WP170 (v1.2) November 19, 2002

CoolRunner-II CPLDs in Secure Applications

By: Jesse Jenkins

In the last year, inconceivable threats have become very real circumstances. Anxieties that once seemed like paranoia now seem like commonplace security concerns. International tragedy, terrorism, and the rapid growth in portable and wireless products have all brought large numbers of people face to face with—among other worries—the potential dangers of compromised data security. Sophisticated cryptography has been a major emphasis in products designed to counter these dangers, but even mathematical algorithms are susceptible to outside theft or interference with common microprocessor systems. Additional precautions must be taken to deter talented hackers and well funded technical thieves from acccessing private information. In recent months, Xilinx CoolRunner™-II CPLDs have drawn attention in the press for their potential use in just such precautionary measures.

This white paper will discuss the features of Complex Programmable Logic Devices (CPLDs) in general, and CoolRunner-II devices specifically, that lend validity to some of these claims. It will cover the basics of tamper resistance as well as detail some of the more interesting data-attack methods. Describing the technical value of CoolRunner-II CPLDs in hindering these attacks, it will conclude by summarizing the degree of safety provided by the use of CoolRunner-II devices and outlining additional defensive steps that can help to ensure data security.

Introduction

Xilinx CoolRunner-II CPLDs are general purpose, high-speed, and low-power CPLDs with the added advantage of built-in tamper-resisting features that help to safeguard designs. Because many industries — like gaming, wireless communication, automotive telematics, and security — depend on reliable design security, this paper will detail the capabilities of CoolRunner-II devices such that designers working in these or related industries will be able to evaluate the benefits these devices can bring to their systems.

CoolRunner-II CPLDs deliver all the standard CPLD capabilities expected of Industry-standard programmable logic. In addition, when it comes to protecting designs from being inspected or copied, CoolRunner-II CPLDs include elements that make designs substantially more secure than do competing products in today's market.

Before going into the details of how these particular features function, however, let's consider some PLD basics and identify weaknesses with other standard methods.

Simple PLD Security

In the past, nonvolatile PLDs had an advantage over volatile ones in that the nonvolatile PLDs did not need to reload their patterns into their chips each time they were powered up. Volatile parts do, and their patterns are exposed and accessible, to some degree, during this necessary reload. Nonvolatile parts, because they are self-contained, configure themselves from bits stored within the device that are never exposed to the outside world. Xilinx advanced Virtex™-II FPGAs circumvent this issue by a clever nonvolatile addition to their volatile logic fabric (using a battery backed up 168-bit encryption key).

Many nonvolatile devices support a read-back port where it is possible to inspect the internal program pattern using a third party programmer or JTAG port. Additionally, they frequently have programming or test pins that accept a “super” voltage, exposing (to external access) pins which exhibit test behavior ("test behavior" meaning circumstances in which normal I/O pins become row/column address and data pins directly attached to the programming array within the chip). Test circuitry is highly beneficial in production because it allows factories to quickly program and test chip innards, but it is also a liability because it gives outside parties the ability to inspect the chip, as well.

To deter this kind of hostile inspection, SPLD providers introduced “protect” bits — namely, read and write protect bits. Read protect, when set, denies access to the internal pattern via the pins. Write-protect, when set, denies programming access to the part, except to erase the entire part. These two actions are considered substantially safer than nothing.

However, protect bits are not complicated cells, and once “hackers” discovered their simplicity they attacked the obstacle with partial erasure. By cutting through the package and exposing the die, SPLDs could be optically erased. EPROM, EEPROM, and Flash devices will all erase if sufficiently exposed to ultra violet light.

A standard experiment to rapidly discover the die position of security bits is to take two die and drop a simple inverter/buffer design into one without setting the read/write protect bits. Then, program the other part similarly, but set the security bits. Whatever bits differ are the security bits. Using a simple search with multiple parts, hackers can expose half the die to light for several hours and attempt readback. If they are successful, they know the exposed half has the read protect bit. If unsuccessful, they know the read protect bit was on the other half of the die, so they then cover one half of that half and repeat.

Binary deduction, aided by the fact that chip designers typically set the read / write protect bits apart from the primary programming circuitry, significantly weaken this "protective" measure. Being separated means they can be inspected by microscope. Tightly aimed light (UV laser) can target just the read / write protect cells and preserve the rest of the design, and, voila! The file is cracked and readable from the JTAG or programming pins. Using a software tool that can parse the resulting JEDEC file, the design is easily converted to equation format. The hackers can also halt at the extraction of the JEDEC file, and optionally copy the SPLD, if that is their goal. If they wish to edit or improve it, they usually add the step involving the creation of equations.

Figure 1 shows the structure of a nonvolatile PLD where there is conceptually a layer of logic and programming cells coupled to at least one layer of metal. The metal is much of the rest of the design.

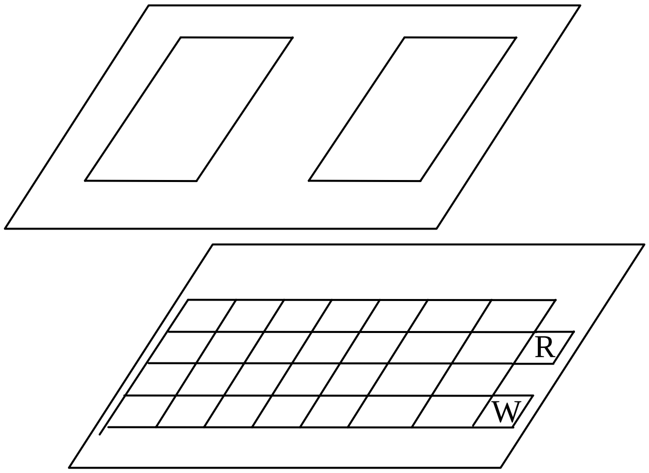

Figure 1: **Logic Layer Above Programming Cells for an SPLD**

Note: "R" and "W" represent protect bits outside the main array.

Complex PLD Security

Most CPLDs offer the same protection — read protect and write protect bits. To some degree, they are more secure than SPLDs simply because their complexity dictates smaller features and more metal layers. The die surface is no longer visible to UV for direct erasure. Most CPLDs have four layers of metal or more and this blocks optical penetration to the die surface for erasure. Hence, a simple read protect / write protect protocol might be sufficient.

The need to have factory access to the pins for testing remains, though, and is still an exposure point for hackers wanting to experiment with super-voltages. This is risky, but can be profitable in fully exposing the addressing / data innards to the pins. However, reverse engineering methods have evolved along with the newer technologies (see **Reverse Engineering**, page 10 for more information on this).

CoolRunner-II Security

CoolRunner-II adds multiple read and write protect bits into the standard CPLD framework. They are placed among the programmable cells that hold the design. They are also placed in a way that requires specific sequencing of signals to set and clear them, as well as charge pumping and other protocols. With today’s bit counts (well above 10,000) simple erasure experiments would take substantial time, and four to five layers of metal rule out top die exposure. See Figure 2.

Because CoolRunner-II CPLDs are designed for a wide range of applied voltages to accomodate their applications in the handheld, portable design world, they are substantially less susceptible to exposing address and data bits via externally applied voltages. CoolRunner-II devices do not use external super-voltages, as programming occurs strictly through the JTAG interface.

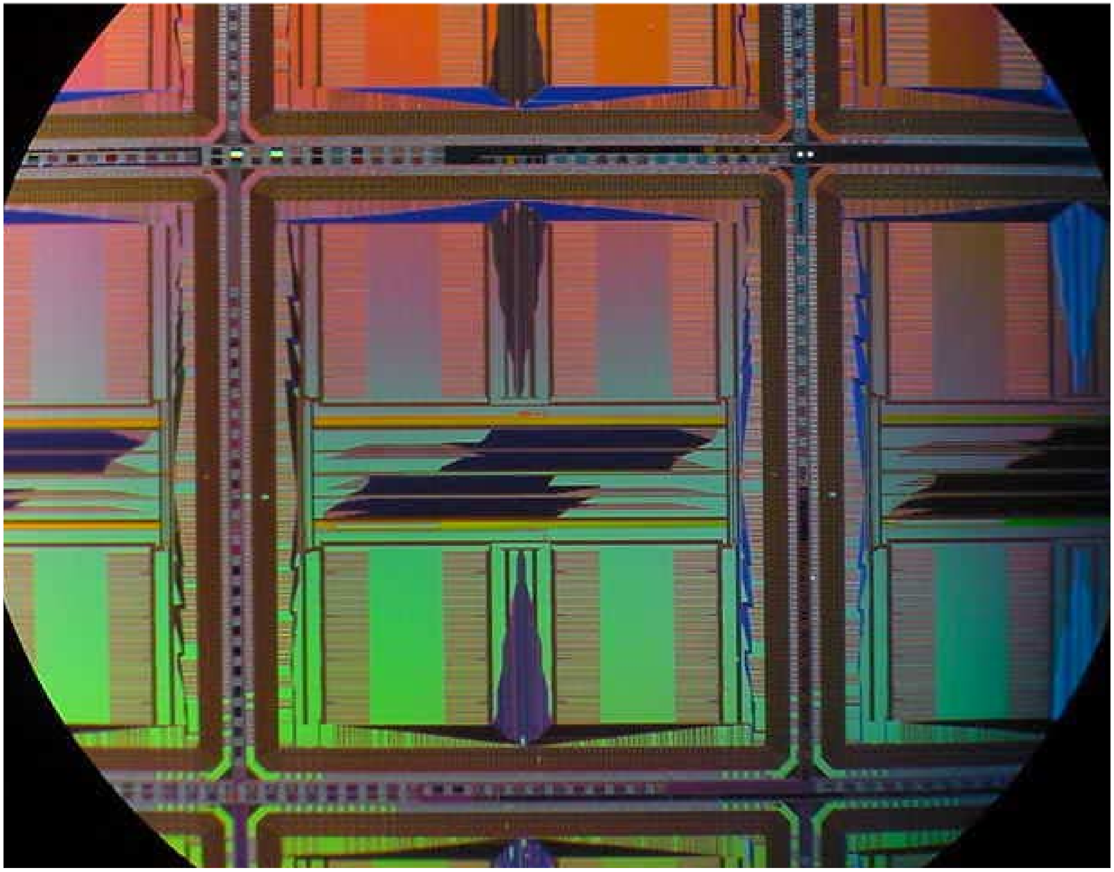

Figure 2: **CoolRunner-II CPLD Top Metal Layers**

CoolRunner-II Basics

CoolRunner-II CPLDs are designed to provide industry standard CPLD capabilities with a focus on both high speed and low power — hence, their wide acceptance in both data communication systems and portable wireless devices. Both marketplaces have an inherent need for security, and CoolRunner-II parts also address that need. However, let’s take a few lines to describe the basic CoolRunner-II device capabilities.

Figure 3 shows the high level architecture of CoolRunner-II devices. Table 1 shows the densities, packages and I/O counts for the various parts. Figure 4 expands the detail for a single function block and Figure 5 exposes the innards for a macrocell. Of particular note for the security conscious are the D flip flop, the EX-OR gate and the substantial product term resources for creating logic. Each macrocell is well suited to standard operations associated with password-checking (wide compares), linear feedback shift registers (pseudo random number generators, CRC, signature analysis, etc.) and boolean operations for encryption and decryption. Feedbacks through the Advanced Interconnect Matrix are well suited for high connection and potential combinational barrel shifting. All of these tasks are common for security design. However, the PLA structure also works efficiently for standard logic creation and state machines. It is also possible to build small, fast microntrollers using CoolRunner-II CPLDs.

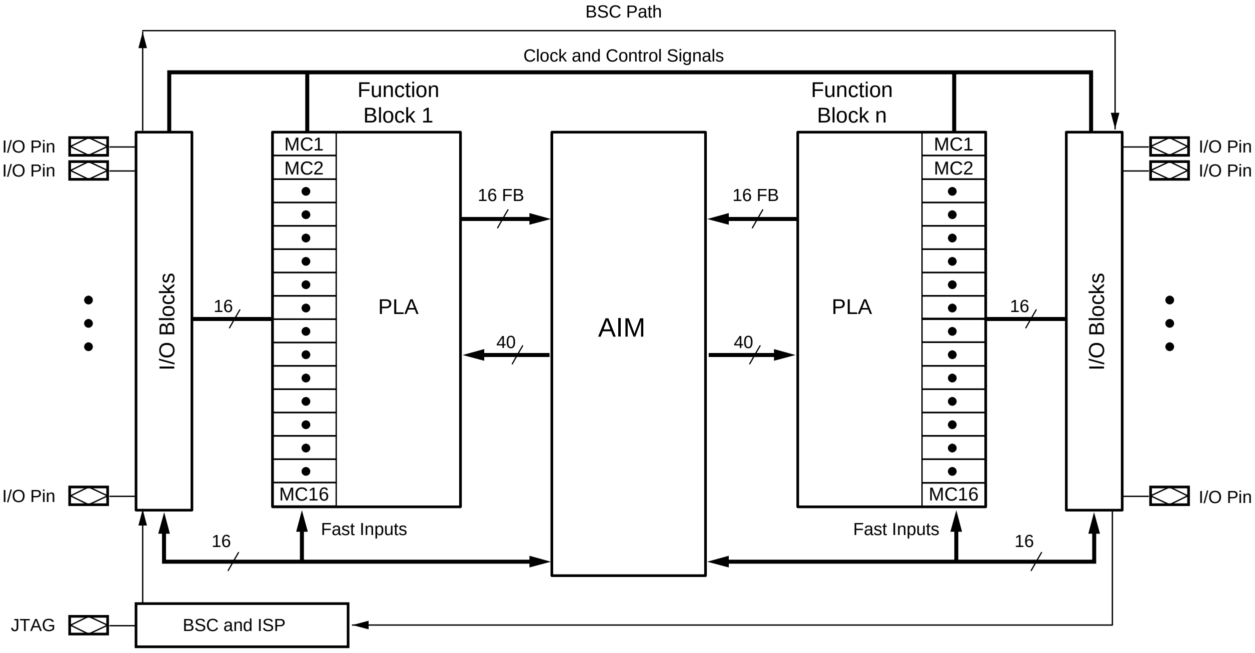

Figure 3: **High Level CoolRunner-II Architechture**

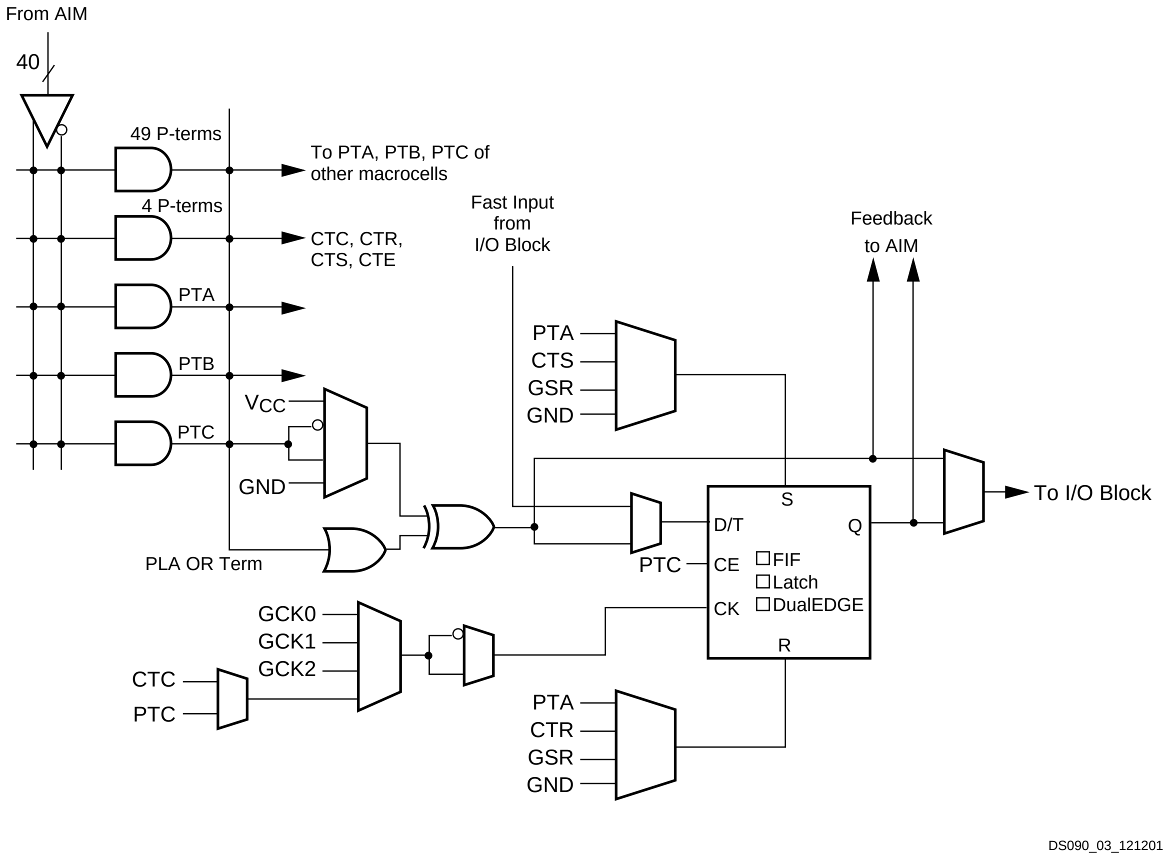

Figure 4: **CoolRunner-II Macrocell**

Table 1: **CoolRunner-II Features versus Macrocell Density**

	XC2C32	XC2C64	XC2C128	XC2C256	XC2C384	XC2C512
IEEE 1532	✔	✔	✔	✔	✔	✔
I/O Banks	1	1	2	2	4	4
Clock Division	-	-	✔	✔	✔	✔
Clock Doubling	✔	✔	✔	✔	✔	✔
DataGate	-	-	✔	✔	✔	✔
LVTTL	✔	✔	✔	✔	✔	✔
LVCMOS15,18,25,33	✔	✔	✔	✔	✔	✔
SSTL2-1	-	-	✔	✔	✔	✔
SSTL3-1	-	-	✔	✔	✔	✔
HSTL-1	-	-	✔	✔	✔	✔
User programmable ground	✔	✔	✔	✔	✔	✔
Quadruple data security	✔	✔	✔	✔	✔	✔
Open drain outputs	✔	✔	✔	✔	✔	✔
Hot plugging	✔	✔	✔	✔	✔	✔

Additional capabilities originate in the device's low power features. Using standard CMOS design methods, CoolRunner-II draws microwatts where other products pull milliwatts or even watts of power. Two of these low power features — DualEDGE flip-flops and DataGATE — are particularly inviting for security.

DualEDGE flip-flops are designed for both power and speed advantage, but add another level of confusion to reverse engineers, since most designers are unfamiliar with double data rate operation when applied to state machines.

DataGATE is the ability to lock out input signals under the control of an internal macrocell. This can isolate the chip for several types of attacks, if desired.

Note: *Be sure to scan the* ***Glossary****, page 17, for a partial list of common and esoteric attacks on digital systems.*

Figure 5 is a representation of the CoolRunner-II programming structure, where the various protect bits are embedded within the main programming array, hidden under layers of metal and distributed throughout the die. There are thousands and thousands of programming bits within the array.

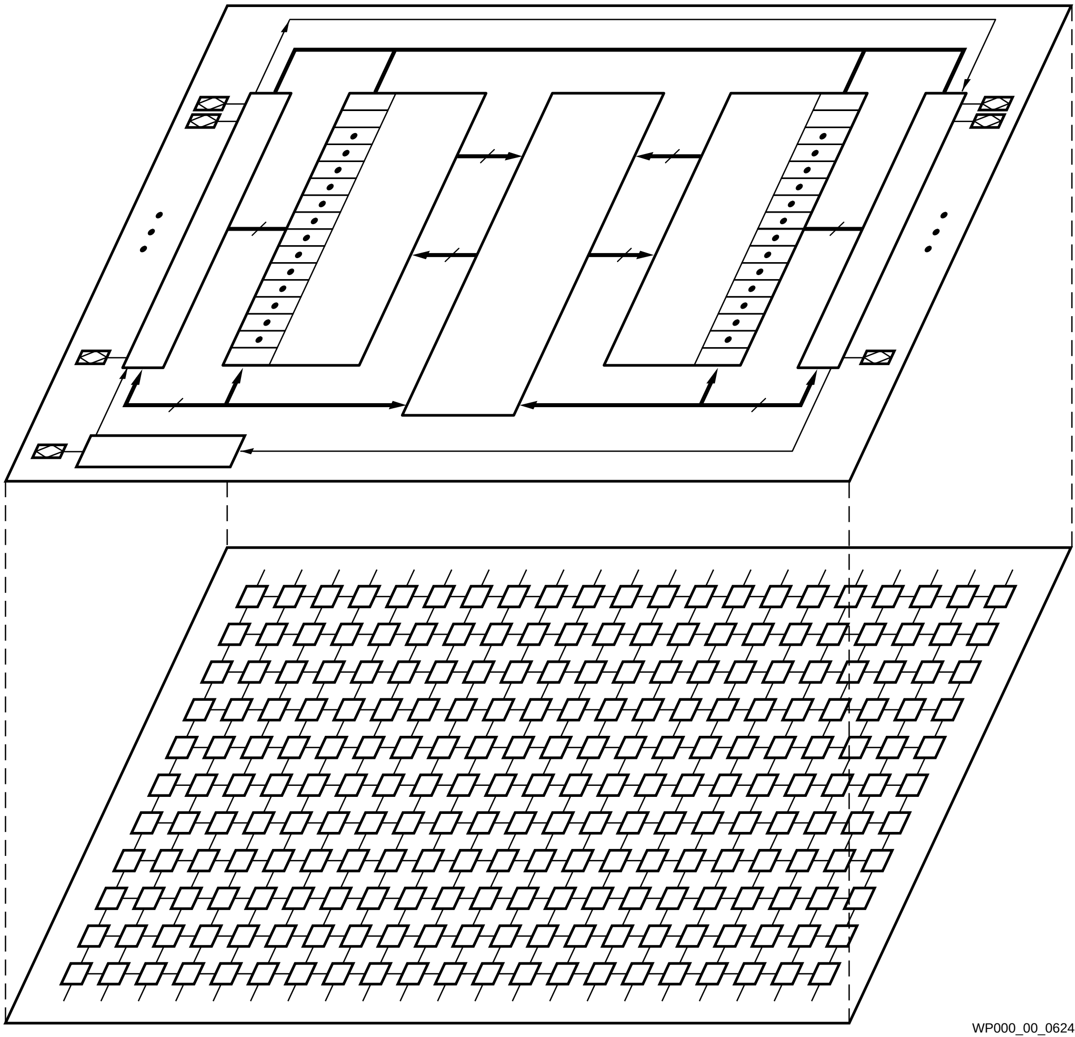

Figure 5: **CoolRunner-II Logic and Programming Array Presentation**

Note: *"R" and "W" are not distinguished among the cells as before.*

There are more CoolRunner-II details available in the following documents:

http://www.xilinx.com/publications/products/cool2/ds_pdf/ds090.pdf

http://www.xilinx.com/publications/products/cool2/apps_pdf/xapp376.pdf

http://www.xilinx.com/publications/products/cool2/apps_pdf/xapp378.pdf

http://www.xilinx.com/publications/products/cool2/apps_pdf/xapp377.pdf

http://www.xilinx.com/publications/products/cool2/apps_pdf/xapp379.pdf

http://www.xilinx.com/publications/products/cool2/apps_pdf/xapp380.pdf

Cracking CoolRunner-II

IBM[3] outlined three categories of security depending on the sophistication and resources of the "cracker". At the first level is the clever outsider, a curious citizen with negligible resources beyond curiosity and energy. This is the largest population, and the people who might crack a single unit. Wholesale piracy is not on their agenda; instead, they simply want access to some capability (free cellphone time, free cable movies, etc.) for personal use. Because there are so many of them, it is important to limit this class, to maintain profits derived from the mass delivery of a controlled access device. Unfortunately, this class has access to the internet, which has become an international resource to broadcast results to other like-minded citizens. These crackers can form a coalition on the internet and share results, so one can become many. Hence, this class is vital to address. They are called Class 1.

Class 2 is similar to Class 1, but a smaller population. They are knowledgeable insiders, and have some resources. Frequently, these are technology students that can get access to "borrowed resources" like an electrical engineering laboratory, with oscilloscopes, microscopes, computers and possibly silicon deprocessing equipment. Again, if they can partially reverse engineer a design, they will do it and broadcast their solution to the world via internet. Usually, their solution involves a level of technology to "do it yourself at home," so their results may be limited to others of Class 2 or higher.

Class 3 is the funded organization. It includes the FBI, CIA, NSA, and any large commercial or national government that can pay the full price for a complete reverse engineering operation. They may even own such an operation. Class 3 can also include organized crime, which may have the financial resources to obtain whatever technical help they need. Talented consultants abound. Full fledged off-shore piracy operations have been discovered and are known to exist. It is believed that anything can be cracked by these folks.

Clearly, Class 1 and Class 2 are key targets to withstand.

As mentioned earlier, read and write protection circuitry, a complex programming and erase protocol, exotic (chip scale / BGA) packages and four to five levels of metal should deter Class 1 and Class 2. Class 3 is another matter.

With enough time and money, bright engineers working for Class 3 could probably reverse engineer a CoolRunner-II part. However, if such an event occurs, these engineers will only have one particular design in hand. CoolRunner-II CPLDs permit a large number of reprogramming cycles. Thus, Class 3 can be dealt with by reprogramming parts to hold a new design.

Nonetheless, let's look at some of today's popular attacks, to see what kinds of tamper resistance CoolRunner-II devices can provide.

Attack Categories

Table 2 outlines a half dozen attacks. We will discuss each to some degree and provide references for deeper study, if desired. The strategy column will be discussed later, also.

Brute Force

Because we are considering general tamper resistance, and not just encryption, "brute force"[1,2] has a very general meaning. For instance, a brute force technique would be one where a "hacker" wants the contents of a chip and is willing to risk the chip's destruction to get it. Applying an external 12V signal to the pins of a 5V part to determine whether that is a super voltage site would be such an action. Grounding pins on a chip to see the resulting changes would be another.

In general, attacks like these are not systematic, and are inconsistent in their resulting gain. Any kind of stress beyond the specified limits of a chip's physical behavior might fit this model. Temperature (hot or cold), high or low voltage, electrostatic discharge (ESD), X-Ray, Single Event Upset (also known as Soft Error Upset or SEU), might all fit this category.

In cryptography, a brute force attack would be defined as the methodical application of a large set of trials for either a key or a password to the system. This is usually done with a computer or an array of FPGAs delivering the patterns at high speed and looking for something to "unlock". Mathematically, the number of applied patterns tracks the length of the key or password, and success frequently occurs at 50% (or less) of the possible patterns.

Table 2: **Some Attacks and Tactics to Reduce Risk**

Category	Typical Attacker	Strategy
Brute Force	Class I, II, III	DataGate / DualEDGE FFs
Power	Class II, III	Bogus Logic / CryptoBLAZE
Glitch	Class I, II, III	DualEDGE / CryptoBLAZE
Tempest	Class II, III	Bogus Logic
Reverse Engineer	Class III	Bogus Logic / Reconfiguration
Toothpick	Class I	Reconfiguration

Power Attack

Although believed to be fairly new, the underlying knowledge[1] for this kind of attack is fairly old. A power attack[7,8] usually targets a microprocessor and develops a time profile of current drawn (ICC). Commercial microprocessors have published instruction sets, and simple experimentation or even technical documentation can determine the current drawn for the execution of a given instruction. To a large degree, witnessing the current profile versus time is like inspecting the flow of instructions versus time. With a small amount of knowledge of a particular cryptographic algorithm's sequence of events, it is possible to determine when the

[1] This author originally worked in computer performance evaluation, with research dating back before the early 1960's. In the 1960's, General Electric produced a computer system (GE 225) which had an ammeter tied into its memory supply. The ammeter was calibrated in MIPs (millions of instructions per second). Hardware analysts could inspect the power drawn at any time and determine the health of the various instructions. Extending this to a time profile is straightforward.

code is inspecting key or password events. Papers have been written in which engineers have divulged the key from the branch behavior of code, simply by watching which bits were accepted (one loop length) and which ones were rejected (another loop length).

One defense against a power attack would be to use a customized processor where the instruction set is not publicly described. More on this later.

Glitch Attacks

Glitch attacks simply violate a chip's specified timing. Overclocking and violating the setup and/or hold time of a flip-flop would be examples of this. By identifying the guaranteed specifications of a chip (microprocessor, memory, CPLD, FPGA, etc.), and delivering signals that violate those specifications, it is possible to discover undocumented, "robust" behavior. In fact, most manufacturers do not know how much they are exposing in this area.

When integrated products are developed, the designers and test personnel work from specifications. Those specifications are ultimately reflected in the published datasheet. Most, if not all chip makers, characterize their product's behavior for the datasheet, with a small amount of additional slack called "guard band." Guard band covers timing specifications, voltage specifications and temperature specifications.

Most gltich-attack discoveries occur beyond both the specified behavior and the guard band behavior. By violating a timing specification, it is possible to get data delivered to the chip's pins. Knowing the underlying architecture helps, but with this information, it is possible to devise multiple experiments to extract internal state information from a chip. The extracted information may or may not bear useful information for the hacker, so long time periods may be needed to progress.

Tempest Attacks

Tempest attacks[10,11,12] go back to World War I. Watching the emission behavior of electrical equipment has been standard fare for the NSA, CIA and FBI, as well as for their international counterparts. (In the novel *Cryptonomicon*, Neil Stephenson romanticized "Van Eyck Phreaking" as a means of reading back the text contents of a laptop computer screen. Mr. Van Eyck managed to enlighten a number of people by demonstrating this and documenting it in his IEEE paper.) Similar to the Power Attack, the Tempest Attack (also known as EMF sideband attack) will divulge distinctive electromagnetic field patterns that can be correlated with data transactions. Methods of controlling the EMF radiation are the domain of tamper resistance.

Reverse Engineering

Reverse engineering[4,5,14,15,16,17,18] may be the ultimate attack. The same technology available to semiconductor manufacturers for creating and testing ICs can be used to discover their inner sanctums, when in the hands of the "reverse engineer." The reverse engineering business is substantial. The ChipWorks and Semiconductor Insights make a livelihood of it. Others exist.

Equipment manufacturers for engineering chips also sell to reverse engineers. As a rule, the reversal process isn't cheap. Today's semiconductor processes are sub-micron, with multiple metal layers (4+). To extract the design involves recording the outer layer metal connections, then careful removal to inner ones, recording them, and so forth down to the substrate. Oxide layers, polysilicon paths and vias must all be recorded. Then, the substrate will need cleaving with successive passes of a focused ion beam (FIB). Ultimately, the process will be revealed, and all the connections —

when reconstructed — will divulge the underlying circuits. Although the professionals use automated computer imaging and registration, the detail required demands substantial human interaction and interpretation. The process is far from being “push button.”

Knowing the underlying circuits is a big step, but not the only one. For PLDs, reverse engineers must also know the internal pattern in order to figure out the end-design function. But there are ways to get this information. A number of techniques, many of which were originally developed at Sandia Laboratories, are available for reading the backside of a die. Optometrix, of Renton, Washington, is one commercial provider of backside imaging equipment that can be used to inspect and discover circuit information from chips, using lasers and other optical equipment.

Toothpick Attack

"Toothpick attack" is an invented term for that unknown attack, the attack-yet-to-come. In May of 2002, Ross Anderson and Sergei Skorobogatov published an attack called “Optical Fault Induction Attacks” where they got a RAM cell to switch by shining intense light from a camera flash bulb on it. Imagine that — attack via a light bulb.

A potential “toothpick” attack on an IC might be using a radiation source to upset memory cells in what is known as Soft Error Upset (SEU). It is known that volatile storage cells can be switched by the insertion of cosmic rays and alpha particles, so introduction of such a source is a potential risk.

On a more whimsical note, an imaginative hacker might believe that a blank PLD might weigh less than a programmed one. With this line of thought, a weighing scale might be an attack that could divulge — to some degree — the internal condition of a programmed versus blank PLD. Heads the world over are still being scratching over this idea.

It is difficult for IC makers to anticipate the myriad ways that attackers will approach their chips, and attack equipment may range from bubble gum and tinfoil to electric hair dryers and, yes, toothpicks. However, at least one hacker has been quoted as insisting that security would be dramatically improved if controlled-access equipment manufacturers simply introduced frequent changes. That way, anything cracked today could be fixed by later today.[1]

Market Needs

The needs of many distinct markets are diverse. For instance, cellphones, PDAs, laptop computers, digital games and set top boxes all have a need for security. Cellphones may need encryption (comp128) over the wireless channel as well as theft deterrence. PDAs need password-protected memory storage as well as wireless encryption and theft deterrence. Laptop computers are similar to PDAs, with the added inherent risks of the 802.11 wireless channel issues. Digital games primarily need design protection, thwarting read-back, but they also need copy protection of the game sources. Set top boxes primarily need encryption and design protection. Currently, each market approaches their need in different ways.

Table 3 shows some of the various encryption needs for various markets. Theft deterrence and information protection are also factors, not shown. It is not hard to envision computer gamers playing over the internet at one level of competition, while

[1] This particular reference is to hacking a set top box decoder, but other notable recent “hacks” have occurred. Cracking RC4 encryption for 802.11b wireless is one example and DES is another.

hackers attempt to crack their encryption — to change the tide of the game — as they operate at another level of competition!

Table 3: **Encryption Needs for Various Markets**

Market	Current Encryption	Status	Future Direction
Cellphones (GSM)	A3, A5, A8, comp128	All cracked; changing	ECC; Kasumi
Digital Cameras	Proprietary; n/a	Not adopted; still vs. motion	ECC
Games			
XBOX	Proprietary	Cracked	Internet capable
Playstation	Proprietary	Cracked	Internet capable
Game Cube	Proprietary	?	Internet capable
Professional Internet	3DES, SSL, SHA-1, MD5	Evolving	AES
Personal Internet	DES, 3DES	evolving	AES
Set-Top Boxes	Proprietary	Multiple; some cracked	Proprietary; AES, 3DES
PDA	N/A	In development	ECC likely
802.11 (wireless laptop)	RC4, WTLS	RC4 Cracked	ECC likely; AES

Simple Design Protection

CoolRunner-II provides substantial design protection for Class 1 and 2 attackers. The read protect methods block all but the Class 3 attackers.

As mentioned earlier, the outlay for reverse engineering is substantial. It is assumed that moderate to large chips will cost in excess of $250,000 and 12 weeks of time, or more.

Raising the Bar

There are some actions that can be taken to make it harder for hackers to crack a system, and with programmable logic, they may be "free." Let's consider some of them. Note that "Bogus Logic", "EMM", and "Lockout" assume the existence of small scraps of remaining logic within a given design that are used to create mystery circuits.

Bogus Logic

Let's assume that a Class 3 reverse engineering has occurred, and that the reverse engineers successfuly discovered the internal program pattern and reconstructed the design equations of the CPLD. That set of equations is relevant in the context of the board to which it is attached. In some cases, the CPLD might be seen to drive memory address and data lines; in other cases, it may be attached to control signals on processors or ASICs. The reverse engineers will have a harder time if it is attached to an ASIC because the ASIC will have an unknown functionality and require an additional expenditure of funds. Many ASICs have unused pins, to which bogus logic within the CPLD could be attached. This adds to the confusion. Going a step further, it would be possible to include bogus logic within the ASIC itself, also. If something like encryption is going on inside either chip (CPLD or ASIC), these connections could be included to confuse that fact. Any action that uses scrap chip area (either CPLD or ASIC) would complicate the reverse engineering process, and increase its expense.

EMM

Electromagnetic masking can help reduce a side channel Tempest attack. In this situation, scrap logic is included to create a whole set of high speed oscillators that sit among the functional circuits on the CPLD. CoolRunner-II CPLDs can build up oscillators that run at 300+ MHz using natural delay connections within the parts. The number included in a design depends on the remaining available logic after the functionality is achieved. This raises the baseline power consumed, but the overall amount will be well below that of other CPLDs doing the same thing.

Lockout

DataGATE is a CoolRunner-II feature that could be thought of as a “tri-state” input. All I/O pins on CoolRunner-II parts greater than 64 macrocells have this capability. If a brute force key attack is sensed, it is possible to block input pin activity by asserting DataGATE.

To explain further: brute force key attacks are usually recognized by high speed successive key trials. If successive trials are perceived (using extra logic), a simple action of blocking inputs could occur, until an overriding reset happens. The overriding reset would be something like erasing the part, power cycling the part or issuing a system reset. These actions dramatically increase the time between key trials, so they serve to hamper design-cracking success.

Cryptography

Most power attacks have been on small, 8-bit microcontrollers with well known instruction sets. It is possible to make a microcontroller from a CPLD. A version of popular Xilinx PicoBlaze™ processor has been placed into an XC2C256, with minor edits. This controller is called CoolBLAZE. Another version, with a selected instruction set for encryption/decryption activities, could also be placed into a 256-macrocell part. This would let designers create an appropriate instruction set, and add in muddling actions like double data-rate clocking. This could occur on some instructions and not others. With advanced planning, multiple versions of a given design could be created and periodically reprogrammed to have different power and timing behavior. Adding EX-OR keys to the instruction and data ports would mean “clear bits” were never present in external EPROM and memory chips. CoolRunner-II allows for this kind of functionality as well as more advanced options.

On the Fly Reconfiguration (OFR)

CoolRunner-II CPLDs support On the Fly Reconfiguration (OTFR). This means that while the CPLD is operating, it is possible to download a new pattern into the part. At the delivery of the appropriate “reload” command, the new pattern takes over within a hundred microseconds. The nonvolatile technology permits this to happen at least a thousand times. It would be possible to reload the pattern every day, for years. For the truly paranoid, this could occur multiple times daily.

Theft Deterrence

Theft deterrence is typically approached by making a stolen item virtually useless when stolen, making the payoff to the thief worthless. Depending on the item stolen, the act of stealing can also trigger apprehension of the thief — as with cellphones and possibly automotive telematics. The standard trick is to require some action be taken by the owner (password, biometric, etc.) that enables the target product to operate. If that action is not taken, the target product becomes useless. To that end, it is vital that the deterring technology be deeply involved in a *mission critical* aspect of the target product operation. Typically, this has involved the operation of a keyboard, mouse or disk. If a password does not match the copy held in the CoolRunner-II CPLD, then, for instance, a keyboard won't work.

There are many ways to achieve this behavior, so another approach would be to change behavior periodically with reprogrammability. Backtracking to the comment on apprehending the thief, a cellphone can relay the successive cell registrations to the authorities as the phone migrates around. An automobile using its cellphone and GPS can actually automatically relay geographical coordinates of the vehicle!

Encryption

Cryptographic design is primarily boolean-based with regard to primitive operations. It typically is the result of "confusion" and "diffusion". To that end, hardware support for encryption and decryption is seldom arithmetic in the standard sense. Typical operations are Exclusive Or, shifting, rotating, byte-wapping, and bit-manipulation applied multiple times using stored and expanded keys. Those operations are easily implemented within the framework of programmable logic, and both state machine solutions as well as custom microcontroller solutions are easy to build and hard to crack. Just one look at the Advanced Encryption Standard (AES) at the **NIST website** will show the manipulations needed to encrypt and decrypt blocks of data.

Reprogrammability

Set top boxes are one market where reprogrammability has become a mainstay of the technology. Being able to enable and disable capabilities from the service provider permits time limited access to various programming material on a "pay per view" basis. Set top box deployment has proven the value and validity of the reprogrammable model in delivering service to those that pay for it. If reprogrammability is clearly thought-through in the design process, it can be very effective. An example where this might have been beneficial was the recent RC4 cracking for wireless 802.11b. If designers had implemented the solution in programmable logic, the design could be altered after discovery of weakness in the algorithm.

Comments on other Technologies

SRAM

Since Xilinx was founded, its premier technology has been SRAM-based FPGAs. To that end, it is well known that the output of the configuration PROM could be intercepted and valuable design information compromised. Knowing users need added security, Xilinx Virtex-II FPGAs offer a battery backed up technology that maintains cryptographic keys and triple DES decryption within the part. In this way, the external configuration holds a fully encrypted bit-stream that is useless to a thief. Only after bringing it within the FPGA does it decrypt into the functional bit-stream, and this is blocked from external inspection by protection logic. To date, the standard DES algorithm has been cracked, but triple DES has not.

Two things should be noted. First, there exists an international specification called the "Common Criteria" (FIPS 1402), which has published guidelines for cryptographic modules. A key capability, which is infrequently implemented, is the ability to "zero out" such a module's internal memory. SRAM-based programmable logic can implement this capability.

The second note to remember regarding SRAM-based FPGAs is that, due to its high speed, high density, and reprogrammability, the SRAM FPGA is the technology of choice for cryptanalysts to use for cracking algorithms. In fact, the study sponsored by the Electronic Frontier Foundation that cracked DES used an array of Xilinx FPGAs. Xilinx has apparently been involved in the security business — one way or another — for quite some time.

Antifuse

Antifuse technology has long claimed an advantage with regard to security implementation. It may have been an oversight to let this claim go unchallenged for so long. The standard argument is that antifuses are tiny, there are vast numbers of them in FPGA parts, they are difficult to inspect, and they are impossible to erase. In fact, the argument lately has taken the position that an antifuse-based FPGA is harder to reverse engineer than an ASIC.

As with a cryptographic attack, it is usually more effective to attack the protocol than the algorithm. Let's first review some ideas about antifuses. Today, there are two manufacturers in the lead for antifuse FPGAs: Actel and QuickLogic. The Actel antifuse is called a Plice™ and the QuickLogic antifuse is called a ViaLink™. Both operate along similar lines in that they are coincident-select technology requiring a fairly high current applied through intersecting X-Y metal lines to make the electrical event called "programming." In each case where the Plice or ViaLink is present, the application of sufficient current will substantially reduce the impedance at the intersection. The physics is interesting, but not important. The logic building blocks within each architecture are configured from identical logic that is configured by programming the antifuses. Ultimately, a design becomes a bitstream that corresponds to programmed antifuses.

In order to develop a design, an engineer would use standard design methods (synthesis, schematics, libraries, etc.) and obtain a programming file. It is appropriate that creating the design with the ability to readback the bit-stream is possible, for debug. Then, for security, there is a separate step for programming the read protection bits.

This is a weakness. It is possible to distinguish where on the chip the programming bits are by observing their programming during that time. Current required for an antifuse to change state is typically several milliamps, which makes an observable thermal event. Inexpensive laboratory equipment can deduce the location of the security bits by thermal inspection (infra red, liquid crystal, etc.).

Antifuse sites are located on outer metal layers. This is a second weakness. When programmed, they change impedance from a high impedance (gigaohms) to a low one (ohms). Their location on outer layers makes them available for erasing — once. Erasing an antifuse means that the low impedance need only be increased. Creating an open circuit at the antifuse site will accomplish that. A number of ways exist to do that. Among them are laser, mechanical probing with an inspection station, and focused ion beam (FIB). The best choice is probably FIB, which would also be the most expensive. However, this is one case where the "toothpick" attack is very close to reality.

Unlike nonvolatile EPROM technologies or SRAM, it is not possible to easily reprogram an antifuse product. This rules out recovery from a bad encryption choice (RC4 for 802.11b), or making a new set top box algorithm. Once programmed, its designer can only hope for the best. It would also rule out complete compliance with the Common Criteria methodology for cryptographic modules, unless that is only restricted to embedded SRAM blocks. Clearly, reprogrammability has substantial value.

Microprocessors

Processors have been the choice for most cryptology systems. This is largely due to calculation facility, reprogrammability and support. Microprocessor attacks are well documented. Their primary weakness stems from the wide publication of their instruction sets and electrical properties. With that information, attackers can systematically uncover information that leads to discovery of their internal operation. Microprocessors are also, to some degree, limited in their "Von Neumann" or Harvard architecture (ie, they must process words or bytes). Embedded processors with programmable fabric contained in the same chip definitely have merit for creating strong cryptographic algorithms and fitting the Common Criteria model for secure cryptographic modules. See **Virtex-II Pro™ FPGA product description and user manuals** for more information.

ASICs

ASICs are either gate arrays or standard cells. In either case, they are expensive commitments. They are fast, and can be low power, and tough to reverse engineer, but they can fall prey to Class III reverse engineering attacks. ASICs also suffer from publication of their electrical properties, so could also fall victim to certain timing attacks.

One big problem with ASICs is that if an internal algorithm or protocol is cracked, ASICs can't easily be changed. Having microprocessors embedded into ASICs improves their reprogrammability, but can do so at the expense of adding power and tempest attacks. As with microprocessors mentioned above, it is a two edged dilemma.

Conclusion

This paper has discussed a broad and eclectic range of topics relating to data security and tamper resistance. Its fundamental point, however, is that reprogrammable logic — particularly the CoolRunner-II CPLD — provides for more secure circuitry design than do other possible security solutions. Table 4 summarizes how the standard solutions of ASIC, microprocessor and reprogrammable logic fare with regard to the set of attacks described in Table 2.

Table 4: **Comparison of Design Technologies versus Attacks**

Attack Method	Design Method		
	Microprocessor	ASIC	Reprogrammable PLD
Brute Force	Susceptible	Maybe	Maybe
Power	Susceptible	Maybe	Maybe
Timing	Susceptible	Maybe	Maybe
Tempest	Susceptible	Maybe	Maybe
Reverse Engineering	Susceptible	Susceptible	Susceptible
Toothpick	Maybe	Maybe	Maybe

Reprogrammability keeps attackers at bay by allowing for an astronomical number of possible designs and re-designs on a single part. If attackers can’t directly access a programmed design, they have no choice but to revert to reverse engineering. Once a device has been reverse engineered, an attacker must still find a way to obtain the bit pattern for the design on the part. Changing designs frequently — hourly, if necessary—seriously hampers an attacker’s ability obtain these patterns. Reprogramming is thus a logical and efficient way to plan for future design attacks (of both known and as-yet-unknown methodologies). And Xilinx CoolRunner-II CPLDs make this a viable solution for design-security concerns..

Glossary

AES – acronym for NIST’ Advanced Encryption Algorithm (see also Rijndael)

Attack – the general term used for adverse action taken with controlled access circuits (see tempest, differential)

BGA – ball grid array package

CoolCLOCK – Xilinx term for halving a global clock and locally doubling it at a macrocell. CoolCLOCK lowers power and improves performance.

CPLD – Complex Programmable Logic Device

CRC – Cyclic Redundancy Coding

CSP – Chip Scale Package; very small BGA package

Comp128 – encryption algorithm used in the cellphone world

Connoisseur Coating – commercially available chip / die coating to thwrt inspection

Cryptanalysis – the science / art of cracking ciphers

Cryptography – basically, the science of constructing ciphers

Cryptology – the combination of cryptanalysis and cryptography

DataGATE – Xilinx feature to conditionally block inputs into a CoolRunner-II CPLD

Differential Fault Analysis – comparative cryptanalysis technique for cracking ciphers

Differential Power Analysis – comparative measurement technique for cracking ciphers,

DualEDGE Clock – Xilinx feature where macrocell flip flops are clocked on both edges of an input clock (see CoolCLOCK)

ECC – eliptic curve cryptography; attractive for difficulty to crack and small key lengths

EEPROM – electrically erasable Programmable Read Only Memory

EPROM – early Programmable Read Only Memory (erased with UV light)

FIB – focused ion beam

Flash – EPROM technology that is electrically erasable

FPGA – Field Programmable Gate Array

Glitch – unwanted, out-of-specifcation signal

GPS – Global Positioning System used for navigation systems

GPRS – General Packet Radio Service

GSM – Global System for Mobile Communications (cellphone tecnology)

JTAG – Joint Test Action Group

JEDEC – Joint Electronic Device Engineering Council

Kasumi – block cipher targeted at 3GPP cellphone standard

Linux – Popular version of Unix

LIVA – light induced voltage alteration

MD5 – Message Digest Algorithm for calculating a digital text “figerprint”

Overclocking – a possible timing attack for both microprocessors and state machines; basically, exceeding the recommended clock speed

PalmOS – Operating System used in Palm Pilots and other PDAs

PDA - Personal Digital Assistant

PLD – Programmable Logic Device (see CPLD and FPGA)

PocketPC – Microsoft PDA operating system derived from Windows CE

Protect bits – additional EPROM cells within a PLD that deny reading or writing using the standard protocol

RC4 – stream cipher used on many wireless devices

Read protect – protect bits specifically denying read access

Rijndael – the inventor’s name for the AES encryption standard

SEU – Soft Error Upset; transient switching of an SRAM storage cell

Side Channel – name given to attacks that are indirect; see tempest, differential power analysis

Side Channel EM – an attack using electromagnetic measurement of emitted signals (usually from a microprocessor)

SmartCard – digital technology widely employed in portable systems

SPLD – Simple PLD

Super-voltage – EPROM technique to program or read-back internal bit patterns where a pin receives a voltage beyond the standard logic levels and assumes a different functional identity

Symbion – British pioneering company with PDAs and cellphones

Telematics – the merging of communications and computing technology to provide driver information, communications and entertainment

Tempest – another term for side channel EM; more formally: Telecommunications Electronics Material Protected from Emanating Spurious Transmissions

TIVA – light induced voltage alteration

WTLS – Wireless Transport Layer Security; current standard for wireless data communictions

Write protect – protect bit specifically denying write access to an EPROM

XIVA – advanced induced voltage alteration

X-RAY – technique for inspecting die; same method as in medicine

References

General Security and Encryption

1. R.J. Anderson, Security Engineering – A Guide to Building Dependable Distributed Systems, Wiley 2001.
2. Bruce Schneier, Applied Cryptography, John Wiley & Sons, 1996
3. D. Abraham, G. Dolan, G. Double, J.Stevens, “Transaction Security System”, IBM Systems Journal v. 30, no. 2 (1991), pp 206-229
4. Ross Anderson, Markus Kuhn, Tamper Resistance – A Cautionary Note, USENIX Electronic Commerce Workshop, Oakland, CA., Nov. 18-20, 1996
5. Ross Anderson, Markus Kuhn, Low Cost Attacks on Tamper Resistant Devices, Security Protocols, 5th International Workshop. LNCS, 1361, pp. 125-136, Springer-Verlag, 1997

Attacks

6. J.R. Rao, P. Rohatgi, H. SCherzer, S. Tinguely, Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards, IEEE Sumposium on Security and Privacy, Oakland, CA, May 2002
7. Paul Kocher, Joshua Jaffe, Benjamin Jun, Differential Power Analysis, Crypto 99
8. T. Masserges, E. Dabbish, R. Sloan, Investigations of Power Analysis Attacks on Smartcards, Proceedings of USENIX Workshop on Smartcard Technology, May 1999, pp. 151-161
9. G. Hachez, F. Koeune, J.J. Quisquater, Timing Attack: What Can Be Achieved by a Powerful Adversary?, UCL Crypto Group, **www.dice.ucl.ac.be/crypto**
10. J. Rao, P. Rohatgi, EMpowering Side-Channel Attacks, IBM whitepaper downloaded from: **http://www.research.ibm.com/intsec/emf.html**
11. Wim van Eck, Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?, Computers & Security, 1985 Vol. 4, North Holland, pp. 269-286. Available at: **www.eskimo.com/~joelm/tempestintro.html**.
12. Markus Kuhn, Ross J. Anderson, Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations, Information Hiding 1998, LNCS 1525, pp. 124-142, 1998.
13. Sergei Skorobogatov, Ross Anderson, Optical Fault Induction Attacks, 2002 IEEE Symposium on Security and Privacy, Oakland, CA, May 2002.

Reverse Engineering

14. Simon Blythe, Beatrice Fabroni, Sanjay Lall, Haroon Ahmed, Ugo de Riu, Layout Reconstruction of Complex Silicon Chips, IEEE Journal of Solid-State Circuits, Vol. 28, No. 2, February 1993, pp. 138 – 145.
15. E.I. Cole, J.M. Soden, J.L. Rife, D.L. Barton, C.L. Henderson, Novel Feature Analysis Techniques Using Photon Probing with a Scanning Optical Microscope, whitepaper, Sandia Laboratories, **www.sandia.com**

16. R. Aaron Falk, Backside Thermal Mapping Using Active Laser Probe, Electronic Device Failure Analysis News, May 2000. Reprint available at: **www.optomet.com**.

17. R. Aaron Falk, E.W. Budiarto, Application of Near IR, Phase-Contrast Imaging to Backside Failure Isolation and Analysis, whitepaper, published at www.optomet.com.

18. R. Aaron Falk, Advanced LIVA / TIVA Techniques, whitepaper, published at www.optometrix.com.

Security Improvement

19. Oliver Kommerling, Markus Kuhn, Design Principles for Tamper- Resistant Smartcard Processors, Proceedings of the USENIX Workshop on Smartcard Technology, Chicago, Illinois, USA, May 10-11, 1999, USENIX Association, pp. 9-20.

20. S. Moore, R. Anderson, P. Cunningham, R. Mullins, G. Taylor, Improving Smart Card Security using Self-timed Circuits, Async 2002]

21. U.S. Patent #5,258,334,Inventor: Leon Lantz, II, Assignee: U.S. Government as by the Director, National Security Agency

Internet Resources

http://theory.lcs.mit.edu/~rivest/crypto-security.html

www.dice.ucl.ac.be/crypto

http://www.counterpane.com

http://www.cryptography.com/

http://www.cacr.math.uwaterloo.ca/hac/

http://www.cl.cam.ac.uk/users/rja14/

http://csrc.nist.gov/encryption/aes/

http://www.nsa.gov

http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

www.eskimo.com/~joelm/tempestintro.html

http://www.protonworld.com

http://www.cryptomathic.com/

http://www.certicom.com/

http://www.cs.berkeley.edu/~daw/crypto.html

http://www.gsmworld.com/index.shtml

http://www.geocities.com/ResearchTriangle/Lab/1578/gsm.htm

http://www.geocities.com/ResearchTriangle/Lab/1578/smart.htm

http://www.smartcardbasics.com/

http://www.smartcard.co.uk/

http://www.aces.att.com/glossary/encrypti.htm

http://www-3.ibm.com/security/index.shtml

http://www.cryptography.com/

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/22/02	1.0	Initial Xilinx release.
11/05/02	1.1	Minor revisions.
11/19/02	1.2	Minor revision