answers to this homework in, on paper, at the beginning of class on March 5th. You are welcome to discuss the problems in groups, but your final write-up must be your own.

Analyze the following block ciphers. In particular, list the maximum number of rounds you can break (i.e., distinguish from a random permutation), using at most 2^{80} work and 2^{40} texts. Your answer should include:

- How many rounds you can break.
- A summary of the complexity of the attack. List the workfactor, the number of plaintexts, whether you need known or chosen plaintexts, and any other parameters that may be relevant[1].

  You don't need to micro-optimize the complexity of your attack (I don't care about the difference between $O(n)$ and $O(n \log n)$), but try to avoid being grossly and unnecessarily inefficiently (e.g., $O(2^n)$ vs. $O(2^{n/2})$, or using chosen plaintexts where the same number of known plaintexts suffice).
- A short phrase summarizing the attack technique you use (e.g., differential cryptanalysis, linear cryptanalysis, meet-in-the-middle, internal birthday collision, etc.).
- Finally, give the detailed description of the attack.

You can make my life easier by writing the first three items above at the top of your answer and circling it.

If you have trouble analyzing any of these ciphers, don't despair.

1 Finite-field ciphers

Fix a representation of elements of the finite field $GF(2^{128})$ as 128-bit strings. Define a round function $R_k(x) = I(x+k)$ where $I(x) = x^{-1}$ is the inversion map in the finite field $GF(2^{128})$ and where $+$ denotes addition in the finite field, i.e., bitwise xor. (You may assume that $I(0) = 0$.) Then the block cipher will iterate the round function n times, using independent round subkeys in each round (so that the key is $128n$ bits long):

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

Remark. It is not hard to show that there are no good differential characteristics for even one round of this cipher. Hence, if you can break this, you can break a cipher that is "provably secure" against differential cryptanalysis.

Remark. If you're not familiar with finite fields, see Section 2.6 of *The Handbook of Applied Cryptography* (referred to on the course web page) for more information.

2 Substitution-permutation networks

(a) At design time, choose 64 random bijective 2-bit S-boxes $S_1, \ldots, S_{64} : \{0,1\}^2 \to \{0,1\}^2$. Fix and publish these S-boxes. Let $T : \{0,1\}^{128} \to \{0,1\}^{128}$ denote application of these in parallel. Let $P : \{0,1\}^{128} \to \{0,1\}^{128}$ denote some re-ordering of the bits chosen randomly by the designer and

[1] For instance, if your attack only works for a fraction of the keyspace, list the fraction of the keys you can break. If you need a huge amount of memory, list the amount of memory. If there is anything else that seems relevant, feel free to mention it.

fixed and published. The round function is $R_k(x) = P(T(x+k))$ where $+$ denotes bitwise xor. The block cipher will iterate the round function n times, using independent round subkeys in each round:

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

(b) The same as in part (a), except we use 3-bit S-boxes (so we get a 192-bit block cipher). As before, we choose 64 random bijective 3-bit S-boxes $S_1, \ldots, S_{64} : \{0,1\}^3 \to \{0,1\}^3$, and so on.

3 A slightly different architecture

Let $f : \{0,1\}^{64} \to \{0,1\}^{64}$ be a highly nonlinear but unkeyed function: I suggest letting $f(x)$ be the first 64 bits of SHA1(x), but you could think of it as any fixed, public, random-seeming function. Let $T : \{0,1\}^{128} \to \{0,1\}^{128}$ denote the transformation $T(u,v) = (u + f(u+v), v + f(u+v))$, where $+$ denotes bitwise xor. Let $U : \{0,1\}^{128} \to \{0,1\}^{128}$ denote the transformation $U(x,y) = (x \lll y, y \lll (x \lll y))$, where $a \lll i$ denotes the result of rotating the bit-string a left by i places. (If i is too large, it is taken modulo 64.) The round function is $R_k(x) = U(T(x+k))$. Note that this is easily computable in both the forward and backward direction. The block cipher will iterate the round function n times, using independent round subkeys in each round (so that the key is $128n$ bits long):

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

4 Keying an unkeyed permutation

Let $f : \{0,1\}^{128} \to \{0,1\}^{128}$ be a public, bijective, highly nonlinear function: I suggest thinking of $f(x)$ as $\mathrm{AES}_0(x)$, i.e., the result of encrypting x under AES with the all-zeros key, but you could think of it as any fixed, public, random-seeming bijective function. If k is a 40-bit key, let (k,k,k,k) be the result of concatenating it with itself enough times to get a 128-bit string (and truncating the result as necessary to get a 128-bit string). Also, define the round function $R_k(x) = f(x + (k,k,k,k))$, where $+$ denotes xor. The block cipher will iterate the round function n times, using independent round subkeys in each round (so that the key is $40n$ bits long):

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

5 Your neighbor's construction

You will find stapled to this homework three randomly chosen ciphers designed by someone else in this class. Analyze each of them, with the number of rounds specified. (You don't have to break it for any more rounds than were specified.)

To make my life easier, I'd like it if you interspersed the cipher description pages with your analysis (I won't remember what the specification of each cipher is unless you include the description with it).

If the cipher specification is at all ambiguous, feel free to pick any reasonable choice for disambiguating the specification that makes your life easier. After all, in security if the specification is ambiguous, some implementors will implement it one way and some another way, and so the cryptanalyst will have an opportunity to break the scheme in the scenario that is as favorable to her as possible, from among all plausible scenarios.