Chapter 5

DES Isn't Strong Anymore

DES is one of two innovations that brought cryptography into the computer age.

In Chapter 1 we point out that during 1977 there were two innovations in computer cryptography that forever changed 3,500 years of message disguise engineering. The introduction of the Data Encryption Standard was one of these changes; the invention of public key cryptography was the other.

DES uses 16 rounds of confusion and diffusion (iterated product cipher) on each group of about eight plaintext letters. Statistical analysis of letter frequency—the mainstay of cryptanalysts for thousands of years—is no help in attacking a well-designed method such as DES. The DES method is so secure that the cryptanalyst has no choice except to attack the keys.[1]

Review: With a strong method, the best attack is to try each key, but a strong method has too many keys to try.

So why is DES no longer strong if the best attack is to try all possible keys and there are an enormous number of potential DES keys? Simply stated, technology has made brute force attacks much faster.

On average, without knowing the right key, the computer cryptanalyst must cycle through half of all possible keys before stumbling on the correct one. And because there are trillions and trillions of possible keys, modern cryptanalysis requires super computer power—so much computer power that, until recently, computer cryptanalysis has been prohibitively expensive for all but the most well-endowed governments. With increasing amounts of computing power in increasing numbers of hands, searching through all those keys is no longer the chore it used to be for many people.

The Historical Need for an Encryption Standard

The new era in cryptography began in the late 1960s, when companies began needing secure ways to send files. Although there were many security

1. DES is susceptible to advanced cryptographic attacks, such as differential and linear cryptanalysis. Compared with trying all the keys, these attacks can be successful with less effort, but they are still impractical. These attacks are complex and beyond the scope of this book.

companies that claimed to have cryptographic computer programs, there was no industry standard—no widely accepted encryption method that had a universal seal of approval.

Financial institutions in particular wanted a standard encryption method that they could have confidence in and could use for secure data exchange. They needed something that was tested by someone they could trust. Whom to trust? How about someone whose reputation was built on keeping secrets?

Because U.S. national security depended on secure cryptography, government agencies involved in cryptography, such as the National Security Agency (NSA), had not been open with their expertise. Secrets have always been the mainstay of the NSA. According to one quip, the NSA was so secret in its beginning that its acronym stood for No Such Agency. Its budget is still secret. But the needs of private enterprise were beginning to pressure computer cryptography to come out in the open.[2]

NIST chooses DES, a cipher developed by IBM.

Finally, in 1972 the National Institute of Standards and Technology (NIST), then known as the National Bureau of Standards, decided to assist in the development of a secure cryptographic method (algorithm). In 1974 it settled on DES, a cryptographic method submitted by IBM. It is also known as the Data Encryption Algorithm (DEA).

Much of the underlying DES cipher research is credited to an IBM researcher, Horst Feistel, who in 1967 began experimenting with a combination of substitution and transposition ciphers. Before computers, transposition was difficult to mechanize and too complex to do by hand. Increases in the size of computer memory made it possible for Feistel to use transposition in his system. Feistel blocks are still used in many (if not most) secret key ciphers.

NSA assists in development.

DES did not evolve into the standard without significant discussion and even suspicion by some participants. NIST requested and received assistance from NSA on DES development. Some people have said that NSA modified DES so that the agency could easily reclaim plaintext without using a brute force attack, installing what is commonly known as a *trapdoor* function. NSA shortened the secret key originally proposed by IBM to 56 bits from 128 bits.

Some people believe that NSA altered IBM's algorithm to ensure that IBM had not secretly included its own trapdoor function. Some people also believe

What's in a Name?

Feistel wanted to call his system Dataseal. IBM shortened the term *demonstration cipher* to *Demon*. Later, Demon morphed to Lucifer, which phonetically contains the word *cipher*. In the end, the name evolved to DES.

2. It's interesting to note that the Internet is driving the development of today's cryptographic tools.

that NSA wanted to control the use of DES and expected DES to be implemented only in hardware, which can be controlled more easily than a software implementation. A hardware-only implementation of DES would have been harder to copy, and thus the method would have been less easily scrutinized. Because software implementations of DES became widely available, people have been able to study a method that NSA claimed was secure.

NSA didn't publicly respond to the criticisms about DES during the 1970s, but a heated debate ensued nevertheless. For example, people complained about the reduction of the key size to 56 bits; but because the DES method was intended to last fewer than 10 years, NSA could have easily justified its decision in that a 56-bit key was considered more than secure for that time period. There were also complaints that NSA changed some of the method's inner workings, perhaps to prevent cryptographic attacks NSA knew about but did not want to disclose.

Despite the controversy, DES was adopted as the federal standard for unclassified documents in 1977 and is the most widely used cryptographic method in history.

Cryptographic methods implemented in software

NIST made the DES design public, and the advent of computer chips permitted faster processing, making software implementation of DES feasible. Because any program that can be implemented in hardware can be implemented in software, DES software implementations began to appear. Disclosing cryptographic design criteria can give adversaries ideas they wouldn't otherwise have thought of, but the best way to ensure a method's security is to publish it and let it withstand attack over time.

Like a certain battery, DES goes on and on and on and . . .

The 1977 DES standard mandated a review every five years. In 1983, DES was approved for five more years. In 1987, the expected end of DES's life span, DES was approved for another five years, with the provision that it would expire before 1992. Then in 1993, DES was again approved for yet another five years. In 1997, NIST solicited candidates for a new secret key encryption standard, Advanced Encryption Standard (AES).

NIST announced the candidates for AES, successor to DES, in 1999, and in October 2000, NIST selected Rijndael. See Epilogue and www.nist.gov/AES.

Machines That Cipher

Computers make it easier to mechanize a secret key method that combines substitution and transposition. Because of a basic difference between substitution and transposition, patent offices prior to the computer age were swamped with applications for mechanical cipher disks based primarily on substitution. That's because transposition requires that the plaintext letters be jumbled in a certain pattern, and the

(Continued)

exact process can depend on the length of the message. Mechanizing this process isn't easy. With substitution, the message length doesn't matter, and it's much easier to mechanize the typical one-to-one correspondence of plaintext to ciphertext.

Cycling Through Computer Keys

DES has withstood all practical attacks against its method.

Although DES has probably been subjected to more cryptanalysis than any other encryption method in history, no useful practical holes have yet been found. As we mentioned, the best attack on DES is to try each possible key.

In 1977, a 56-bit key was considered a good defense. A cryptanalyst without the key trying all the combinations of 56 1's and 0's (more than a quadrillion possible keys) at one million keys per second would have to work for more than 1,000 years to try all of them.

But even though the number of possible DES keys confounds most humans, it's just a bunch of memory cells (transistors) to a computer. And the reason DES is no longer strong enough has a lot to do with those memory cells.

Only 30 years ago, human technology put a man on the moon, at the time the pinnacle of technology. Still, the computer that was used had less than one megabyte of memory (at a time when even most of the tech-savvy world didn't know what computer memory was). The NASA computer filled a small room and was tended to by people in white coats with engineering degrees. As recently as 10 years ago, most tech-savvy people didn't know what a megabyte was. Now *megabyte* is in most computer users' vocabularies, and most home computers have at least 64 MB of memory.

Moore's Law

More remarkable than the ability of DES to withstand statistical attack was an observation made by Gordon Moore around 1970. Moore, who co-founded Intel, said that the number of transistors on a silicon chip would double every 18 months without increasing the price of the chip. Widely known as Moore's Law, this dictum is often interpreted as doubling a computer's power every 18 months for the same cost.

Moore's Law makes DES obsolete.

What does all this have to do with DES? A 1975 computer could try about half of all possible DES keys in about 100,000 days (300 years). That's a respectably secure secret. But during the past quarter-century, Moore's Law has turned out to be correct: Computers have become about 100,000 times more powerful, while their cost has remained the same. So a computer made in 2000 can do in one day what a 1975 computer would have been able to do in 100,000 days for the same price. This means that DES is no longer the bastion of security it was created to be. Although it's difficult to estimate, those with cryptographic expertise claim that a $10,000,000 machine could find the DES key by brute force in a few minutes or even less!

DES Crackers

RSA Data Security Inc., named after the inventors of the RSA public key encryption algorithm (see Chapter 12), issued its first "DES Challenge" in January 1997. The first-prize winner, Rocke Verser, cracked DES by recovering the secret key in 96 days. Less than one year later, in February 1998, a team from Distributed.net cracked DES in 41 days.

Four months later, in July 1998, a team from the Electronic Frontier Foundation (EFF) and Distributed.net, using a machine valued at less than $250,000, cracked DES in 56 hours. Half a year later, in January 1999, the same team accomplished the feat in less than 24 hours.

The pattern is clear. If you need a strong cryptographic method, DES doesn't (or soon won't) do it anymore.

Probably no one is going to spend the time and effort to crack your 2001 holiday gift list. But if your livelihood depends on e-commerce or you make your living negotiating million-dollar deals, DES may not be strong enough. In January 1999, NIST reiterated a statement (in FIPS 46-3) it made in 1988: "Organizations needing security beyond that provided by the DES could use Triple DES."

Double and Triple DES

Until a new encryption standard is agreed upon, the de facto interim solution is Double or Triple DES. Double DES refers to the use of two DES encryptions with two separate keys, effectively doubling the DES key from 56 bits to 112 bits. This dramatic increase in key size much more than doubles the strength of the cipher. Each increase of a single bit doubles the number of keys. This means that a 57-bit key is twice as big as a 56-bit key; a 58-bit key is four times as big as a 56-bit key, and so on. It seems that this should stop an ambitious high schooler, or nearly anyone else, from completing a successful brute force attack. But wait.

Double DES: Probably not what you want

There's an attack on Double DES that reduces its effective number of keys to about double the number in DES (in math lingo, $2^{57} = 2 \times 2^{56}$). Known as the *meet-in-the-middle* attack, this approach was discovered by Whitfield Diffie and Martin Hellman, the same folks who invented public key cryptography. So although Double DES is an improvement over DES, Triple DES is better.

With Triple DES, you use three DES encryptions with three separate keys. Managing three keys is more difficult, so Walter Tuchman, another member of IBM's DES development team, proposed the current popular Triple DES implementation using just two keys.

DES (and Other Block Cipher) Modes

DES is a *block* cipher.[3] A block cipher takes a certain number of letters (actually bits[4]) and encrypts them all at once. DES encrypts 64 bits (about eight letters) at a time. Each chunk of 64 bits is called a block.

DES (and other block ciphers) also employs a *mode* method when a message is longer than one block. When DES is used in its simplest mode, called electronic codebook (ECB), it gives an adversary too many patterns. Why? It's because each block (chunk) of equivalent plaintext characters always encrypts to the same value. For example, suppose a message contains many blocks of the plaintext phrase `Your CIO`. Using DES (or any block cipher) in ECB mode will always encrypt the block `Your CIO` to the same ciphertext value, and that's often too many clues to give to a cryptanalyst. A short message without repeating plaintext can use ECB. But longer messages that have repeating phrases need to ensure against plaintext-ciphertext repetition.

The three other common modes ensure that repeated phrases are enciphered differently each time they are encrypted. The modes are cipher block chaining (CBC), cipher feedback (CFB), and output feedback (OFB). CBC is probably used most often, followed by CFB. OFB is a more complex variant of CFB.

The Avalanche Effect

DES and other secret key block ciphers also add something called the *avalanche effect*. Essentially, this means that each small change in plaintext implies that about half the ciphertext changes.

For example, suppose that two plaintext messages (message 1 and message 2) are almost identical; they differ by only a single bit (a single binary 0 or 1). DES (and other block ciphers) promise that although the underlying plaintext is almost identical, the ciphertext is very different.

Plaintext message 1:	100000000001 000101010001
Plaintext message 2:	100001000001 000101010001
Ciphertext of message 1:	100100001111 010111000001
Ciphertext of message 2:	101001111001 011111011000

3. As opposed to *stream* ciphers, such as Caesar's cipher or Vigenére's cipher, which encrypt a single character or bit at a time. Much more network encryption is done with block ciphers.
4. For a review of binary numbers, see the section "Supplement: Binary Numbers and Computer Letters" at end of this chapter.

DES illustrates that with every new cryptographic system comes new problems to be solved and that the problems are moving targets over time. Today's cryptography cloaks the electronically interconnected fishbowl in which we now live. Yesterday's cryptography is a good way to see how we got here. In Chapter 6 we'll take a look back with an overview of cryptography's evolution before we rush head-on into the field's current twists.

Supplement: Binary Numbers and Computer Letters

If you're already comfortable with binary numbers, you can skip this section. Otherwise, here's a brief explanation.

In a nutshell, letters (`a, b, c, . . . z`) are represented in the computer as binary numbers. At their core, computers work only with 0's and 1's. Computer memory is analogous to billions of tiny light bulbs; some are off, and others are on. The electronic circuit is either open or closed.

Of course, this isn't new; Morse code does almost the same thing. A telegram specialist transforms each plaintext letter to dots and dashes. Think of the dot as 0, or off, and the dash as 1, or on. For example, Morse code represents `B` as dash, dot, dot, dot (1,0,0,0) and `C` as dash, dot, dash, dot (1,0,1,0). At the destination, each group of dashes and dots is converted back to plaintext.

Morse code uses as many as four dashes and dots to represent each letter. Most computers use American Standard Code for Information Interchange (ASCII) code characters (see Figure 5-1). ASCII represents each letter with some combination of eight 0's and 1's. Each 0 or 1 is called a *bit*. Eight bits is called a *byte*.

Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
0	0	0	1	0	1	0	1
Byte = 1 letter (character)							

Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
0	1	0	1	0	0	1	0
Byte = 1 letter (character)							

Figure 5-1 Two letter characters represented as binary data.

Review

Openly publishing a cryptographic method is a good way to ensure its security. DES, the published cryptographic standard since 1977, has withstood attack over the years. The DES algorithm was strong, so cryptanalysts had no choice except to attack the keys. This means trying, on average, half of all possible keys—some number of trillion keys.

However, advances in computer hardware have compromised the strength and security of DES. That's because it's easier to search through all those keys now than in 1977.

The candidates for AES, DES's successor, were announced by NIST in 1999, and the new standard is scheduled for selection. If you believe your adversary has sophisticated knowledge of cryptography and if your secret is valuable, don't use DES. Until the new encryption standard is anointed, it's safer to use a DES variant called Triple DES.

Rijndael was selected to replace DES in October 2000. See our Epilogue and NIST's site www.nist.gov/AES for additional information.