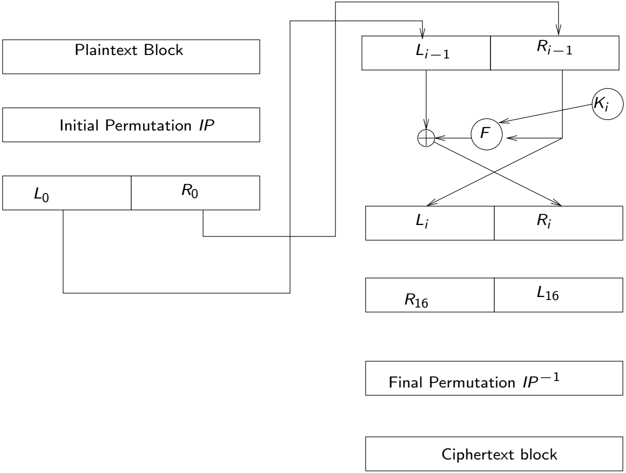
Network Security and Cryptography
Symmetric-key cryptography

Lecture 3: “2DES” and 3DES

Mark Ryan

Overview of DES



DES is not secure by today's standards

In any practical encryption system, such as DES, an attacker could try to enumerate all the keys, and test them all. What prevents this in practice is that it would take too long. How long depends on the key size.

In the 1970s, the assumption was that you could test at most 1 million keys per second. In that case it would take you more than 2000 years to crack a DES key.

DES keys are too short for today's standards. In 2012, a system with 48 Xilinx Virtex-6 LX240T FPGAs was announced, each FPGA containing 40 fully pipelined DES cores running at 400 MHz, able to test 8 x 10^{11} keys/sec. The system can exhaustively search the entire 56-bit DES key space in about 28 hours.

DES, “2DES” and 3DES

DES a good design, but as it only has 56 bit keys, it has only approximately 2^{56} security. (There are some cryptanalytic attacks on DES, but not very serious ones, so let’s say its security is about 2^{56}.)

How about using DES twice? Take a 112-bit key, split it into two keys K_1 and K_2 and encrypt M like this:

$$\mathsf{Enc}_{K_1}(\mathsf{Enc}_{K_2}(M))$$

Would that give us 2^{112} security?

“2DES” is not significantly more secure than DES

Suppose we have a pair (M, C) consisting of a valid plaintext-ciphertext pair. With approximately 2^{57} work, we can find the 112-bit key $K_1 K_2$ used in 2DES. Here is how to do it.

- Try all 2^{56} possible keys K_2, and store all the results $\mathrm{Enc}_{K_2}(M)$. Sort them in order. This is 2^{56} work for the encryption, and $2^{56} \log(2^{56})$ for the sorting.
- Try all the 2^{56} possible keys K_1, computing $\mathrm{Dec}_{K_1}(C)$. For each such value, check if it is one of the stored $\mathrm{Enc}_{K_2}(M)$. That is 2^{56} work for the Dec, and $\log(2^{56})$ work for the checking.

The total work is not much more than 2^{57}.

3DES is good, but slow

3DES takes the same idea, but uses DES three times. That gives us a 168-bit key. Take the 168-bit key, split it into three keys K_1, K_2 and K_3, and encrypt M like this:

$$\mathrm{Enc}_{K_1}(\mathrm{Dec}_{K_2}(\mathrm{Enc}_{K_3}(M)))$$

3DES is good, but slow

3DES takes the same idea, but uses DES three times. That gives us a 168-bit key. Take the 168-bit key, split it into three keys K_1, K_2 and K_3, and encrypt M like this:

$$\mathrm{Enc}_{K_1}(\mathrm{Dec}_{K_2}(\mathrm{Enc}_{K_3}(M)))$$

- Why Enc-Dec-Enc instead of Enc-Enc-Enc?
  Enc-Dec-Enc gives us an option of setting $K_1 = K_2 = K_3$, which is then equivalent to DES. So if you have 3DES, you can make it do DES. This could be useful in some circumstances.
- How much security does 3DES give us? It doesn't give us 2^{168} of security, because the same meet-in-the-middle attack as we had for "2DES" is possible. It is said to give us 2^{118} of security.

AES replaces DES

DES considered insecure; 3DES considered too slow.

A NIST competition for the DES successor was held in 1997
15 submissions 1998; 5 finalists 1999

Rijndael was winner, named after its two inventors, two Belgian cryptographers, Vincent Rijmen and Joan Daemen.
Rijndael was adopted as the recommended successor to DES in 2000, and is now called AES.

AES has 128 bit keys. That is vastly more keys than DES. Even if you could build a system capable of testing 8 x 10^{11} keys/sec, it would take 25,000 years to test them all.