Model Eliciting Activity: Symmetric Key Cryptography

Individual Reading

Modern Symmetric key cryptographic protocols evolved from the classical ciphers discussed in the “Simple Ciphers” module. As cryptanalysis techniques have evolved from simple substitution and transposition to more complex mathematical relationships, and processing power has evolved from manual processing to automated computer processing, more complex symmetric ciphers became necessary. These ciphers used both **more mathematically complex algorithms** and **longer keys**. This evolution is illustrated by the evolution of DES as an encryption standard and its eventual replacement. The legal fight between the FBI and Apple over iPhone encryption is an example of how work in symmetric key cryptography since the 1970s has improved the strength of this type of cryptography.

The Evolution of DES

The origins of DES go back to the early 1970s. In 1972, after concluding a study on the US government's computer security needs, the US standards body NBS (National Bureau of Standards)—now named NIST (National Institute of Standards and Technology)—identified a need for a government-wide standard for encrypting unclassified, sensitive information.[1] Accordingly, on 15 May 1973, after consulting with the NSA, NBS solicited proposals for a cipher that would meet rigorous design criteria. None of the submissions, however, turned out to be suitable. A second request was issued on 27 August 1974. This time, IBM submitted a candidate which was deemed acceptable—a cipher developed during the period 1973–1974 based on an earlier algorithm, Horst Feistel's Lucifer cipher. The team at IBM involved in cipher design and analysis included Feistel, Walter Tuchman, Don Coppersmith, Alan Konheim, Carl Meyer, Mike Matyas, Roy Adler, Edna Grossman, Bill Notz, Lynn Smith, and Bryant Tuckerman.

On 17 March 1975, the proposed DES was published in the *Federal Register*. Public comments were requested, and in the following year two open workshops were held to discuss the proposed standard. There was some criticism from various parties, including from public-key cryptography pioneers Martin Hellman and Whitfield Diffie,[2] citing a shortened key length and the mysterious "S-boxes" as evidence of improper interference from the NSA. The suspicion was that the algorithm had been covertly weakened by the intelligence agency so that they—but no-one else—could easily read encrypted messages.[3] Alan Konheim (one of the designers of DES) commented, "We sent the S-boxes off to Washington. They came back and were all different."[4] The United States Senate Select Committee on Intelligencereviewed the NSA's actions to determine whether there had been any improper involvement. In the unclassified summary of their findings, published in 1978, the Committee wrote:

In the development of DES, NSA convinced IBM that a reduced key size was sufficient; indirectly assisted in the development of the S-box structures; and certified that the final DES algorithm was, to the best of their knowledge, free from any statistical or mathematical weakness.[5]

However, it also found that

NSA did not tamper with the design of the algorithm in any way. IBM invented and designed the algorithm, made all pertinent decisions regarding it, and concurred that the agreed upon key size was more than adequate for all commercial applications for which the DES was intended.[6]

Another member of the DES team, Walter Tuchman, stated "We developed the DES algorithm entirely within IBM using IBMers. The NSA did not dictate a single wire!"[7] In contrast, a declassified NSA book on cryptologic history states:

In 1973 NBS solicited private industry for a data encryption standard (DES). The first offerings were disappointing, so NSA began working on its own algorithm. Then Howard Rosenblum, deputy director for research and engineering, discovered that Walter Tuchman of IBM was working on a modification to Lucifer for general use. NSA gave Tuchman a clearance and brought him in to work jointly with the Agency on his Lucifer modification."[8]

NSA worked closely with IBM to strengthen the algorithm against all except brute force attacks and to strengthen substitution tables, called S-boxes. Conversely, NSA tried to convince IBM to reduce the length of the key from 64 to 48 bits. Ultimately they compromised on a 56-bit key.[9][10]

Some of the suspicions about hidden weaknesses in the S-boxes were allayed in 1990, with the independent discovery and open publication by Eli Biham and Adi Shamir of differential cryptanalysis, a general method for breaking block ciphers. The S-boxes of DES were much more resistant to the attack than if they had been chosen at random, strongly suggesting that IBM knew about the technique in the 1970s. This was indeed the case; in 1994, Don Coppersmith published some of the original design criteria for the S-boxes.[11] According to Steven Levy, IBM Watson researchers discovered differential cryptanalytic attacks in 1974 and were asked by the NSA to keep the technique secret.[12] Coppersmith explains IBM's secrecy decision by saying, "that was because [differential cryptanalysis] can be a very powerful tool, used against many schemes, and there was concern that such information in the public domain could adversely affect national security." Levy quotes Walter Tuchman: "[t]hey asked us to stamp all our documents confidential... We actually put a number on each one and locked them up in safes, because they were considered U.S. government classified. They said do it. So I did it".[12] Bruce Schneier observed that "It took the academic community two decades to figure out that the NSA 'tweaks' actually improved the security of DES.

Despite the criticisms, DES was approved as a federal standard in November 1976, and published on 15 January 1977 as FIPS PUB 46, authorized for use on all unclassified data. It was subsequently reaffirmed as the standard in 1983, 1988 (revised as FIPS-46-1), 1993 (FIPS-46-2), and again in 1999 (FIPS-46-3), the latter prescribing "Triple DES" (see below). On 26 May 2002, DES was finally superseded by the Advanced Encryption Standard (AES), following a public competition. On 19 May 2005, FIPS 46-3 was officially withdrawn, but NIST has approved Triple DES through the year 2030 for sensitive government information.[14]

The algorithm is also specified in ANSI X3.92 (Now, X3 is now known as INCITS and ANSI X3.92 as ANSI INCITS 92),[15] NIST SP 800-67[14] and ISO/IEC 18033-3[16] (as a component of TDEA).

Another theoretical attack, linear cryptanalysis, was published in 1994, but it was the Electronic Frontier Foundation's DES cracker in 1998 that demonstrated that DES could be attacked very practically, and highlighted the need for a replacement algorithm. These and other methods of cryptanalysis are discussed in more detail later in this article.

The introduction of DES is considered to have been a catalyst for the academic study of cryptography, particularly of methods to crack block ciphers.

Group Activity 1: [Work as a group but report the findings individually]

Brute force attack on DES (**KeySearcher_DES.cwm** on Blackboard)

<u>Step 1.</u> Assuming part of the keys are leaked and a hacker is trying to decrypt the key. Investigate how long does it take for the hackers to crack the key if they know the first few bits of the HEX of the key are all 1s. Put it differently, the settings of the key searcher is:

11-11-11-11-1*--**-****

Record the time it takes to complete searching the key. Would the number of CPU cores involved in the calculation show any differences? Prove your answer.

<u>Step 2:</u> Assuming the hacker has a knowledge of four more bits, how much faster can she/he crack the key? Would the number of CPU cores allocated change the results? Prove your answer. [key is: **11-11-11-11-11-**-**-****]

<u>Step 3:</u> What if the hacker has a knowledge of another four more bits, how much faster can she/he crack the key? Would the number of CPU cores allocated change the results? Prove your answer. [key is: **11-11-11-11-11-1*-**-****]

Step 4: Base on what you observed, predict the fastest time that your computer can crack an unknown DES key using the brute force approach (cipher text only).

Instructor-led introduction on AES.

Step 5: Group discussion: What made DES a strong encryption algorithm then and what made it a weak encryption now? What does it imply for the dominant AES encryption?

Instructor led discussion on Block Mode

Group Activity 2: [Work as a group but report the findings individually]

Can you ‘re-use’ the key for a different block? (**Modes.cwm** on Blackboard)

Step 1: Complete the diagram so that the picture can be encrypted using ECB mode

Step 2: Ran the diagram a few times. Notice that the Key is random. Were the scheme successfully hide the images? What went wrong?

Step3: Given that CBC mode can be illustrated as: $C_{-1} = IV, C_i = E_k(C_{i-1} \oplus m_i),$ encrypt the image using the CBC mode. Run the program a few times and compare the results with Step2.

Step 4: Group discussion: What mode should you adopt? Why? And what’s the disadvantages of your choice of block mode?

Group Activity 3: **[Work as a group but report the findings individually]**

Can you use space in exchange of time? (**2DES_meet_in_the_middle.cwm** on Blackboard)

Step 1: Each DES encryption has a key size of 56 bits. If decrypting a single DES uses only 1 minute, how long does it take, in theory, to decrypt double DES?

Step 2: Assuming there is a sufficient hard drive space to store information, the following operations can be used to attack 2DES:

1. Search key k_1 and store $E(k_i, M)$ pair in table 1(***not*** described in the file)
2. Search key k_2 and store $D(k_j\ , C)$ pair in table 2(***not*** described in the file)
3. Compare table1 and table2 to find (k_1, k_2)

Group discussion:

Suppose all the bits for key1 and key2 are unknown for 2DES, what's the consumption time for meet in the middle attack?

Compare to single DES, how do you evaluate the safety level of 2DES?

Individual Report

Scenario:

Your country has just developed and demonstrated its ability to use a nuclear weapon. A rivalrous nation is close to testing a similarly destructive weapon and plans to deploy it to their forces worldwide. Your leadership also wants to deploy these weapons across your country and to its forces stationed worldwide in order to balance this threat. Both countries routinely monitor each other's communications; so your leaders need to be able to communicate commands to those forces without its rivals being able to know the content of those messages. If the content of those messages are known to your country's rivals, they would be able to neutralize your country's forces before they could act. This could also happen if your country's leaders can't communicate quickly enough with its forces as threats emerge. Your country's leaders recognize your cryptographic brilliance and have asked you to design a system that allows it to communicate efficiently and secretly with its 50 force commanders. Assume that the encrypted traffic can only flow from the command node to the receiver nodes, not in the reverse.

1. Describe how you would implement a system that meets the specified requirements. Discuss in detail how you would manage your cryptographic keys and how you would get messages to the 50 commanders.

2. Draw your command node and ONE of the 50 receiver nodes. Illustrate your transmission method and write an equation beneath both nodes that describes how the messages will be enciphered and deciphered.