Cryptanalysis of the Full AES Using GPU-Like Special-Purpose Hardware

Alex Biryukov and Johann Großschädl
University of Luxembourg
Laboratory of Algorithmics, Cryptology and Security (LACS)
6, rue Richard Coudenhove-Kalergi, L–1359 Luxembourg
{alex.biryukov,johann.groszschaedl}@uni.lu

Abstract: The block cipher Rijndael has undergone more than ten years of extensive cryptanalysis since its submission as a candidate for the Advanced Encryption Standard (AES) in April 1998. To date, most of the publicly-known cryptanalytic results are based on reduced-round variants of the AES (respectively Rijndael) algorithm. Among the few exceptions that target the full AES are the Related-Key Cryptanalysis (RKC) introduced at ASIACRYPT 2009 and attacks exploiting Time-Memory-Key (TMK) trade-offs such as demonstrated at SAC 2005. However, all these attacks are generally considered infeasible in practice due to their high complexity (i.e. $2^{99.5}$ AES operations for RKC, 2^{80} for TMK). In this paper, we evaluate the cost of cryptanalytic attacks on the full AES when using special-purpose hardware in the form of multi-core AES processors that are designed in a similar way as modern Graphics Processing Units (GPUs) such as the NVIDIA GT200b. Using today's VLSI technology would allow for the implementation of a GPU-like processor reaching a throughput of up to 10^{12} AES operations per second. An organization able to spend one trillion US$ for designing and building a supercomputer based on such processors could theoretically break the full AES in a time frame of as little as one year when using RKC, or in merely one month when performing a TMK attack. We also analyze different time-cost trade-offs and assess the implications of progress in VLSI technology under the assumption that Moore's law will continue to hold for the next ten years. These assessments raise some concerns about the long-term security of the AES.

Keywords: Advanced Encryption Standard, Cryptanalysis, Cryptanalytic Hardware, Graphics Processing Unit, Performance and Energy Evaluation.

1 Introduction

Research on special-purpose hardware for cryptanalysis has a rich and illustrious history stretching back almost hundred years [23, 37]. In 1938, Polish mathematicians led by Marian Rejewski constructed the *Bomba Kryptologiczna* (or Bomba for short), an electromechanical machine that allowed them to break the German Enigma cipher by exhaustively trying all 17,576 rotor positions. This success was expanded by British cryptographers (most notably Alan Turing and Gordon Welchman), who designed ingenious cipher-breaking machines enabling the Allied forces to read Enigma-encrypted messages during World War II [40]. A parallel effort of cryptanalysis of another German cipher, the Lorenz SZ40/42, resulted in the construction of *Colossus*, one of the world's first programmable computers. Colossus contained 1,500 thermionic valves (vacuum tubes) and was able to process 5,000 characters per second.

In the 1980s, Pomerance et al [33] designed a hardware architecture called *Quasimodo* for factoring large integers using the quadratic sieve algorithm. Quasimodo was actually built but never functioned properly. The *DES Cracker* (also known as *Deep Crack*) is a parallel key-search machine developed by the Electronic Frontier Foundation (EFF) in the late 1990s with an overall budget of just 210,000 US$ [13]. Deep Crack consists of about 1,500 custom chips and needs at most nine days to find a 56-bit DES key by "brute force." Also in the late 1990s, Shamir [35] proposed *TWINKLE*, an electro-optical device for performing the sieving step of the Number Field Sieve (NFS) algorithm. He estimated that a single chip of the size of a 6-inch GaAs wafer would allow one to factor a 512-bit number in reasonable time and, as a consequence, break 512-bit RSA keys. *TWIRL*, a successor of TWINKLE, could reduce the total sieving time for 512-bit numbers to less than ten minutes [36]. Even though both TWINKLE and TWIRL are purely hypothetical devices that were never built due to technical issues (e.g. too large chip area) and high cost, they received considerable attention in the cryptographic community and initiated a slew of follow-up research [15, 16]. Recent attempts to implement cryptanalytic devices mainly use FPGAs as underlying hardware platform [28, 29]. A typical example is *COPACOBANA*, an FPGA-based parallel machine for cryptanalysis that was successful in breaking some symmetric ciphers with a key size of up to 64 bits, e.g. KeeLoq, DES, and A5/1 [25, 20].

The advent of powerful yet inexpensive multi-core processors, especially Graphics Processing Units (GPUs), has triggered a large body of research to analyze their capabilities for cryptanalytic purposes [19]. GPUs are particularly well suited for the implementation of multiplication-intensive cryptanalytic algorithms that can be mapped efficiently onto a highly parallel architecture [2, 3]. Other cryptosystems performing mainly logical operations have also been successfully attacked on various high-end graphics platforms [11, 30]. While such software-based cryptanalysis on multi-core processors is relatively easy to implement, it does not reflect the potential of custom hardware, simply because GPUs are optimized for graphics (or multimedia) processing and not for cryptanalysis. The same holds, although to a lesser extent, for FPGA-based cryptanalytic hardware: An ASIC designed and optimized from the ground up to break a certain cryptosystem can reach higher clock frequencies (and consumes less power) than an FPGA implementing the same functionality.

In this paper, we study the cost of a hardware-based attack on AES-128 and AES-256 assuming that the attack complexity is bounded by 2^{100} computations. We are motivated by the recent progress in the cryptanalysis of AES-256 [6], which has provided an attack with a time and data complexity of $2^{99.5}$ and a memory complexity of 2^{78} in the related-key scenario (we will call this attack RKC, an abbreviation for *Related-Key Cryptanalysis*). While it is clear that implementing this highly specific attack has little practical impact due to its reliance on related keys, we believe that a more threatening secret-key attack of complexity 2^{100} would not be much different in terms of hardware implementation cost, and thus we can use the existing attack a case study. We also notice that the same hardware can, to a large extent, be re-used for a *Time-Memory-Key* attack (TMK, also known as multiple target attack) on AES-128 [8]. In such an attack it is assumed that a fixed plaintext is encrypted under many different secret keys, and the goal of the attacker is to find one of these keys. Given e.g. 2^{32} targets, the TMK attack has a one-time pre-computation complexity of 2^{96}, after which each new secret key can be found with a time complexity of 2^{80} and a memory complexity of 2^{56}.

We evaluate the cost of these two attacks on the full AES assuming special-purpose hardware in the form of multi-core AES processors that are realized in a similar way as modern Graphics Processing Units (GPUs) such as the NVIDIA GT200b [31]. Using state-of-the-art VLSI technology, it is possible to implement a GPU-like processor reaching a throughput of up to 10^{12} AES operations per second at a cost of only about 30 US$. An organization able to spend one trillion US$ (which is roughly a single-year defense budget of the US [38]) for designing and building a large-scale supercomputer based on such optimized processors could theoretically break the full 256-bit AES in a time frame of as little as one year when using RKC or another attack of similar complexity. One tenth of this budget (100 billion

US$) would be sufficient to mount a TMK attack on 2^{32} targets. This attack requires a pre-computation phase of one year done once, after which a new key can be recovered every 2^{80} AES operations, or every eight minutes, on this "smaller" supercomputer.

Our contribution in this paper is twofold. First, we present the architecture of a GPU-like multi-core AES processor optimized for RKC and TMK attacks, and discuss how the requirements for this special processor differ from that of standard high-speed AES hardware with respect to pipelining options, support for key agility, and memory bandwidth. Second, we analyze the production cost and performance of large-scale cryptanalytic hardware built of GPU-like processors, and estimate the running time of the RKC and TMK attack on the full AES. More precisely, we try to provide a realistic lower bound for the time and energy required to perform these attacks when using a cryptanalytic supercomputer consisting of 10^{10} optimized AES processors. This supercomputer, which we call *CAESAR* ("Cryptanalytic AES ARchitecture"), is a hypothetical machine (like TWINKLE and TWIRL) since an actual implementation of the proposed GPU-like AES processor is beyond our resources. However, we point out that, unlike TWINKLE, our AES processor can be implemented with present-day VLSI technology since its silicon area, clock frequency, and power consumption are quite similar to that of a commodity GPU. We also analyze different time-cost trade-offs and try to assess the implications of progress in VLSI technology under the assumption that Moore's law will continue to hold for the next ten years.

2 Cryptanalytic Attacks on AES

The cipher Rijndael has been a subject of intensive cryptanalysis already during the AES-competition and in the past ten years after it has been adopted as a new encryption standard by NIST. In this section we highlight the main advances in cryptanalysis of the AES and describe two of these approaches in more detail.

The first cryptanalysis of 6-round AES was provided by its designers [12], who have shown how to break six rounds of AES-128 using a *multiset attack* (historically called Square attack). During the AES competition two other attacks were described. The first was the *partial-sum* approach [14], which used the same ideas as the designer's attack, but managed to reduce the time complexity for a 6-round attack from 2^{70} to 2^{44}. The second was a novel *functional-collision* technique [17], which also falls into a class of multiset attacks and is capable of breaking up to seven rounds of AES-128 marginally faster than an exhaustive key search. Rijndael was announced as a NIST standard in November 2001. In the following eight years there were many attempts of cryptanalysis (boomerang attack, impossible differentials, algebraic attack, various related-key attacks); however, the progress was very slow and mainly restricted to related-key attacks on 192 and 256-bit AES. The best of these attacks reached seven rounds (out of ten) for AES-128, and ten (out of 12 or 14) rounds for AES-192 and AES-256, all with complexities close to that of an exhaustive search.

In 2009, new related-key and open-key attacks capable of breaking the *full* AES-192 and AES-256 were discovered. The attack on 256-bit AES initially had a complexity of 2^{96} data and time and worked for one out of 2^{35} keys, i.e. it had a total complexity of 2^{131} steps [7]. This attack used simple key relations of the form $K' = K \oplus C$, where K, K' are two unknown but related keys and C is a constant chosen by the attacker. In Section 2 of the same paper a practical chosen-key distinguisher for AES-256 was demonstrated. Later in the same year, the attack on AES-256 was significantly improved to run with a time and data complexity of $2^{99.5}$ using a boomerang related-key attack and the *related subkey* setting in which $K' = F^{-1}(F(K) \oplus C) = R_C(K)$, where function F is one round of the AES-256 key-schedule and C is a constant chosen by the attacker [6]. Even though these attacks reveal a certain structural weakness of the key-schedule of AES-192 and AES-256, they are of no immediate threat in practice due to two

Table 1: Summary of attacks on AES.

Cipher	Attack/Result	Rounds	Data	Workload	Memory	Reference
AES-128	Multiset	6	2^{33}	2^{70}	2^{32}	[12]
	Collisions	7	2^{32}	2^{128}	2^{80}	[17]
	Partial sum	6	2^{35}	2^{44}	2^{32}	[14]
	Partial sum	7	$2^{128}-2^{119}$	2^{120}	2^{32}	[14]
	Boomerang	6	2^{71}	2^{71}	2^{33}	[5]
	Impossible diff.	7	$2^{112.2}$	$2^{117.2}$	2^{109}	[27]
	Boomerang - RK	7	2^{97}	2^{97}	2^{34}	[9]
AES-192	Rectangle - RK	9	2^{64}	2^{143}	?	[18]
	Rectangle - RK	10	2^{125}	2^{182}	?	[24]
	Boomerang - RK	12	2^{116}	2^{169}	2^{145}	[9]
AES-256	Rectangle - RK	10	2^{114}	2^{173}	?	[4, 24]
	Subkey Diff.	10	2^{48}	2^{49}	2^{33}	
	Differential - RK[a]	14	2^{131}	2^{131}	2^{65}	[7]
	Boomerang - RK	14	$2^{99.5}$	$2^{99.5}$	2^{78}	[6]

[a] The attack works for a weak key class, and the workload includes the effort to find related keys from the class.

factors: First, the related-key scenario is a very strong attacker model and, second, the attack requires a huge amount of both time and data. Nonetheless, they are the first attacks on the AES that have broken through the psychological 2^{100} complexity barrier, which may motivate cryptanalysts to pay increased attention to the AES in the coming years. A summary of attacks on AES is given in Table 1.

A completely different approach to cryptanalysis of block ciphers is possible by exploiting a Time-Memory-Key (TMK) trade-off. Such trade-offs can be used to invert any one-way function. The original Time-Memory trade-off was introduced by Hellman [21] and required a single pre-computation equal in complexity to the full exhaustive search. Later, Biryukov and Shamir [10] presented a Time-Memory-Data trade-off as a generalization of Hellman's method, which added more flexibility by introducing an extra *data* parameter into the trade-off equations and, as a consequence, allowed to considerably reduce the heavy pre-computation phase of the original trade-off. In [8], an application of trade-off attacks to multiple target attacks on block ciphers has been studied. The introduced TMK attack requires a fixed plaintext to be encrypted under D unknown secret keys and the goal of the attacker is to find one of these keys. It was shown that, in this scenario, it is impossible for a cipher with a keylength of n bits to stand to its complexity guarantee of 2^n since any cipher can be broken in time $O(2^n/D)$ and with considerably less memory (i.e. a lot better than what a straightforward birthday trade-off would suggest).

The following two subsections summarize the details of the related-key and trade-off attack on the full AES that are needed to understand our estimates for a large-scale hardware implementation of these attacks. The reader is referred to the original papers [6, 8] for a full description.

2.1 Summary of the Related-Key Boomerang Attack on AES-256

This type of attack is embedded in a scenario of four secret related keys; it needs $2^{99.5}$ time and data to find these keys. The attack works as follows. Repeat the following steps $2^{25.5}$ times:

1. Prepare a structure with all possible values in column 0, row 1, and the main diagonal (nine bytes in total), and constant values in the other bytes (2^{72} plaintexts).

2. Encrypt it on keys K_A and K_B and keep the resulting sets S_A and S_B in memory.

3. XOR Δ_C to all the ciphertexts in S_A and decrypt the resulting ciphertexts with K_C. Denote the new set of plaintexts by S_C.
4. Repeat previous step for the set S_B and the key K_D. Denote the set of plaintexts by S_D.
5. Compose from S_C and S_D all the possible pairs of plaintexts which are equal in certain 56 bits.
6. For every remaining pair check if the difference in $p_{i,0}, i > 1$ is equal on both sides of the boomerang quartet (16-bit filter).
7. Filter out the quartets whose difference can not be produced by active S-boxes in the first round (one-bit filter per S-box per key pair) and active S-boxes in the key schedule (one-bit filter per S-box), which is a $2 \cdot 2 + 2 = 6$-bit filter.
8. Gradually recover key values and differences simultaneously filtering out the wrong quartets.

The time and memory complexity of this attack can be evaluated as follows. From 2^{72} texts per structure we could compose 2^{144} ordered pairs, of which $2^{144-8\cdot 9} = 2^{72}$ pairs pass the first round. Thus, we expect one right quartet per $2^{96-72} = 2^{24}$ structures, and three right quartets out of $2^{25.5}$ structures. Let us now calculate the number of noisy quartets. Roughly $2^{144-56-16} = 2^{72}$ pairs come out of step 6. The next step applies a 6-bit filter, which means we get $2^{72+25.5-6} = 2^{91.5}$ candidate quartets in total. Note that we still *do not store these quartets*. For each quartet we check what key candidates it proposes on both sides of the boomerang; this process allows us to gradually reduce the number of candidate quartets to $2^{72.5}$ as shown in [6]. Each candidate quartet proposes 2^6 candidates for 11 key bytes for each of the four related keys. However, these bytes are strongly related, and so the number of independent key bytes on which the voting is performed is significantly smaller than 11×4, namely about 15. The probability that three wrong quartets propose the same candidates does not exceed 2^{-80}. Thus, it can be estimated that the complexity of the filtering step is 2^{78} in time and memory. In total, we recover $3 \cdot 7 + 8 \cdot 8 = 85$ bits of K_A (and 85 bits of K_C) with $2^{99.5}$ data and time and 2^{78} memory. The rest of the key bits can be recovered with negligible complexity compared to the main phase of the attack.

2.2 Summary of a Time-Memory-Key Attack on AES-128

Hellman's trade-off [21] is a well-known way to invert arbitrary one-way functions. The main idea is to iterate a one-way function on its own output, thereby computing a chain, and then to discard all the computed points keeping only the start-end point pairs in memory, sorted by the end points. During the attack, the attacker iterates the function starting from the given ciphertext and checks at each step if he has hit one of the end points[1]. He then picks the corresponding starting point and iterating from it finds the pre-image from the function. In the context of block ciphers with reasonably long keys this attack is typically not considered to be of a threat since the pre-computation time needed to build the tables containing all start-end points is the same as the exhaustive search of the key.

The main idea of the Time-Memory-Key (TMK) trade-off is that we can cover only a fraction of the search space if multiple targets are available. This is typically the case when messages (or files) with the same constant header are encrypted under different keys. Let us denote by $N = 2^n$ the size of the search space ($n = 128$ for AES-128). Given encryptions of a fixed plaintext under D_k different keys, we need to cover only N/D_k points of the space. Hence, we will use t/D_k tables instead of t, which means the

[1] The need to perform memory access at each iteration can be avoided by using the idea of so-called distinguished points, i.e. the attacker does not perform memory access until a point that he obtained has some distinguishing feature, e.g. l leading zeroes.

Table 2: Comparison of TMD attacks on various ciphers.

Cipher	Key size	Keys (Data)	Time	Memory	Preprocessing
DES	56	2^{14}	2^{28}	2^{28}	2^{42}
Triple-DES	168	2^{42}	2^{84}	2^{84}	2^{126}
Skipjack	80	2^{32}	2^{32}	2^{32}	2^{48}
AES	128	2^{32}	2^{80}	2^{56}	2^{96}
AES	192	2^{48}	2^{96}	2^{96}	2^{144}
AES	256	2^{85}	2^{170}	2^{85}	2^{170}
Any cipher	k	$2^{k/4}$	$2^{k/2}$	$2^{k/2}$	$2^{3k/4}$
Any cipher	k	$2^{k/3}$	$2^{2k/3}$	$2^{k/3}$	$2^{2k/3}$

Table 3: Trade-off attacks on 128-bit key AES (and any other 128-bit key cipher).

Attack	Data Type	Keys (Data)	Time	Memory	Preprocessing
BS TMD	FKP	2^{8}	2^{120}	2^{60}	2^{120}
BS TMD	FKP	2^{20}	2^{100}	2^{58}	2^{108}
BS TMD	FKP	2^{32}	2^{80}	2^{56}	2^{96}
BS TMD	FKP	2^{43}	2^{84}	2^{43}	2^{85}

memory requirements drop to $M = mt/D_k$ (here m is the number of start-end points in one Hellman's table). The time requirements $T = t/D_k \cdot t \cdot D_k = t^2$ are the same as in Hellman's original trade-off (we have less tables to check, but for more data points). Finally, the matrix stopping rule is $N = mt^2$, which is the condition to minimize the "waste" in the matrix coverage due to birthday paradox effects. Using this matrix stopping rule and eliminating the parameters m and t, we get the trade-off formula

$$N^2 = T(MD_k)^2.$$

Taking AES with 128-bit key as example and assuming an attacker given 2^{32} encryptions of a fixed text under different unknown keys, he can recover one of these keys after a single pre-processing of 2^{96} steps and using 2^{56} memory entries for table storage (2^{61} bytes) and 2^{80} time for the actual key-search.

Another important observation is that the attack is not exactly a chosen plaintext attack since the specific value of the fixed plaintext is irrelevant. Thus, in order to obtain an attack faster than exhaustive search, the attacker should first try to find out which plaintext is the most frequently used in the target application, collect the data for various keys, and then perform the actual attack. The results summarized in Table 2 compare favorably with the best attacks on such ciphers as DES, Triple-DES, Skipjack, and AES. Moreover, the scenario of TMD attacks is much more practical than that of related-key attacks. We provide several trade-off points for AES-128 in Table 3.

3 GPU-Like AES Processor for Cryptanalysis

In this section we introduce the architecture and functionality of a high-speed AES processor optimized for cryptanalytic attacks following the TMK trade-off and the RKC method as discussed above. From an architectural point of view, this AES processor is basically a homogenous multi-core system with local cache and shared memory, similar to present-day Graphics Processing Units (GPUs) such as NVIDIA's GT200b [31]. It consists of a large number of programmable high-speed AES engines that can work in parallel, controlled by a small number of general-purpose processing units. The AES engines are optimized for the requirements and characteristics of the cryptanalytic attacks described in Section 2.

The AES engines of our processor differ greatly from a "conventional" high-speed AES hardware implementation such as the one that Intel recently announced to include in the forthcoming Westmere micro-architecture. For example, in our processor the plaintexts to be encrypted do not need to be loaded from "outside," but are either constant (in case of the RKC attack) or can be easily generated on chip (in case of the TMK trade-off). The same holds for the keys; they are either constant over a large number of encryptions or can be generated on-chip. Another major difference between our AES engine and general-purpose AES hardware is that we do not need to support a mode of operation, which allows for pipelining the datapath.

There exists a rich literature on high-speed AES hardware architectures, targeting both FPGA and standard-cell implementations. Hodjat and Verbauwhede present in [22] the design of an AES datapath that fulfills most of the requirements for use in cryptanalysis mentioned above. Their datapath has a width of 128 bits and implements both inner-round and outer-round pipelining, which means that a new plaintext can be fed into the circuit every clock cycle. Every round is performed in four cycles and the plaintext has to pass through a total of 10 rounds, which results in a latency of 41 clock cycles altogether (including one cycle for the initial key addition). Hodjat and Verbauwhede also report implementation results based on a 0.18 μm standard cell library. Due to the massive pipelining, the AES datapath can be clocked with a relatively high frequency of 606 MHz, yielding a throughput of 77.6 Gbit/s. The overall silicon area is about 473k gates for a 10-round implementation; the area of a 14-round datapath (for keys up to 256 bits) can be estimated to be roughly 660k gates. However, the throughput is independent of the length of the datapath (and also of the key size) since the plaintexts are always processed at a rate of one 128-bit block per cycle.

As mentioned before, the proposed multi-core processor for cryptanalysis of the AES follows the architectural model of modern GPUs such as the NVIDIA GT200b. The GT220b architecture is based on a scalable processor array and consists of 240 streaming-processor cores (so-called "shader" cores), each of which can execute up to three floating-point operations per cycle (e.g. two multiplications and one addition). Assuming that the shader cores are clocked with a frequency of 1476 MHz (e.g. GeForce GTX 285 video card [31]), the theoretical performance of the GT200b exceeds 1000 single-precision GFLOPS. For comparison, a high-end CPU such as the Intel Core-i7 reaches just slightly more than 100 single-precision GFLOPS when clocked at its maximum frequency of 3.33 GHz, i.e. the performance gap between current-generation CPUs and GPUs is about an order of magnitude. The GT200b consists of 1.4 billion transistors (i.e. 350M gates) covering a 470 mm^2 die area built on a 55 nm TSMC CMOS process [31].

Our AES processor is a multi-core system consisting of 500 AES engines based on Hodjat's design as sketched above. Each AES engine has an area of 660k gates, which amounts to a total of 330M gates for 500 engines. When including other building blocks (e.g. host interface, small local memory, interface to external memory, etc.), it can be expected that the overall silicon area of our AES processor will be roughly comparable to that of the GT200b. However, we assume the AES engines to be clocked with a frequency of 2.0 GHz, which should be easily possible when considering that Hodjat's implementation reached a frequency of 606 MHz on basis of an old 0.18 μm process that is significantly slower than the recent 55 nm TSMC technology. Of course, cranking up the clock speed will also increase power consumption, but the additional heat can be controlled by better cooling, as will be discussed in more detail in Section 5. Each AES engine can encrypt plaintexts at a rate of one 128-bit block per cycle, yielding an overall throughput of $500 \times 2 \cdot 10^9 = 10^{12}$ AES operations per second. Interestingly, these 10^{12} AES operations per second match exactly the 1000 GFLOPS per second of the GT200b.

4 Optimized Memory and Storage

Looking at the raw complexity figures of the RKC and TMK attack, it is obvious that memory capacity (and throughput) is the most critical issue after computation time. In fact, the main bottleneck of many high-performance applications is memory bandwidth, i.e. the speed with which data can be transferred between memory (i.e. RAM) and the functional units of the processor where the actual computation is performed. However, in analogy to our argumentation from previous section, we have to point out first that both cryptanalytic attacks considered in this paper have very special requirements with respect to memory and storage, which differ greatly from the requirements of "general-purpose" applications. This difference is especially pronounced with respect to key agility. While support of key agility is important for many "conventional" hardware implementations, it is not an issue for the RKC attack since there are only four keys at all. In case of the TMK attack, key-agility is important but the keys can be generated on chip. The plaintext is kept fixed during the whole attack.

Given the high performance of our processor (500 AES operations per clock cycle), one may expect that memory bandwidth is a limiting factor since plaintexts can hardly be loaded at a rate of $500 \times 16 = 8000$ bytes per cycle. Fortunately, the plaintexts processed in both the RKC and TMK attack do not need to be loaded from off-chip memory, but can be generated on-chip. In the former case (RKC attack), the plaintexts can be generated in a very straightforward way using a plain counter. The TMK attack, on the other hand, executes encryption chains in the pre-computation phase, i.e. the output of an AES operation is input to the next one, whereby a simple modification of the output (e.g. a bit permutation) is carried out in between. However, this modification can be easily implemented in hardware and does not impact the throughput of the AES processor. The situation is similar for the ciphertexts. When performing an RKC attack, only very few ciphertexts are actually stored, i.e. the throughput with which data is moved to off-chip storage is several orders of magnitude lower than the AES processing rate.

Even though only a small fraction of the ciphertexts are actually stored, the storage requirements of the RKC attack are still enormous, namely 2^{78} bytes. This amount is needed for storage of four 2^{72} structures of 16-bytes and is unavoidable unless a better differential is found. The attack also needs an array of 2^{78} counters in the final stage, but this part can be optimized to consume less memory. Storage of such size can only be realized in a distributed fashion, e.g. by attaching a high-capacity harddisk to each AES processor. The CAESAR supercomputer described in the next section consists of $3 \cdot 10^{10}$ AES processors (and therefore we also have $3 \cdot 10^{10}$ harddisks), which means each harddisk must provide a capacity of slightly less than 10 TB. However, the state-of-the-art (as of 2010) are harddisks with a capacity of 2.5 TB, costing 100 US$ when purchased in large quantities. Nonetheless, we argue that the enormous storage requirements do not render the RKC attack infeasible, at least not when taking into account recent technical innovations. For example, Hitachi and other harddisk vendors are striving to make a technology called Thermally-Assisted Recording (TAR) ready for mass production, which could boost storage density to about 10 TB per inch. Researchers at Stanford University are experimenting with electron quantum holography. By superimposing images of different wavelengths into the same hologram, a density of $3 \cdot 10^{18}$ bytes per inch can be reached. Therefore, it is not unrealistic to assume that 100 TB harddisks could be mass produced for just 100 US$ within the next 5–10 years. Such a 100 TB harddisk could be shared by 10 AES processors. Consequently, the overall cost of storage for the RKC attack (i.e. $3 \cdot 10^9$ harddisks of 100 TB each) would amount to roughly 300 billion US$.

The storage requirements of the TMK attack are $2^{56} \cdot 16 \cdot 2 = 2^{61}$ bytes, which will cost in 5–10 years only 2.3 million US$ (or 92 million US$ with present-day technology and prices). As mentioned in Section 2, only the start and endpoints of the encryption chains generated in the pre-computation phase of the TMK attack are actually stored. The situation is similar for an RKC attack since only a very small fraction of the ciphertexts actually needs to be stored. In order to simplify the estimation of the time required for an RKC or TMK attack, we assume that storing data on the harddisks, as well as any subsequent operation accessing the stored data, does not slow down the attack (i.e. the overall attack time is primarily determined by the AES operations and not by accesses to memory or storage). This assumption is justified in the context of the present paper for two reasons. First, the rate at which data is transferred to and from storage is very low compared to the AES processing rate. Second, as stated in Section 1, we aim to estimate a *lower bound* for the time and energy required to perform an attack.

5 Evaluation of Attack Time and Energy

To assess the feasibility of the RKC and TMD attack using "GPU-like" special-purpose hardware, we make the following assumptions about the adversary. We assume an extremely powerful adversary with huge resources in terms of both financial means and expertise in cryptanalysis, which can be expected to be the case for the national security agencies and/or defense departments of certain countries. More precisely, we assume that the adversary has a budget of 1 trillion (i.e. 10^{12}) US$ at its disposal for the design and manufacturing special-purpose hardware (i.e. GPU-like processors). Based on these highly optimized processors, the adversary can build a large-scale supercomputer for cryptanalytic attacks on the AES; we call this supercomputer *CAESAR* (short for Cryptanalytic AES ARchitecture). A budget of 1 trillion US$ is not completely unrealistic when considering that the overall amount the US spends for military and national defense is estimated to be between 880 billion and 1.03 trillion US$ in fiscal year 2010 [38].

We furthermore assume that the adversary has additional funds to cover other expenses such as designing the AES processor, designing and implementing dedicated storage for the TMD attack, operating the CAESAR supercomputer for a certain period of time (which is primarily energy costs), and so on. The exact amount of money needed for these additional expenses depends on many different factors (e.g. the resources of the adversary). For example, the adversary could be an organization that has its own power plants, which significantly reduces the cost for operating large server farms which house the CAESAR supercomputer. In any case, it can be expected that these additional costs will be considerably below the 1 trillion US$ we assume for the manufacturing of AES processors; a reasonable estimation is 500 billion US$. Again, a total funds of 1.5 trillion US$ is not completely unrealistic when taking the annual budget deficit of the US as reference, which is expected to exceed 1.4 trillion US$ in the fiscal year 2009 [1].

The next question to answer is how many AES processors can be produced for 1 trillion US$. We take again the NVIDIA GT200b as reference since our optimized processor housing 500 AES engines has roughly the same silicon area as the GT200b, which means that the manufacturing costs should be very similar. Unfortunately, we were not able to find a reliable source for the manufacturing cost of a GT200b processor. However, what is publicly known are the retail prices of complete graphics/video cards containing the GT200b. For example, the GeForce GTX 285 [31], a graphics card equipped with a GT200b processor clocked at 1476 MHz, retails for less than 300 US$. The GeForce GTX 295 [32] is a graphics card housing two GT200b processors that costs less than 400 US$. However, it must be considered that both are complete graphics cards that do not only contain GT200b chips, but also large amounts of fast memory and several other components. Furthermore, we have to take into account that the quoted retail prices include gains for the producer and retailer(s), NRE costs, as well as other costs such as VAT. Therefore, it can be assumed that manufacturing a GT200b chip costs significantly less than 100 US$. The cost of one of our AES processors will be even much lower since, for example, the NRE costs are negligible when producing a very large number of chips. Taking all this into account, we can estimate a lower bound of 30 US$ for the manufacturing cost of a single AES processor.

Having a budget of 1 trillion (i.e. 10^{12}) US$ for chip production (and assuming a reasonably high yield) means that the adversary gets a total of about $3 \cdot 10^{10}$ AES processors, each of which can perform 10^{12} AES operations per second (see Section 3). Consequently, the overall throughput of all processors of CAESAR amounts to roughly $3 \cdot 10^{22}$ AES operations per second. The RKC attack as described in Section 2 requires the adversary to perform $2^{99.5} \approx 9 \cdot 10^{29}$ AES operations, which can be accomplished in just $3 \cdot 10^{7}$ seconds (i.e. approximately one year) on the CAESAR supercomputer. Of course, these estimations are based on "best-case" (yet not unreasonable) assumptions and should be considered as a lower bound for the execution time of this key-recovery attack given a budget of 1 trillion US$ for chip production.

The situation is similar for the TMK trade-off attack in the sense that the AES operations dominate the overall execution time by far. However, TMK attacks, as mentioned in Section 2, are performed in two phases; an off-line (i.e. pre-computation) phase in which tables containing the start and endpoints of encryption chains are generated, and an online phase in which pre-images of data points are searched in the tables. Let us consider an example of a TMK attack in which the adversary is given 2^{32} encryptions of a fixed plaintext under different (but unknown) keys. To recover one of these keys, the adversary has to carry out 2^{96} AES operations during the pre-computation phase, as well as 2^{80} AES operations in the online phase. Similar to the RKC attack, the inputs for the AES operations carried out in the pre-computation phase do not need to be loaded from an external source, but can be generated on-chip[2]. Only the start and end-point of the encryption chain are actually stored in the table, which means that the memory bandwidth can be orders of magnitude lower than the AES throughput. In our case, the pre-computed tables contain 2^{56} data points (i.e. 128-bit ciphertexts) altogether. If we assume again $3 \cdot 10^{10}$ GPU-like AES processors with an overall throughput $3 \cdot 10^{22}$ AES operations per second, the 2^{96} AES operations carried out in the off-line phase take about 30.6 days. The 2^{80} AES operations of the on-line phase are negligible in relation to the execution time of the off-line phase, which means the overall attack time is primarily determined by the pre-computation of the tables. However, this pre-computation is a one-time effort because the same set of tables can be used to recover other keys. If one is willing to wait for one year until the pre-computation is finished, then he needs less than one tenth of the attack budget. On such a "smaller" supercomputer, solutions will still be generated at an amazing speed of eight minutes per 128-bit key.

5.1 Further Considerations

Besides execution time and memory requirements, there are a number of other factors that need to be taken into account when studying the feasibility of a large-scale supercomputer for cryptanalysis of the AES like CAESAR. In the following, we try to estimate the time it takes to manufacture $3 \cdot 10^{10}$ AES processors and the energy these processors consume when clocked with a frequency of 2.0 GHz.

A state-of-the-art fab for chip production, such as the one mentioned in [34], has an overall capacity of 300,000 wafers per month. Given a diameter of 300 mm, the silicon area of a single wafer amounts to 70,685 mm^2. In Section 3 we argued that a GPU-like processor housing 500 AES engines would have roughly the same gate count as the NVIDIA GT200b, hence it is sensible to assume that its silicon area will be in the same range, namely 470 mm^2 on basis of the 55 nm TSMC technology. Consequently, 150 AES processors can theoretically be obtained from a 300 mm wafer. However, given a typical yield of 75% and taking edge dies into account, it can be estimated that we get out some 100 AES processors per wafer. A high-capacity fab would be able to produce $3 \cdot 10^{7}$ chips in one month, or $3.6 \cdot 10^{8}$ chips

[2] More precisely, the input (i.e. plaintext) of a given AES encryption is always fixed and the output (i.e. ciphertext) of the previous AES encryption is used as a new key, after a simple modification (e.g. a fixed bit permutation). This simple modification of the output bits can be easily implemented in hardware and does not impact the throughput of the AES processor.

per year. Consequently, the total production time for $3 \cdot 10^{10}$ AES processors amounts to approximately 83 years. Accordingly, the time would drop to one year when the chip production is distributed to 83 high-capacity fabs. Note that the fab mentioned in [34] was constructed in 18 months and required an investment of 1 billion US$. A well-funded adversary (as assumed in this paper) may even consider to construct its own high-capacity fabs and operate these fabs solely for the production of GPU-like AES processors.

NVIDIA's GeForce GTX285, a graphics card equipped with a GT200b processor, has a maximum power consumption of 204 W [31]. In this card, each of the 240 shader cores of the GT200b is clocked with a frequency of 1476 MHz. On the other hand, the GeForce GTX295 houses two GT200b, but their shaders are clocked with a slightly lower frequency of 1242 MHz. Its maximum power consumption is 289 W as specified in [32]. However, it must be considered that these figures refer to the power consumption of the "whole" graphics card, which includes besides the GT200b processor(s) also several other components, in particular large amounts of memory. Therefore, it can be estimated that a GT200b processor clocked at 1476 MHz consumes approximately 100 W. Our AES processor is operated at a slightly higher frequency (2.0 GHz[3] instead of 1476 MHz) and, as a consequence, its power consumption will rise by the same factor to 135 W. The power consumed by all $3 \cdot 10^{10}$ AES processors amounts to a whopping 4 TW, i.e. $4 \cdot 10^{12}$ W. For comparison, the average total power consumption of the US was 3.34 TW in 2005 [39]. In summary, it can be concluded that the most limiting factor of attacking AES using special-purpose hardware is neither the computation time nor the memory requirements, but the power consumption of the hardware.

Does this enormous power consumption of our CAESAR supercomputer render RKC (respectively TMK) attacks with a complexity of $2^{99.5}$ (respectively 2^{96}) completely impossible? Not necessarily when we consider Moore's law: transistor sizes (and also power consumption) shrunk significantly with every new process generation that was introduced during the past two decades. It is expected that Moore's law will continue to hold—and transistor sizes will continue to shrink—for another ten years, though at a slightly lower rate than in the past [26]. For example, TSMC estimates a transistor size of only 7 nm in 2020, which is eight times smaller than the transistor size of the 55 nm TSMC technology under which the GT200b processor is produced. Using a 7 nm technology for our AES processor would result in a power consumption that is only a fraction of the 135 W we used for the evaluation above. Estimations beyond the the-year horizon are rather difficult since future VLSI technology must not necessarily be silicon-based. However, the following historical example may help to understand the progress in VLSI technology. The first supercomputer that reached a performance of 1 TFLOPS (i.e. 1000 GFLOPS) was the ASCI Red, built in 1997 by Intel and operated by Sandia National Labs. ASCI Red housed almost 10,000 Pentium Pro processors and had a power consumption of roughly 500 kW. Today, a single GPU like the GT200b reaches the same performance, but does so at a power consumption of only 100 W (see above). Consequently, the power consumption per TFLOPS dropped from 500 kW in 1997 to 100 W in 2010, which corresponds to a factor of 5,000.

5.2 Outlook into the Future

In this subsection, we briefly mention some factors that can significantly decrease the cost of hardware attacks on AES in the future (10–20 years from now). These factors are:

- Moore's law continuing for another ten years, albeit at a slightly reduced speed.

[3] The increased heat due to the higher clock frequency can be handled through better cooling, e.g. a liquid cooling system.

Table 4: Main characteristics of CAESAR and summary attack complexities.

One AES engine:	660K gates	2GHz clock speed	
One AES processor:	500 AES engines[a]	10^{12} AES ops/ s	30 US$[b]
CAESAR supercomputer:	$3 \cdot 10^{10}$ AES processors	$3 \cdot 10^{22}$ AES ops/s	1 trillion US$
AES chip production:	83 high capacity fabs	approx. 1 year	83 bln US$
Power consumption:	135 W per processor	4 TW = $4 \cdot 10^{12}$ W	
RKC Attack:	$9 \cdot 10^{29}$ AES ops	approx. 1 year	
Storage RKC:	2^{78} bytes		300 bln US$[c]
TMK Attack (2^{32} targets):	$0.8 \cdot 10^{29}$ pre-computation	30.6 days	
TMK Attack (2^{32} targets):	10^{24} ops per AES key	negligible	
Storage TMK:	2^{61} bytes		92 mln US$

[a] A 470 mm^2 die on 55nm TSMC CMOS process with 330M gates.

[b] This is a lower bound.

[c] Estimate for the next 5-10 years.

- Cryptanalytic breakthroughs can entail spectacular reductions in attack complexity. However, the cryptanalytic progress for AES does not follow a steady and predictable flow. It is hard to make any predictions based on the time-line of the past attacks since they were very sporadic.
- Computers based on spin (so-called magneto-electronics or spintronics) may significantly reduce power consumption.
- The use of optical computers may also significantly reduce the power consumption of large-scale cryptanalytic hardware.
- 3D optical data storage is one of the technologies that could increase storage density and hence decrease the memory cost of attacks. For example, DVD-size optical disks of 1 TB (and thus of similar price) are conceivable. These disks will use more than 100 optical layers.
- Quantum holography: Superimposing images of different wave length into the same hologram on copper medium can increase memory density spectacularly. For example, a density of 3 ExaBytes (i.e. $2^{61.5}$ bytes) per square-inch was demonstrated in 2009 using this technique[4].

6 Conclusions

In this paper, we investigated the feasibility of large-scale hardware attacks on AES-128 and AES-256 bounded by a time complexity of 2^{100}. We described CAESAR, a hypothetical supercomputer consisting of $3 \cdot 10^{10}$ GPU-like AES processors, each of which can reach a throughput of 10^{12} AES operations per second. CAESAR could be built with a total budget of roughly 1.5 trillion US$ (or with 1 trillion US$ solely spent for chip fabrication) and would be capable of performing up to $3 \cdot 10^{22}$ AES operations per second, or approximately $9 \cdot 10^{29} \approx 2^{99.5}$ AES operations in a year. Table 4 summarized the main characteristics and capabilities of CAESAR along with the complexities of the TMK and RKC attack on

[4] Overall, in the field of digital storage there seem to be several competing technologies, which are of very different physical nature and in which progress happens in sudden leaps, rather than a monotone growth. There is also a negative effect of well-developed technologies that come close to their physical limits, but still act as a barrier to the development of new revolutionary ideas due to high initial costs.

AES-128 and AES-256, respectively. Our evaluation shows that a TMK trade-off attack on AES-128 using 2^{32} targets is well within reach with current VLSI technology. CAESAR requires about 30 days for the pre-computation phase, after which each new 128-bit key out of the pool of 2^{32} targets can be found in negligible time. A smaller variant of CAESAR costing 100 billion US$ is able to break a new key every eight minutes, but requires a full year for the pre-computation of tables. We also studied the RKC attack on AES-256 and found it prohibitively expensive because of the huge memory complexity of 2^{78}, even though CAESAR could perform the required $2^{99.5}$ AES operations in roughly one year. In summary, our work shows that the main bottlenecks of large-scale cryptanalytic hardware for breaking the AES are neither execution time nor production cost, but rather power consumption and high memory complexity. Therefore, we recommend cryptanalysts to focus on attacks with a time complexity of up to 2^{100}, and a memory and data complexity of less than 2^{70}.

References

[1] BBC News. US deficit hits record $1.4tn. Available online at `http://news.bbc.co.uk/2/hi/8296079.stm`, 2009.

[2] D. J. Bernstein, H.-C. Chen, M.-S. Chen, C.-M. Cheng, C.-H. Hsiao, T. Lange, Z.-C. Lin, and B.-Y. Yang. The billion-mulmod-per-second PC. In *SHARCS '09: Special-Purpose Hardware for Attacking Cryptographic Systems*, pages 131–144, Lausanne, Switzerland, Sept. 2009.

[3] D. J. Bernstein, T.-R. Chen, C.-M. Cheng, T. Lange, and B.-Y. Yang. ECM on graphics cards. In A. Joux, editor, *Advances in Cryptology — EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 483–501. Springer Verlag, 2009.

[4] E. Biham, O. Dunkelman, and N. Keller. Related-key boomerang and rectangle attacks. In R. Cramer, editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 507–525. Springer Verlag, 2005.

[5] A. Biryukov. The boomerang attack on 5 and 6-round reduced AES. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *Advanced Encryption Standard — AES 2004*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer Verlag, 2005.

[6] A. Biryukov and D. Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In M. Matsui, editor, *Advances in Cryptology — ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer Verlag, 2009.

[7] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and related-key attack on the full AES-256. In S. Halevi, editor, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer Verlag, 2009.

[8] A. Biryukov, S. Mukhopadhyay, and P. Sarkar. Improved time-memory trade-offs with multiple data. In B. Preneel and S. E. Tavares, editors, *Selected Areas in Cryptography — SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 110–127. Springer Verlag, 2006.

[9] A. Biryukov and I. Nikolic. Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and others. In H. Gilbert, editor, *Advances in Cryptology — EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer Verlag, 2010.

[10] A. Biryukov and A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In T. Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 1–13. Springer Verlag, 2000.

[11] J. W. Bos, T. Kleinjung, R. Niederhagen, and P. Schwabe. ECC2K-130 on Cell CPUs. In D. J. Bernstein and T. Lange, editors, *Progress in Cryptology — AFRICACRYPT 2010*, volume 6055 of *Lecture Notes in Computer Science*, pages 225–242. Springer Verlag, 2010.

[12] J. Daemen and V. Rijmen. *The Design of Rijndael: AES – The The Advanced Encryption Standard*, volume ?? of *Information Security and Cryptography*. Springer Verlag, 2002.

[13] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly Media, 1998.

[14] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved cryptanalysis of Rijndael. In B. Schneier, editor, *Fast Software Encryption — FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230. Springer Verlag, 2001.

[15] W. Geiselmann, F. Januszewski, H. Köpfer, J. Pelzl, and R. Steinwandt. A simpler sieving device: Combining ECM and TWIRL. In M. S. Rhee and B. Lee, editors, *Information Security and Cryptology — ICISC 2006*, volume 4296 of *Lecture Notes in Computer Science*, pages 118–135. Springer Verlag, 2007.

[16] W. Geiselmann and R. Steinwandt. Non-wafer-scale sieving hardware for the NFS: Another attempt to cope with 1024-bit. In M. Naor, editor, *Advances in Cryptology — EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 466–481. Springer Verlag, 2007.

[17] H. Gilbert and M. Minier. A collision attack on 7 rounds of Rijndael. In *Proceedings of the 3rd Advanced Encryption Standard Candidate Conference*, pages 230–241. National Institute of Standards and Technology, 2000.

[18] M. Gorski and S. Lucks. New related-key boomerang attacks on AES. In D. Roy Chowdhury, V. Rijmen, and A. Das, editors, *Progress in Cryptology — INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 266–278. Springer Verlag, 2008.

[19] R. E. Graves. *High Performance Password Cracking by Implementing Rainbow Tables on NVIDIA Graphics Cards (IseCrack)*. M.Sc. Thesis, Iowa State University, Ames, IA, USA, 2008.

[20] T. Güneysu, T. Kasper, M. Novotný, C. Paar, and A. Rupp. Cryptanalysis with COPACOBANA. *IEEE Transactions on Computers*, 57(11):1498–1513, Nov. 2008.

[21] M. E. Hellman. A cryptanalytic time-memory tradeoff. *IEEE Transactions on Information Theory*, 26(4):401–406, July 1980.

[22] A. Hodjat and I. Verbauwhede. Speed-area trade-off for 10 to 100 Gbits/s throughput AES processor. In *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers (ACSSC 2003)*, volume 2, pages 2147–2150. IEEE, Nov. 2003.

[23] D. Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, rev. sub. edition, 1996.

[24] J. Kim, S. Hong, and B. Preneel. Related-key rectangle attacks on reduced AES-192 and AES-256. In A. Biryukov, editor, *Fast Software Encryption — FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 225–241. Springer Verlag, 2007.

[25] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler. Breaking ciphers with COPACOBANA – A cost-optimized parallel code breaker. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems — CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 101–118. Springer Verlag, 2006.

[26] R. Kwong. TSMC warns Moore's law may have 10 years left. Fiancial Times Tech Blog, available online at `http://blogs.ft.com/techblog/2010/04/tsmc-warns-moores-law-may-have-10-years-left`, Apr. 2010.

[27] J. Lu, O. Dunkelman, N. Keller, and J. Kim. New impossible differential attacks on AES. In D. Roy Chowdhury, V. Rijmen, and A. Das, editors, *Progress in Cryptology — INDOCRYPT 2008*, volume 5365 of *Lecture Notes in Computer Science*, pages 279–293. Springer Verlag, 2008.

[28] N. Mentens, L. Batina, B. Preneel, and I. M. Verbauwhede. Time-memory trade-off attack on FPGA platforms: UNIX password cracking. In K. Bertels, J. M. Cardoso, and S. Vassiliadis, editors, *Reconfigurable Computing: Architectures and Applications — ARC 2006*, volume 3985 of *Lecture Notes in Computer Science*, pages 323–334. Springer Verlag, 2006.

[29] G. Meurice de Dormale, P. Bulens, and J.-J. Quisquater. Collision search for elliptic curve discrete logarithm over GF(2^m) with FPGA. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems — CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 378–393. Springer Verlag, 2007.

[30] K. Nohl, E. Tews, and R.-P. Weinmann. Cryptanalysis of the DECT standard cipher. In S. Hong and T. Iwata, editors, *Fast Software Encryption — FSE 2010*, volume 6147 of *Lecture Notes in Computer Science*, pages 1–18. Springer Verlag, 2010.

[31] NVIDIA Corporation. GeForce GTX 285: A Powerful Single GPU for Gaming and Beyond. Specification, available online at `http://www.nvidia.com/object/product_geforce_gtx_285_us.html`, 2010.

[32] NVIDIA Corporation. GeForce GTX 295: A Powerful Dual Chip Graphics Card for Gaming and Beyond. Specification, available online at `http://www.nvidia.com/object/product_geforce_gtx_295_us.html`, 2010.

[33] C. B. Pomerance, J. W. Smith, and R. S. Tuler. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. *SIAM Journal on Computing*, 17(2):387–403, Apr. 1988.

[34] Samsung Electronics Co. Ltd. Samsung and Siltronic start joint production of 300mm wafers in Singapore. Press release, available online at `http://www.samsung.com/us/aboutsamsung/news/newsIrRead.do?news_ctgry=irnewsrelease&news_seq=9345`, 2008.

[35] A. Shamir. Factoring large numbers with the TWINKLE device (Extended abstract). In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 2–12. Springer Verlag, 1999.

[36] A. Shamir and E. Tromer. Factoring large numbers with the TWIRL device. In D. Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 1–26. Springer Verlag, 2003.

[37] S. Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor Books, 2000.

[38] Wikipedia. Military Budget of the United States. Available online at `http://en.wikipedia.org/wiki/Military_budget_of_the_United_States`, 2010.

[39] Wikipedia. Orders of Magnitude (Power). Available online at `http://en.wikipedia.org/wiki/Orders_of_magnitude_(power)`, 2010.

[40] J. E. Wilcox. *Solving the Enigma: History of the Cryptanalytic Bombe*. Center for Cryptologic History, National Security Agency, 2001.