BRUTE FORCE

CRACKING THE DATA ENCRYPTION STANDARD

Matt Curtin

INTERHACK CORPORATION

Copernicus Books

AN IMPRINT OF SPRINGER SCIENCE+BUSINESS MEDIA

Published in the United States by Copernicus Books,
an imprint of Springer Science+Business Media.

Copernicus Books
Springer
233 Spring Street
New York, NY 10013

www.springeronline.com

Library of Congress Cataloging-in-Publication Data
Curtin, Matt.
Brute force : Cracking the data encryption standard / Matt Curtin.
p. cm.
Includes bibliographical references and index.
ISBN 0-387-20109-2 (alk. paper)
1. Computer security. 2. Data encryption (Computer science) I.Title.
QA76.9.A25C873 2005
005.8′2—dc22
2004058919

Manufactured in the United States of America.
Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

ISBN 0-387-20109-2 SPIN 10958636

Contents

Foreword

A big battle over privacy was fought in the 1970s, 80s, and 90s, and most people didn't even know it was happening.

The U.S. government deliberately restricted the ways in which people could protect their own privacy. They did this with laws, with regulations, and by threatening prominent activists like Ron Rivest and Phil Zimmermann with censorship and prosecution. Most of it was unconstitutional, though they got away with it for decades. But most importantly, they restricted our privacy by keeping us ignorant and by lying to us.

A good way to keep information private is to safeguard it with encryption, a mathematical technology that scrambles information. You set it up so that the only people who have the "key" to unscramble it are the people that the owner intends to give access to. The government wanted to keep a monopoly on information about encryption. This would let the government hide information from its citizens (and from foreigners), but its own citizens (and foreigners) could not hide information from the government. The government had already threatened prominent academic researchers, tried to cut off National Science Foundation funding for research in encryption, and had built a "voluntary" censorship system for research papers.

It seemed to some people that freedom to do research, freedom to publish the results, and privacy were fundamental values of society that were more important than any particular government desires. The early academic researchers of cryptography, like David Chaum, Ron Rivest, and Whitfield Diffie, were such people. The Cypherpunks, who came along a few decades later, were also such people. I co-founded the Cypherpunks, an open group who educated ourselves and each other about encryption, and encouraged each other to write encryption software for free public use. Our goal was to re-establish the freedoms that the government had silently taken away, do the research, and publish the results, to transform society's expectations about privacy.

Part of the lies and ignorance created by the government was about a system called DES—the Data Encryption Standard. The government claimed that it was secure and private. Independent researchers claimed that it was too easy for governments to break into the privacy of DES. But mere claims were not enough to stop it, and the government succeeded in getting almost everyone to use DES worldwide. Banks used it to secure billions of dollars of money transfers. Satellite TV companies used it to keep their transmissions to their customers private. Computer security products used it. ATMs used it to guard the phone line that connects them to their bank and tells them when to deliver cash.

DES was deliberately designed by the U.S. government to be flawed. The government could read what was encrypted by DES, merely by spending enough money to build a machine that would break it. And the amount of money that it took went down every year, both as technology evolved, and as the designer learned more about how to build such machines. All that knowledge was hidden in the same secretive government agencies who deliberately weakened DES.

As personal computers and chip technology rapidly became cheaper and faster, ordinary people working together could rival the machine-building power of the government. This book is the story of how they proved the government was lying, twenty years after the lie, and by doing so, energized the public to take its privacy into its own hands. The end result was not only that government policy about encryption and privacy was changed. Also, the process of building networks of people and machines to do calculations by "brute force" taught us a lot about collaboration, about social structures in volunteer groups, about how the world is changed by the broad distribution of consumer products that compute. And about how to break down certain kinds of intractable problems into small pieces, such that many people can do a piece and thus contribute to the solution.

The panicky public reaction to the attack of 9/11 was unable to upset the balance of relatively sane encryption policy that it had taken decades to set right. However, the abdication of responsibility that took hold of both the Congress and the bulk of the public has let a corrupt administration get away with murder—literally, in the case of hundreds of thousands of civilians in Iraq. Civil rights and moral standards as basic as the prohibition on torture, the freedom to move around in your own country, and the universal condemnation of unprovoked attacks on other countries have all fallen by the wayside.

Yet computers and networks have shown even more interesting ways for millions of people to collaborate to solve big intractable problems like this. As I write this, thousands of people are working for a few days from their homes, phoning up strangers to encourage them to go out and vote in the upcoming U.S. election. A computer network, programmed by a small number of people, has collected and connected both the callers and the people who they should call.

We will continue to be surprised by the capabilities that human societies have, when thousands of people network through their computers to accomplish a common purpose.

John Gilmore
Electronic Frontier Foundation
October 31, 2004

Preface

In the past fifty years, society has undergone a radical shift in the storage and processing of information, away from the physical and toward electronic representation. Important information is no longer written on a sheet of paper and stored in a locked file cabinet or safe. Information necessary to care for our health, our finances, and the institutions, public and private, that support society is now stored electronically, in little ones and zeroes. Encryption technology—the mathematical system used to protect electronic information—was developed to protect all of those data from prying eyes.

In the late 1970s, the U.S. government decided to create a national data encryption standard in order to bring order to a market that had generated a multitude of competing and rarely complimentary encryption products. The standard the government settled on called the data encryption standard or DES was immediately criticized for being too weak by many security and computer experts. For years the critics demanded stronger cryptography and for years the government ignored their requests.

In 1997 a security company, RSA, answered DES's critics. They launched a contest, challenging cryptographers and computer enthusiasts to show the government just how weak DES was. *Brute Force* tells the story of DES: how it was established, challenged, and ultimately defeated. But more than the longevity of DES or the definition of the standard was at stake.

Even while technologists argued over how strong the cryptographic standard had to be, lawmakers in the United States were busy debating the government's role in the regulation of cryptography. At the heart of the debate was whether or not the government would permit American companies to export products that they couldn't break overseas, and whether private citizens would be permitted to use cryptography that would shield their information from the eyes of government. Libertarians, cryptographers, and security experts wanted to be able to use and export the most robust encryption possible. While some in Congress supported this view, many other members of the government, including the Clinton administration, were wary of strong encryption, fearing it would fall into the hands of criminals and terrorists. *Brute Force* tells the story of the legislative battle over DES as well.

Although cryptographic specialists will likely be familiar with parts of this story and be eager to learn what happened behind the scenes, this is not only a story for technologists. What happened in 1997 affects people everywhere, even today, and will do so for years to come. So long as we store and transmit private information on computers, we will need to protect it from those who would try to steal it.

Events of this story fall into one of three major topics: the technology of secret writing, the story of how people who never knew each other came together to defeat the global standard for secret writing, and the wrangling over public policy on cryptography. The story is told not by recounting events in a strictly chronological order but as chains of events that place different parts of the story into context and allow the reader to see how these events finally came crashing together, changing the face of information management forever.

Acknowledgments

This book is the product of tremendous work by many people. Thanks must go to Peter Trei for suggesting the demonstration of a brute force attack on the Data Encryption Standard and to RSA for sponsoring the contest that at long last demonstrated the weakness of DES. I also offer my heartfelt thanks to Rocke Verser for his work in starting and running the DESCHALL project that participated in RSA's contest. Justin Dolske, Karl Runge, and the rest of the DESCHALL developers also put in many hours to ensure our project's success and were as pleasant and interesting as one could hope for. Not to be forgotten are the thousands of people who participated by running the DESCHALL client programs on their computers, telling their friends about our project, and giving us access to the tremendous computational power needed to verify that strong cryptography makes the world a safer place. Telling the story of this significant period in the history of cryptography in the form of the book that you are now holding proved to become another sizable project. Gary Cornell at Apress got me connected with the right people at Copernicus Books. I appreciate the connection as well as the help that Anna Painter, Paul Farrell, and the rest of the folks at Copernicus Books provided in moving the book from a raw manuscript into its final, published form. Thanks are also due to John Gilmore for resurrecting a recording of Martin Hellman and Whitfield Diffie arguing with government representatives the need for a stronger standard than what became codified in DES. The recording and other electronic resources of interest are available at:

http://ergo-sum.us/brute-force/.

Finally I thank my wife Nicole for her continued support and thoughtful interest in my work.

Matt Curtin
December 2004

To the Cypherpunks—
making the networks safe for privacy…

1

Working Late

June 17, 1997, 11:51 P.M.
Salt Lake City, Utah

A modest desktop computer quietly hummed along. It sat in the offices of iNetZ Corporation, a Web services company started just a few months earlier. This machine, just an ordinary machine with a 90 MHz Intel Pentium processor, was still hard at work in the darkness of an office that had closed for the day several hours earlier. Running a program called DESCHALL—pronounced "DESS-chall" by some, and "dess-SHALL" by others—this computer was trying to read a secret message. After all, it was practically the middle of the night, and the machine had nothing else to do.

The secret message was protected by the U.S. government standard for data encryption, DES. Largely as a result of the government's fiat, DES was used to protect sensitive data stored on computers in banking, insurance, health care, and essentially every other industry in nearly every part of the world. It was a U.S. standard, but in a world of international corporations and global trade increasingly conducted by computer, it was in everyone's interest, or so it seemed, to standardize on DES.

The slowest of eight iNetZ machines on which system administrator Michael K. Sanders installed DESCHALL, the quiet little computer was trying to find the single key out of more than 72 quadrillion (72,000,000,000,000,000) that would unlock the secret message. Applying one key after another to the message and checking the output for something intelligible, the machine was trying some 250,000 keys per second. It did not falter. It did not quit. It just kept banging away at the problem.

Quite suddenly, just before midnight, the computer's DESCHALL program came to a halt.

When Sanders came to work at iNetZ the following morning, this unassuming computer was displaying an urgent message on its screen.

Information security would never be the same.

2 Keeping Secrets

Cryptography is quite simply the practice of secret writing. The word itself comes from two Greek words, *kryptos* ("hidden") and *graphein* ("writing"). With a history going back at least 4000 years, cryptography has long been surrounded by mystery and intrigue.

Ancient Egyptians used cryptography in hieroglyphic writing on some monuments, thus protecting some proper names and titles. Some 2000 years ago, Julius Caesar used a simple system of substituting one letter for another to send secret messages to his generals. In the thirteenth century, English mathematician Roger Bacon wrote of systems to write in secret in his "Concerning the Marvelous Power of Art and of Nature and Concerning the Nullity of Magic." In that document, Bacon enumerated seven methods for secret writing and famously opined, "A man who writes a secret is crazy unless he conceals it from the crowd and leaves it so that it can be understood only by effort of the studious and wise."

Throughout its history, cryptography has primarily been a tool of government elites because they were the ultimate keepers of military and diplomatic secrets. Code makers and breakers alike have thus almost always been employed by governments to discover others' secrets while protecting their own.

Cryptography is important because it enables information to be stored and transmitted secretly. The ability to control the flow of information, to enforce who may and may not know a particular fact is precisely the kind of power that traditionally governments and increasingly private businesses seek to wield against adversaries and competitors. Especially when the keepers of a secret are not able to meet together, out of the range of eavesdroppers and spies, there is a need for communicating secretly right in the open. As had been demonstrated in numerous wars of the twentieth century, anyone can intercept radio signals. Telephone lines can be tapped. This is where cryptography comes into play—locking up information so that it will remain secret while it is being transmitted via a medium that is open to all.

Once we had passed the age of the trusted courier and locked box, new telegraph and especially radio technologies created the need for reliable encryption machines. In the early twentieth century, enterprising inventors saw an opportunity and before 1920 had invested four such devices. At the heart of these machines was a series of three or four rotors—wired code wheels, each with twenty-six different electrical contacts on each side. To encrypt a message, the user would type a letter on the keyboard, such as A, and electrical current would flow through the machine, going through the rotors, and printing a completely different letter, such as V. The rightmost code wheel would then advance one position, and the user pressing A again would result in another letter being printed, such as T, before the code wheel rotated again. Once the rotor went through all twenty-six positions, the rotor next to it would also advance, much like an analog odometer on an automobile.

In this way, the user would type the original message, while the machine would produce *ciphertext* that could safely be sent as a radio signal. The intended recipient of the message would have a matching cipher machine that would turn the signal back into human-readable *plaintext.* In the United States, Edward H. Hebern invented his machine in 1917, Germany's Arthur Scherbius invented his in 1918, and 1919 saw the invention of a machine in the Netherlands by Alexander Koch and in Sweden by Arvid Gerhard Damm. Scherbius called his machine Enigma, and it would become the only financially successful cipher machine from the era.

Enigma was patented by Scherbius, an electrical engineer, and E. Richard Ritter, a certified engineer. After the eventual transfer of patent rights, Engima would come to be marketed commercially by *Chiffriermaschinen Aktien-Gesellschaft* (Cipher Machines Stock Corporation), whose board of directors included Scherbius and Ritter. Several governments began to investigate Engima, with variations of the original design eventually coming into use throughout the German, Italian, and Japanese armed forces.

Despite the best efforts of its producers, Engima was not generally accepted in the world of business. Its commercial success came as a result of the Axis use of the machine to protect military and diplomatic communications.[1]

With the rise of radio technology in government and military communications in the early twentieth century, the danger of messages being intercepted increased dramatically. Instead of having to get physical access to communications circuits such as telephone or telegraph lines, operatives could simply point high-powered antennas toward their targets and start listening. Governments throughout the world developed “signals intelligence” groups, chartered to intercept radio communications sent by other nations, and to report their findings to their own leaders. To protect their own communications from foreign signals intelligence efforts, governments began to encrypt their radio signals.

Governments would not easily give up the ability to read others’ messages. Signal intelligence came to mean not just message interception but also breaking the encryption used to protect the messages. In the years leading up to World War II, the United States maintained an active signal intelligence operation even while hoping to avoid being drawn into the global conflict. In 1938, the Japanese empire began to use a machine they called “Alphabetical Typewriter 97” for their diplomatic messages—a rotor machine like Germany’s Enigma. Unable to read those messages, the U.S. Army Signals Intelligence Service (SIS) began a project to break the Japanese system, which they had codenamed, “Purple.”

In the late 1930s, SIS cryptanalysts (code breakers) under the direction of cryptographic pioneer Frank Rowlett spent eighteen months studying intercepted Japanese diplomatic messages, looking for any clue that would help them to unlock Purple’s secrets. One day in September 1940, SIS cryptanalyst Genevieve Grotjan made a critical discovery. She found important and previously undiscovered correlations among different messages encrypted with Purple. After Grotjan brought her discovery to the attention of the rest of the SIS Purple team, they were able to build a duplicate of a machine they had never seen—the Alphabetic Typewriter 97.[2]

Putting its new machine to work right away, SIS discovered that Purple was used not simply for routine traffic, but the most sensitive of the Japanese empire’s secrets. Intelligence gathered from intercepted and decrypted Purple messages was so valuable that those decrypted intercepts came to be called “Magic” within SIS.

When Rowlett returned to his office from a meeting at midday on December 3, 1941, he picked up a Magic decrypt from his in-box. That message, intercepted just that morning, was directed to Japan's embassy in Washington. Rowlett read the bizarre orders for Japanese diplomats to destroy their code books and even one of the two Purple machines they had. Without their code books and with only one working Purple machine, the Japanese embassy simply would not be able to operate normally. Colonel Otis Stadtler, who was responsible for distributing Magic decrypts arrived as Rowlett was reading the message. After some discussion, Stadtler realized the meaning of the order: Japan was preparing to go to war with the United States.

On the evening of December 6, U.S. president Franklin D. Roosevelt received analysis of the intelligence: war with Japan was inevitable, and the Magic decrypts were used to support the conclusion. As the Japanese military used different codes from the Japanese diplomats, President Roosevelt had no way of knowing that on the very next day, Japan would attack Pearl Harbor and kill over 2300 Americans. Only five years later would there be enough time for SIS cryptanalysts to look at the military intercepts in the months before the strike on Pearl Harbor. Their efforts to break those messages proved successful, and they anguished over the results of their work. Though not naming Pearl Harbor explicitly, the Japanese military had been ordered to be on a footing for war with the United States by November 20, 1941.[3]

Private industry, driving much of the revolution in communication technology of the twentieth century, also developed its interest and expertise in cryptography. Claude E. Shannon at AT&T Bell Telephone Laboratories made several critical contributions to modern communication, computing, and cryptography. Shannon joined Bell Labs in 1941, after completing his Ph.D. in mathematics at the Massachusetts Institute of Technology. At Bell Labs, Shannon worked as a research mathematician and came to be known for "keeping to himself by day and riding his unicycle down the halls at night."[4]

In 1948, Shannon published "A Mathematical Theory of Communication" in the *Bell System Technical Journal.*[5] The paper was a breakthrough, founding the study of information theory, and coining the term "bit" to describe a BInary uniT. Up to that time, communication was thought to require electromagnetic waves down a wire or radio waves toward a receiver, but Shannon showed how words, pictures, and sounds could be sent across any medium that would carry a stream of bits. The following year, Shannon applied his work directly to cryptography in a paper entitled, "Communication Theory of Secrecy Systems."[6]This paper founded modern mathematically-based cryptography outside of government intelligence agencies.

Fig. 1. Claude E. Shannon, c. 1952. Property of AT&T Archives. Reprinted with permission of AT&T.

The rise of the computer and the rise of cryptography have gone hand in hand. Computing technology has made exchanging information easier, making communication and collaboration easier. Since people still want—and in an ever-growing number of cases, are legally obligated—to stay in control of information in their stewardship, people need cryptography.

Code makers and code breakers agree: the computer is both friend and enemy. For cryptographers, computer technology makes the implementation and use of flexible cryptography easier, while frustrating the efforts of cryptanalysts. For cryptanalysts, the computer improves efficiency in the analysis of encrypted messages and building systems to undermine cryptography, thus making it easier to exploit any flaw in the cryptographers' creations.

Cryptosystems before the twentieth century required tedious manual processing of messages, using code books to match what was written to what was to be communicated, or perhaps a great deal of scratch paper to perform the necessary text substitution and transposition. The process of encrypting and decrypting messages essentially consisted of taking a handwritten message, looking up the correct corresponding symbol on a chart, and writing the symbol on the paper that would actually be delivered to the recipient, who would in turn look at the chart and convert the ciphertext back to the plaintext by hand, one letter at a time.

Later systems like Enigma, though more convenient than the "old way," were still cumbersome and slow. (Early Enigma promotion material boasted that the machine could process 300 characters per minute.)

Though the internal mechanics were much more complicated, the user of the Enigma might liken its operation to a typewriter where the keys are randomly reassigned. The sender would type the letter according to the keys written on the keyboard, knowing that when an `A` is struck, a `V`, for example, will be written. The recipient will then need to know the keyboard layout used by the sender in order to recognize that the `V` in the message was created by striking the `A` key, and write "A" on a scratch pad. Working letter by letter, the sender's message becomes visible. Enigma handled this substitution work automatically, preventing operators from needing scratch paper.

Now, with computers, recipients can often click a few buttons and have huge amounts of deciphered information almost instantly turned into the sender's original message.

Perhaps no one understood the challenge and opportunity that emerged in the post-war era better than the researchers at IBM. In the 1950s and 1960s, with its systems designed to handle the heaviest information processing needs of both corporations and government agencies, IBM had to give serious consideration to the handling of sensitive data.

One of the earliest applications for computers was in the handling of government information—some of which was protected by law. Security was just as much a requirement for early computer systems as the ability to store and to process information accurately.

The trend to establish standards for data security in automated information systems became an important issue for IBM and its customers. The possibility of computerized records being abused was not lost on Americans, who were fascinated with computers and technology, but also worried about the implications of their use in society. One of the key figures in helping IBM realize a workable, powerful security scheme was a German émigré by the name of Horst Feistel. Feistel had arrived in the United States decades earlier, in 1934. Despite his interest in cryptography, he avoided working in the field during World War II to avoid suspicion by the American authorities.

After the war, Feistel found employment at the U.S. Air Force Cambridge Research Center, where he worked on identify friend-or-foe (IFF) systems. IFF systems were (and still are) used on the battlefield to avoid “friendly fire” incidents, where forces attack allied units instead of the enemy. Radar systems with IFF capability, for example, report not only the position of units in range, but whether they are friendly or hostile—thanks to the use of cryptography.

In the middle of the twentieth century, the highly secretive U.S. National Security Agency (NSA) had a virtual monopoly on cryptographic research and were trying hard to maintain it. Feistel’s Air Force project was canceled—though details are shrouded in military secrecy, NSA is generally credited with ensuring its hasty demise.

Feistel attempted to continue his work at Mitre Corporation in the 1960s, but again ran afoul of NSA’s plans. Dependent on Department of Defense contracts, Mitre had little choice but to ask Feistel to direct his energies elsewhere—presumably also at NSA’s behest.

Determined to apply his hard-earned expertise in cryptography, Feistel joined IBM before 1970, where he was finally free to continue his work, and headed up a research project known as Lucifer. The goal of Lucifer was to develop cryptographic systems for use in commercial products that would address the growing need for data security. IBM consequently was able to offer clients a means of protecting data stored in its computers.

Commercial users of computers were finally seeing the need to protect electronic information in their care, and an explosion began in the commercial availability of cryptographic products. In the late 1960s, fewer than five companies were offering cryptographic products, but by the early 1970s, more than 150 companies were active in the marketplace—and more than fifty of them were from outside of the U.S.

During this time, Feistel published an article in *Scientific American*, describing cryptography and how it relates to protecting private information in computers. Although much of the article focused on cipher machines of the sort that were used in World War II, it also contained some descriptions for mechanisms for computer software to encrypt information. Those methods, known as Feistel Networks, are the basis of many cryptosystems today.

Because the government kept their cryptographic technology under lock and key, commercial cryptographers could only guess at what their counterparts within government research facilities like NSA had achieved. These commercial cryptographers began with the fragments that could be assembled from historical literature and began to lay the foundation for open (i.e., not secret) cryptologic research.

At this time, though, very little was understood about how well various cryptographic techniques could withstand analysis. For example, one might believe that an encrypted message would be twice as resistant to analysis if encrypted twice. Only after years of research did cryptographers come to realize that for many kinds of ciphers, double encryption is no stronger than single encryption. Many questions played into a system's strength. How strong would a rotor-based system be if it used four rotors instead of three? How strong is strong enough? How strong is a rotor-based machine system by comparison with an encryption system implemented entirely in software?

In the early 1970s, no one outside of government cryptology knew the answers to questions like these, and it would be years before sufficient work in the field would be done to find answers. Thus, the availability of cryptographic products was of little help—people simply didn't know how good any of it was, and making meaningful comparisons was impossible. Even worse, no two vendors could agree on a system, requiring that both sender and receiver use the same equipment. It would be like buying a Ford only to discover that the nearest gas station sold only fuel to work with Chrysler cars.

Knowing that information needed to be protected, computer system managers had little choice but to buy *something* and hope for the best.

3

Data Encryption Standard

In the United States, the National Bureau of Standards (NBS) began undertaking an effort aimed at protecting communications data. As part of the Department of Commerce, NBS had an interest in ensuring that both its own systems and those of the commercial entities with which it dealt were adequately protecting the information under their stewardship.

The NBS effort included the establishment of a single standard for data encryption, which would allow products to be tested and certified for compliance. The establishment of a single standard would solve three major problems in the chaotic encryption marketplace. First, products compliant with the standard would have to meet security specifications established by experts in cryptography; individual amateurish efforts at merely obfuscating information would not pass muster. Second, compliant products from different vendors would be able to work with one another, allowing senders and recipients to buy from the vendors of their choosing. And third, the tremendous costs incurred by vendors in the creation of cryptographic systems could be reduced, since they would be able to focus on making the systems convenient to use, rather than spending huge amounts of money on development of the cryptographic underpinnings.

Requirements for the standard cryptographic algorithm—the definition of the series of steps needed to turn plaintext into ciphertext and back again—were published in the *Federal Register*. Among the requirements were a high level of security, complete and open specification, flexibility to support many different kinds of applications, efficiency, and exportability to the global marketplace.

NBS received many responses, though it ultimately determined that none of the algorithms submitted satisfied all of these requirements. Despite this apparent setback, NBS did not consider the effort to be a complete loss since it demonstrated that there was a substantial interest in cryptography outside of military circles. The large number of responses, in and of itself, was taken as a firm and positive step in the right direction.

NBS published a second request in the *Federal Register* on August 27, 1974. Once again, several serious submissions were made. Some were too specialized for the purposes NBS envisioned. Others were ineffective. One, however, showed great potential.

IBM's Lucifer project had an algorithm simply named "Lucifer," that was already in the latter stages of its development. IBM submitted a variation of the algorithm, one with a 112-bit key, to NBS.

Before the significance of the 112-bit key can be fully appreciated, it is important to note that modern computers are binary. That is, they store and process data in bits, the binary units Claude E. Shannon described in 1948. Anything with two settings can be used to represent bits. Consider a light bulb. It has two settings and two settings only: on and off.

All data in binary computers are represented in terms of bits, which are represented as 0 or 1. Absolutely everything, to be stored into a computer, must ultimately be represented with these two, and only these two, digits.

The easiest way to grasp the security of algorithms like IBM's Lucifer is to imagine a simple bicycle tumbler lock. Usually, such locks are made up of four or five tumblers, each with ten positions, labeled 0 through 9. In digital computers, however, a cryptosystem with a 112-bit key is like having a lock with 112 tumblers, each with two settings, `0` and `1`.

IBM's algorithm therefore had a total of 2^{112} possible settings, only one of which was the "key" to the system, the equivalent of the setting of a bicycle lock which would allow its opening. Seeing that number written out—5,192,296,858,534,827,628,530,496,329,220,096—shows why scientists prefer to use exponents when talking about large numbers. The difference is even more pronounced (pardon the pun) when you hear the numbers spoken. “One hundred twelve bit” is much easier to say than “five decillion one hundred ninety-two nonillion two hundred ninety-six octillion eight hundred fifty-eight septillion five hundred thirty-four sextillion eight hundred twenty-seven quintillion six hundred twenty-eight quadrillion five hundred thirty trillion four hundred ninety-six billion three hundred twenty-nine million two hundred twenty thousand ninety-six.” Such a vast number of possible solutions made the Lucifer algorithm a powerful means to protect information—satisfying two important NBS criteria at once: high security and security coming from the key.

NBS saw IBM’s submission as promising, but it had a serious problem—the algorithm was covered by some IBM patents which ruled out interoperability. IBM agreed to work out rights for the patents, such that even competitors would have the ability to produce systems that implemented the algorithm without the need to pay IBM licensing fees. Once this legal obstacle was removed, NBS went to work on evaluation of the system itself.

Lacking a staff with its own cryptographic expertise, NBS turned to the greatest source of cryptographic expertise known to exist—in other words, NSA—for help in evaluating the strength of the Lucifer algorithm. After careful analysis, NSA proposed two significant changes.

The first was a change in the algorithm’s S-boxes. S-boxes are the part of the algorithm that control how the data are permutated as they move from step to step along the process of being converted from the readable message to the encrypted result (or vice-versa), much like the rotors of Enigma.

The second, and more drastic, was the reduction of key length from 112 to 56 bits. This recommendation came as a result of debate inside of NSA. While the code-making part of NSA wanted to produce a standard that was strong and could protect U.S. interests, the code-breaking part of NSA was concerned that if the standard were too strong, it could be used by foreign governments to undermine NSA’s foreign signal intelligence efforts. Ultimately, 56 bits was the key size that won out as those two concerns were balanced.[7]

The difference in key size is significant. Because we’re talking about “tumblers” that are binary here—we’re working with a base of 2. That means that each digit added to the key doubles the key strength. That is, the number of possible settings, only one of which is the key to unlocking the encrypted message. Consider Table 1.

Power	*Conventional Notation*
2^{1}	2
2^{2}	4
2^{3}	8
2^{4}	16
2^{5}	32
2^{9}	512
2^{56}	72,057,594,037,927,936
2^{112}	5,192,296,858,534,827,628,530,496,329,220,096
2^{128}	340,282,366,920,938,463,463,374,607,431,768,211,456

Table 1. Powers of Two

The key of IBM's original cipher would be not just double or triple the strength of NSA's modification, but *fifty-six times* the strength. The reduction of the key rate caused a significant stir among the nascent group of civilian cryptographers.

In 1975, two cryptographers from Stanford became particularly critical of the 56-bit key. Whitfield Diffie, one of the two cryptographers, took the notion of an independent cryptographer to a new level. Not only was Diffie free from the restraints of secret government research, but he also developed his work free of the influence of large corporations. Having graduated from MIT with a degree in mathematics in 1965 and performed computer security work for several companies since then, Diffie found himself becoming recognized as an expert by his peers even without the help of a powerful support system.

Cryptographic systems long had a serious problem: getting the keys sent between the sender and recipient of encrypted messages. After all, if you can safely send a key in secret, why not use the same method to send the message itself? In practice, this problem was addressed through procedures, such as having the sender and recipient agree on a series of keys in person. The first message would be encrypted with the first key, the second with the next key, and so on, until they had exhausted their supply of keys, at which point they would need again to exchange a list of keys—whether in person or through a trusted source like a secured courier.

Being fascinated with the problem of the distribution of cryptographic keys, in particular key distribution over a global Internet, Diffie spent a lot of time thinking about this problem. While still forming his ideas on key distribution, Diffie visited IBM's Thomas J. Watson Laboratory to deliver a talk on cryptography, with particular emphasis on how to manage keys safely.

After his presentation, he learned that Martin Hellman, a professor of electrical engineering from Stanford had spoken at the same laboratory on the same topic not long before. Diffie took particular interest in Hellman because most cryptographers at the time were enamored with the algorithms themselves, leaving few to give the problem of key distribution any serious consideration.

That evening, Diffie got into his car and started driving across the country to meet Hellman. After arriving in Stanford, Diffie called Hellman, who agreed to a meeting. The two were impressed enough with each other that they looked for a way to work together. Because Hellman did not have the funding to hire Diffie as a researcher, he took Diffie on as a graduate student instead. Thus began the partnership of Diffie and Hellman at Stanford University.[8]

After the criticisms Hellman and Diffie leveled against the 56-bit key of the developing standard for data encryption throughout 1975 were ignored by NBS, the Stanford pair authored a letter published in *Communications of the ACM*. In that letter, they outlined their objections to the key size and its ramifications. Because the Association for Computing Machinery (ACM) is the oldest and largest association of computer scientists and engineers, its *Communications* is well-read and highly-regarded, seen by effectively everyone working in computing at the time.

Hellman and Diffie knew that the help of this group would be critical in forcing NBS to address their concerns. Even so, they recognized that the issue of the algorithm's security would be so far-reaching that their concerns would be of interest to the American public. The algorithm would protect data about the medical histories, finances, and official records of Americans from all walks of life.

If the standard could not withstand attack, it would be the American people who would suffer. Recognizing the difficulty of bringing such an obscure (albeit important) matter to the attention of the public, Hellman and Diffie wisely enlisted the help of David Kahn, author of the highly regarded 1967 book *The Codebreakers.*[9] Kahn wrote an Op-Ed piece for *The New York Times* that was published on April 3, 1976. In that article, Kahn wrote of the proposed standard, "While this cipher has been made just strong enough to withstand commercial attempts to break it, it has been left just weak enough to yield to government cryptanalysis."

By this time, experts from IBM, Bell Labs, and MIT had also weighed in on the matter: 56-bit keys were too small, they all declared. As Kahn noted in his article, "one major New York bank has decided not to use the proposed cipher" in part because of the criticisms of its key size.

The uproar was sufficient to cause the U.S. House of Representatives' Government Information and Individual Rights Subcommittee to look into the matter. NBS was forced to recognize that the field of cryptanalysis existed beyond the walls of government, that the concerns are real, and they must be addressed if the effort to standardize the proposed 56-bit system was to succeed.[10]Consequently, NBS decided to hold two workshops on the cipher proposed as the "data encryption standard" (DES).

NBS held two workshops in 1976 to deal with the objections raised by Hellman and Diffie. These were working meetings where cryptographers from across the country would be able to discuss the thorny issues around the proposed data encryption standard face-to-face. As part of their objections, Hellman and Diffie proposed the design of a special-purpose computer that would use a technique called brute-force to crack DES-encoded keys quickly. The first NBS workshop was composed of hardware experts who considered the proposed special-purpose DES cracker.

Some participants argued that the proposed DES cracking machine would not work because design and control costs would exceed the cost of the hardware. Hellman and Diffie countered that cracking DES keys would not be one large job, but many small jobs that could be performed independently. As such, there was no need for the microprocessors—the "brains" of the computer—to interact with one another. Each could be given tasks to perform independent of the others. This, Hellman and Diffie responded, meant that the objection to the feasibility of a brute-force attack on the basis of design and control costs did not stand.

Another matter of concern was the reliability of the computer—a more visible concern in the computing technology of the 1970s than it is today. The reliability of computers is directly tied to the number of components needed to construct them. Some of the NBS workshop participants performed calculations for a DES cracker with 1 million components—parts for handling computer working memory, storage, central processing, arithmetic logic, and all of the electronics to hold it all together. Based on the average time it would take electronic equipment of the day to fail, the million-component machine would not be able to run for more than a single day before failing in some way. Such a large system, with that level of failure, would be too big and too complex to operate.

The Diffie-Hellman design for a DES cracker, however, called for far fewer components—only 16,000. Furthermore, rather than using a large number of parts that would be used only a few times in the machine, the Diffie-Hellman design called for construction involving fewer types of parts—allowing any parts that fail to be easily replaced, getting the system back up and running in under ten minutes. Such a system would give error-free operation with a relatively small number of spare parts.

Another objection on the million-chip machine was its size: 6000 large cases—known as "racks"—that were 6 feet high. Hellman and Diffie responded with a proposal for a million chip machine in only 64 racks, suggesting that even were 1 million chips necessary, the size of the machine was being seriously overestimated.

Still basing assumptions on the large, million-chip, 6000-rack machine, power requirements were the next objection raised by NBS and others. Simply providing the electricity for such a machine to run would exceed any "reasonable budget," apparently without specifying what would constitute "reasonable." Hellman and Diffie proposed the use of chips manufactured in a newer and more cost-effective manner that would bring the operating cost to under $1500 per day, observing that power costs could be reduced five times with newer technology.

Looking at the speed with which a message could be encrypted with DES on readily available (general-purpose) chips, some participants determined that those chips would be too slow and cost too much when purchased in the quantity needed to test DES keys quickly. Looking at available technology, Hellman and Diffie suggested that complaints about chip speed and cost could be overcome by using a special chip, designed for the specific purpose of searching for DES keys. A special-purpose chip would dramatically increase the speed of the operation. Such chips, they observed, could be produced in quantity for $10 each.

In the course of this dispute, NBS even offered some of its own alternatives to increasing the key size. One approach they suggested was to develop a system that made use of frequent key changes. Rather than reusing the same key from one message to another, such a system would give each message a unique key. That way, the illicit discovery of a key would compromise only one message, rather than every message encrypted with that machine. Hellman and Diffie responded by observing that rather than cracking the message immediately after it was sent, some attackers might have the ability to intercept a message and then to spend the time necessary to break any particular message. (Interestingly, while cryptographers like Hellman and Diffie had no way to know it at the time, this is precisely what happened when SIS cryptanalysts could not keep up with the flow of Japanese military communications in the run-up to the attack on Pearl Harbor. Recall that SIS decrypted those messages five years after they were intercepted.) Hellman and Diffie went on to observe that medical records needed to remain private for ten years—that kind of long-term privacy requirement could not be met by a system where a single message encrypted with a relatively small key could be broken in a ten-year period.

Looking at the costs that would need to be borne by anyone implementing commercial cryptography, some argued that increasing the proposed standard's length of a key to 128 or 256 bits—as Hellman and Diffie suggested—would greatly increase the costs. The expense, in turn, would make the construction and use of such systems less attractive while also decreasing the overall use of encryption. Hellman and Diffie countered these assertions by observing that the computing power needed to perform encryption is much less than needed to perform brute-force search. (This works much like a scavenger hunt. Hiding twenty items—akin to encryption—is not significantly harder than hiding ten items, though finding those twenty—akin to brute-force decryption—would take dramatically more time than finding ten.) The difference in the cost of operation of a 128-bit system and a 56-bit system was negligible, but the payoff in terms of greater security was significant.

Finally, NBS argued that there simply was no way to tell for sure when the right key had been found in a brute-force search, even if someone took an encrypted message and used that key to turn it into a