Levi Seibert

00086809

lseiber1@my.athens.edu

ITE 523 – Dr. Yahia Fadlalla

Data Encryption Standard

The Data Encryption Standard, commonly referred to as DES, was designed by IBM researchers in the early 1970s and was formalized by the National Institute of Technology in 1977 (with some debatable "assistance" from the National Security Agency) [1]. DES was the first encryption standard in the United States; until that time, encryption was solely a military and academic concept, or at least was advertised that way. It is speculated that the NSA didn't necessarily care for the idea of citizens have data that they didn't have access to. Regardless of its controversy, DES was revolutionary and was "the" standard of encryption for around two decades.

Although it was once the golden standard of encryption, DES is no longer considered secure; several vulnerabilities have been found that make it dangerous for use, specifically attacks involving brute force. In response to this issue, 3DES (or triple-DES) was designed. 3DES uses three iterations of the DES algorithm in order to increase the security. In the most secure version of 3DES, a separate key is used for each of the three iterations, enlarging the key length to 168 bits (56 * 3). Unfortunately, this modification still does not add enough security to make the DES methodology effective and safe. In fact, 3DES will formerly be disallowed for service later this year by NIST, meaning it cannot be used for cryptographical protection (DES has been disallowed for several years already). The biggest issue with 3DES is the potential for brute force attacks, which was the original issue that it tried to remediate. In addition, because of the possibility of meet-in-the-middle attacks, the advertised 168-bit key length is really only effective up to 112 bits [1].

The Data Encryption Standard algorithm can be considered at multiple levels of abstraction. The first thing to know is that DES is a block cipher. A block cipher is a type of cryptographical cipher that encrypts a plaintext by breaking it into certain size blocks (set by the algorithm at hand) and encrypting each block at a time, rather than bit by bit as is done with stream ciphers. In DES, the plaintext is split into 64-bit blocks and the key is 56 bits long (technically, it is 64 bits long, but the first 8 bits are used for parity checking, and thus are not part of the effective key) [1]. At the highest level of abstraction, DES consists of a standard encryption mechanism: a plaintext block and a key are inputs into an encryption "box" (or function) and the ciphertext is the output. When one looks a bit deeper into the algorithm and opens up the box, they would find that the inside of this encryption box is a feistel network. A feistel network is a method of encryption cipher that breaks up a plaintext block into two equal portions and then iteratively performs several rounds of encryption [2].

For DES, the fiestel network has a unique design that is repeated in 16 rounds, each round identical in form to the previous. Before the first round, however, an initial permutation is performed on the 64-bit block (note that a final permutation also occurs at the ending). The reasoning behind this initial permutation is not clear; however, it is speculated that was put into

the specification as solely a hardware design feature, since it does not add any security to the encryption. For each of the 16 rounds, the 64-bit block is split into two 32-bit blocks, called L (for the left-hand side or first 32 bits) and R (for the right-hand side or last 32 bits). R is technically not encrypted during each round, but it does serve as an input for the encryption that takes place on L. At the end of each round, the previous R becomes the new L (so it will be encrypted in the following round).

L, however, does get encrypted. First, there is an "f" function that is calculated by takings R and part of the key as input and outputs a 32-bit result. This result is then XORed with L and L becomes the next round's R value. Before looking into the "f" function, it is important to note that the algorithm does manipulate the key in order to get a workable size, as well as to further obscure its identity. As was previously mentioned, the key for DES is 64 bits long, with 8 of those bits used for parity checking. For each of the 16 rounds, a 48-bit subkey is produced from the 56 remaining bits. So, the complete key is not used in any round, and each round's subkey is different (but related to the original key, based on a series of selections and permutations as defined by the DES standard) [1].

Inside each "f" function, four different operations take place. First, the 32 bits from R are enlarged to become 48 bits (a permutation also takes place, so bits in certain positions in R are then put into new locations, and half of them are copied into two positions in the new bit stream). Next, these 48 bits are XORed with the corresponding subkey. Third, the resulting value is split into 8 groups of 6 bits each. Each of these groups of 6 bits are fed into a substitution table, which outputs a series of 4 bits, reducing the number of bits from 48 (6 * 8) to 32 (4 * 8). Finally, this 32-bit sequence goes through another permutation, which rearranges the bits. The result of this permutation is the output of the "f" function and is XORed with the L value to encrypt it and produce the next round's R value.

The substitution boxes and permutation orders are well-known; anybody can look them up online. There is debate as to why IBM chose the tables they did, as the ordering seems very random and unorganized; however, regardless of their intentions, it has been proven that the tables are very strong and served their purpose of protecting DES for several years. As long as a programmer has access to these substitution boxes and permutation tables, writing code to implement DES is not difficult. To view the specifications for the substitution boxes and permutation tables (including initial and final permutations, expansions, and subkey generations), see [3].

The beauty of the design of the feistel network for DES is that it is reversible. All operations work in reverse. Just as the algorithm begins with a permutation, it also ends with a permutation (this is just the inverse of the initial permutation table), so the two offset each other. Each round of the fiestel network relies on input from the previous round, but the way DES was designed, this input works regardless of the direction of cryptography that is taking place. This is one of the benefits of the XOR operation, all calculations are reversible (for example, an input value XORed with a key value returns an output value; this output value XORed with the same key value returns the original input value). Thus, once one understands the encryption process, the decryption process becomes trivial (assuming they have access to the original key).

Ultimately, DES failed because its key is too short. 56 bits is not nearly enough to protect ciphertext in today's world of cryptanalysts who delight in the easy pickings of DES-encrypted data. With 56 bits, there are about 72 quadrillion (2^{56}) possibilities for the key. To the non-

cryptographer, this number may seem plenty big. Unfortunately, however, it is not nearly large enough. A key of this size can be brute forced in a matter of several hours. Even the 3DES algorithm, with a maximum of 2^{168}, is too short for security in today's world. Yes, it is exponentially more difficult to brute force than DES, but in reality, if any vulnerability is known, it can't really be claimed to be secure.

By the late 1990s, researchers had determined that the 56-bit key was too small and in 1997, NIST began a search to find a more secure successor to DES. In 2001, they selected the Rijndael cipher and modified it to produce the AES standard, which is one of the current accepted encryption standards still in use today. One of the key features of AES is that it allows for varying key sizes, either 128, 192, or 256 bits long (all of which are exponentially more secure than DES). The algorithm is also significantly more complex than DES.

DES did its job well for two decades and, since then, has proven beneficial in the academic realm as it is a fantastic example cipher to help teach cryptography. DES is simple enough that most undergraduate computer science students can grasp its basics, while complex enough that it actually represents a real encryption mechanism that was standardized and used for several years.

DES and 3DES can still be used for personal cryptography, like encrypting files on a home computer; however, the user just needs to be aware that these algorithms are not as secure as once thought and should be used with caution. For personal use, complete security is not often a necessity, so using DES and its counterpart would not likely cause any issues. Since the algorithms are (or will be) disallowed, there is always the potential that someone could be viewing things that were intended to be kept private.

Ever since DES was formalized, there has been speculation that the government was behind the algorithm and had their own private "backdoor" into the mechanism so they could decrypt messages as they desired without a key, thus the reason why the substitution boxes are specified the way they are. They point to the NSA's involvement in the formalization of DES as suspicious and a cause for concern. It is unknown whether or not this has any validity, but there are scholars who hold this as truth. In fact, some believe other encryption algorithms, like Elliptic Curve Cryptography, are also subject to government interference [4].

Note: Most of the above content comes from [5].

References

[1] P. Loshin and M. Cobb, "What is Data Encryption Standard?," *Security*, 20-Aug-2021. [Online]. Available: https://www.techtarget.com/searchsecurity/definition/Data-Encryption-Standard#:~:text=The%20Data%20Encryption%20Standard%20is,it%20into%2064%2Dbit%20blocks. [Accessed: Feb. 28, 2023].

[2] "What is a Feistel Network? - Definition from Techopedia," *Techopedia.com*, 2019. https://www.techopedia.com/definition/27121/feistel-network. [Accessed: Feb. 28, 2023].

[3] "DES supplementary material," *Wikipedia*, Jan. 09, 2023. https://en.wikipedia.org/wiki/DES_supplementary_material. [Accessed: Feb. 28, 2023].

[4] T. C. Hales, “The NSA Back Door to NIST,” *Notices of the American Mathematical Society*, vol. 61, no. 02, p. 1, Feb. 2014, doi: https://doi.org/10.1090/noti1078. [Accessed: Feb. 28, 2023].

[5] Introduction to Cryptography by Christof Paar, “Lecture 5: Data Encryption Standard (DES): Encryption by Christof Paar, *Youtube*, Jan. 30, 2014. [Online]. Available: https://www.youtube.com/watch?v=kPBJIhpcZgE&t=3300s. [Accessed: Feb. 28, 2023].