Combining Security Assurance and High Performance in Hostile Environments

Pierre Capillon (pierre.capillon@c-s.fr)
Antoine Casanova (antoine.casanova@c-s.fr)
C-S Communication & Systèmes,
22 avenue Galilée,
92350 Le Plessis-Robinson, France

Abstract

With growing global bandwidth consumption and increasing cyber-attacks, information security actors are in constant need for scalable, high performance products that still provide a high level of security assurance.

The French national project “SHIVA” aims at developing a new security architecture providing multiple services and such performance and security assurance levels. Based on research and development from various fields, this paper presents usages of technologies from the high performance computing systems (HPC clusters), FPGA-based reprogrammable devices and the use of formal methods to provide additional assurance to be tested under most standard evaluation criteria. A very high level of security assurance is targeted, hence high attack potentials are assumed as per the Common Criteria Vulnerability Analysis assurance requirements (CC AVA_VAN.5):

A distributed architecture using scalable InfiniBand interconnect is discussed as a new interconnect method for cryptographic devices. New usages and advantages of relying on such an architecture are presented, as well as various security considerations on threats, attacks and how reprogrammable devices bring innovative solutions to cryptographic initialization process on hostile platforms, as well as optimizations and opportunities opened by the use of pre-processing and formally designed software in handling operational data flow and critical information.

INTRODUCTION

Communication technologies are witnessing a massive evolution: growing bandwidth, larger contents. The last few years saw the ever increasing mainstream access to information and the evolution of everyone's online, digital identity.

As the technologies, performances and usages evolve, the need for information security becomes more and more compelling. Required security assurance levels are being raised in order to assess the constant evolution of threats and potential attacks on confidential information or highly critical systems. Protecting these by still keeping high security assurance levels while still providing efficient and high performance services is a challenging tasks.

Current networking technology standards are evolving towards higher bandwidths available closer to mainstream audience. Home users see fiber-optic broadband internet access becoming a reality, which will obviously make new services become more and more resource-demanding. The global bandwidth consumption grows by 50% every year[1].

All communication services are growing. As growth doesn't come without more exposure, in 2008, the number of cyber-attacks increased by 40%[2]. This challenges security products to be more and more efficient, with higher protection standard, as infrastructure becomes more and more important.

These observations also translate quite well to less mainstream fields, such as military or governmental usages, or critical commercial applications (banking, intellectual property, etc.). Nowadays, transmission of audio, video streams with high quality, becomes more and more popular as the network capacities evolve. Think about transmission of live on-field video-feeds with high resolution, or online processing of the increasing number of secure transactions. Multi-function secure devices are now key elements of information technologies.

In this context, a French national project was launched by the Ministry of Industry[3] and Minalogic[4], to propose and develop evolutions and enhancement to hardware cryptographic modules and devices used to ensure information security. A new Secure Hardware Immune Versatile Architecture (SHIVA) is currently being developed as part of this project, with consortium partners from industry, security and academic fields.

This paper exposes some of the technical challenges faced by modern cryptographic devices, and discusses some of the proposed evolutions, new features or usages for high-performance cryptographic devices providing a high level of security assurance.

It will go through methodology and approaches used to tackle the aforementioned problematics, as well as design and development methods used to satisfy industry and military evaluation security assurance standards such as FIPS 140-2 recommendations or ISO/IEC 15408 Common Criteria. Also, architectural advantages of the solution will be presented, along with new usages and possibilities. Attacker opportunities and techniques will not be omitted as they will be considered every time an item is discussed.

This paper acts as a small survey of state-of-the-art techniques in security countermeasures, current tech-

[1] ITR news, "Qu'est-ce qui va changer dans l'Internet du futur?", http://www.itrnews.com/articles/83173/est-va-changer-internet-futur.html

[2] Le Monde, "L'administration américaine de plus en plus visée par des cyber-attaques", http://www.lemonde.fr/ameriques/article/2009/02/18/l-administration-americaine-de-plus-en-plus-visee-par-des-cyber-attaques_1156857_3222.html

[3] DGCIS: Direction générale de la compétitivité, de l'industrie et des services, http://www.pme.gouv.fr/presentation/sommdgcis.php

[4] Micro/nano-technologies global competitive cluster, http://www.minalogic.net

nologies and presents an innovative solution combining high performance technologies, formal methods and security measures from various fields to develop a new security architecture.

1 DEVELOPING A NEW SECURITY ARCHITECTURE WITH SHIVA

The challenges to overcome in developing cryptographic devices may be resolved by bringing slight changes in the system architectures involved in handling information security. These changes could lead to better versatility, performance or security assurance, and are flexible enough to assess various security needs. In particular, multi-function devices assessing the risks of increasing cyber-attacks on systems and growing needs in bandwidth.

The French National Project “SHIVA” (Secured Hardware Immune Versatile Architecture) aims at providing such improvements. This paper describes the advances and proposals of the project, as well as discusses various information security-related considerations.

1.1 Threats on a cryptographic device

Such an architecture may not be designed without considering potential attacker options. As the project aims at versatility, this implies dispatching features and capacities throughout multiple different components, instead of aggregating and concentrating features in a small module. However, such a task segregation sometimes has severe impacts on handling of information security, e.g. securing communication channels between key components. Thus, this section reviews some potential threats on a cryptographic device, with an emphasis on threats that will be assessed or accounted for in a way depending on the proposed architecture. It also defines the architecture’s goals, for which proposal and proof-of-concepts are discussed in a further section.

Although it doesn’t provide an extensive list of potential attacks, these will always be taken into account and discussed when proposing improvements and solutions throughout this paper. The attacker is always assumed to have access to unlimited resources, which covers potential governmental threats.

1.1.1 Calculating attack potential

In the scope of the project, an attacker is defined by Information Technology Security Evaluation Criteria (ITSEC) and Common Criteria standards as someone determined to retrieve or compromise protected information.

A high attack potential is assumed. Several hypothesis on the attacker are considered, as per Common Criteria Vulnerability Analysis assurance requirements (CC AVA_VAN.5). This means that the attacker has access to unlimited resources, a high level of technical and scientific knowledge, unlimited time to perform the attack and an available window of opportunity (is aware of a vulnerability before public knowledge).

Considering both direct and indirect attacks (e.g. when the result doesn’t carry compromising information but is used to derivate it or perform another attack), as well as both hardware and software platforms potentially hostile, the strategy developed here progressively goes towards the most sensible level of operation, depending on the security assurance of the environment.

1.1.2 Physical attacks

With cryptographic devices being the key component of modern information security, attacks become more and more focused on these products. Indeed, successful attacks could result in compromising either confidentiality of information, or its availability.

Denial of service attacks target the availability of information, whereas other attacks aim at compromising cryptographic material in order to decrypt eavesdropped communications or to inject malicious information into a system.

While designing a security architecture, one needs to take into account various physical attacks, some of which are now widely known and frighteningly efficient. These attacks can be classified depending on the amount of invasive techniques implied in the process, which directly impacts countermeasure design and development as non-invasive methods will be difficult to detect:

- Side-channel attacks are passive methods, which do not modify or tamper with the device in any way. These passive attacks are difficult, if not impossible to detect as they solely rely on the observation of a running system. A system will have to employ preventive countermeasures to deter side-channel attacks attempts, and render the observed variations unrelated to the actual values being computed or handled.
- Physical tampering on the other hand is the most invasive class of attacks, in that it will attempt at penetrating or physically modifying a device in order to change data paths, retrieve stored data, or reverse-engineer the device.
- Fault injection attacks stand between the previous extreme methods, as they consist in forcing a device to make erroneous calculations, and observing changes in the system's behavior. These changes are used to reveal or correlate variations to the data involved in computations or even the actual algorithms.
- Environmental attacks focus on the operating environment of the device, changing parameters to either allow other attacks to take place (cooling a device to extract stored data afterward), or causing denial of service.

Known side-channel attacks involve the observation of various system characteristics and signatures and correlates them with operations being performed:

- Power consumption is often correlated with the amount of power necessary to change bit states in a chip. A set of power consumption traces could allow an attacker to retrieve keys involved in a computation, as each different key will result in different results. Indeed, the power consumption of a particular algorithm will not be the same depending on the keys involved in the computation. Common known attacks include Simple and Differential Power Analysis (SPA, DPA) [15].
- Electromagnetic emissions can reveal various informations to an attacker. In the same fashion as power analysis attacks, Simple and Differential Electromagnetic Analysis (SEMA, DEMA) reveal involved cryptographic material as a result of varying electromagnetic emissions [5]. These emissions can also reveal the location of computing units on a chip's layout, which could facilitate reverse-engineering. This attack can also target specific locations of a device, thus eliminating noise in the observed fluctuations, whereas power analysis is limited to the global consumption of a particular chip or device. Lastly, these attacks may be carried out by an attacker without direct access to the hardware, from a small distance.

- Timing attacks focus on observing how much time it takes to perform specific calculations (i.e. generating a key, signing data or encrypting a packet), and correlate variation in the duration of calculation with the properties of handled data. Coupled with fault injection techniques, these attacks can be significantly sped up, requiring less data to be collected [6].

On the other hand, invasive methods are much more powerful and direct, but may require much more resources and training to be properly executed. However, since the SHIVA project targets governmental usage, and aims at protecting from government-level threats, such possibilities are also considered.

Invasive physical tampering techniques include:

- Fault injection attacks, in an attempt to corrupt computed data and analyze the system's reaction fluctuations. These attacks may also lead to denial of service, but the main purpose is either to bring a device to an unknown or unstable state or to observe fluctuations in computation timings, power consumption or emanations. Faults can be injected through various ways, either by directly tapping into a device, chip (for instance, using lasers [3]), or by varying its external resources: power variations can induce erroneous calculations on both hardware and software implementations, as recently demonstrated by the attacks on software RSA implementations [12].
- Probing of internal and external communication channels, which could lead to the compromise of critical information exchanges (memory/register transfers, bus transfer of sensitive material...).

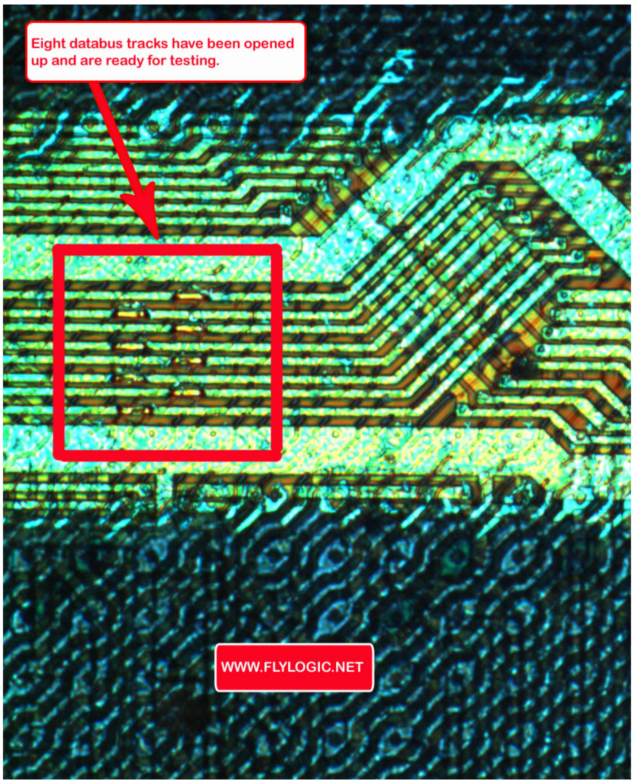
Eight databus tracks have been opened up and are ready for testing.
WWW.FLYLOGIC.NET

Figure 1: Internal bus prepared for probing. Also appearing on the image is the chip's tamper shielding. Source: http://www.flylogic.net

- In-memory data extraction or modification to reveal private information, inject faults in computed data or to change the execution flow of binary code to weaken the device or unlocking specific features.

- Modification of logic circuitry to unlock features or change data/execution flow.

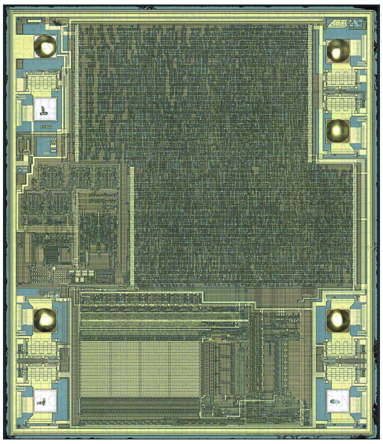

Figure 2: A decapsulated ATMEL CryptoMemory AT88SC153. FlyLogic discovered a backdoor unlocking access to secured memory areas. Source: http://www.flylogic.net

The last class of physical involving environmental modifications may be used to facilitate the attacks described earlier. For instance, changes in temperature will affect physical properties of electronic chips, such as memory remanence. Recent work demonstrated that cooling volatile memory chips considerably extends the period during which data is retained after a loss of power. This allows memory extraction and has been demonstrated to result in the compromising of such materials as disk encryption keys for personal computers [10]. The same attacks performed on FPGA chips could allow retrieving the content of the configuration memory, thus leaking the chip's layout (bitstream). Slow or abrupt environmental variations could either bring a device to unknown or unstable states, or circumvent detection mechanisms and countermeasures.

1.1.3 Logical attacks

While physical attacks may provide significant results, the same goal could be achieved through logical attacks on the device, involving weaknesses in running software or protocol handling. They are dubbed logical in that they do not necessarily require physical access to the device or use physical properties to be carried out.

Viruses, worms and other malicious computer programs are the most well known threats against computer systems, and exploit vulnerabilities of software codes and implementations to take control of the system. While these threats are obviously considered, much more subtle attacks could also have devastating effects.

Fuzzing protocols could reveal mishandling of malformed information and an attacker gaining elevated privileges on a system. State machine fuzzing also aims at bringing a subsystem to an unstable, unknown or otherwise unauthorized state. These include playing with an authentication system, or sending specifically crafted data to a particular function or entry-point to circumvent protection measures.

Logical attacks can also easily target a system's or information availability, leading to a denial of service. They can aim at resource exhaustion, by exploiting features: deliberately flooding a device with corrupted data may exhaust computational resources involved in integrity checking, hence the decrease of overall performance of the device for legitimate traffic.

Information disclosure may also be triggered by such attacks. Simple examples of such disclosure is attacking a network switch by overflowing its MAC tables, which, on vulnerable devices, results in the disclosure of other traffic not initially intended to be relayed on the attacker's port.

In short, logical attacks are hard to define, as they are often much more targeted at a specific system, specifically carried out for that system. This implies careful design and comprehension of the vulnerability surface, to identify and mitigate potential attacks on available services and capacities. Returning to the previous example, describing this vulnerability surface implies taking into account the full data path, including all functions processing packets (integrity checking, filtering) throughout the process, and not limiting to the general description of the functionality (e.g. providing a simple data transfer capacity).

1.1.4 On the relevance of using Application-Specific Integrated Circuits (ASICs)

As chips are more and more subject to physical attacks, is it still relevant to use ASICs compared to the flexibility and versatility of Field-Programmable Gate Arrays (FPGAs)?

ASICs were originally used by nations willing to protect their cryptographic algorithms from disclosure or attacks, which dispatched the efforts between developing protections, obfuscation, countermeasures for classified algorithms and sensitive material storage protection. As an attacker will most likely go for encryption keys, which are one of the most important part of security, it is critical to properly protect them for disclosure, and concentrate appropriate efforts on key management and protection. For instance, Russian GOST 28147-89 standards [8] describe a public, known algorithm, and most effort is spent toward protection of critical parameters, such as S-boxes or keys.

Nowadays, chips can be easily decapsulated, which allows a quick reverse-engineering when it comes to ASIC chips [1]. If the cryptographic device is to implement custom or classified algorithms, it is preferred to deter reverse-engineering in the event of a compromise of the device. Shielding techniques can also be circumvented, and doors are open to internal bus probing [13], fault injection or tampering using lasers, focused ion beams attacks [14].

On the other hand, FPGAs bring a number of interesting features that allow overcoming these problems. At first glance, decapsulation of the chip is not enough to allow reverse-engineering the general architecture of a subsystem or specific logic function. In fact, an FPGA will keep the same appearance regardless of the programmed bitstream. An attacker attempting to reverse-engineer would need to access the bitstream, either through the configuration memory layer or its non-volatile storage area elsewhere on the module. Current FPGA technology also allows partial reprogramming of the chip or dynamically changing part of the chip's layout which opens new opportunities for obfuscation techniques.

However, a system designer is still dependent on manufacturers [7], and all parties involved from conception to distribution of chips. What security assurance does one have on these processes? What assurance is there that no backdoor, exploit, or eavesdropping functionality has not been put in the device by some party during the process? How could designs and intellectual property be effectively protected without using methods only approved by manufacturers? The SHIVA project aims at using the dynamic nature of FPGA-based system to help develop better mitigation techniques to these threats than an ASIC could.

Combining these features bring new mitigation techniques and possibilities, which are simply not possible using ASIC components. Of course, such features do not come without drawbacks, but the proper use of cryptographic and security features as well as an appropriate design bring solutions to overcome the security implications. These new possibilities made the choices lean towards an FPGA-based architecture, and spawned research on mitigation and obfuscation techniques or countermeasures developed for FPGAs.

1.2 SHIVA architecture features

The developed architecture aims at providing a broad variety of cryptographic features, ranging from network encryption to certificate and key management. Although the main purpose is to provide network encryption (either IPsec services for OSI Layer 3 encryption or Layer 2 services at the MAC level [5]), the SHIVA module could provide services normally performed by a dedicated Hardware Security Module (HSM), like onboard secure generation, handling sensitive material or cryptographic offload.

Research and development efforts focus on developing a cryptographic module that would offer the following services:

- Network data flow encryption/decryption in either a Line Encryption or VPN use case;
- Cryptographic material management (key generation, signature-related operations);
- Information tagging relative to their security clearance;
- Filtering gateway based on tagging and source/destination clearance;
- Cryptographic acceleration engine;
- Red / Black separation.

Such a module could be integrated in a platform restricted to a subset of the proposed features, thus being customized based on customer needs.

Besides general services, some specific features are being developed for versatility and customization purposes:

- A customer-specific cryptographic algorithm could be loaded into the module, without intervention from the manufacturer. This allows the development and use of completely proprietary algorithms, instead of tweaking variations of an initial known algorithm.
- Integration in an existing infrastructure, wether it is about cryptographic material, identity management or if it implies diverse network interconnection technologies.

Of course, while designing and developing all of these features the attacker's standpoint has to be constantly kept in mind.

1.2.1 Performance and security assurance goals

Network infrastructure are switching to 10 Gbits/s Ethernet, with the next standards being 40 and 100 Gbits/s. For network applications, the targeted throughput shall handle multigigabits links, starting at 10 Gbits/s interconnect, and is designed to be scalable.

As the project aims at governmental use, the product design process considers certification and evaluation criteria very closely. Efforts are made to comply with main recommendations of industry and military standards, such as FIPS 140-2 (considered up to the level 4) or Common Criteria, with some subsystems being designed by methods allowing evaluations at levels EAL5 and more.

[5] Draft IEEE standards for LinkSec: IEEE 802.1ae and 802.1af

The strategy here is to design software and hardware core components using formal methods. Starting with a small subset, such as the device's state machine, administration and supervision components, this design will gradually expands to cover multiple capacities, up to IPsec and Internet Key Exchange (IKE) protocol handling done with a formally designed implementation of the protocol stacks.

1.2.2 Technical challenges, bottlenecks and blocking issues

In light of previous considerations and goals, the main objective is to design an highly reusable architecture, which allows easy personalization to the customer needs, and easy integration to an existing infrastructure. Technical choices made in regard of this objective affect further developments with different kinds of blocking issues.

Numerous challenges to overcome are identified:

- While the cryptographic module shall be scalable to accommodate networks of different performances, or appliances for specific usages (e.g. offering products of various performance ranges based on the same architecture for multiple price tags), it also needs to be prepared for future technology evolutions, and consider performances way beyond 100 Gbits/s of encrypted traffic. Therefore, a new product needs to be ready to sustain those performances while technology is not yet available to handle such an amount of traffic.
- While the cryptographic module itself will sport the same base architecture for all product lines, its integration will have to take into account the various usages of its features. A cryptographic accelerator used for SSL web services will be integrated in a backend network, while VPN gateways are usually put as front-ends near the public network access, and a certificate management infrastructure may be isolated. Versatility comes with the ability to reuse the same cryptographic module for these various usages.
- Handling network data flows at very high transfer rates comes with various bottlenecks that would have to be worked around. Among these bottlenecks, here are the most troubling ones:
  - hardware and software interrupt handling, which quickly becomes a CPU hog, limiting the number of packets per second a device is able to process,
  - protocol encapsulation and fragmentation issues have a strong impact on performance, as they require deeper dissection of data packets,
  - IP forwarding and network address translation complicates the use of some hardware optimizations such as offload engines computing checksums,
  - parallelization of packet processing brings scheduling, pipelining and buffering issues, as well as indirect issues such as packet re-ordering and rendezvous points.

1.2.3 Personalized cryptographic algorithm

As stated earlier, an important feature of the project is the ability to load customer-specific cryptographic algorithms.

This provides another level of security assurance to the customer within the boundaries of the Wassenaar Arrangement which limits exportation of appliances with potential double-use.

2 ARCHITECTURE DESIGN AND CONSTRAINTS

The SHIVA architecture is designed from the ground up, trying to bring new applications and high performance while retaining a high security assurance level.

On the cryptographic module, multiple tasks are performed besides the main purpose of the device. Monitoring, administration and management, as well as alert handling and tamper detection shall all be running at the same time, while the cryptographic core needs to communicate with other subsystems to function properly, such as a session management subsystem in a VPN use-case (security associations, security parameters and encryption keys for a particular tunnel).

Clever task segregation also allows using different development techniques, with hardware handling critical capacities while software can support some other features, thus facilitating developments and security efforts.

Modern FPGAs have the ability to implement one or several soft-core CPUs, allowing the use of general-purpose embedded operating systems which greatly simplifies development. Models such as Xilinx Virtex 4 and Virtex 5 families also provide the option of embedding hard-core ASIC CPU cores. As running such operating systems comes with implications on security, tasks being performed by the operating system will have to be properly secured.

In the following section, the main characteristics, technical choices and proposed advancements are discussed and consider the impacts on the security level of the device.

2.1 General architecture

The new architecture tries to segregate tasks as much as possible. This brings a number of sub-systems to be considered and how to make them interact securely, while still providing performance enhancements.

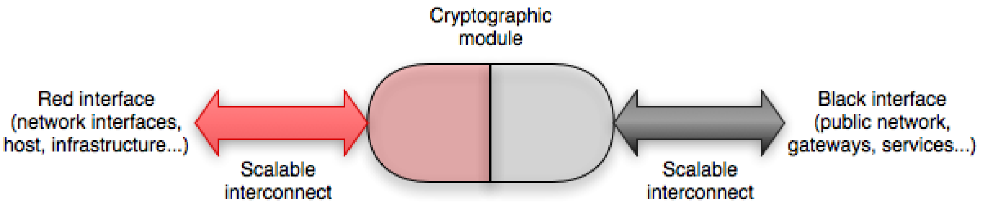

Figure 3: SHIVA architecture with a dedicated cryptographic module, embedding a security engine and a cryptographic engine, and its separated interfaces

2.2 Interconnection

While designing the SHIVA cryptographic module, a choice was to be made regarding its interconnection with appliances. The need for a fast, scalable, reliable technology, with low processing overhead and straightforward developments arose.

As it aims to be interoperable with a large variety of equipments, with reusability in mind, emerging or unpopular technologies were excluded. There are other computing areas which require very high performance. Such an interesting field is supercomputers and HPC (High-Performance Computing). Very large clusters of computing nodes require tremendous amounts of bandwidths being transferred from point to point with very low latency tolerances.

An interesting fact about HPC technologies, is that interconnection standards are converging towards a mixed usage of Ethernet-based networking and InfiniBand Infrastructures[6].

While Ethernet is a widely-known and mainstream standard, InfiniBand is somewhat regarded as an HPC-only technology[7]. This is without considering the fact that its specifications served as a basis of the now widely deployed PCI-Express standard. It features a switched fabric for communications with quality of service features, failover capacities and great scalability: by aggregating several communication channels in an adaptable fashion, one can set up InfiniBand interconnect with throughput ranging from 2 Gbits/s up to 96 Gbits/s of useful bandwidth.

Beside the performance advantages of an HPC technology, InfiniBand is a very interesting technology to provide scalable architecture, devices and versatile solutions. Indeed, applications of different scales could only implement one, four, eight or twelve aggregated InfiniBand transfer lanes for their interconnection, with current useful data-transfer rates of 2, 4 or 8 Gbits/s per lane (respectively using SDR, DDR or QDR signaling rates). The InfiniBand Trade Association is planning evolution towards bidirectional rates of 480 Gbits/s in the next three years[8] by only enhancing signaling rates, not multiplying the number of lanes. This alone is a major advantage regarding scalability and future evolution, as general architecture design would require only slight adaptations while interconnection bandwidth evolves.

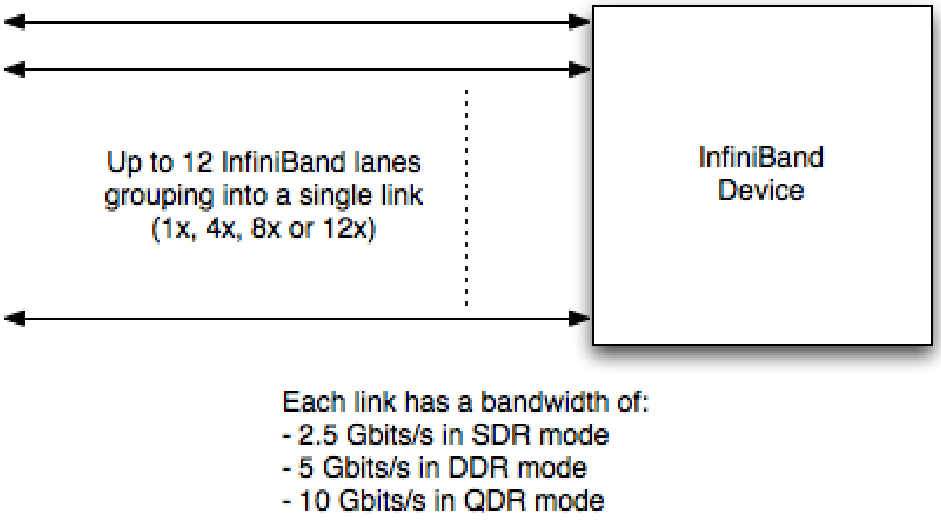
Up to 12 InfiniBand lanes
grouping into a single link
(1x, 4x, 8x or 12x)
InfiniBand
Device
Each link has a bandwidth of:
- 2.5 Gbits/s in SDR mode
- 5 Gbits/s in DDR mode
- 10 Gbits/s in QDR mode

Figure 4: Composition of an InfiniBand link

Choosing this technology opens up a whole new range of applications and reusability of a cryptographic device. One could imaging integrating such a cryptographic device in a small cluster of computers. These computers could act as service interfaces with the high-performance cryptographic module, such as cryp-

[6] Most InfiniBand vendors such as Mellanox Technologies or Voltaire are providing gateway interfaces between Ethernet and InfiniBand infrastructures for datacenters.

[7] The top500.org website keeps statistics about the world's most powerful supercomputers to date. Infiniband interconnect is credited with a 36.20% share, and Gigabit Ethernet with 51.80 % as of November 2009. These shares have been on the rise since 2002 (gigabit Ethernet) and 2004 (InfiniBand)

[8] InfiniBand Trade Association Roadmap, http://www.infinibandta.org/content/pages.php?pg=technology_overview

tographic requests (certificate and key management), or multi-level security services. The main use-case toward which the proof-of-concept efforts are focused could be a very-high performance network blacker device, offering VPN tunnel services (such as IPsec) handled by the cryptographic device. Multiple 10 Gbits/s ethernet networks could be connected to a gateway located on the red network side of the platform, passing the operational flow to the cryptographic device which performs encryption and forwards the frames to another InfiniBand node, which in turn will act as a gateway on the black network side of the platform.

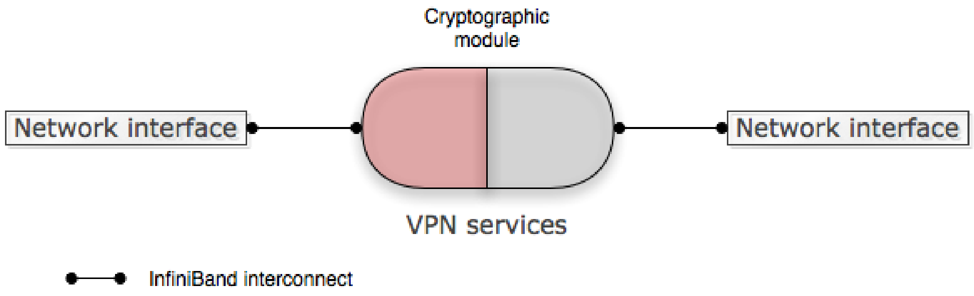

Figure 5: SHIVA architecture in a VPN configuration

Multiple gateways could then be connected to a single SHIVA module to provide an unprecedented overall throughput performance.

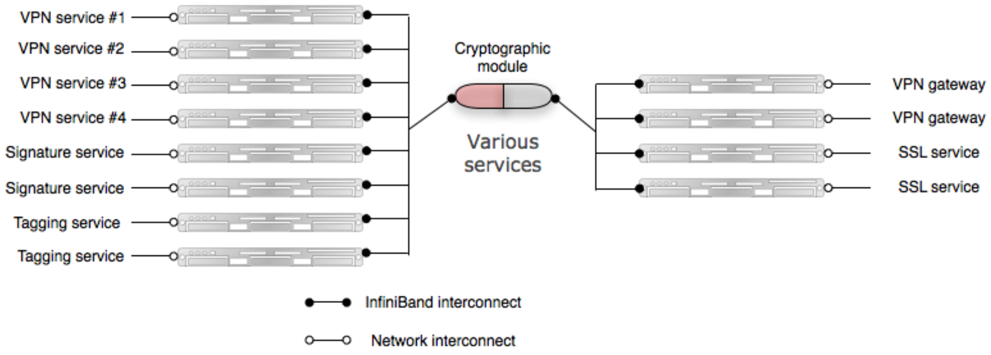

Figure 6: SHIVA module in an architecture offering multiple services to maximize throughput of a single cryptographic module

InfiniBand interconnect is available on both copper and optical carriers. Optical interconnection offers many advantages, such not being prone to external interferences, and not having compromising emissions (e.g. TEMPEST requirements). They will also not be able to carry compromising signals from or to the cryptographic module

2.3 Network on a Chip

Pursuing the goal of using formal methods applied to security verification procedures, hardware design relies on formal models developed by TIMA [2]. Network on a Chip (NoC) architectures offer several advantages over conventional designs:

- A high level of parallelism can be obtained, and an higher level of scalability can be provided compared to conventional bus systems;
- Formal models provide proof of correctness of the internal communication architecture;
- An internal networking stack can provide protection against faults and errors with consistency and integrity checks [11].

These advances help us develop robust and scalable hardware designs to support planned capacities in a secured manner.

2.4 Cryptographic capacities

General cryptographic capacities of the module are handled by a dedicated FPGA implementing the cryptographic engine, and communicates with a security engine. The security engine runs the embedded operating system which performs various monitoring and management tasks. This security engine also communicates with the cryptographic engine handle requests for secured information stored in dedicated memory, such as security associations and parameters, as well as checking the component's state and integrity.

When the cipher algorithm is not to be disclosed, it can be implemented in the cryptographic engine as a partial bitstream, which can be dynamically loaded or changed through dynamic partial reconfiguration of the device [16], i.e. changing part of the logic configuration while the rest of the chip is still operating. An example of such application would be the french standard RIP6 (Réseaux IP Sécurisés Interopérables Standardisés) or NATO HAIPIS (High Assurance Internet Protocol Interoperability Specification), an hardened IPsec evolution [4]. While these applications were originally developed using ASIC components, the FPGA-based architecture allows a customer to load his own algorithm with his own countermeasures and security features using a dedicated Application Programming Interface (API). This algorithm is then also isolated from the other components of the system.

2.5 Key and session management

As an embedded operating system such as Linux may be used on the security engine, doing so is only relevant if sufficient security assurance is provided. In the case of an IPsec VPN gateway, using this operating system greatly simplifies handling key and session management protocols such as IKE.

Using a hardened embedded Linux OS on FPGA, development of a proof-of-concept is currently going on using modified versions of strongSwan which eliminates keys from the scope of the program. Keys are then externally stored and only accessed by dedicated hardware co-processors, which transmit them to the cryptographic engine in a black form upon request. The IKE daemon is restrained to handling keys only through slot identifiers passed to the custom co-processors. No software, including the operating system, ever has access to those materials, which also allows only relevant memory to be made available to the OS.

2.6 Monitoring and management

System monitoring and management are again easily developed when based on a general-purpose operating system. Doing so allows using higher level languages which can bring formal proof of correctness or security properties obligations.

Separating cryptography from management and monitoring also eliminates interferences of having many sub-systems packed into a single entity.

3 OPERATIONAL FLOW HANDLING

With such high goals of performance, processing of the operational flow can prove challenging. To overcome some issues and keep the cryptographic module from performing non-critical operations, a solution is to dispatch the architecture between pre-processing interfaces and the cryptographic core. This allows limiting the number of functionalities and protocol-handling the cryptographic module would have to perform as well as keeps its surface of vulnerability smaller than an all-in-one solution.

3.1 Pre-processing and optimisations

Depending on the type of the operational data flow, several pre-processing or post-processing could be done before handing it over to the cryptographic module. If handling IPsec traffic, a possibility is, for instance, to pre-process most of the network encapsulation, filtering or post-process decapsulation to relieve the cryptographic module from such tasks, thus limiting the number of tasks implemented in the module which are unrelated to the main objective: performing critical cryptographic operations.

Having these problematics in mind, the SHIVA architecture allows such pre-processing to occur, by placing dedicated processing equipments in front of the device. While directly handling network interconnection, these devices also perform preliminary or post-processing tasks such as:

- Network protocol encapsulation / decapsulation;
- Enforcing basic security policy, such as content filtering or firewalling;
- Data flow segregation, quality of service enforcement
- Filtering and forwarding remote management commands or monitoring data form or to the cryptographic module.

This spawned research and development of IPsec encapsulation optimizations, and software tools for offloading tasks from the cryptographic module to the pre- and post-processing units.

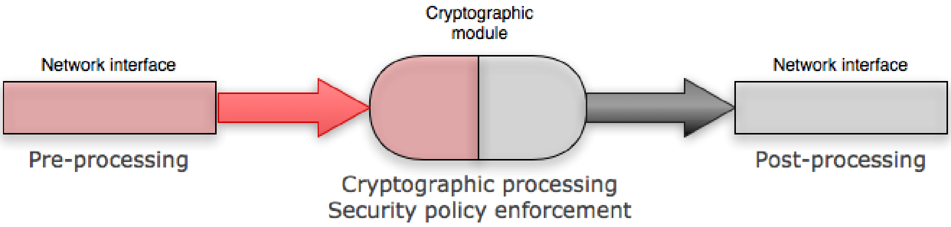

Figure 7: Preprocessing and post-processing when handling outgoing traffic in a Virtual Private Gateway (VPN) gateway configuration

However, this does not eliminate all bottlenecks stated in §1.2.2, especially interrupt handling problems. Testing has proven reasonable bandwidth could be attained with mainstream hardware and software with

IPsec forwarding and encapsulation, using blank cipher encryption. The focus on software packet handling frameworks led to the testing and development of in-house solutions to perform highly-efficient packet encapsulation and forwarding between IP and InfiniBand networks. Where most IP over InfiniBand protocol implementations could not get performance beyond 5 Gbits/s, these solutions are currently capable of handling about 25 Gbits/s[9] of traffic with packet sizes of less than 1500 bytes. This work-in-progress solution have not yet been tested beyond, and is yet still scalable, using full potential of multicore architectures.

Latest technological advance is the appearance of multi-queue 10 Gbits/s network interface cards[10], allowing both interrupt handling and packet processing to occur on multiple CPU cores simultaneously. While initially developed for virtualization by affecting queues to virtualized operating system, this technology allows parallelization of interrupt handling for network flows

These solutions coupled with parallelization of packet pre-processing brings new opportunities for better performance, leaving only critical cryptographic processing to the SHIVA cryptographic module, which is then relieved of most network-specific processing.

3.2 Cryptographic module interconnect

By using InfiniBand interconnect, the solution can rely on existing features to easily perform several tasks:

- early traffic segregation, such as management traffic directed to the security engine or operational flow which will exclusively be processed by the cryptographic engine,
- early dispatching of operational flow to parallel processing units,
- basic quality of service (QoS) management by controlling buffer priorities

In the same manner virtual network IDs (VLAN tags) may be used to segregate Ethernet traffic, InfiniBand provides a virtual lane feature, which allows tagging traffic with a particular identifier. This tagging is performed at pre-processing before the flow is sent to the cryptographic module, which dispatches is on various processing units in respect of the QoS priority policy.

This features also help ensure management commands or data are not mixed with operational flow, and helps speeding up packet interpretation. Signed management packets would be routed to the signature verification unit (thus preventing injection of malicious management packets) and forwarded to the security engine right away, or immediately dropped if failing the tests. Sufficient resources could be dedicated to these critical management data flows, and pre-processing would normally ensure no operational flow gets transmitted over these dedicated virtual lanes.

In the same manner, basic traffic shaping can also be applied to different virtual lanes, based on the InfiniBand flow control features, as well as memory segregation: incoming operational flows, pending processing, are directed to dedicated and secured external memories, while management flow could be handled internally.

[9] Tests, benchmarks and optimizations in progress. We obtained 25 Gbits/s of forwarding from multiple 10 Gbits/s ethernet link to a single InfiniBand link with software solutions developed in-house, nearing theoretical limits of about 20 millions of packets per seconds being handled by the system in a standard use case.

[10] Intel® Virtualization Technology for Directed I/O (Intel VT-d), http://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices/

4 FORMAL METHODS APPLIED TO SECURITY

Formal methods are a required when developing high security assurance appliances. These are stated as the basis from conception to development by several information security standards, such as FIPS 140-2 recommendations or ISO/IEC 15408 Common Criteria. But their use could go a step further than limiting to security features or cryptographic modules. Research and experimentations are being conducted on the SHIVA project to make intensive use of such methods.

4.1 Protocol handling

As the primary focus is developing the Virtual Private Gateway (VPN) applications of the SHIVA cryptographic module, formal methods could bring additional security assurance in protocol implementations such as IKE or even IPsec.

While these protocols have already been studied and extensively challenged, current implementations may always suffer from mishandling or programming errors. It is also impossible to evaluate them at high standards of security assurance due to the lack of high level specifications or formalism.

Based on different formal methods, an implementation entirely derived from formal specifications of networking and IPsec protocols is proposed, formalizing security goals of the protocol, enforcing them, as well as generating a robust implementation less prone to human programming errors.

Using such an implementation in an existing operating system is possible through the use of packet handling frameworks, in which formally designed components can be integrated. It is indeed quite difficult to base such developments solely on the Linux kernel API, for instance. However, the use of user-land frameworks allow easier integration of generated code with no modifications besides interfacing. It also help keeping these tasks in a secluded environment.

An example of such a framework would be Click[11], which can easily modified to accommodate a large variety of protocols, not limited to IPsec.

4.2 Supervision and administration

Formally designed and verified administration and monitoring tools are currently in development using formal methods like the B-Method[12] or the Coq Proof Assistant[13]. These tools communicate with a management core, also formally designed and verified, which handles critical operations:

- User authentication and role management;
- Mandatory access control to available features;
- General state transitions;
- Security events and alerts handling;
- Communication with remote management stations.

[11] The Click Modular Router Project, http://read.cs.ucla.edu/click/

[12] Atelier B, http://www.atelierb.eu/

[13] The Coq Proof Assistant, http://coq.inria.fr

This management core provides proof of correctness and enforces several security properties, and checks machine configuration coherency.

Management core, authentication, role management, configuration handling

5 SECURITY MEASURES

While generic countermeasures are implemented (bus protections, DPA and SPA mitigation techniques, continuous monitoring...), some are being developed with the SHIVA architecture in mind. While algorithm-specific countermeasures can only be implemented by the entity having access to its specifications, a number of improvement over straightforward implementations are still provided. Default algorithms, such as the Advanced Encryption Standard (AES), are implemented with multiplicative masks to deter power analysis attempts. Otherwise, some advantages of FPGA-based platforms are put to good use with innovative solutions.

As stated in §1.1.1, a high attack potential is considered. Therefore, both software and hardware platforms are not trusted. This implies careful countermeasure implementations and FPGA technology helps in providing suitable methods for gradually rising the device's security level.

5.1 Cryptographic initialization

The cryptographic initialization of a device is one of the most critical phase, since eavesdropping or reverse-engineering may compromise the whole architecture and stored secrets.

The solution proposes the implementation of a staged initialization process, where each stage performs integrity, authenticity and clearance level checks before being allowed to continue with the next one.

A three-stage process allows an acceptable level of security assurance:

1. The first stage after power-up loads basic configuration software to the security processor. This software runs basic hardware self-tests, and allows input of security parameters to unlock the second stage.

2. The second stage decrypts and loads the security engine bitstream as well as its operating system, using keys computed at the earlier stage, and initiates a secure communication channel with the cryptographic engine by loading it with an initial bitstream. After performing self-tests, authenticating with the security engine and input of secret parameters by the operator, the third and last stage is unlocked.

3. The final stage allows decrypting and loading of operational bitstream to the cryptographic engine, including algorithms, after the hardware architecture has been verified and deemed to provide enough security assurance. Final tests should ensure the proper functioning of the cryptographic capacities.

Below is an illustration summing up the different stages of the cryptographic initialization process:

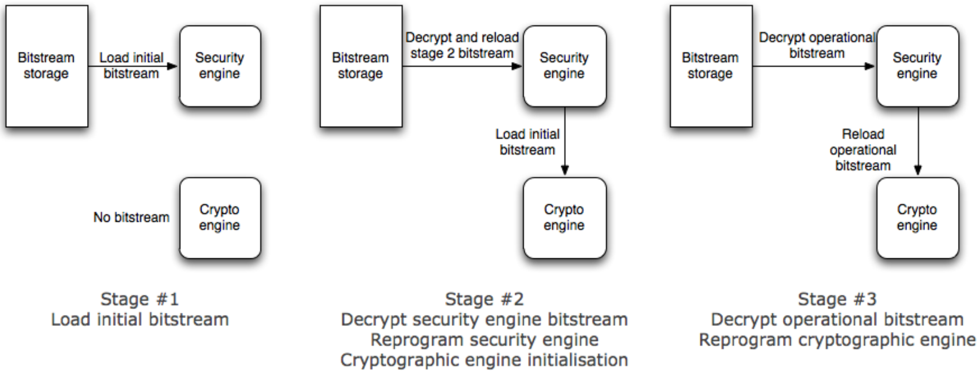
Bitstream storage
Load initial bitstream
Security engine
No bitstream
Crypto engine
Stage #1
Load initial bitstream
Bitstream storage
Decrypt and reload stage 2 bitstream
Security engine
Load initial bitstream
Crypto engine
Stage #2
Decrypt security engine bitstream
Reprogram security engine
Cryptographic engine initialisation
Bitstream storage
Decrypt operational bitstream
Security engine
Reload operational bitstream
Crypto engine
Stage #3
Decrypt operational bitstream
Reprogram cryptographic engine

Figure 8: 3-stage cryptographic initialization process, with progressive advance to the highest security level

This three-stage process tries to ensure the platform is safe before proceeding with the loading of information of higher importance. A stage failure will not allow decrypting the stored data for the next step.

Raising the security level at each step also raises the level of emergency zeroization performed in case of failure. These failure conditions will enable putting the device to a blocked state by erasing critical information which will prevent subsequent initialization, rendering stored secret informations useless since the device's master encryption keys would be lost.

5.2 Storage area rotation

It is widely known data remanence can compromise encryption keys or other critical information. While attacks taking advantage of physical properties of different memory technologies are common on mainstream hardware (RAM memory modules, flash storage drives), some other properties of silicon-based chips can prove dangerous if not taken into account.

It has been proven that data-remanence can appear after electromigration or ionic contamination of silicon-based transistor devices occurs [9]. Early signs of this migration can be detected after a few hours of operation. Observation through adequate equipment which may compromise secret, red information if the device does not take appropriate measures.

For these reasons the SHIVA module implements a storage area management system that allows rotation of storage areas where red keys are placed after a few hours of operation. Switches occur between all RAM blocks and registers of the FPGA, which prevents an attacker from blindly going after a memory location in hopes to find critical material. Partial reprogramming capacities of modern FPGAs allow such rotation to be also performed on implemented algorithms, through live placement of logical blocks (given the cryptographic engine is dedicated to these algorithms).

These features are also combined with the last advantages of FPGAs discussed below.

5.3 FPGA dynamic reconfiguration

The ability to partially or fully load or unload a bitstream at any time on a FPGA allows the introduction of a rotation feature between multiple chips.

Indeed, the SHIVA architecture proposes the use of multiple FPGAs:

- Two of these FPGAs are at any time performing the same cryptographic engine operations using differently implemented functions, and compare the results on output to detect errors or tampering of a single processor;
- One stays idle for the rotation period to account for the previously-discussed remanence issues;
- One provides security engines capacities to the module.

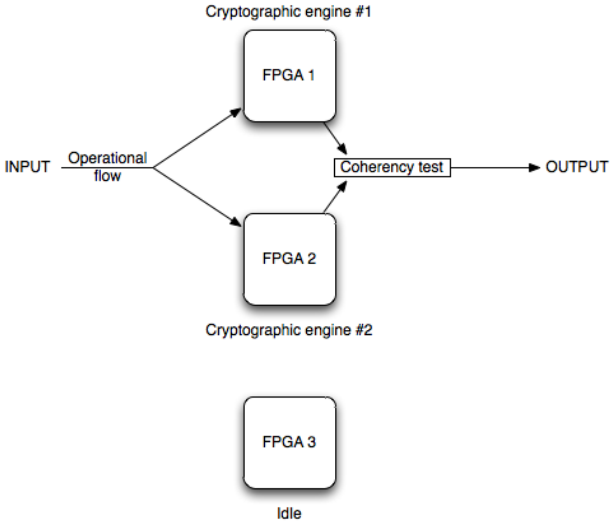

Figure 9: Configuration of cryptographic engines while handling operational flow

Such a rotation provides an additional level of security assurance while keeping performance and reliability achievable.

6 CONCLUSION

While tackling the difficult task of combining high performance equipments with high security assurance levels, the SHIVA project provides solutions to achieve this goal, as well as providing multi-service and multi-level enhancements.

While the potential attacker is always considered having access to unlimited resources, an architecture designed to handle performance issues in this hostile environment is being developed. Using HPC technologies such as InfiniBand, it provides very fast, scalable and easily evolutive cryptographic modules. Optimizations are also explored in the form of task segregation, pre- and post-processing or parallelization techniques.

Formal methods bring a high security assurance being used for software and hardware conception. They are also used to re-implement protocol handling and to develop security management and monitoring tools. Careful control of the device's state machines also ensure a proper level of security, even when considering the environment hostile by proper security measures, 3-stage initialization procedures and FPGA-based dynamic reconfiguration and operational flow handling techniques.

These solutions should bring security appliances to a next level in terms of service and security assurance.

References

[1] L. R. Avery, J. S. Crabbe, S. Al Sofi, H. Ahmed, J. R. A. Cleaver, and D. J. Weaver. Reverse Engineering Complex Application-Specific Integrated Circuits (ASICs). In *Diminishing Manufacturing Sources and Material Shortages Conference*, 2002.

[2] D. Borrione, Amr Helmy, Laurence Pierre, and Julien Schmaltz. Executable formal specification and validation of NoC communication infrastructures. In *Proceedings of the 21st annual symposium on Integrated circuits and system design*, 2008.

[3] Gaetan Canivet, Jessy Clédière, Jean Baptiste Ferron, Frederic Valette, Marc Renaudin, and Régis Leveugle. Detailed analyses of single laser shot effects in the configuration of a virtex-ii fpga. In *IOLTS '08: Proceedings of the 2008 14th IEEE International On-Line Testing Symposium*, pages 289–294, Washington, DC, USA, 2008. IEEE Computer Society.

[4] Committee on National Security Systems. *National Policy Governing the Use of High Assurance Internet Protocol Encryptor (HAIPE) Products*, Feb. 2007.

[5] E. De Mulder, S. B. Örs, B. Preneel, and I. Verbauwhede. Differential power and electromagnetic attacks on a fpga implementation of elliptic curve cryptosystems. *Comput. Electr. Eng.*, 33(5-6):367–382, 2007.

[6] J.F. Dhem, J. f. Dhem, F. Koeune, P. Mestré, F. Koeune, P.-A. Leroux, P. a. Leroux, J.-J. Quisquater, J. j. Quisquater, J. l. Willems, J. l. Willems, and Bld E. Jacqmain. A practical implementation of the timing attack, 1998.

[7] Saar Drimer. Volatile FPGA design security – a survey (v0.96), April 2008.

[8] Government Committee of the USSR for Standards. *Cryptographic Protection for Data Processing System, Gosudarstvennyi Standard of USSR (In Russian)*, 1989.

[9] Peter Gutmann. Data Remanence in Semiconductor Devices, 1998.

[10] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In Paul C. van Oorschot, editor, *USENIX Security Symposium*, pages 45–60. USENIX Association, 2008.

[11] H. Paulitsch, C. Paukovits, C. El Salloum, and H. Kopetz. Fault Isolation with Intermediate Checks of End-to-end Checksums in the Time-Triggered System-on-Chip Architecture. *DATE Friday Workshop on Diagnostic Services in Network on Chips (DSNOC'09) - Test, Debug, and On-Line Monitoring*, Apr. 2009.

[12] A. Pellegrini, V. Bertacco, and T. Austin. Fault-Based Attack of RSA Authentication. In *Proceedings of 2010 Design, Automation and Test in Europe Conference*. University of Michigan, 2010.

[13] Jörn-Marc Schmidt and Chong Hee Kim. A Probing Attack on AES. In *WISA*, pages 256–265, 2008.

[14] Sergei P. Skorobogatov and Ross J. Anderson. Optical Fault Induction Attacks. pages 2–12. Springer-Verlag, 2002.

[15] François-Xavier Standaert, François Macé, Eric Peeters, and Jean-Jacques Quisquater. Updates on the security of fpgas against power analysis attacks. In *ARC*, pages 335–346, 2006.

[16] Xilinx. *Virtex-5 FPGA Configuration Guide*, 2009.