DES Cracking on the Transmogrifier 2a

Ivan Hamer and Paul Chow

Department of Electrical and Computer Engineering
University of Toronto
Toronto, Ontario, Canada M5S 3G4
ivan.hamer@utoronto.ca
pc@eecg.toronto.edu

Abstract. The Cryptographic Challenges sponsored by RSA Laboratories have given some members of the computing community an opportunity to participate in some of the intrigue involved with solving secret messages. This paper describes an effort to build DES-cracking hardware on a field-programmable system called the Transmogrifier 2a. A fully implemented system will be able to search the entire key space in 1040 days at a rate of 800 million keys/second.

1 Introduction

The RSA Cryptographic Challenges sponsored by RSA Laboratories [1] have provided some interesting opportunities for those in the computing area to become involved in the mystery and intrigue of discovering secret messages. One of the challenges was to break a straightforward version of the Data Encryption Standard, more commonly known as DES [2]. The brute-force approach is to search the entire key space consisting of 2^{56} or about 7.2×10^{16} keys.

This paper describes a project to implement a DES cracking system in a general-purpose programmable hardware system called the Transmogrifier 2a (TM-2a) [3, 4]. The TM-2a is a unique system of field-programmable gate arrays being developed at the University of Toronto that is intended for doing prototyping of hardware. A brief description of the TM-2a will be given in Section 2.

In the remainder of this section, a brief overview of DES will be given and a review of other attempts at cracking DES will be given. Section 3 will describe our implementation of DES on the TM-2a. We will conclude and give some future work in Section 4.

1.1 Overview of DES

The simplest form of the DES algorithm takes a 56-bit encryption key and uses it to encode a 64-bit block of plain text data into a 64-bit block of output cipher text. Between an initial and final permutation, there are 16 essentially identical stages. In the first stage, one half of the data as well as the key goes through a function, **F**, and the result is exclusive-ored with the other half. For each successive stage, the same thing happens with the halves reversed. Figure 1 shows the data flow. Function **F** is shown in Fig. 2. The data and the key first go through the expander and reducer that do simple selection and/or replication of the input bits to generate two 48-bit words. These two words are then exclusive-ored to form a single 48-bit word, which then goes through a table lookup called the *S-box substitution*. The *S-box* substitution is shown in Fig. 3. It consists of eight 6-bit input, 4-bit output lookup tables. The lookup tables are predetermined functions that, along with the permutations, does most of the coding of the data. The same algorithm is used for decoding. Hence, if we run the output through the circuit again, we should get the same as what we started with.

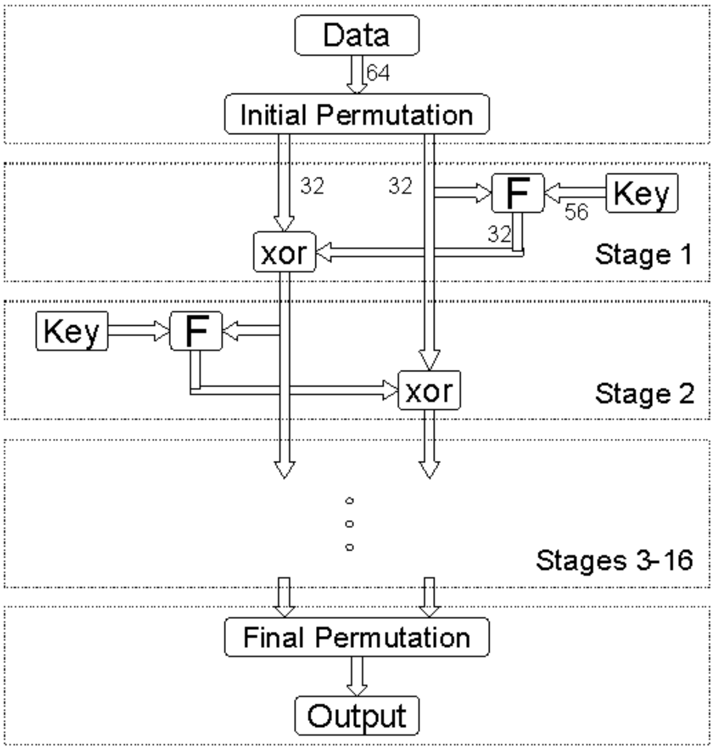

Fig. 1. The basic DES pipeline.

1.2 Other Attempts

The DES standard has long been criticized as being susceptible to an exhaustive key search and there has been much discussion and many recent attempts to show that it is weak.

One of the earliest analyses of a practical machine for doing this was done by Wiener [5] in 1993. In his appendix, there is a very detailed gate-level design

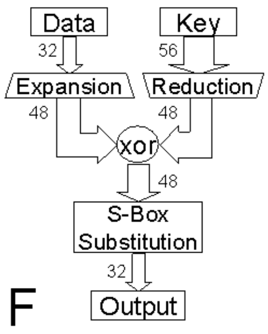

Fig. 2. The F function.

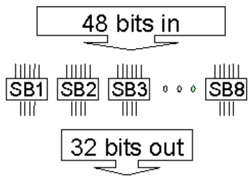

Fig. 3. The S box substitution.

of a chip that could be implemented in a CMOS technology. He estimates that the chip can test keys at a rate of 50 million keys per second. This chip can be used as the basis of a machine that can reduce the search time down to hours or minutes depending on the available budget. A review of numerous other designs was also given by Wiener.

Recently, the evolution of the world-wide web has made it possible to network together thousands of computers, ranging from low-cost personal computers to high-end workstations, all working on portions of the key space [6, 7]. This was how the first RSA DES Challenge was solved in about 4 months [6].

A real hardware system, called *Deep Crack*, was constructed by the Electronic Frontier Foundation (EFF) for under $250,000 and it was able to win the second RSA DES Challenge in 56 hours [8, 9].

A world-wide web group, hosted by Distributed.Net [7], and EFF combined their technologies to solve the final DES Challenge in a record 22 hours and 15 minutes [10].

The use of FPGAs as a means of building hardware to crack cryptosystems has been suggested by many in the past and we only cite a few here [11, 12]. FPGAs are an obvious technology because of the relatively low cost. Although our system of FPGAs will not come close to meeting the speeds of the EFF *Deep Crack* or Distributed.Net systems, we present it here as another data point showing what can be done with some programmable hardware, which puts it somewhere between an application-specific hardware approach, and a large network of general-purpose computers.

We first describe our hardware system.

2 The Transmogrifier 2a

The Transmogrifier 2a (TM-2a) [3, 4] is a second-generation field-programmable system that is constructed with field-programmable gate arrays (FPGAs). The TM-2a is a flexible rapid-prototyping system that offers high capacity and high clock rates. It is intended to be flexible enough to implement a wide variety of systems. A simple way to visualize the TM-2a is to think of building a large FPGA using existing FPGAs and field-programmable interconnect chips (FPICs).

Figure 4 shows the resources available on one TM-2a board. There are two Altera 10K100 [13] logic devices and four I-Cube IQ320 [14] FPICs. Attached to each FPGA is up to 4MB of RAM. The FPICs provide a programmable routing network that can be used to connect the FPGAs on the board to each other and to FPGAs on other boards. Each board also has a low-skew, programmable clock generator and a *housekeeping* FPGA that is used to monitor the system and communicate over a bus to the host system. When the host is on a network, then the TM-2a can be programmed and run remotely.

There can be up to 16 boards in a system. Assuming that each FPGA can hold a circuit of about 60K gates, the size of a 16-board system is about 2-million programmable gates.

The TM-2a is being used at the University of Toronto to prototype a number of hardware concepts such as 3-dimensional procedural texture mapping, head tracking, and image processing. When the RSA DES Challenge was announced, the TM-2a seemed like an obvious system for building a DES cracker.

3 DES on the TM-2a

In this section we describe the implementation of our DES cracking system on the TM-2a hardware. We first give a small primer on the Altera 10K series logic devices architecture and the design methodology that we use. Some of the history behind the development of the hardware is given before we describe the final implementation. We end with a summary of our results.

3.1 The Altera 10K Logic Device

The main building block of the Altera 10K logic device is called a *Logic Element* or LE. Each LE has a number of resources of which the important ones for us were the 4-input, 1-output look-up table (4-LUT), the cascade chain, and the programmable register. The LEs are grouped in blocks of eight called LABs with local routing within the LABs. The LABs are arranged in the chip as a matrix with another routing structure connecting the LABs. A 10K100 has 52 columns and 12 rows for a total of 4992 LEs.

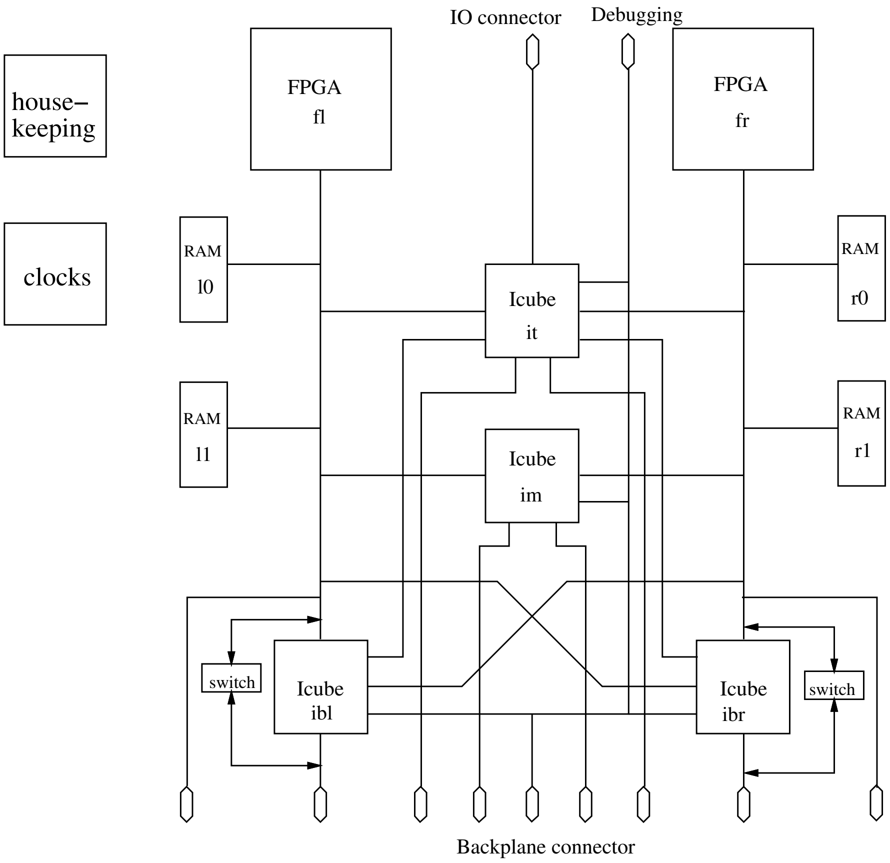

Fig. 4. Resources available on one of the boards in the TM-2a.

There are several ways to describe circuits that will be programmed in the device. These include schematics and various hardware description languages (HDLs). We chose to use AHDL, which is Altera's proprietary HDL, instead of a language such as VHDL. With AHDL, it is easier to control the logic mapping and therefore get more efficient and faster designs than with a more generic language. The actual synthesis and place and route is done using Altera's design system called Max+Plus II.

3.2 Early DES Work on the TM-2

Based on the work of Wiener [5] we understood that the goal was to build a pipeline capable of having a throughput of one key crack per cycle. Our first attempt [15] was based on an earlier version of the hardware, called the TM-2. The TM-2 was built at a time when the largest available FPGA was the Altera 10K50, which has roughly half the capacity of the 10K100 used in the TM-2a. Our TM-2 system has two boards, and four 10K50 FPGAs. On this system it was only possible to build half of the DES pipeline in a single 10K50. Therefore, we could only build two complete pipelines on the original TM-2 system. At that time the TM-2 only ran at 6.25 MHz, which was the limiting factor. This meant that we could crack keys at the rate of 12.5 million keys per second taking about 183 years to search the space.

Further analysis [16] of the work by Bernier showed that there were two areas that would limit the performance of the circuit. One was in the *S-box* circuitry and the other was in the interface circuitry that was used to communicate with the host. The interface could be easily decoupled from the rest of the circuit while the *S-box* needed more thought. A more serious problem we discovered was that the 10K100 did not really have double the logic of the 10K50 despite what the part numbers might suggest! The reason has to do with how the FPGA manufacturer counts its gates. This meant that we could not simply combine our two 10K50 circuits to form a single DES pipeline in one 10K100. More analysis and optimization of the circuit area was required.

3.3 The TM-2a DES Implementation

The goal of the TM-2a implementation was to implement a complete DES pipeline in a single 10K100, make it run as fast as possible, and then replicate it so that we could have 32 pipelines running in parallel. By doing this, we would not be limited by the interconnect network and crossing chip boundaries. It would also be much easier to replicate the pipelines across the system.

The top-level organization in a single chip is shown in Fig. 5. The *key maker*, which is some sort of counter, is used to produce keys. The plain text is coded with each key and then compared with the given cipher text. The circuit stops when a match is found.

There are enough resources to build all 16 stages as a large combinational circuit, but clearly this would be very slow. The next obvious step is to pipeline the logic by separating each stage with pipeline registers as shown in Fig. 6.

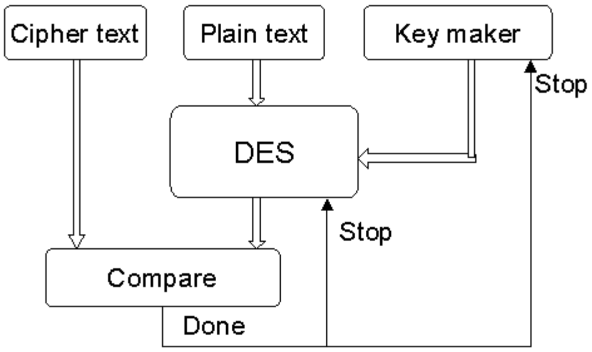

Fig. 5. Top-level structure of a chip.

The problem with this design is that there are not enough resources. As the computation proceeds down the pipeline, it is necessary to also keep the key for that stage in a register meaning that 16 keys will have to be stored. This uses almost 20% of the available LEs in the 10K100. We needed to find a *key maker* that would remember the 16 most recent key values without using so many registers. The next step would be to try and make the S-box logic go faster.

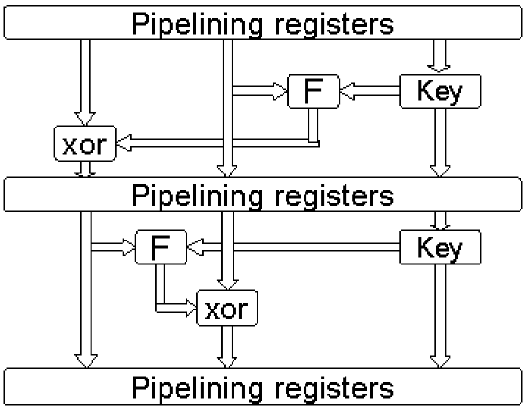

Fig. 6. Simple DES pipeline.

The Key Maker Our solution to the resource problem was to use a Linear-Feedback Shift Register (LFSR). By choosing the feedback taps correctly it is possible to generate each key exactly once. To remember previous keys, it is only necessary to extend the shift register by 15 registers as shown in Fig. 7. As each key is generated, the older ones can be found by sampling the values at a shifted offset from the new key. A possible disadvantage of this is routing the extended bits to the rest of the pipeline, but this ended up not being a factor.

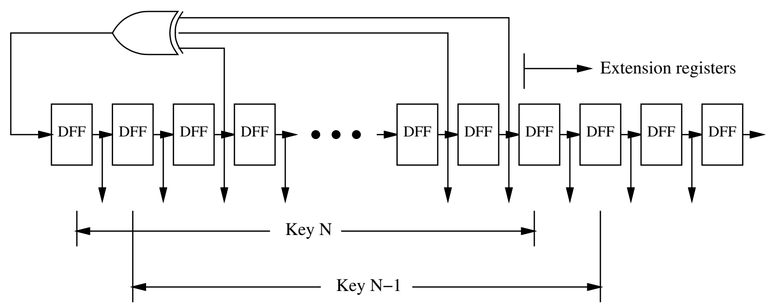

Fig. 7. LFSR with extension to save old keys.

Other advantages result from using an LFSR. An LFSR is much faster and simpler than binary counters, although in our case the key generation was not the critical path. Also, it is straightforward to serially preload the counter without additional logic and extra pins.

A slight disadvantage of the LFSR is because it does not count in a linear sequence. This means that we have to be a bit more careful when dividing the key space across the chips. The simple solution is to fix the key space for each chip by pre-setting the high order five bits when we are using 32 chips. We then build an LFSR that is only 51 bits long instead of 56 bits long.

Pipelining Possibilities Based on our previous work we knew that the *S-box* was the important critical path. Figure 8 shows one stage of the basic 16-stage pipeline and more details of how one bit of the *S-box* is constructed.

An *S-box* is a 6-input, 4-output lookup table, which can be thought of as four 6-input, 1-output lookup tables (6-LUT). The Altera device only has 4-LUTs so we had to find an efficient way to build the 6-LUTs. The straightforward solution is to have four 4-LUTs and a 4:1 multiplexer. The 4:1 multiplexer can be implemented as two levels of 2:1 multiplexers, which means that three levels of 4-LUTs are needed. A solution that uses only two levels and one fewer 4-LUT is shown in Fig. 8. This takes advantage of the AND gate that is available as part of the cascade chain in the LEs. An extra inversion is necessary at the output of the modified S-box but this can be absorbed transparently in the next level of logic.

The modified *S-box* structure can be easily pipelined, almost for free because the output of each 4-LUT can be latched at no extra cost. Only two additional registers are needed to pipeline the 2-bit bus that is connected to the inputs of

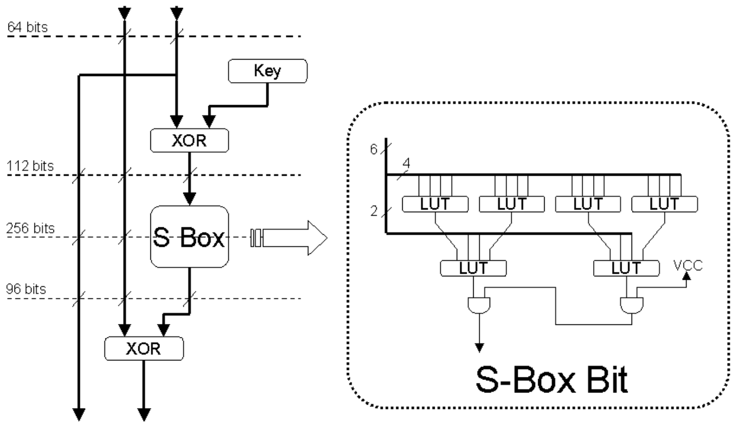

Fig. 8. S-Box details and pipelining options.

the second level of LUTs. However, the true cost of an additional pipeline stage must consider the context of the *S-box* in the full pipeline.

The simple pipeline puts a register between each of the 16 stages. If we wish to add an additional pipeline stage, then there are three possibilities as shown in Fig. 8. The dashed line at the top shows the existing 64-bit pipeline register. The labels on the dashed lines show the number bits that have to be registered if pipelining were done at that level. We do not have enough resources to add registers at all of the levels. The most economic place is at the level that goes through the *S-box* because many of the register bits come for free as mentioned above. However, it is still necessary to register the 64 other bits at that level that do not go through the *S-box*. Unfortunately, adding this extra pipeline stage exceeded the resources available to us so we are left with the original simple pipeline.

The Complete System The full system will consist of 32 complete DES pipelines, each running in one of the 10K100s on the TM-2a. Software running on a host machine will communicate with the hardware to monitor the status of each chip. In addition there is a separate daemon program that monitors the status of the TM-2a. Since the TM-2a is available to everyone on our network, it is essentially a common resource. Users make calls to the monitor to gain access to the machine and to load their circuits. The actual utilization of the TM-2a for other projects is low so we have modified the monitor to determine when the TM-2a is idle. During the idle periods, the DES cracker can be loaded and run. When the TM-2a is needed, then the current state, which is just the current key in the LFSR, is saved so that it can be restored the next time the circuit is loaded.

3.4 Results and Status

Our final design uses 4300 of the 4992 available LEs, which is about 86% of the resources. Adding an extra level of pipelining in the *S-box* was just 4% larger than what could fit in our FPGAs. It would have easily worked if we had 10K130 devices.

The maximum clock speed reported by Max+Plus II is 25MHz. Since we are able to process one key per clock cycle, this gives us 25M keys/second per chip. By using all 32 chips available on TM-2a, a total throughput of 800M keys/second is achieved. To search through the whole key space of 2^{56} keys, it would take 90.1 million seconds, or 25 thousand hours, which is about 1040 days. While this is clearly not fast enough for practical use, it represents a tremendous speed increase compared to what conventional computers can do within the same volume of space. If we could have improved the pipeline with one extra stage in the *S-box*, the speed would have been over 40 MHz giving around 650days to search the key space.

Since much of the structure of the circuit is reasonably regular and the data flows in one direction, we would have liked the option of hand-placing the logic to reduce routing delays. Evidence from other work using other devices shows that amazing speeds can sometimes be obtained, such as a 250 MHz cross-correlator [17]. Hand-placement was not an option with our devices. We do not know how much difference this would have made, given the hierarchical routing structure of the Altera 10K devices but it would have been nice to try. We feel that with a different FPGA architecture, we could have more easily optimized the design for speed.

The TM-2a is estimated to cost about US$3300 per board and about US$60,000 for the 16-board system using prices from the fall of 1998[1]. If the desire is to always be using the current state-of-the-art FPGA then the above numbers are probably a good estimate for a starting point.

However, this is much more than would be needed for a dedicated system of 32 chips. A single-board system with 32 chips using similar technology to ours is estimated to be less than half the cost of a 16-board TM-2a system. The TM-2a is also using technology that is about 2 years old. When we revised the TM-2 design to use the 10K100s, we could have used larger and faster parts but this would have caused too much change to our design, given our desire to make the revision quickly. We would have had to redo our routing network because there would have been more pins, and the faster parts run at lower voltages, meaning our board design would have had to change too much.

It is clear that as the density and speed of FPGAs continues to improve, it will become easier and easier to build a small fast machine to crack DES.

We have successfully run the system on a two-board (four-FPGA) version of the TM-2a. At this point in time, summer of 1999, our 16-board system is being

[1] Our numbers are very approximate because we have always been fortunate that Altera was willing to donate the devices that we needed so we have been sheltered from a lot of the true costs.

commissioned. Our DES cracking circuit is the first application to run on it that uses all of the boards.

4 Conclusions and Future Work

In this paper, we have described the implementation of a DES cracking system on a general-purpose field-programmable hardware system. The goal was to demonstrate the capabilities of field-programmable hardware and, in particular, the capabilities of our particular TM-2a field-programmable system. Although the system cannot compete with those that actually were able to solve the DES Challenges, our implementation does show how close technology is to being able to build machines capable of cracking DES without the aid of special-purpose custom hardware or the organizational requirements of coordinating a large number of computers on a network. This technology is available to everyone.

A 16-board TM-2a system can achieve a throughput of 800 million keys per second, which is still about 300 times slower than the last DES Challenge winner that was a combination of the EFF *Deep Crack* custom hardware and Distributed.Net's roughly 100,000 computers. They were testing 245 billion keys per second when the key was found [9]. When compared to just the *Deep Crack* hardware, which can test over 88 billion keys per second, the TM-2a is about 110 times slower. Based on our estimate of about US$30K for a dedicated 32-chip system, spending the same amount as EFF did would give us 8 times more performance, so that the FPGA system would only be about 14 times slower. By using a tool that allows more manual placement and routing and a similar generation of technology to *Deep Crack*, it is possible we could find another factor of 2 to 3 in performance. The difference between programmable and custom hardware then becomes even smaller.

With very few modifications, our DES cracker can be used as an ordinary high speed DES encoder/decoder.

For our own research into FPGA architectures and systems, the DES cracker circuit has given us a large benchmarking circuit. In future we plan to investigate more sophisticated ciphers such as RC5 [18].

Finally, it is clear that DES cracking hardware is quickly becoming within reach of many institutions because of the rapid improvement in FPGA technology.

5 Acknowledgements

Thanks go to Marcus van Ierssel and Dave Galloway for keeping the machines running. The Transmogrifier project benefits from the support of Micronet, a National Centre of Excellence in Canada, ATI Technologies, Altera Corporation, Cypress Semiconductor, and the Natural Sciences and Engineering Research Council of Canada.

References

1. RSA Laboratories. http://www.rsa.com/rsalabs/.
2. Data Encryption Standard. National Bureau of Standards (U.S.), Federal Information Processing Standards Publication 46, National Technical Information Service, Springfield, VA, 1977.
3. David M. Lewis, David R. Galloway, Marcus van Ierssel, Jonathan Rose, and Paul Chow. The Transmogrifier-2: A 1 Million Gate Rapid Prototyping System. *IEEE Transactions on VLSI Systems*, 6(2):188–198, June 1998.
4. http://www.eecg.toronto.edu/EECG/RESEARCH/FPGA.html.
5. Michael J. Wiener. Efficient DES Key Search. In W. Stallings, editor, *Practical Cryptography for Data Internetworks*, pages 31–97. IEEE Computer Society Press, 1996. First presented at the Rump session of Crypto '93 and also available by searching the WWW.
6. http://www.frii.com/~rcv/deschall.htm.
7. http://www.distributed.net.
8. Electronic Frontier Foundation, editor. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1998.
9. http://www.eff.org/descracker.html.
10. http://www.rsa.com/rsalabs/des3/index.html.
11. Electronic Frontier Foundation, editor. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*, chapter 11. O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1998.
12. Tom Kean and Ann Duncan. DES Key Breaking, Encryption and Decryption on the XC6216. In *IEEE Symposium on FPGAs for Custom Computing Machines*, pages 310–311, 1998.
13. http://www.altera.com.
14. http://www.icube.com.
15. Carolynn Bernier. DES Cracking on the TM-2. Undergraduate summer project report, 1997.
16. Kathleen Lam. Implementation and Optimization of a DES Cracking Circuit on the Transmogrifier-2 and the Transmogrifier-2a. B.A.Sc. thesis, Division of Engineering Science, Faculty of Applied Science and Engineering, University of Toronto, supervised by Professor Paul Chow, 1998.
17. Brian von Herzen. Signal Processing at 250 MHz using High-Performance FPGAs. In *International Symposium on Field Programmable Gate Arrays*, pages 62–68. ACM/SIGDA, 1997.
18. Bruce Schneier. *Applied Cryptography*. John Wiley and Sons, New York, 2nd edition, 1996.