A Dispatch From the Crypto Wars

A. MICHAEL FROOMKIN

Reviewing MATT CURTIN, BRUTE FORCE: CRACKING THE DATA ENCRYPTION STANDARD (2005).

ABSTRACT

Matt Curtin's Brute Force is a primarily personal account of one early effort to harness the power of distributed computing. In 1997, Mr. Curtin and other members of the DESCHALL (DES Challenge) project built, distributed, and managed software that united thousands of computers, many of them ordinary personal computers, in the search for a single decryption key among 72 quadrillion possibilities. The DESCHALL project sought to demonstrate that DES, then the U.S. national standard encryption algorithm, was no longer as secure as advertised. While Brute Force also offers some background on encryption regulation, export control policy, and other aspect of the Crypto Wars, it succeeds best as an almost diaristic account of the technical and organizational challenges at the heart of one of the earliest large-scale widely dispersed volunteer computing projects. The DES cracking project chronicled in Brute Force exemplifies the interplay between technology and politics. More importantly, Brute Force reminds us that although we survived one round of the Crypto Wars without actual controls on the use of cryptography, and indeed with some substantial relaxation of the export control regime that stood in the way of the routine adoption of strong crypto in many types of software, that result was not inevitable – and might again come under threat.

Matt Curtin's Brute Force: Cracking the Data Encryption Standard (*Brute Force*)[1] is a personal account of a battle in the first round of the Crypto Wars. It provides the fullest account in print of the cracking of a DES message in 1997 by the DESCHALL (DES Challenge) project. In addition, it offers a thumbnail sketch of the state of the political debate over permissible cryptographic use and its regulation as it stood towards the end of the 20th century. However, other books preceding *Brute Force* provided a much fuller account of the surrounding policy debates and their implications.[2] And if *Brute Force* was trying to be a

---

[1] Matt Curtin, Brute Force: Cracking the Data Encryption Standard (2005).

[2] *See, e.g.,* Whitfield Diffie & Susan Landau, Privacy on the Line: The Politics of Wiretapping and Encryption (1998). *See also* Stefan A. Brands, Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy (2000).

thriller along the lines of *The Cukoo's Egg*[3] or *The Eudaemonic Pie*[4] -- and there are some quite thrilling, or at least quite entertaining, stories to be told about the Crypto Wars -- this account, although very clear and workmanlike, probably will not be beach reading.

The decryption of a DES message by brute force - relying on distributed computing power provided by an army of volunteers coordinated over the Internet - was an important milestone in the movement to make strong cryptography more generally available (and exportable from the U.S.), although only one of many. The decryption signaled the vulnerability of the existing standard, and helped pave the way for a new and better one. As it happens, however, there are increasing signs that the second round of the Crypto Wars may soon be upon us. *Brute Force* is thus a timely reminder of the role of individuals in the battle to secure free access to the cryptographic tools that make strong electronic privacy possible.

The numbers of people and machines involved in the various competing DES key searches was a peculiar social phenomenon. As Whit Diffie, one of the fathers of modern asymmetric encryption put it:

> Exhaustive key search is a surprising problem to have enjoyed such popularity. To most people who have considered the problem, it is obvious that a search through 2^{56} possibilities is doable if somewhat tedious. If it [is a] mystery why so many of them, myself included, have worked to refine and solidify their estimates, it is an even greater mystery that in the late 1990s, some people have actually begun to carry out key searches.[5]

But search they did, reaching a rate of over 600 trillion keys per day.[6]

---

[3] Clifford Stoll, The Cukoo's Egg: Tracking A Spy Through the Maze of Computer Espionage (1989) (giving a dramatic account of an early attempt to fight a wily hacker).

[4] Thomas A. Bass, The Eudaemonic Pie (1985) (detailing the author's attempt to beat a casino with high-tech tools).

[5] Whitfield Diffie, *Foreword* to Electronic Frontier Foundation, Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design xi (1998), http://web.archive.org/web/19981202092932/http://www.eff.org/pub/Privacy/Crypto_misc/DESCracker/HTML/19980716_diffie_crackingdes_foreword.html.

[6] Curtin, *supra* note 1, at 252.

There are 72 quadrillion possible DES keys, but a brute force attacker would only have to test them all if she were exceedingly unlucky. The expected value of the number of keys one needs to test is 50% of the total keyspace, although the actual number could be anything from one to the full lot. In the event, the DESCHALL group was somewhat lucky: They found the key on June 18,[7] after testing about a quarter of the keyspace over a five month period.[8] Even the peak rate of 600 trillion keys represented less than 1% of the total keyspace per day. But the testing rate was rising rapidly, and Mr. Curtin reports that DESCHALL projected that it would have taken only 36 days to test the remaining three quarters of the keys.[9]

Although few are aware of it, almost every person in the developed world relies on encryption. From interbank financial transactions to ATM cards, from web connections to military communications, from car door openers to building security passes, modern activities depend on the ability to secure information and to release it only to users who can demonstrate they are entitled to have it. Encryption is a shield against eavesdroppers and intruders, but it is a shield that functions as a kind of sword: The ability to secure communications empowers not only those with lawful purposes, but potentially those with illicit goals as well. The same cryptographic authentication technologies that enable us to ensure that other people are who they claim to be can also with, at most minor changes, be enlisted to cloak identity online.

At their heart, cryptographic formulae (and the products which instantiate them) rely on abstract principles of set theory, and on cryptographers with a special sort of mathematical insight that builds strong codes and understands how to find their potential weaknesses. These formulae then have to be translated into software, introducing new rounds of potential vulnerabilities.[10]

Ciphers come in two basic varieties, symmetric and asymmetric. In either case, to encrypt a message one use the cipher to encrypt a message by applying a unique encryption "key." In general, well-designed algorithms of both sorts share the property that a longer encryption key makes for a more securely encrypted message. And,

---

[7] *Id*. at 259-260.

[8] The actual number of keys tested was 18,859,645,992,960. *Worldwide effort cracks DES*, ZDNet (June 23, 1997), http://us.cryptosoft.de/snews/jun97/23069705.htm.

[9] Curtin, *supra* note 1, at 254.

[10] *See generally* Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C (1994).

again speaking in very general terms, code-breaking happens in one of three ways: (1) attackers find a weakness in the algorithm (or its implementation); (2) attackers get a hold of the key needed to decrypt the message by, for example, suborning the keyholder; or (3) attackers subject the encrypted messages to a 'brute force' attack - in modern days this means programming computers to try every possible combination of keys until one produces something that looks like it could be the message.[11] Since programming computers to know that a decrypted message is in fact a real message and not gibberish is itself a considerable programming and computational chore, brute force attacks - especially the theoretical ones used to model the relative strengths of two different ciphers - are usually 'known plaintext attacks' based on searching for some known character string in the decrypted message.

In 1977, the U.S. government selected a symmetric cipher to be the "Data Encryption Standard" for private and non-classified use.[12] The single standard, the government reasoned, would enhance interoperability, which it had previously concluded was threatened by the emergence of competing cryptographic products that were unable to communicate with each other.[13] The lack of interoperability among commercial cryptographic products deterred firms from using encryption as much as they should.

DES is a single-key symmetric cipher: the sender and the receiver use the same key to encrypt and decrypt the message.[14] The strength

---

[11] *See* A. Michael Froomkin, *The Metaphor Is The Key: Cryptography, The Clipper Chip, And The Constitution*, 143 U. Pa. L. Rev. 709 app. at 885-889 (1995), *available at* www.law.miami.edu/~froomkin/articles/clipper.htm.

[12] DES, issued as FIPS 46 in January 1977, was reviewed, slightly revised, reaffirmed for federal government use in 1983 and 1987, and reissued as FIPS 46-1 in January 1988; on September 11, 1992, NIST announced a third review of FIPS 46-1, DES, and reaffirmed it for another five years as FIPS 46-2. Revision of Federal Information Processing Standard (FIPS) 46-1 Data Encryption Standard (DES), 58 Fed. Reg. 69,347, 69,347-48 (Dec. 30, 1993). DES was approved for use by the government for its sensitive information, but not for classified information. *Id.* at 69,348.

[13] Encryption Algorithm for Computer Data Protection, 40 Fed. Reg. 12,134 (Mar. 17, 1975) ("In order to insure compatibility of secure data, it is necessary to establish a data encryption standard and develop guidelines for its implementation and use.").

[14] With a symmetric cipher, whatever you do to encode a message, you just do the reverse to decode it. It follows that if an attacker knows the algorithm used to encode a message, and he knows the key used to encode it, the attacker has everything he needs to decode the message. Symmetric ciphers, even quite complicated and sophisticated ones, can be fast enough to use for real-time applications such as telephones as well as for asynchronous communications such as email. *See generally* Froomkin, *supra* note 11, at 890-94.

of a cipher is measured by its binary key length.[15] All other things being equal, longer keys mean stronger ciphers, with each additional bit raising the theoretical complexity of a brute force cracking effort (one that tries every possible key) by a power of two. DES keys are fifty-six bits (about eight ASCII characters) long.[16] This means that there are seventy-two quadrillion (actually 72,057,594,037,927,936) different possible keys.[17] But even the longest key in the world is of no use unless the underlying algorithm is sound - and whether that is the case is much harder to determine than the key length.

The government stated that by certifying the quality of the algorithm used in DES, it would reassure potential users that the system was strong enough to resist attack, something that most users would be unable to determine for themselves. Government endorsement had value to the private sector: Cryptography is hard; subtle errors can make a seemingly formidable system turn out to be much more vulnerable than ever suspected. As the National Security Agency (NSA) is acknowledged to have an excellent cryptography staff, government endorsement would give confidence that a cipher was unlikely to have hidden vulnerabilities. Similarly, government endorsement provided financial institutions and others with a degree of legal cover; were the cipher to prove vulnerable, the fact that it carried the U.S. government imprimatur likely would protect against a negligence claim. In contrast, reliance on any other system created a heavy burden of due diligence, not to mention the prospect of genuine risk.

The trouble was that while DES had to be algorithmically strong, its keys could not be *too* strong or DES would get in the way of the government's desire to be able to decrypt private messages for law enforcement and intelligence purposes. An earlier version of the cipher used a key with well over one hundred bits,[18] yet the published standard fixed the key at fifty-six bit. This raised fears that the government intended the shortened key to frustrate corporate

---

[15] Actually, not all ciphers use keys in the same way, so to compare the relative strengths of different ciphers it is sometimes necessary to use a conversion factor.

[16] FIPS 46-2, *supra* note 12, at 69,348.

[17] Gilles Garon & Richard Outerbridge, *DES Watch: An Examination of the Sufficiency of the Data Encryption Standard for Financial Institution Information Security in the 1990s*, 15 Cryptologia 177, 179 (1991).

[18] Schneier, *supra* note 10, at 221.

eavesdroppers while still permitting the NSA to break it if necessary.[19]

These fears about the weakness of DES intensified as computers got faster. Good cryptographers are some of the most professionally paranoid people on earth, so it does not take much in the way of a possible vulnerability to scare them. And by the late 1990s, experts were worried that DES might be vulnerable to brute force attacks by private attackers armed with purpose-built machines, as well as attacks by governments. In an effort to draw attention to these concerns, in early 1997, RSA Data Security, a company that sold strong cryptography tools, offered $10,000 to anyone who could decode a message that RSA had encrypted with DES.

The RSA challenge inspired competing efforts to crack the DES message using distributed processing in which a very large number of computers, from PCs on up, linked together via the Internet. Mr. Curtin, along with Rocke Verser and Justin Dolske, played a key role in organizing DESCHALL, the one which succeeded.

In 1997, Matt Curtin was a young software security expert working for an Internet startup.[20] A crypto-hobbyist from an early age, he was intrigued when, on January 28, 1997, RSA put out thirteen challenges to the code-breaking community, each at a different level of difficulty, and each with an increasingly large prize.[21] The message for the 40-bit challenge, the easiest, was decrypted by Ian Goldberg in only 3.5 hours. Foreshadowing the method used to crack DES, Goldberg took advantage of the computing power of 250 workstations in order to find the message encrypted by RSA: "The unknown message is: This is why you should use a longer key."[22] Within thirteen days, the 48-bit message (i.e. 2^8 times more difficult than the 40-bit challenge) had also been decoded by a parallel-computing project.[23]

DES seemed almost within reach - or was it? The government said the difficulty of breaking the 48-bit key showed that brute forcing DES was still very hard. After all, thirteen days of massively parallel computing could not be deployed casually to decrypt just anything.

---

[19] James Bamford, The Puzzle Palace: A Report on America's Most Secret Agency 347 (1982).

[20] Curtin, *supra* note 1, at 43.

[21] *Id.* at 44-45.

[22] *Id.* at 45.

[23] *Id.* at 46.

Moving to a 56-bit key from a 48-bit key added another 2^8 times more difficulty over the 48-bit key. If the work proceeded at the pace of the 48-bit crack, it was forecast to take nine years. The government contemporaneously estimated that a $30 million purpose-built computer could do the job in about 15 months.[24] Even that hardly suggested a cipher on its last legs. The private sector, reading the trend lines, said it was only a matter of time, and that even the cracks of the shorter keys showed that DES was near the end of its useful life.[25]

Spurred by the RSA challenge, DES became the next big target. Mr. Curtin decided to have a hand in cracking it, helping to found an effort that came to be called DESCHALL, for DES Challenge. The story of DESCHALL occurs on three levels in *Brute Force*: a personal narrative, a technological struggle, and the effort's political context.

PERSONAL

For a field founded on some fairly abstract number theory, cryptography seems surprisingly well populated with outlandish figures who are almost larger-than-life. One thinks of Whit Diffie, who along with the Martin Hellman and Ralph Merkle discovered -- or rather, since we now know that the British secret signals intelligence agency, Government Communications Headquarters (GCHQ), had already found it, rediscovered[26] -- public key cryptography. There is also Bruce Schneier, author of the indispensable *Applied Cryptography*,[27] and Phil Zimmerman, perhaps strong cryptography's greatest popularizer thanks to his "Pretty Good Privacy" Encryption Program, not to mention the makers of RSA, the leading commercial purveyor of strong crypto. Matt Blaze and several others found flaws in highly touted products. David Chaum patented some brilliant ideas for electronic cash and then held the patents so close to his chest that he almost killed off the market. And, of course, there were Eric Hughes, Tim May, "Lucky Green," "Black Unicorn" and the rest of the amazingly heterogeneous crew who populated the Cypherpunks

---

[24] *Id*. at 53.

[25] *Id*. at 54-55.

[26] *See generally Public-key Cryptography*, Wikipedia, http://en.wikipedia.org/wiki/Public_key (last visited Jan. 31, 2006).

[27] Schneier, *supra* note 10.

mailing list. The Cypherpunks thought that building systems that would allow people to communicate without fear of eavesdroppers, 'blacknet' networks of anonymous untraceable speech, and anonymous tradable e-cash would change the world. If some basic institutions such as the IRS or the FBI were subverted along the way, a significant minority might have cheered (at least back in the days before they got their current well-paid jobs as security professionals).

Unfortunately, as few of these people had a role in the organization of DESCHALL project, almost none of these characters appear in *Brute Force*, and if they do, they are at most a tangential presence. While we learn something about DESCHALL's competitors, especially SolNET, they are seen from their public statements and e-mails. This viewpoint reflects the day-to-day perspective of the insiders at DESCHALL, but one would have liked to hear other views more directly. There seems to have been no systematic attempt to interview the participants in the competing projects such as SolNET, for example, and let them tell their stories in their own words.

Instead, we have only Mr. Curtin - or rather, we have his professional side. While Mr. Curtin's diary-like account of the DESCHALL project is primarily concerned with his and his colleagues' experiences with regards to the DES cracking project, there are relatively few biographical or personal details in *Brute Force* regarding him[28] or other characters involved in the Crypto Wars. The DES crack was clearly an engrossing affair for those who ran it; it was also of interest to those who, like me, just joined the effort by running code-cracking program, but it is unclear to me how much the rest of the world is going to care about the minutiae of the effort. The unfortunate fact is that *Brute Force* succeeds best as raw material for some future historian, less well as a suspenseful tale, and much less well as an introduction to either the high or even the low politics of the time.

That said there are fun stories in the book, some which seem not just true, but familiar. I was particularly amused by the accounts of fruitless attempts to persuade authorities at Yale and Northwestern to allow students to run the DESCHALL client in the labs. The Yale computer lab administrator ignorantly objected that "we don't want to wear out the processors;" at Northwestern the objection was that there was no reason why the university should be "helping" RSA, a private company.[29] It certainly made me feel a little better about my

---

[28] Mr. Curtin's homepage suggests he's quite an interesting person. *See* http://www.interhack.net/people/cmcurtin/ (last visited Jan. 31, 2006).

[29] Curtin, *supra* note 1, at 124-25.

unsuccessful attempt to get the staff to run a similar client, distributed.net,[30] at the University of Miami. Our system administrators objected that they did not know what effect it would have on the network, and also did not want to be involved in anything that had a monetary prize attached. At the time, that felt like an unreasonable decision and it is interesting to learn that there was a lot of it around. It is also interesting to learn of the tensions between DESCHALL and the "Bovine project" behind distributed.net: Curtin's group thought that the Bovine project should have waited until the DES challenge was finished before targeting a different 56-bit cipher, RC-5, so as to avoid competition for computing resources. Plus, Curtin explains that he thought that RC-5 was "not an interesting target" because it secured mostly web communications rather than the financial and commercial data secured with DES. In addition, RC-5 could use longer keys; DES could not. A crack to 56-bit RC-5 could be remedied by using longer keys. A response to a DES crack would require a whole new encryption standard.[31] Although computationally equivalent, Mr. Curtin argues cracking DES was politically significant in a way that cracking 56-bit RC-5 never could be.

TECHNICAL

The technical challenge in a brute force attempt to test every key was substantial because the DES-cracking effort was not, as had been the case for so many code-cracking exploits in history,[32] the effort of a small team of canny cryptographers or the harnessing of a single massive supercomputer. Instead, the DES effort harnessed the power of a massively parallel and nationally distributed volunteer army of computers, ranging from the desktop PC to the large university laboratory machine, probably then the largest effort of its kind in history.

The way it worked was deceptively simple. Participants would download a "client" program that would do the actual work of trying keys to see if they were the one, which decoded the message. The

---

[30] Why I picked Distributed.net instead of DESCHALL or some other consortium, I no longer recall. I do remember that the distributed.net client had a nice cow logo and sent funny messages every time it got keys from the server. Distributed.net, http://distributed.net (last visited Jan. 31, 2006).

[31] Curtin, *supra* note 1, at 166.

[32] *See, e.g.* David Kahn, The Code Breakers: The Story of Secret Writing (1967).

client program ran in the background, and it got out of the way whenever the user wanted to actually do something with her computer. But since many machines, especially those in corporate and academic labs, sit idle for long periods (at night and during holidays), there was a massive amount of potential computing power just going to waste and ripe for DESCHALL and its competitors to harvest.

At the heart of the scheme was a "server" which kept track of which keys had been served out to clients, and which had been returned as tested. The idea was to ensure that every key got one - but only one - trial. Keys were served in blocks adjusted for the speed of the machine asking for work to do, but even when lumped into large blocks, seventy-two quadrillion keys take a fair amount of management.

The organizational and administrative efforts required to harness the power of otherwise idle computers is indeed a story worth telling, and one with ongoing implications. Added to that, there was indeed a need to optimize code so that it would run faster and faster - and the optimization problem was multiplied by several times given the divergent architectures of the various machines harnessed to the task. *Brute Force* does a good job of explaining the technical aspects of this challenge and how it was surmounted. Indeed, the DESCHALL code itself was a remarkable achievement, "especially with the Pentium and Pentium Pro processors, [Rocke] Verser's code was able to run at a phenomenal speed, allowing modest desktop computers with Intel processors to run circles around $20,000 scientific workstations."[33]

Managing the clients was a big job, too: To test as many keys as possible, the software needed to be optimized at quite a low level for the peculiar features of the architecture of the various computer chips in wide circulation. Code optimized for an Intel chip was not going to work nearly as efficiently on an AMD chip or on one of Sun Microsystem's powerful SPARCstations. And even if it worked at all, sub-optimal code widely disbursed could represent a lost opportunity to do millions of extra key trials. As the code changed and improved, participants had to be encouraged to update their clients; meanwhile, whenever possible, the central registry had to remain backwards compatible with older clients in order not to waste any work.

While its general aim was to get as many machines as possible running the fastest clients possible, the DESCHALL project was hobbled in one important way. U.S. export control laws had long treated DES as if it were a component for a dangerous weapon. DES

---

[33] Curtin, *supra* note 1, at 94.

could not be exported without a license, which required an onerous application form; the penalty for an illegal export included five years in jail. The rules were loosening, but not wanting to take risks the DESCHALL project organizers decided to limit downloads of their clients to persons who could claim a U.S. address and whose computer had a U.S. internet protocol (IP) number. This complied with U.S. export control regulations for software.[34] As a result, DESCHALL was a U.S.-only operation; SolNET, its major competitor, faced no such limit since it was based abroad and there is no law preventing the import of cryptographic software to the U.S. Worse, the addition of the IP number check into the equation introduced another point of failure - and it did fail quite spectacularly at least once, blocking an unknown number of downloads from legitimate users.[35]

Curtin's own role was primarily in working on the software on the server end, which harnessed all this horsepower for the common task. This, too, was no small feat. While the idea had been around on paper for some time, the infrastructure for organizing a very disparate set of machines had to be built almost from scratch. The problems of sending all the machine keys to work on recording the results, preventing duplication while making sure that no possible combinations were skipped, and ensuring that the computers available wasted as little time as possible waiting for instructions were all formidable ones; and even more so in the days - can it only be eight years ago? - when many people, even those with access to high-powered computers, relied on slow dial-up services to link their computers. Curtin does a very good job of explaining the problems and their resolution in terms that should easily be accessible to the non-specialist.

Perhaps of even greater significance than cracking one DES message was the demonstration of the sheer power that could be harnessed by distributed computing. It is indeed true that "quantity has a quality all its own"[36] And while a lot of the heavy lifting was done by very large machines, a very respectable fraction of the search was conducted by desktop computers. Annoyingly, Curtin does not organize his statistics in ways that make inter-temporal comparison easy, but he does tell us that even quite late in the project, when the larger machines were doing a greater fraction of the key searching,

---

[34] *Id.* at 68.

[35] *Id.* at 250-52.

[36] *Id.* at 116 (quoting Justin Dolske).

"daily statistics showed that roughly forty percent of the work was being done by small domains - those contributing less than one percent of the total processing power. Another ten percent was being done by even smaller domains - those contributing less than one-half percent of the total."[37] Indeed, the machine that actually found the key was a Pentium 90, not a supercomputer.[38]

POLITICAL

RSA's challenge had a political aspect, one not lost on the hard-core participants in DESCHALL. The organizers of the DES-crack campaign wanted the crack to make a statement about the need for stronger mass-market cryptography, as well as a geeky desire to surmount a challenge and win a contest; a substantial fraction of the people who lent their computers to the effort undoubtedly shared one or both impulses.

Cryptography was once the domain of military code-makers and especially code-breakers,[39] but it has broken out into the mainstream of computer science and mathematical research. A surprisingly swashbuckling cast of characters caused this transformation, people who together moved modern cryptography from a specialist field dominated by spies and counter-spies into something far more mainstream, but also more unruly. Even if most of the recent cryptographic pioneers were not actual revolutionaries, they were often the sort of strong libertarians who believed deeply in individual autonomy. They self-consciously saw their work as helping to preserve and to expand human freedom, as heading off otherwise powerful tendencies of states to use new technologies to create Orwellian systems of information control.

This attitude quickly brought them into conflict with intelligence agencies and with law enforcement. Intelligence agencies thought that the rise of consumer cryptography would result in a lethal threat to the so-called 'national technical means' of information gathering.[40] Law enforcement agencies, in addition to carrying the spear for the more reclusive intelligence agencies, argued loudly that allowing criminals

---

[37] *Id*. at 253 (quoting Rocke Verser).

[38] *Id*. at 261.

[39] *See* Kahn, *supra* note 32.

[40] *See* Froomkin, *supra* note 11.

to encrypt their phone and email communications would dangerously undermine wiretaps and other similar investigative techniques that the police required to detect and to prosecute organized crime and worse. The specter of wiretaps becoming useless also worried prosecutors, who well understood the dramatic impact of evidence of a conspiracy that could be played to the jury in the defendant's own voice; if the bad guys started encrypting all their telephone calls, these dramatic moments would be no more.

Thus, the battle over the use and regulation of modern cryptography arose, the so-called Crypto Wars.[41] On one side, a heterogeneous group of mathematicians, businessmen, civil (and even uncivil) libertarians, students and even the occasional anarchist, creating and seeking to deploy cryptographic tools. On the other side were government officials trying to protect surveillance and intelligence techniques against the specter of wide-spread military-grade cryptography which might be too difficult to break.

The battles took place at many levels. For example, the government sought at one point to argue that some ideas are 'classified at birth,' that whenever anyone works them out, even from unclassified sources, their very importance to national security means that the result should nonetheless be considered classified.[42] The obvious First Amendment problems with this prior restraint on speech, not to mention the terrible publicity it caused,[43] doomed it to failure.[44]

The primary means that the U.S. government used to slow the spread of strong cryptography relied on U.S. export control law. By preventing the export of cryptography, the government not only slowed its adoption abroad, but also at home. There were and are no legal controls on the production or use of strong cryptographic products by U.S. citizens or residents within the U.S., nor have there ever been in peacetime. However, for many years the U.S. limited the

---

[41] *Cf.* A. Michael Froomkin, *It Came From Planet Clipper: The Battle Over Cryptographic Key "Escrow*," 1996 U. Chi. Legal F. 15 (1996), *available at* www.law.miami.edu/~froomkin/articles/planet_clipper.htm.

[42] *See* Bamford, *supra* note 19, at 354-58.

[43] *See id.*

[44] Under the Inventions Secrecy Act of 1951, codified at 35 U.S.C. §§ 181-188 (2005), the government retains the ability to apply secrecy orders to patented inventions, even those produced by private parties. At the end of fiscal year 2005, there were 4,915 secrecy orders in effect, almost all of these likely applied to government-produced inventions. FAS Project on Government Secrecy, *Invention Secrecy*, http://www.fas.org/sgp/othergov/invention/ (last visited Jan. 31, 2006). *See* Bamford, *supra* note 19, at 355-56.

export of cryptographic systems with keys over 40 bits and also put tight limits on sharing the know-how to produce them.[45] Although banks and a few other high-value users were allowed to use a stronger variant, 3DES, even basic DES was officially rated as so strong that the government deemed it a dual-use technology, ranking it as dangerous as the trigger that might be used for an atomic bomb. Thus, under the International Traffic in Arms Regulations (ITAR),[46] DES had been treated as something so dangerous that it could only be exported with a license.

At the time of the events in *Brute Force*, the U.S. was in the process of de-controlling the export of DES, yet it was still fighting a rear-guard action against the spread of consumer cryptography.[47] By making the grant of export permission chancy and less than instantly rapid, the government introduced a substantial speed bump into the production cycle of software. Contrary to its caricature, U.S. export control policy was not premised on the false idea that foreigners cannot program (although, in fact, implementing cryptographic protocols without introducing exploitable errors is a difficult task[48]), but rather relied on the reluctance of vendors to take risks that might slow the shipping of products, or worse might require them to support different versions of the same product for the domestic and export markets.

A seemingly more promising governmental strategy (for a while at least) was to offer the private sector an improved version of government-endorsed strong cryptography in exchange for acceptance of a government 'back door' which would allow the U.S. government to decrypt private messages using the cipher with relative ease.[49] The

---

[45] There are, however, reasons to doubt the constitutionality of these limits. *See Bernstein v. Dept. of State*, 176 F.3d 1132 (9th Cir. 1999) (declaring portions of ITAR relating to cryptographic source code were unconstitutional prior restraint in violation of the First Amendment), *reh'g granted withdrawn sub nom. Bernstein v. Dept. of Justice*, 192 F.3d 1308 (9th Cir. 1999). *Cf.* EFF, http://www.eff.org/Privacy/Crypto_export/Bernstein_case/20000303_bernstein_pr.html (last visited Jan. 31, 2006) (explaining circumstances of subsequent remand to District Court); *Cf. Bernstein v. U.S. Dept. of Commerce*, No. C 95-0582 MHP, 2004 WL 838163 (N.D. Cal. Apr. 19, 2004) (noting case had become moot due to changes in ITAR and due to government's determination that Bernstein was not subject to prosecution under the new rules). *See also* Froomkin, *supra* note 11.

[46] International Traffic in Arms Regulations, 22 C.F.R. §§ 120.1-120.20 (2005).

[47] *See* Froomkin, *supra* note 41.

[48] *See* Schneier, *supra* note 10.

[49] *See generally*, Diffie & Landau, *supra* note 2; Froomkin, *supra* note 11; Froomkin, *supra*

algorithm, known variously as the Clipper Chip, Fortezza, and SKIPJACK, never achieved significant commercial adoption in part because of domestic and foreign mistrust of the U.S. government.[50] Gradual loosening of the export control regime combined with the widespread availability of both foreign-produced strong crypto and domestic escapees from the increasingly porous ITAR regime also reduced potential demand. Ultimately, however, the U.S. government did accept that DES was no longer secure enough for the private sector, and in 2001 it adopted the "Advanced Encryption Standard" (AES) as a DES replacement.[51] Interestingly, the National Institute of Science and Technology began the process that five years later would result in the adoption of the AES on January 2, 1997. That is just *before* RSA issued the DES challenge.

LESSONS LEARNED

Brute Force is especially timely. The DES-cracking experience teaches us several lessons with particular relevance today.

The first is that technology influences politics. The 1997 "crack" recounted in *Brute Force* did not in itself render DES transparent: it decoded only one message, and only after some substantial effort. The 72 quadrillion possibilities amongst which DES concealed encrypted texts remained a formidable barrier for any but the rare determined and powerful attacker. But by showing that DES was no longer invulnerable, and that as computing power increased by leaps and bounds it could only become more vulnerable, the DES crack effort marked one of the turning points in the struggle to spread strong cryptography, itself a project that remains controversial to this day.

And, indeed, the threat to DES quickly became much more real. In July 1998, shortly after the DESCHALL effort, John Gilmore, Paul Kotcher, and the EFF built an optimized DES-cracking computer

---

note 41.

[50] *See* Froomkin, *supra* note 41.

[51] Announcing Approval of Federal Information Processing Standard (FIPS) 197, Advanced Encryption Standard (AES), 66 Fed. Reg. 63,369 (Dec. 6, 2001). *Cf.* CSRC, AES http://csrc.nist.gov/CryptoToolkit/aes/ (last visited Jan. 31, 2006) (documenting history of adoption of AES). AES can have a keyspace of 128, 196, or 256 bits, making it monumentally stronger than DES. DES, however, was only officially retired last May. *See* National Institute of Standards and Technology Announcing Approval of the Withdrawal of Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard (DES); FIPS 74, Guidelines for Implementing and Using the NBS Data Encryption Standard; and FIPS 81, DES Modes of Operation, 70 Fed. Reg. 28,907 (May 19, 2005).

("Deep Crack") for about $250,000.[52] Working alone, Deep Crack found a DES key in only 56 hours.[53] A subsequent effort uniting the efforts of Deep Crack and a distributed processing team managed the feat in 22 hours and fifteen minutes.[54] DES was not just on the ropes, it was exposed as insecure.[55]

The second is that cryptography is a critical technology. There has never before been so much information collected and collated about so many. As I have argued elsewhere, the only defense to privacy-busting technology available to most of us is to keep our information private in the first place.[56] For electronic communications, strong crypto is our first line of defense - and maybe our only defense - against third parties seeking to capture our data.

The third is that not all "cracking" is bad. The Digital Millennium Copyright Act (DMCA), which among other things criminalizes the cracking of copy-protection systems, exemplifies a contrary stance, one we would do well to question. Suppose that DES had been protected by a DMCA-like rule, and that attempts to meter its increasing vulnerability had thus been illegal. The public, Congress, and policymakers all would have run the risk of remaining in the dark as to the real vulnerabilities of their tools, and might have enjoyed a dangerous false sense of security as a result. As Curtin puts it:

> DMCA prohibits any *attempt* to defeat an "effective" technical means of copyright enforcement. Putting the obvious logical question aside--an effective mechanism would withstand attack, so what's the point of prohibiting attack?--we are still left with a troubling question. If

---

[52] The story of the construction of this DES cracker is told (complete with wiring diagrams) in EFF, Cracking DES Secrets of Encryption Research, Wiretap Politics & Chip Design: How federal agencies subvert privacy (1998), *available at* http://cryptome.org/cracking-des.htm.

[53] Curtin, *supra* note 1, at 272.

[54] *Id.* at 273.

[55] But consider the wise words of Whit Diffie: [C]ryptosystems have nine lives. The most convincing argument that DES is insecure would not outweigh the vast investment in DES equipment that has accumulated throughout the world. People will continue using DES whatever its shortcomings, convincing themselves that it is adequate for their needs. And DES, with its glaring vulnerabilities, will go on pretending to protect information for decades to come. Diffie, *supra* note 5.

[56] *See* A. Michael Froomkin, *The Death of Privacy?*, 52 Stan. L. Rev. 1461 (2000), *available at* http://osaka.law.miami.edu/~froomkin/articles/privacy-deathof.pdf.

> consumers cannot independently verify the security of such systems and if we cannot understand how these systems are likely to fail, how are we supposed to ensure their validity?[57]

Fourth, export controls on software can impose real costs on U.S. companies and academics. Due to U.S. export rules, the DESCHALL team felt compelled to limit themselves to U.S. computers. Being based outside the U.S., its foreign competitor was not subject to U.S. export control rules and thus could solicit both U.S. and foreign participants. DESCHALL managed to win without a level playing field, but such U.S. victories are far from inevitable in the future, especially as computer resources are more widely spread around the world.

Fifth, and perhaps most important at this juncture, *Brute Force* can serve as a reminder that we have survived one round of the Crypto Wars without actual controls on the use of cryptography, and indeed with some substantial relaxation of the export control regime that stood in the way of the routine adoption of strong crypto in many types of software, notably consumer operating systems. But that was something of a close-run thing, a result due in part to the activism of people like Matt Curtin. And it could be that the forces favoring domestic cryptography regulation might make a comeback.

Just recently the FCC issued a Report and Order[58] and accompanying policy statement[59] as part of its attempt to assert broad jurisdiction over Internet telephony and other Internet-based communication services. As part of its sweeping vision of the scope of federal regulation over computer-based communications, the FCC stated it believed that, "to encourage broadband deployment and preserve and promote the open and interconnected nature of the public Internet, consumers are entitled to run applications and use services of

---

[57] Curtin, *supra* note 1, at 280.

[58] FCC, Communications Assistance for Law Enforcement Act and Broadband Access and Services, 70 Fed. Reg. 59,664 (Oct. 13, 2005) (adopting "a rule establishing that providers of facilities-based broadband Internet access services and providers of interconnected voice over Internet Protocol (VoIP) services--meaning VoIP service that allows a user generally to receive calls originating from and to terminate calls to the public switched telephone network (PSTN)--must comply with the Communications Assistance for Law Enforcement Act (CALEA)").

[59] FCC, Policy Statement, Appropriate Framework for Broadband Access to the Internet over Wireline Facilities et al. (2005), http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-05-151A1.pdf.

their choice, *subject to the needs of law enforcement*."[60] Of course, this is just a policy statement, not a legally binding decision … for now. And there are reasons to doubt whether the FCC's Communications Assistance for Law Enforcement Act (CALEA) authority stretches nearly as far as it wants us to believe,[61] much less into the realm of desktop software.[62] The FCC statement is nonetheless significant, for it cannot have come out of nowhere: It must reflect the agenda of at least a portion of the law enforcement community. Most likely these are the same people who made maximalist demands in the original CALEA process, seeking to optimize the telephone system for wiretapping, and who are now making similarly broad demands in regard to Voice Over IP (VoIP) and other communications that travel over the Internet such as Instant Messenger (IM).[63] The FCC's concern here seems to be to prevent end-users from installing software that would let them do VoIP-like things while evading the infrastructures that the FCC intends to mandate to make VoIP wiretap-friendly. As the FCC stated in the preamble to its recent regulation:

> The overwhelming importance of CALEA's assistance capability requirements to law enforcement efforts to safeguard homeland security and combat crime weighs heavily in favor of the application of CALEA obligations to all facilities-based broadband Internet access service providers…It is clearly not in the public interest to allow terrorists and criminals to avoid lawful surveillance by law enforcement agencies by using broadband Internet access

---

[60] *Id*. at 3.

[61] *See* FCC, First Report and Order, Communications Assistance for Law Enforcement Act and Broadband Access and Services 13 (2005), http://hraunfoss.fcc.gov/edocs_public/ attachmatch/ FCC-05-153A1.pdf (concluding that CALEA applies to facilities-based broadband Internet access providers and providers of interconnected VoIP service).

[62] The FCC's recent First Report and Order purports to apply CALEA standards to VoIP. This order is being challenged in the DC Circuit. *See* Cynthia Brumfield, *EFF, CDT and Pulver to Appeal FCC's CALEA Rules*, IP & Democracy, Oct. 24, 2005, http://www.ipdemocracy.com/archives/2005/10/24/index.php#000653.

[63] *See* Susan P. Crawford , *Shortness Of Vision: Regulatory Ambition In The Digital Age*, 74 Fordham L. Rev. 695, 714-24 (2005); Joshua E. Adrian, *Recent Developments In Administrative Law VoIP On Tap: Whether The FCC Should Apply Wiretapping Standards To Voice Over Internet Protocol*, 57 Admin. L. Rev. 647 (2005).

services as a substitute for dial-up service.[64]

Whether the FCC has the authority under current law or would need some new authority, once we have articulated the goal of trying to ensure that users cannot use VoIP without the possibility of being wiretapped, it is only one more step down the same path to regulating cryptography as well. After all, strong end-user cryptography threatens the ability of law enforcement and other government agencies to extract meaning even when armed with a legal wiretap order. Is it too far-fetched to worry that perhaps the government might seek some form of 'trusted computing' solution in which cryptographically secured devices seek to ensure that secure communications only happen in permissible ways? If it were to come to pass, round two of the Crypto Wars will have begun, and we may be needing some Matt Curtins again.

Brute Force documents one of the early efforts to harness the distributed processing power of the Internet. In the case of DESCHALL, the 'intelligence' being harnessed was silicone-based, and lay sleeping in computer labs around the nation.[65] More recently, in projects like Wikipedia, the intelligences being harnessed are carbon-based. At present it seems that the many-hands-make-light-work approach to writing and problem-solving will be one of the most important uses of the Internet. It is useful to have an eyewitness account of how one of those early efforts came to fruition.

---

[64] FCC, First Report and Order, supra note 61, at 18.

[65] Current projects that work on a similar basis include SETI@Home, which harnesses distributed computation to search signals for signs of alien intelligences.